

Capítulo

6

Introdução à Teoria e Prática da Interação Humano Computador fundamentada na Engenharia Semiótica

Raquel Oliveira Prates¹, Simone Diniz Junqueira Barbosa²

Abstract

The goal of this chapter is to briefly introduce the area of Human-Computer Interaction (HCI) and to describe Semiotic Engineering, a theory of HCI that presents a broader account to the activities involved in the conception, modeling and evaluation of interactive systems, and also instruments these activities with methods grounded in the theory. Semiotic Engineering was developed in Brazil in the 90's and is currently an internationally known theory. In 2006, the Brazilian HCI community recommended that Semiotic Engineering be included in the syllabus of undergraduate and graduate HCI courses.

Resumo

O objetivo deste capítulo é introduzir brevemente a área de interação humano-computador (IHC) e se aprofundar na Engenharia Semiótica, uma teoria de IHC que apresenta uma caracterização mais ampla das atividades envolvidas na concepção, modelagem e avaliação de sistemas interativos, e instrumenta essas atividades com métodos ancorados na teoria. A Engenharia Semiótica foi desenvolvida no Brasil nos anos 90 e é atualmente uma teoria reconhecida internacionalmente. Em 2006, a sua inclusão na ementa de disciplinas de IHC passou a ser recomendada pela comunidade brasileira de IHC.

¹ Departamento de Ciência da Computação, UFMG.

² Departamento de Informática, PUC-Rio.

6.1. Introdução

A Ciência da Computação foi definida pelo comitê de fundamentos da Ciência da Computação do Conselho Nacional de Pesquisa dos Estados Unidos como sendo “o estudo de computadores e o que eles podem fazer – as capacidades e limites inerentes a computadores abstratos, o design e características de computadores reais, e as inúmeras aplicações de computadores na resolução de problemas”³ [CFCS 2004, p. 12]. Assim o comitê descreve a sua área de atuação como indo desde pesquisa sobre propriedades de dispositivos eletrônicos até características relacionadas à compreensão humana, abordando tanto aspectos teóricos, quanto pragmáticos desta ampla gama de questões. Dentre as diversas especialidades que constituem a Ciência da Computação podemos descrever a área de Interação Humano-Computador (IHC) como sendo aquela que considera todos os aspectos relacionados com a interação entre pessoas e computadores [Preece et al. 1994].

Por várias décadas depois de criados os primeiros computadores, era necessário que o usuário entendesse a fundo como a máquina funcionava para poder lhe dizer o que fazer. Porém, na década de 70, observou-se um rápido crescimento e barateamento da tecnologia, que possibilitaram a criação dos computadores pessoais, seguida de uma popularização do seu uso por pessoas leigas em computação [Dourish 2001]. Ao mesmo tempo, várias universidades e centros de pesquisa começaram a investir em pesquisas e desenvolvimento de sistemas que pudessem ser utilizados por usuários não especializados e lhes fossem úteis [Preece et al. 1994]. Assim é criada na década de 80 a área de Interação Humano-Computador.

Em 1981, Moran definiu formalmente a interface de um sistema com o usuário como sendo “a parte de um sistema computacional com a qual a pessoa entra em contato – física, perceptiva ou conceitualmente” [Moran 1981]. Em meados da década de 80 foi então cunhado o termo interação humano-computador (IHC)⁴ para definir esta nova área de estudo, cujo foco era não apenas o projeto de interface, mas todos os aspectos relacionados com a interação entre usuários e sistemas [Preece et al. 1994]. Por envolver não apenas os computadores, mas também as pessoas que os utilizam, IHC é multidisciplinar e se encontra na interseção das ciências da computação e informação e ciências sociais e humanas. Em 1982 foi criado na ACM (*Association of Computing Machinery*) o grupo de interesse em IHC, o SIGCHI⁵ (*Special Interest Group in Computer Human Interaction*) [Borman 1996], que há vários anos é o 2º maior grupo de interesse da ACM [Konstan 2007]. No Brasil, os primeiros trabalhos na área de IHC surgiram ainda na década de 80. No entanto, a comunidade de IHC só se organizou no final da década de 90. Em 1997, foi criada uma lista de interesse em IHC e uma página

³ Tradução livre das autoras do original: “*Computer science is the study of computers and what they can do — the inherent powers and limitations of abstract computers, the design and characteristics of real computers, and the innumerable applications of computers to solving problems.*”

⁴ Foi definido na comunidade internacional o termo *Human-Computer Interaction*, e seu acrônimo HCI. Apenas em 1998 foi que a comunidade brasileira de IHC definiu formalmente como deveria ser o termo em português.

⁵ O SIGCHI é atualmente o 2º maior grupo de interesse da ACM, ficando atrás apenas do SIGGRAPH (grupo de interesse em Computação Gráfica) [Konstan 2007].

contendo as informações de contato das pessoas trabalhando na área. Nesta época foi criado um “potencial” grupo local de interesse em IHC associado ao SIGCHI (*Prospective Local Chapter*, o BRCHI). Em 1998, foi organizado o primeiro Workshop de Fatores Humanos em Computação Gráfica, junto ao SBES em Maringá, PR, o IHC 1998.

Em 2000, o BRCHI se tornou um grupo consolidado junto à ACM/SIGCHI e em 2002 foi criada a Comissão Especial de IHC (CEIHC) junto à SBC. O evento da comunidade, IHC, foi organizado anualmente e se tornou um Simpósio em 2002. Neste mesmo ano, a comunidade se decidiu por tornar o evento bianual, e revezá-lo com a Conferência Latino-Americana de IHC (CLIHC), que teve sua primeira edição em 2003 no Rio de Janeiro. A comunidade de IHC atualmente está bem consolidada no Brasil, e a área de IHC tem se firmado no Brasil no desenvolvimento de pesquisas de qualidade, na formação de profissionais de computação, sendo oferecida nos cursos de Ciência da Computação, Engenharia de Computação e Sistema de Informação como disciplina optativa ou obrigatória, e também na indústria, com o crescimento da preocupação com a qualidade do sistema para o usuário [Silveira 2007].

6.1.1. Interação e Qualidade de Uso

A interface é a parte do sistema computacional com a qual o usuário se comunica, ou seja, aquela com a qual ele entra em contato para disparar as ações desejadas do sistema e receber os resultados destas ações, que o usuário então interpreta para em seguida definir suas próximas ações. A este processo de comunicação entre usuário e sistema se dá o nome interação [Preece et al. 1994]. Ver Figura 1.

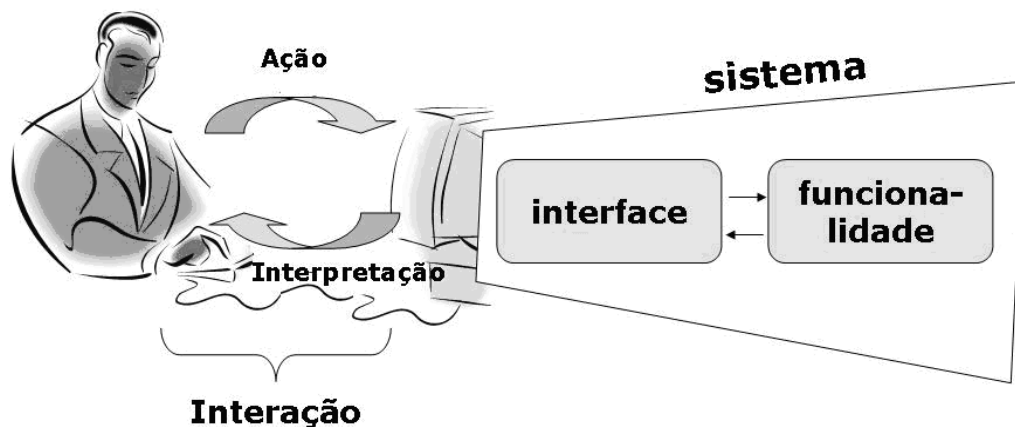


Figura 1. Interação entre usuário e sistema

O usuário interage com o sistema através do software (e.g. janelas de diálogo e linha de comando) e do hardware (e.g. teclado, mouse, monitor). Durante a interação, o usuário entra em contato com o sistema fisicamente (ao manipular dispositivos de hardware), perceptivamente (ao perceber o que lhe é apresentado pela interface), e conceitualmente (ao interpretar e raciocinar sobre a interação e seus resultados) [Moran 1981].

Ao projetar um sistema interativo, uma das preocupações do designer deve ser com a qualidade de uso associada à interação do usuário com a interface. A **usabilidade** de um sistema foi a primeira propriedade definida relativa a esta qualidade, na década

de 80, e leva em consideração a facilidade e eficiência com a qual um usuário consegue utilizar um sistema [Gould & Lewis 1985]. A usabilidade é atualmente a qualidade de uso mais amplamente difundida e utilizada tanto por pesquisadores, quanto profissionais da área. Para se avaliar a **usabilidade** de um sistema deve-se levar em conta os seguintes fatores [Nielsen 1993, Preece et al. 2002]:

- **facilidade de aprendizado:** se refere ao tempo e esforço necessários para que os usuários aprendam a utilizar uma determinada porção do sistema com determinado nível de competência e desempenho.
- **facilidade de uso:** está relacionado não apenas com o esforço cognitivo para interagir com o sistema, mas também com a facilidade de completar a interação sem cometer erros durante este processo.
- **eficiência de uso e produtividade:** analisa se o sistema consegue fazer bem aquilo a que se destina, e se o usuário completa suas tarefas de forma rápida e eficaz.
- **satisfação do usuário:** enfatiza a avaliação subjetiva do sistema feita pelo usuário, incluindo suas preferências pessoais e emoções (positivas ou negativas) que possam surgir durante a interação.
- **flexibilidade:** considera o quanto um sistema é capaz de acomodar caminhos distintos para se atingir um mesmo objetivo, apoiando assim as preferências e modo de trabalho individuais dos usuários.
- **utilidade:** relativo ao conjunto de funcionalidades oferecidas ao sistema para que os usuários realizem suas tarefas.
- **segurança no uso:** se refere ao grau de proteção de um sistema contra condições desfavoráveis ou até mesmo perigosas para os usuários, envolvendo desde aspectos de recuperação de condições de erro até impacto no seu trabalho ou saúde.

A usabilidade foi proposta com base no conhecimento empírico relativo ao uso de sistemas interativos. À medida que a comunidade de IHC se consolidou e a pesquisa e aplicação de métodos avançaram, surgiram novas abordagens e teorias para dar conta dos processos de design e interação [Carroll 2003]. Com elas surgiram também novas propriedades relevantes relativas à qualidade de uso.

A teoria da Engenharia Semiótica, como veremos na próxima seção, se concentra na comunicação entre o designer e usuário sendo feita através da interface de um sistema [de Souza 2005]. Assim, para poder avaliar a qualidade de uso da interação, a Engenharia Semiótica define a propriedade de comunicabilidade. A **comunicabilidade** de um sistema se refere à capacidade de o projetista conseguir transmitir aos usuários, através da interface, o design tal como concebido por ele [Prates et al. 2000]. Em outras palavras, se ao utilizar o sistema os usuários conseguem entender, através da interface, para que o sistema serve, a quem ele se destina, quais as vantagens de utilizá-lo, como ele funciona e quais são os princípios gerais que definem as possibilidades de interação com ele.

Além das propriedades que surgem com as novas teorias, o crescente avanço da tecnologia e de sua adoção no cotidiano das pessoas também requer que novas propriedades sejam consideradas. Por exemplo, em sistemas de apoio ao aprendizado o mais importante é saber se o aluno, usuário do sistema, consegue de fato utilizá-lo para

aprender; ou em jogos o relevante é a sua propriedade de entreter e divertir os usuários; ou ainda em uma comunidade online a prioridade é sua capacidade de apoiar às relações sociais entre seus membros. Dentre tantas propriedades que estão sendo consideradas atualmente, a **acessibilidade** merece um destaque especial. A propriedade de acessibilidade está relacionada com a capacidade que o sistema tem de permitir que pessoas com deficiências⁶ possam perceber e entender o sistema, e utilizá-lo [Decreto-Lei 5296, W3C/WAI]. Vários países, dentre eles o Brasil, atualmente têm leis que exigem que os sistemas Web de administração pública disponibilizados aos cidadãos sejam acessíveis [Decreto-Lei 5296]. Relacionado à acessibilidade tem-se também o conceito de usabilidade universal que propõe que os sistemas devem poder ser utilizados em uma ampla diversidade de usuários (incluindo aqueles que têm deficiência), tecnologias (e.g diferentes equipamentos novos ou obsoletos e sistemas operacionais) e ambientes físicos (e.g. disposição dos móveis em uma sala) [Shneiderman & Plaisant 2005, Vanderheiden 2000].

6.2. Teoria da Engenharia Semiótica

A teoria da Engenharia Semiótica (EngSem) [de Souza 2005] é uma teoria explicativa de IHC, ou seja, uma teoria que nos permite entender os fenômenos envolvidos no design, uso e avaliação de um sistema interativo. Como tal, ela tem por objetivo esclarecer a natureza e aspectos envolvidos nestas atividades. A análise fundamentada na teoria enriquece o entendimento que os designers têm do problema que estão tentando resolver com o sistema, e assim amplia as suas considerações em relação a possíveis soluções. Não é objetivo desta teoria prever os resultados de uma ação ou levar o designer a buscar “a” solução correta para o problema em questão.

O primeiro aspecto relevante para o qual a EngSem chama a atenção é que um software é um artefato intelectual. Mas o que é um artefato intelectual? Todos os artefatos gerados por humanos são de fato produto de sua criatividade e exercício intelectual. No entanto, de Souza (2005) define como um **artefato intelectual** o produto gerado a partir da interpretação de um projetista sobre um problema e sua concepção de solução, que é então apresentada em uma codificação lingüística. Para um artefato ser considerado intelectual ele deve ter as seguintes características:

- Codificar uma determinada interpretação de uma situação;
- Codificar um conjunto de soluções para a situação em questão;
- A codificação da situação e das suas soluções é lingüística, ou seja, baseada em um sistema de símbolos que possa ser interpretado por regras semânticas consistentes;
- O objetivo do artefato só pode ser alcançado se os usuários podem formulá-lo no sistema lingüístico no qual o artefato foi codificado. Em outras palavras, os usuários devem ser capazes de entender o sistema e usar a codificação utilizada para explorar os efeitos das soluções disponibilizadas através do artefato.

Considerando esta definição, uma cadeira e um bule não são artefatos intelectuais, uma vez que não possuem uma codificação lingüística associada. Por outro lado, uma tabela

⁶ Deficiência se refere a deficiências motoras, perceptivas ou neurológicas, permanentes ou temporárias (W3C/WAI).

periódica é um artefato intelectual, pois podemos pensar nela como uma solução proposta para se entender a relação entre os diversos elementos químicos existentes, e se obter rapidamente algumas de suas características básicas. A codificação lingüística apresentada envolve a posição dos elementos na tabela e dentro de cada célula. Esta codificação deve ser entendida pelo usuário para se conseguir utilizar a tabela periódica.

Assim, a EngSem caracteriza um sistema interativo como um artefato intelectual. Neste caso, algumas particularidades devem ser levadas em consideração no seu projeto: (1) o artefato deve ser descrito em alguma linguagem artificial que seja processada por um computador; (2) a linguagem de interface com a qual o usuário vai interagir é sempre única e, logo, nova para o usuário; (3) o artefato se caracteriza como sendo de meta-comunicação.

Um artefato de meta-comunicação é aquele cuja mensagem é relativa à própria comunicação. Assim, na visão da EngSem, a interface de um sistema é entendida como um caso de meta-comunicação, uma vez que comunica ao usuário a visão do projetista sobre a quem ela se destina; que problemas ela pode resolver; e como interagir com ela, ou seja, como trocar mensagens com o sistema através da sua linguagem de interface para conseguir executar as tarefas desejadas. Em outras palavras, a interface de um sistema é uma mensagem do designer para o usuário cujo conteúdo é [de Souza 95:84]:

*“Esta é a **minha interpretação** sobre **quem você é**, o que eu entendi que **você quer ou precisa fazer**, de que **formas prefere fazê-lo e por quê**. Eis, portanto, o sistema que conseqüentemente concebi para você, o qual você **pode ou deve usar assim**, a fim de realizar uma série de objetivos associados com esta (minha) visão.*

Esta mensagem sendo transmitida pela interface é indireta, pois o usuário deve compreendê-la à medida que interage com o sistema. A própria interface é quem comunica a visão do projetista, assumindo então um papel de seu representante ou **preposto**. Além disso, a mensagem é unidirecional, pois o usuário no contexto de interação não pode dar continuidade àquela comunicação com o projetista.

Para ilustrar o conceito de meta-comunicação, vamos utilizar o sistema chamado Student Life [Tesoro], que se propõe a auxiliar o aluno na organização de suas atividades escolares e até mesmo aspectos de sua vida social. O aluno ao visitar o site do sistema para baixá-lo receberá esta informação publicada pelo designer do Student Life para todos os potenciais interessados. Ao instalar o sistema e iniciar seu uso, o aluno inicia a sua interação com a interface proposta pelo designer para cumprir este objetivo. A Figura 2 mostra a tela inicial do Student Life com a qual o aluno interage. Imaginemos a primeira interação de um estudante brasileiro com o Student Life: Quando a tela inicial é aberta (Figura 2), o aluno imediatamente percebe as opções (1) e (2) e entende que, para poder selecionar uma disciplina, ele deve primeiro inseri-la. Vendo as informações disponíveis na aba (3), ele percebe que são as informações relativas a uma disciplina. Ele então imagina se pode incluir a informação sobre a disciplina diretamente nesta aba. No entanto, ao passar o mouse sobre ela, percebe que ela não está habilitada para edição e conclui que o caminho para incluir uma disciplina deve ser clicando sobre o botão *Add Class* (2). Ao ver as opções disponíveis do lado direito, o aluno percebe que antes de incluir uma disciplina não pode dizer nada ao sistema sobre seus trabalhos ou provas (4). No entanto as outras opções não são dependentes de uma disciplina. Ele então percebe o botão *My Life* (5) e fica curioso tentando imaginar que aspectos de sua vida esta opção lhe permite organizar. Finalmente, os 3 últimos botões

(6), mais o fato de a interface estar em inglês lhe levam a crer que o sistema foi feito para um estudante americano. Ainda assim, pode ser que lhe seja útil, ele então continua a sua interação com o sistema para melhor entendê-lo.

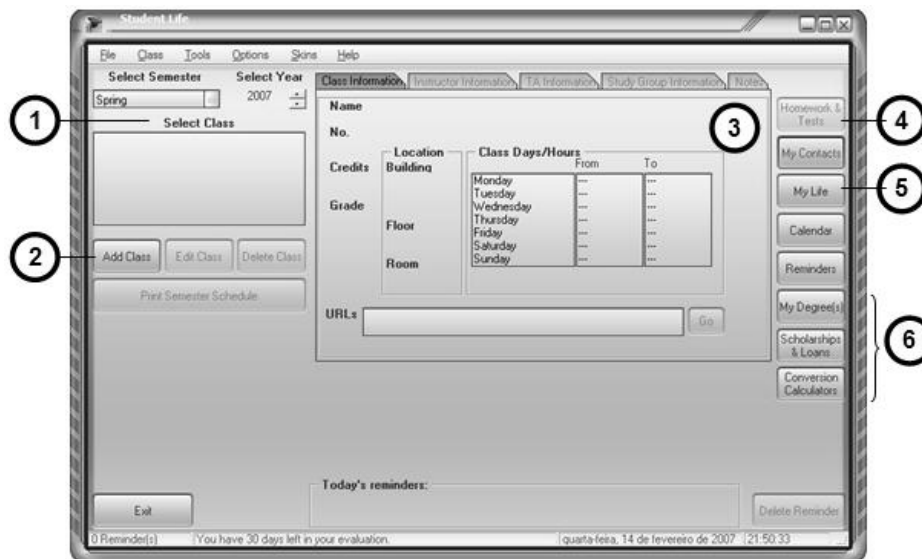


Figura 2. Tela inicial do Student Life

À medida que o aluno interage com o sistema, ele vai entendendo para que serve o sistema, a quem ele se destina, como utilizá-lo e que vantagens terá no uso deste sistema. Em outras palavras, ele vai entendendo a solução proposta pelo designer, ou seja, a meta-mensagem sendo enviada a ele pela interface. Para o Student Life, a meta-mensagem do designer para o usuário seria algo como:

“Caro estudante, eu entendo que você pode ficar sobrecarregado com suas atividades acadêmicas. Assim, estou lhe propondo uma forma de manter suas atividades e prazos organizados. Você registra as informações relativas às disciplinas e seus prazos, e o sistema lhe mostra seus horários e agendamentos do dia ou do mês. Para as disciplinas e trabalhos, você pode cadastrar seus contatos, possibilitando um acesso rápido às suas informações. Além disso, você pode solicitar que lembretes dos prazos mais importantes lhe sejam enviados com antecedência. Como você também tem uma vida social, você pode incluir aqui também seus compromissos e contatos pessoais. Este sistema foi feito principalmente para alunos cursando o sistema universitário americano e o inglês é nossa língua comum de comunicação. De todo jeito, ele pode ser útil também para alunos em outros países que entendam inglês.”



A EngSem é fundamentada na Semiótica, uma disciplina que estuda fenômenos de significação e comunicação. Um dos principais conceitos na semiótica é o conceito de signo. Um **signo** é tudo aquilo que significa algo para alguém [Houser & Kloesel (1992-1998)], assim um signo relaciona uma representação, denominada **representamen** (e.g. a palavra carro, ou a foto de um carro, ou o som de um carro passando), a um **referente** (o veículo) e a uma idéia criada na mente da pessoa ao perceber a representação (e.g. o carro desta pessoa, ou o seu primeiro carro, ou mesmo o filme Carros exibido nos cinemas), à qual se dá o nome de **interpretante**.

Na Figura 3 podemos ver que a mensagem transmitida pelo emissor contém o signo carro, gerando um interpretante na mente do seu receptor. Através de um processo de associação de idéias, cada interpretante pode gerar novos interpretantes, e a este processo dá-se o nome de **semiose ilimitada** [Eco 1976]. A Semiótica considera este fenômeno ilimitado, pois teoricamente o receptor da mensagem pode gerar tantos interpretantes quanto queira, não sendo possível definir quando ou se este processo termina. No entanto, na prática este processo normalmente é interrompido pelo receptor, seja quando ele se dá por satisfeito com o interpretante gerado, seja por falta de recursos – como por exemplo tempo ou paciência. Por exemplo, na Figura 3 quando o emissor comentou sobre seu próprio carro, o receptor se lembrou do seu carro, e então do carrinho de brinquedo que prometera a seu filho e em seguida do seu aniversário. O receptor poderia continuar gerando associações, ou seja, em semiose ilimitada, ou interromper esta semiose, por exemplo por se dar conta que deveria levar o carrinho para o filho neste mesmo dia, saindo então para comprá-lo.



Figura 3. Semiose ilimitada

Em uma interface, os signos são os elementos da interface. No exemplo mostrado na Figura 3, o botão *Add Class* é um signo, assim como o texto *Selected Year*. Os interpretantes gerados na mente do usuário são justamente os pensamentos narrados no exemplo que surgem a partir da interação.

Entendidos estes conceitos básicos, entendamos o foco da Semiótica: significação e comunicação. **Significação** é o processo através do qual expressão e conteúdo de signos são estabelecidos com base em convenções sociais e culturais conhecidas das pessoas que vão utilizá-los, produzindo e interpretando signos. A esta codificação entre expressão e conteúdo se dá o nome de **sistemas de significação** [Eco 1976]. Note-se que em IHC esta codificação pode ser feita artificialmente na definição da linguagem de interface. Por exemplo pense-se na linguagem padrão do sistema operacional Windows e nas aplicações que a utilizam, que normalmente associam o elemento  à ação de fechar uma janela ou  à de desfazer. Estas associações não surgiram de processos culturais ou sociais, mas foram criadas em um determinado momento pelos designers do

Windows. **Comunicação** é o processo através do qual pessoas produzem mensagens formadas por signos utilizando um ou mais sistemas de significação com o intuito de expressar determinados conteúdos. Vale ressaltar que uma parte importante da comunicação consiste em não apenas se apropriar desses signos e sistemas de significação, mas também em se fazer um uso criativo deles, subvertendo o que estava prescrito pelos sistemas de significação ou ainda inventando novos signos. A mensagem criada é enviada através de um canal, para outro interlocutor – o receptor, que deve então ser capaz de interpretá-la.

Sendo o processo de design de uma interface um ato comunicativo, ele requer que o designer entenda o problema, avalie as oportunidades de solução, e tome decisões sobre esta solução. Estas decisões incluem desde aspectos relativos a que pontos do problema serão endereçados, até a definição dos signos e sistemas de significação que comporão a interface e que permitirão a interação usuário-sistema e, logo, a transmissão da meta-mensagem do designer ao usuário. Este processo de design na EngSem é visto sob a perspectiva de reflexão durante a ação (*reflection-in-action*) [Schön 1983], na qual cada problema é visto como sendo um problema único e mutável. Único porque cada problema é caracterizado por elementos do seu contexto que definem o problema. Mutável porque, não apenas não existe uma única solução “correta”, mas também porque o entendimento do problema pode evoluir à medida que o projetista (e usuários) pensam mais sobre ele e exploram diferentes caminhos de solução. Para a área de IHC, Schön propõe que a reflexão dos projetistas seja guiada pela questão de qual artefato eles pretendem criar e como o tornar usável [Schön e Bennett 1996]. Na perspectiva da reflexão durante a ação, propõe-se munir o projetista de **ferramentas epistêmicas**, ou seja, ferramentas que o permitam levantar hipóteses sobre o problema, experimentar diferentes possibilidades de solução e avaliar os resultados. Na EngSem, o papel destas ferramentas epistêmicas é permitir ao projetista da interação refletir sobre questões relacionadas ao artefato de meta-comunicação e comparar diferentes soluções.

6.3. Avaliação da Propriedade de Comunicabilidade

Na perspectiva de meta-comunicação da EngSem, a qualidade de uso de uma interface é dada pela sua comunicabilidade. Como vimos na subseção 1.1.1, a comunicabilidade é definida como a propriedade de um sistema transmitir ao usuário de forma eficaz e eficiente as intenções e princípios de interação que guiaram o seu design [Prates et al. 2000]. Quando o usuário não é capaz de entender a comunicação pretendida pelo designer, ocorrem então rupturas de comunicação que podem dificultar ou até mesmo impossibilitar a meta-comunicação ou uso do sistema. A Figura 4 mostra a tela do sistema Student Life que permite ao aluno incluir uma disciplina no sistema, e algumas potenciais rupturas que podem surgir na comunicação usuário-sistema.

Ao clicar na tela principal (Figura 2) no botão *Add Class*, é aberta para o aluno esta tela. O aluno então começa a criar o cadastro de sua disciplina. Em (1) o aluno vê uma lista vazia de cursos que deveria permitir a ele dizer a qual curso está associada a disciplina. No entanto, não há na tela nenhuma indicação de como inserir os cursos. O texto não pode ser clicado, e mesmo o botão da direita clicado sobre a lista não oferece nenhuma opção. Desta forma, o aluno vivencia sua primeira ruptura, sem saber exatamente como incluir o curso em que está matriculado para o qual a disciplina conta. Em seguida, em (2) o aluno deve dizer a que ano do curso a disciplina está associada. O

sistema apresenta para ele 12 anos como opção, o que ele não entende, já que não consegue imaginar cursos que chegariam a 12 anos de duração. Ele procura alguma explicação, mas não encontra nem mesmo no guia do usuário, então ele imagina que talvez seja uma particularidade do ensino superior nos EUA. Como o seu curso tem duração de 4 anos, ele desiste de entender a que se referem os 12 anos e marca o ano que faz sentido para ele. O aluno segue então e preenche o nome da disciplina, seu código, e os dias e horários em que é ministrada (3). O aluno então tenta salvar a disciplina, e qual não é sua surpresa quando vê que o botão para salvar está desabilitado (4). Ele tenta entender por quê e vê no canto inferior direito um texto informando que os campos marcados com asterisco (*) são obrigatórios (5). Ele então verifica os campos marcados com * e acredita que tenha preenchido todos, e fica então sem saber o que fazer para habilitar o botão. Depois de muito tentar o aluno seleciona o botão Cancel, e começa tudo novamente. Na 2ª tentativa ele percebe que, ao preencher o horário da disciplina (3), ele não havia clicado no botão *Set*, para que o horário fosse confirmado para os dias selecionados, e por isso o botão para gravar não tinha sido habilitado na 1ª tentativa.

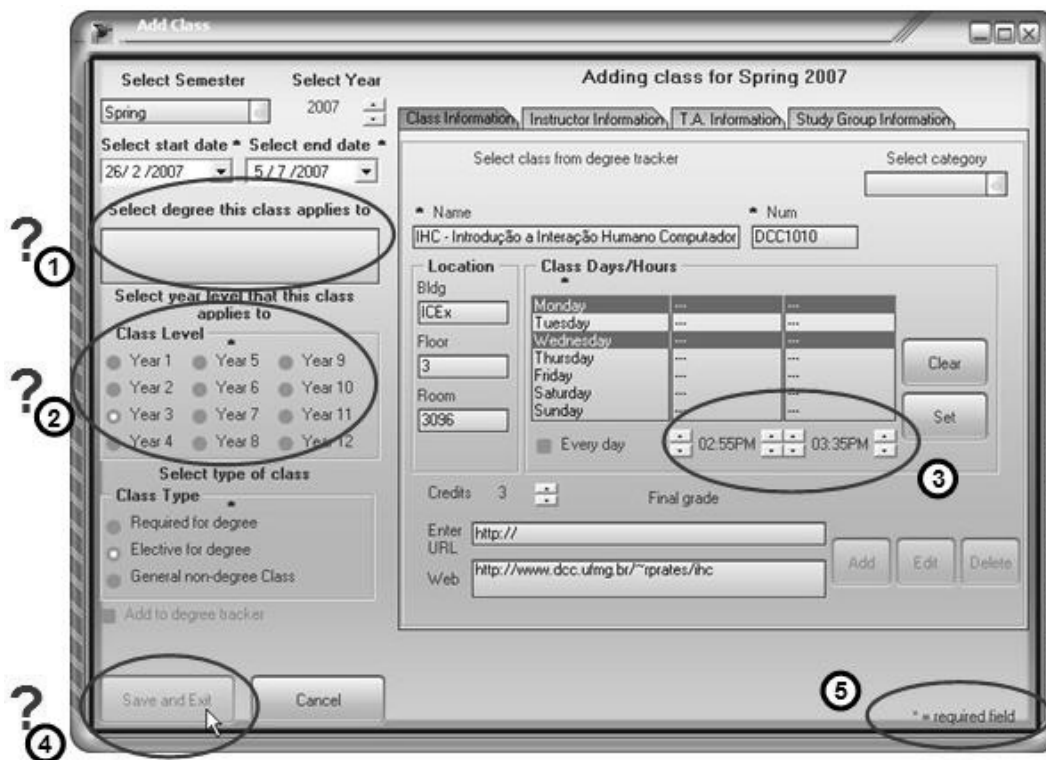


Figura 4. Student Life - Tela para Inserção de Disciplina, com potenciais rupturas de comunicação

Vale ressaltar que cada momento descrito que o aluno não entende a mensagem do designer representa uma ruptura na comunicação designer-usuário. Quanto mais rupturas, ou quanto mais severas as rupturas, mais baixa a comunicabilidade da interface. Atualmente, existem dois métodos para se avaliar a comunicabilidade de uma interface: o Método de Inspeção Semiótica e o Método de Avaliação de Comunicabilidade. O **Método de Inspeção Semiótica (MIS)** é um método antecipativo, ou seja, método em que

um especialista percorre a interface e identifica (i.e. antecipa) potenciais rupturas de comunicação que poderiam surgir na interação usuário-sistema. O **Método de Avaliação de Comunicabilidade (MAC)** é um método que envolve a observação de usuários em um ambiente controlado (e.g. um laboratório de testes) por um especialista. O especialista analisa a interação do usuário com o sistema, identificando então as rupturas vivenciadas pelos usuários. Os dois métodos são **qualitativos**, ou seja, eles têm como resultado indicadores (problemas identificados) sobre a qualidade da meta-mensagem sendo enviada do designer para o usuário – a interface⁷. Outra característica de métodos qualitativos é que os resultados dependem diretamente da interpretação, e logo da experiência, cultura e valores, dos avaliadores aplicando os métodos. A seguir apresentamos em detalhes cada um destes métodos.

6.3.1. MIS – O Método de Inspeção Semiótica

No MIS, o avaliador examina a meta-comunicação do designer para o usuário com o objetivo de identificar se existem rupturas de comunicação e gerar a reconstrução desta mensagem [de Souza et al. 2006]. Como vimos na seção 1.2, esta mensagem é formada por signos de um ou mais sistemas de significação⁸. Logo, para inspecionar a meta-mensagem, o avaliador examina os signos utilizados pelo designer na sua comunicação em 3 diferentes níveis. Para isso a EngSem identifica três níveis de signos: os de meta-comunicação, os estáticos e os dinâmicos.

Os **signos de meta-comunicação** são aqueles que estão presentes na documentação online (e.g. sistema de ajuda, website) ou offline (e.g. manuais de usuário, material impresso de divulgação). Por exemplo, entrando no manual do usuário tem-se a seguinte informação disponível para o usuário⁹ (ver Figura 5).

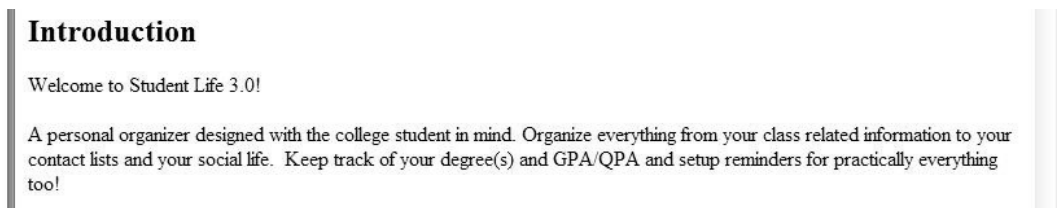


Figura 5. Trecho do Guia do Usuário do Student Life

Observe que neste trecho o designer comunica diretamente ao usuário sobre o Student Life e apresenta alguns aspectos relevantes sobre a sua solução: ela é voltada para alu-

⁷ Métodos qualitativos se opõem a métodos quantitativos que têm por objetivo gerar dados numéricos a partir de métricas e medidas.

⁸ Por exemplo, palavras e imagens normalmente pertencem a sistemas de significação que vêm da cultura fora do contexto específico da interação humano-computador, enquanto apontadores de mouse e janelas de diálogo pertencem a um contexto de significação nativo a aplicações computacionais.

⁹ Tradução livre das autoras do texto da Introdução: Bem-vindo ao Student Life 3.0! Uma agenda pessoal projetada com o aluno universitário em mente. Organize tudo, de informações relacionadas às suas disciplinas, a suas listas de contato e sua vida social. Gerencie seu(s) curso(s) e GPA/QPA e crie lembretes para praticamente tudo também!

nos universitários, ela permite mais do que gerenciar atividades relacionadas aos cursos sendo feitos, mas também da vida social e contatos. O fato de aparecerem as siglas GPA/QPA, sem maiores explicações, indica a expectativa do designer que o aluno-usuário saiba o que significam, e logo o aluno considerado é o americano.

Signos estáticos são aqueles que expressam o estado do sistema. Estes signos normalmente podem ser captados apenas olhando-se para a interface ou mesmo uma imagem desta. Por exemplo, na Figura 6(a) a lista (*Select Class*) mostra o número e nome das disciplinas (até 8) cadastradas. Além disso, o botão *Add Class* habilitado indica que o usuário pode incluir uma disciplina neste momento, mas ele não pode editar ou remover uma disciplina (botões *Edit Class* e *Delete Class* desabilitados).

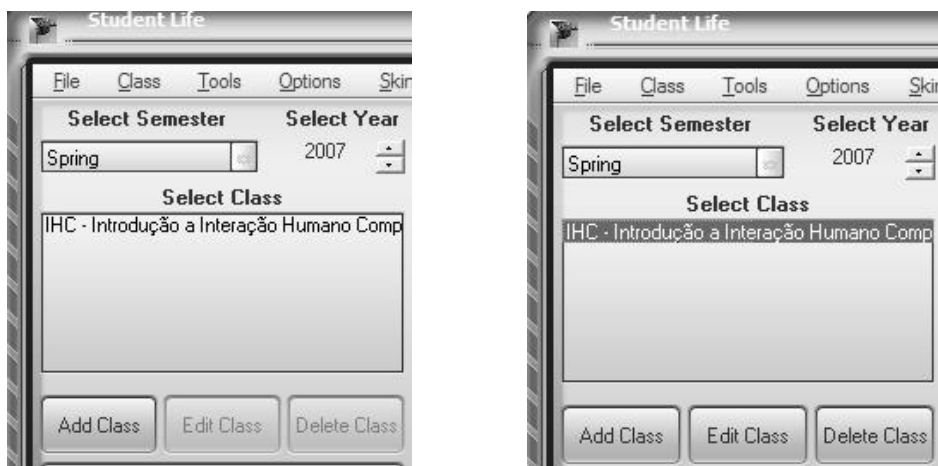


Figura 6. Parte da tela inicial do Student Life. (a) Antes de selecionar disciplina; (b) Depois de selecionar disciplina disponível

Finalmente, **signos dinâmicos** expressam o comportamento do sistema, e só podem ser percebidos quando o usuário interage com o sistema. Por exemplo, ao se selecionar a disciplina de IHC, os botões para editar e remover a disciplina (botões *Edit Class* e *Delete Class*) se tornam habilitados (ver Figura 6(b)). Este comportamento é um signo dinâmico da interface do Student Life.

Para inspecionar a interface, o **Método de Inspeção Semiótica (MIS)** propõe 5 passos a serem seguidos pelo avaliador:

Passo 1: Inspeção dos signos de meta-comunicação presentes na documentação e sistema de ajuda: O principal objetivo neste passo é a reconstrução da meta-mensagem do designer. Para auxiliar nesta etapa o avaliador pode utilizar a paráfrase da meta-mensagem do designer como *template* a ser preenchido:

*“Esta é a **minha interpretação** sobre **quem você é**, o que eu entendi que **você quer ou precisa fazer**, de que **formas prefere fazê-lo e por quê**. Eis, portanto, o sistema que conseqüentemente concebi para você, o qual você **pode ou deve usar assim**, a fim de realizar uma série de objetivos associados com esta (minha) visão.”*

No trecho que examinamos do guia do usuário do Student Life, coletamos informação sobre o “**quem você é**”, que seria um estudante universitário americano. Observe que enquanto “estudante universitário” está explícito no texto do guia, “americano” fica

apenas implícito a partir do conhecimento esperado que o usuário compartilhe com o designer (e.g. GPA/QPA). Da mesma forma, o “**que você quer ou precisa fazer**” é explicitado no uso do sistema para organização das tarefas associadas ao(s) curso(s) e também aspectos da vida social do estudante. Ao ler o guia de usuário e se deparar com o trecho “A maior parte das funções do Student Life pode ser usada de imediato, sem que seja necessário uma configuração prévia” o avaliador poderia entender que o designer também considera que o estudante **gostaria de fazer** isso sem muito trabalho extra. Além disso, toda a parte que explica como fazer cada coisa utilizando o sistema, assim como a definição do estilo de interação selecionado (e.g. manipulação direta x linha de comando), está associada com a expectativa do designer sobre o seu entendimento em relação às preferências do usuário sobre a execução. Finalmente, o **porquê** de o usuário querer usar o sistema aparece na forma de benefícios que ele pode atingir (e.g. guia do usuário: “Gerencie seu curso(s) e GPA/QPA e crie lembretes para praticamente tudo também!”). Muitas vezes questões estratégicas relacionadas com o porquê de utilizar um sistema, ou escolhê-lo em detrimento de outro, aparecem apenas no website ou documentos relacionados com a divulgação e/ou venda do produto (e.g. website do Student Life: “Obtenha melhores resultados a partir da sua experiência na universidade através do aproveitamento do seu tempo¹⁰.”). Vale ressaltar que nesta etapa a mensagem pode ter vários autores além do designer do sistema, como por exemplo profissionais de propaganda ou relações públicas. Esta perspectiva integrada é interessante, já que o usuário real normalmente recebe todas estas mensagens.

Passo 2: Inspeção dos signos estáticos: Neste passo o avaliador inspeciona os signos estáticos da interface, e com base neles faz também a reconstrução da metacomunicação designer-usuário, utilizando o mesmo *template* do passo 1. Note-se que esta meta-mensagem reconstruída idealmente deveria complementar ou elaborar sobre aquela obtida no passo 1. Por exemplo, ao observar os signos estáticos na tela inicial do Student Life (texto dos botões (6) e possibilidade de 12 anos (2)) (Figura 2), eles confirmam para o avaliador a mensagem de que o foco é o aluno universitário americano. O avaliador deve estar atento também para pontos onde os signos estáticos estão inconsistentes com a mensagem obtida pela inspeção dos signos de meta-comunicação. Estas inconsistências devem ser registradas para elaboração sobre elas no Passo 4. Por exemplo, o fato de a tela de calendário não conter nenhum signo estático que indique uma forma rápida de navegar entre semanas ou meses (Figura 7), pode ser inconsistente com a parte da meta-mensagem relacionada com facilidade de organização e gerenciamento de tarefas que permite ganhar tempo.

¹⁰ Tradução livre das autoras do original “Get superior results from your college experience by making the most out of your time.”

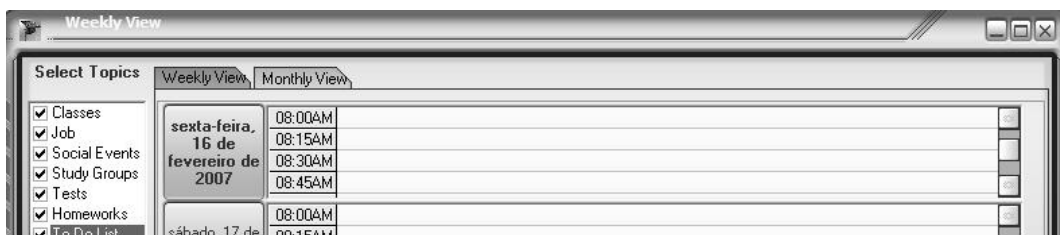


Figura 7 . Tela de Calendário não apresenta signo estático para navegação entre semanas

Passo 3: Inspeção dos signos dinâmicos: Neste passo, o objetivo do avaliador é, novamente, fazer a reconstrução da meta-mensagem usando mais uma vez o *template* utilizado no passo 1, mas desta vez a partir da inspeção dos signos dinâmicos da interface. Caso no processo ele identifique inconsistências entre as meta-mensagens, deve registrá-las. Como vimos, os signos dinâmicos representam o comportamento do sistema, e logo têm um papel extremamente relevante na comunicação designer-usuário. É através deles que os usuários conseguem confirmar ou elucidar o significado antecipado para os signos estáticos. Por exemplo, o usuário imagina que o botão *Add Class* lhe permitirá cadastrar as informações sobre uma disciplina. Ao clicar neste botão e perceber que surge uma janela contendo as informações sobre a disciplina a serem preenchidas, então o seu interpretante para aquele signo estático é ratificado. Um exemplo de uma inconsistência acontece ao se interagir com a tela de cadastro de tarefas mostrada na Figura 8. Ao ver o signo estático marcado para automaticamente acrescentar um lembrete ao cadastro da tarefa (em destaque na Figura 8), o estudante acredita que ao clicar no botão *Add Homework* lhe serão solicitadas informações sobre o lembrete. No entanto, isso não acontece e o usuário fica sem saber se o lembrete foi criado ou não, e, se foi, para quando e com que texto. Assim, o signo dinâmico não está consistente com o interpretante gerado pelo estático, e além disso, ainda poderia estar inconsistente com partes da mensagem transmitidas pelos signos de meta-comunicação relacionados com facilidade de uso, e possibilidade de criar lembretes para tudo.

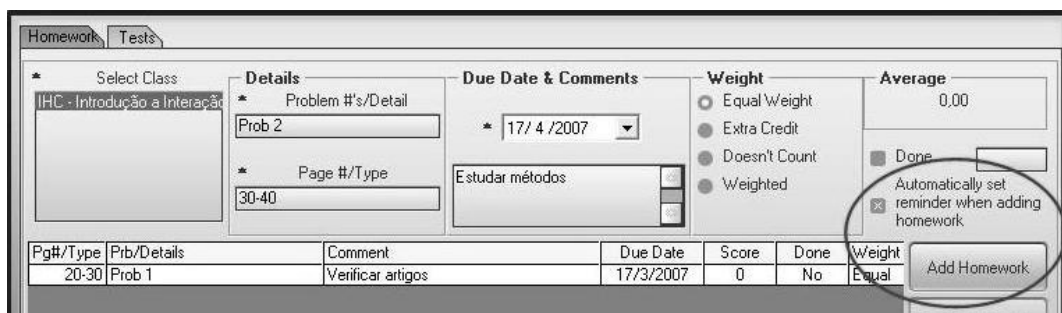


Figura 8. Tela de cadastro de tarefas

Passo 4: Contrastar e comparar as mensagens de meta-comunicação: Analisando os diferentes tipos de signos nos passos 1, 2 e 3, observa-se que eles não têm o mesmo poder de expressão. Isto é o esperado, já que eles pertencem a diferentes sistemas de significação. Assim, enquanto signos meta-comunicativos podem ser expressos em linguagem natural, os signos estáticos se expressam através de elementos como ícones, textos e botões. Desta forma, não se poderia esperar que as meta-mensagens reconstruídas fos-

sem idênticas. No entanto, para a que a comunicação designer-usuário tenha sucesso, é necessário que elas sejam consistentes. Neste passo, o avaliador analisa as inconsistências identificadas nos passos anteriores. Além disso, ele deve intencionalmente procurar rever as meta-mensagens buscando inconsistências e ambigüidades, e explorar a possibilidade de o usuário atribuir significados contraditórios aos signos que constituem as mensagens em cada um dos níveis. Para auxiliá-lo nesta tarefa, ele pode fazer uso das seguintes perguntas:

1. É plausível que o usuário interprete este signo ou mensagem de forma diferente? Como? Por quê?
2. Esta interpretação está consistente com a intenção de design?
3. A cadeia interpretativa me lembra outras cadeias interpretativas que gerei durante a inspeção? Quais? Por quê?
4. As classes de signos estáticos e dinâmicos¹¹ podem ser identificadas pela análise? Quais?
5. Existem signos estáticos ou dinâmicos que estão na classe errada de acordo com as classes propostas em 4? Isto pode afetar a comunicação com o sistema? Como?

Outras perguntas poderiam ser adicionadas a estas para auxiliarem o avaliador na identificação de inconsistências. A interface de um sistema compõe um sistema de significação único (composto pelos signos dos 3 níveis) projetado para transmitir suas qualidades e características únicas. Desta forma, consistência e regularidade são fundamentais para evocar padrões ou significações que sejam familiares aos usuários. Assim, vale ressaltar que, em algumas situações, ambigüidades não necessariamente configuram um problema, já que as pessoas têm costume em lidar com elas na linguagem natural. Quando o sistema lida com a ambigüidade de forma clara para o usuário, então não é um problema, como por exemplo a função “Salvar” que em vários sistemas pode funcionar como “Salvar como...” na 1ª vez em que é acionada.

Passo 5 – Apreciando a qualidade da meta-comunicação: Este é o último passo do método, no qual o avaliador produz um relatório contendo sua apreciação final resultante de sua inspeção. O relatório deve ser composto pelas seguintes partes:

1. Uma breve descrição do método para auxiliar o leitor a entender a apreciação feita;
2. Os critérios utilizados para selecionar porções do sistema a serem inspecionadas (quando a inspeção não examinar todo o sistema)
3. Para cada um dos níveis de signos inspecionados, descrever:
 - a. A identificação dos signos relevantes (listar e justificar sua relevância)
 - b. A identificação das classes de signos utilizadas
 - c. Uma versão unificada da meta-comunicação designer-usuário

¹¹ As classes de signos se referem aos tipos de signos usados em cada nível. Por exemplo, as classes de signos estáticos poderiam constar de elementos como botões, menus e apontadores; enquanto as de signos dinâmicos referenciam tipos de comportamento, como abrir uma janela ou disparar a execução de um *applet*.

4. A apresentação e explicação sobre os problemas de comunicabilidade encontrados que possam dificultar ou prevenir o usuário de entender a mensagem pretendida pelo designer, e interagir produtivamente com o artefato.

A Figura 9 apresenta uma visão geral do Método de Inspeção de Semiótica apresentado. Antes de aplicar o método, algumas considerações devem ser feitas pelo avaliador. Primeiramente, como o MIS requer um exame detalhado da interface nos três níveis de signos, dependendo do tamanho do sistema ou do objetivo da avaliação pode não ser viável ou necessária a inspeção de todo o sistema. Neste caso, o avaliador (ou o designer) deve definir qual parte do sistema deve ser avaliada, por exemplo uma parte crítica para o uso do sistema ou cuja meta-comunicação seja mais elaborada.

Em métodos de inspeção, o avaliador fala em nome do usuário. Assim, podemos associar o seu papel ao de um advogado que, conhecendo a visão do seu cliente e as leis, fala por ele em processos jurídicos. O avaliador deve então conhecer a visão do usuário e ser especialista em EngSem para então falar por ele, identificando os problemas de comunicabilidade que ele poderia vivenciar durante a interação. Para isso, o avaliador deve adotar uma perspectiva única para o usuário para uma inspeção (e.g. em um sistema educacional, a perspectiva de usuários alunos e professores é distinta). Além disso, a intenção do usuário e seu contexto são fundamentais na definição de caminhos plausíveis de interpretação, e logo devem ser fornecidos ao avaliador de antemão por meio de cenários¹².

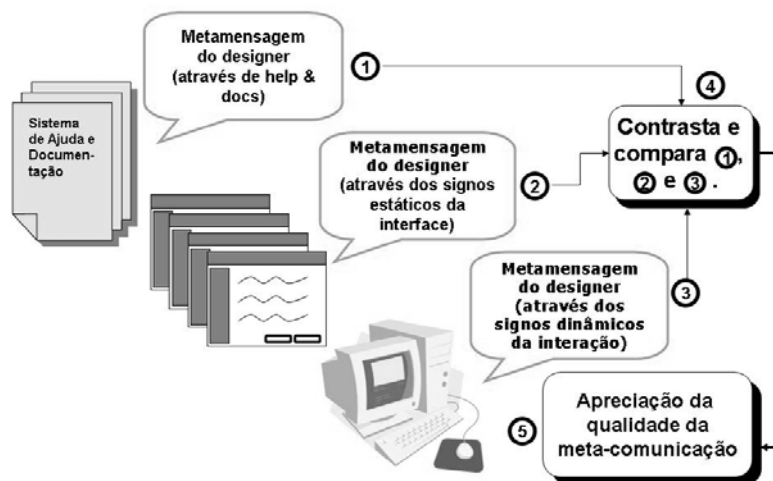


Figura 9. Visão geral do Método de Inspeção Semiótica [de Souza et al. 2006]

Finalmente, alguns métodos de inspeção requerem a participação de mais de um avaliador para que seja possível consolidar o resultado da avaliação. Sob o ponto de vista da EngSem, cada avaliador gera um conjunto de possíveis interpretantes sobre a

¹² Cenários são narrativas textuais plausíveis contextualizadas e detalhadas que descrevem determinada situação [Carroll et al. 1994]. Na seção 6.4.1 este conceito será apresentado em mais detalhes.

meta-comunicação. Não há necessidade de mais de um avaliador, já que todos os possíveis caminhos interpretativos que possam ser gerados por um ou mais avaliadores são plausíveis. No entanto, a análise por mais de um avaliador permite que se identifique alguns caminhos interpretativos mais salientes e se enriqueça o relatório com visões distintas. Em casos de sistemas que envolvem usuários em diferentes papéis (e.g. professor e aluno), cada avaliador poderia ficar responsável por inspecionar a interface sob a perspectiva de um dos papéis.

6.3.2. MAC – O Método de Avaliação de Comunicabilidade

O **Método de Avaliação de Comunicabilidade (MAC)** [Prates et al. 2000, de Souza 2005], diferente do MIS, é um método que envolve usuários na avaliação. A apreciação da comunicabilidade do sistema no MAC é feita a partir da comunicação usuário-sistema, com base na qual o avaliador simula a comunicação do usuário para o designer sobre a meta-mensagem. Como veremos, isto é feito utilizando um conjunto de expressões para identificar as rupturas de comunicação com o sistema vivenciadas pelo usuário, como se o avaliador “colocasse palavras na boca do usuário”. O usuário participa na execução de tarefas utilizando o sistema em um ambiente controlado (i.e., ambiente em que o usuário poderá se concentrar no sistema, sem correr o risco de ser interrompido ou ter sua atenção desviada das atividades do teste). Em etapa posterior, o avaliador identifica os problemas de comunicabilidade, com base na sua análise da experiência do usuário durante sua interação com o sistema.

A etapa de **coleta de dados** envolve o planejamento da avaliação e a geração do material necessário, seguidos da participação do usuário em executar as tarefas previstas no laboratório e uma entrevista (opcional) com o usuário sobre o que ocorreu. A preparação para a coleta deve ser minuciosa para garantir a coleta de dados úteis para análise, ou seja, que permitam observar aquilo que se deseja. Neste capítulo apresentamos brevemente os passos envolvidos nesta preparação (para uma descrição detalhada ver [Prates e Barbosa 2003 ou Preece et al. 2002]):

1. Determinação do objetivo do teste: Na avaliação com o usuário, não é possível se avaliar todo o sistema, e logo o foco do teste deve ser definido. Para isso, o avaliador deve levar em consideração os objetivos de design e os pontos críticos do sistema. Como a propriedade em questão é a comunicabilidade, objetivos do sistema estão relacionados com a intenção de comunicação do designer, como por exemplo “O usuário do Student Life consegue entender que as informações mínimas a serem fornecidas para utilizar o sistema são sobre o seu curso e disciplinas?”¹³. Pontos críticos normalmente são as partes do sistema que o designer prevê que sejam utilizadas com maior frequência (e.g. cadastro e visualização de tarefas associadas à disciplinas), fundamentais para o uso do sistema (e.g. cadastro de disciplinas), ou ainda cuja comunicação o designer ache elaborada (e.g. contém ambigüidades que devem ser compreendidas pelo usuário).

2. Seleção das tarefas para teste: Com base nos objetivos específicos do teste, o avaliador deve definir quais tarefas deverão ser executadas pelos usuários durante a sua

¹³ Alguns objetivos podem não ser passíveis de apreciação no MAC. Por exemplo, a questão “O aluno consegue terminar com rapidez o cadastro de uma disciplina?” requer uma avaliação de usabilidade.

interação com o sistema. As tarefas devem ser típicas e realistas, e devem permitir a coleta de dados sobre os objetivos determinados no passo 1. Alguns cuidados devem ser tomados durante a definição da tarefa: (1) especificá-las bem, para que toda a informação necessária para sua execução seja fornecida ao usuário, ou seja, parte do conhecimento prévio esperado do usuário; (2) o teste normalmente não deve durar mais do que uma hora, e a execução de cada tarefa não mais do que 20 minutos; (3) para cada tarefa deve ser gerado um cenário que a contextualize em uma situação de uso plausível, permitindo assim ao participante se colocar em um contexto e ser capaz de se comportar de acordo com motivações e intenções reais.

3. Seleção dos participantes: A seleção dos (potenciais) usuários a participarem da avaliação deve ser feita com base no perfil de usuários a que o sistema se destina. Por exemplo, para o Student Life a avaliação deveria ser feita com alunos universitários, e não com profissionais atuando no mercado de trabalho nem alunos do ensino fundamental. Normalmente é recomendado que de 5 a 8 usuários participem do teste [Prates e Barbosa 2003, Nielsen 2000]. Quando o sistema é destinado a mais de um perfil de usuário (e.g. alunos e professores) então deve-se fazer uma avaliação envolvendo cada um destes perfis, e neste caso pode-se envolver menos usuários (não menos que 3) de cada perfil [Nielsen 2000].

4. Considerações sobre aspectos éticos: No Brasil, a resolução 196/96 do Conselho Nacional de Saúde [CNS 196] regulamenta pesquisas envolvendo pessoas, e se aplica à avaliação de sistemas interativos com usuários [Leitão e Dias-Romão 2003]. Para esses sistemas, os principais aspectos desta resolução são: o caráter voluntário e consentido da participação do usuário no teste; a preservação do anonimato dos participantes; a proteção de grupos vulneráveis (e.g. crianças e alunos); a garantia de bem-estar dos participantes e de seu direito de interromper sua participação a qualquer momento durante a avaliação. Além disso, o participante deve dar seu consentimento por escrito, através da assinatura de um documento que especifique as condições acordadas sobre sua participação, assinado também pelo avaliador.

5. Geração do material impresso para uso durante avaliação: Depois do planejamento, o avaliador deve gerar o material impresso que será utilizado durante o teste: o questionário para seleção de participantes do teste (caso haja mais voluntários do que o necessário); questionário pré-teste (para coletar informações sobre características pessoais e conhecimento prévio dos participantes); *scripts* de apresentação e explicação do processo de teste aos usuários (para garantir que todos terão as mesmas informações); formulários de consentimento de participação (contendo as condições da avaliação às quais o usuário deve consentir); cenários de apresentação do sistema (descrevendo o que se quer apresentar do sistema ao usuário) e das tarefas (descrevendo e contextualizando cada tarefa em uma situação de uso); formulário de acompanhamento de teste pelo avaliador (contendo a descrição do teste e a identificação do usuário para facilitar as anotações do avaliador durante a execução dos testes); roteiros de entrevista ou questionários para coleta pós-teste das opiniões do usuário sobre o sistema.

6. Execução do Teste Piloto: O teste piloto permite ao avaliador apreciar a qualidade do material gerado. Para isso, executa-se a avaliação planejada, de preferência com pessoas do perfil desejado, e observa-se se o participante é capaz de entender todo o material apresentado, se o tempo de execução do teste está de acordo com o tempo previsto, e se as tarefas propostas geram indicadores relevantes sobre o objetivo específico

da avaliação a que estão associadas. Assim, garante-se que os dados coletados permitirão de fato avaliar os aspectos desejados do sistema, e não causarão perda de dados ou, no pior caso, invalidação do teste. Com base na experiência obtida durante o teste piloto, faz-se ajustes no material e, se necessário, conduz-se novos testes-piloto. Os dados coletados durante os testes-piloto são usados apenas para avaliação do material, e não devem ser considerados para análise e apreciação do sistema.

Uma vez terminada a etapa de preparação, segue-se então para a **execução dos testes** com os usuários selecionados. Isto deve ser feito em ambiente controlado, idealmente um laboratório de testes contendo uma sala para os usuários e outra de observação para os avaliadores. O vidro para observação deve ser espelhado para deixar o usuário mais à vontade. Da sala de observação, o avaliador deve ser capaz de acompanhar a interação do usuário (e.g. através de um monitor na sala de observação ligado ao computador usado para teste), ver suas reações durante a interação e ouvir seus comentários¹⁴. O MAC requer que toda a interação do usuário com o sistema de interação seja gravada através da utilização de um software de captura de tela. Esta gravação é fundamental para a etapa de análise. Gravação de vídeo ou áudio do teste são opcionais, mas podem ser úteis para se tirar ambigüidade ou se rever comentários ou expressões do usuário. Além disso, é recomendável que o avaliador faça anotações durante a execução do teste de ações dos usuários que possam auxiliar posteriormente a análise, como por exemplo comentários espontâneos sobre suas dificuldades ou dúvidas, indicações de sua intenção durante um dado momento, ou pontos que o avaliador já identifica como ambíguos ou interessantes a serem discutidos com o usuário durante a entrevista. Ao fim da execução, passa-se para a entrevista com o usuário sobre a sua experiência sobre os testes. Embora a entrevista seja opcional, ela é bastante útil e portanto recomendada.

Executados os testes, o avaliador passa então à etapa de **análise dos dados** coletados. O MAC divide esta etapa em 3 passos: (1) Etiquetagem: identifica-se no filme da interação as rupturas de comunicação e associa-se a cada uma delas uma expressão que a caracterize; (2) Interpretação da Etiquetagem: com base na etiquetagem identifica-se classes de problema de comunicação designer-usuário ou interação; (3) Geração do Perfil Semiótico: faz-se a reconstrução da meta-mensagem designer-usuário. A seguir apresentamos em detalhe cada um destes passos.

Etiquetagem

Neste passo, o avaliador vê o filme gravado da interação do usuário com o sistema com o objetivo de identificar rupturas na comunicação, ou seja, pontos em que o usuário não foi capaz de entender a comunicação sendo feita pelo designer através da interface. O avaliador deve fazer uso de suas anotações durante o teste e entrevista para resolver ambigüidades que porventura surjam. Caso tenha gravado o vídeo e/ou áudio do teste pode fazer uso destes também. A cada ruptura identificada, o avaliador seleciona a expressão que a caracteriza (a partir de um conjunto de 13 expressões). As expressões têm por objetivo serem expressões naturais que seriam plausíveis de serem manifestadas pelo usuário (e por vezes o são) durante a interação. Desta forma, o efeito de associar uma determinada expressão a uma seqüência de interação que representa uma ruptura é o de

¹⁴ Se o avaliador não tem um laboratório disponível, ele pode criar um ambiente controlado usando um espaço de acesso restrito, que lhe permita observar o usuário sem interferir na execução da tarefa.

simular a comunicação do usuário para o designer sobre a interface. Por exemplo, se o usuário procura na interface como executar determinada ação, o avaliador associaria a esta ruptura a expressão “Cadê?”. A seguir apresentamos na Tabela 1 cada uma das expressões do conjunto disponível ao avaliador.

Tabela 1. Descrição das expressões para etiquetagem do MAC.

<p>Cadê? Ocorre quando o usuário sabe a operação que deseja executar mas não a encontra de imediato na interface. O principal sintoma desta ruptura é a procura pela operação na interface, inspecionando diversos elementos de interface sem ativá-los (e.g. abrindo e fechando menus e submenus ou passando o cursor de mouse sobre botões para ver a dica associada).</p>
<p>Ué, o que houve? Identificado quando o usuário não percebe a resposta dada pelo sistema a uma ação sua (e.g. a resposta é muito sutil, ou mesmo inexistente) ou não é capaz de entendê-la. Os sintomas típicos incluem repetir a ação ou buscar uma forma alternativa de alcançar o resultado esperado.</p>
<p>E agora? O usuário não sabe o que fazer e procura descobrir qual é o seu próximo passo. Os sintomas incluem vagar com o cursor do mouse sobre a tela e iniciar um caminho aleatório de interação.</p>
<p>Epa! O usuário realiza uma ação indesejada e, ao perceber isto, imediatamente desfaz a ação. Os sintomas incluem o acionamento imediato do <i>Undo</i> ou o cancelamento de um quadro de diálogo aberto indevidamente. Observe-se que este 2º sintoma poderia ser visto também como parte de uma busca (“Cadê?”). O “Epa!” se diferencia por ser uma única ação e não parte de uma sequência maior.</p>
<p>Assim não dá. O usuário realiza uma sequência de ações e acredita estar seguindo por um caminho improdutivo, interrompendo-o e cancelando-o. Os sintomas incluem o acionamento de <i>Undo</i> repetidas vezes, a interrupção de um caminho guiado pelo sistema ou ainda o cancelamento quadros de diálogos relacionados. A diferença entre o “Assim não dá.” e o “Epa!” é que o primeiro envolve várias passos, enquanto o “Epa!” envolve apenas um.</p>
<p>Onde estou? O usuário tenta efetuar operações que não são apropriadas para o contexto em que se encontra, mas o seriam para outros contextos do sistema, indicando uma confusão em relação ao contexto com o qual está interagindo (e.g. tenta editar um elemento da interface disponível apenas para visualização). Um sintoma típico é desfazer a ação incorreta e mudar em seguida para o contexto desejado.</p>
<p>O que é isto? Ocorre quando o usuário não sabe o que significa um elemento de interface. O principal sintoma consiste em deixar o cursor do mouse sobre o elemento por alguns instantes, esperando que uma dica seja apresentada. Outro sintoma é quando o usuário abre menus e submenus ou quadros de diálogos para ver a que se referem. Observe que este sintoma pode acontecer também para a expressão “Cadê?”. A diferença está na intenção do usuário. Quando o usuário está procurando algo então este mesmo sintoma seria um “Cadê?”, se está explorando a interface, então seria “O que é isto?”.</p>
<p>Por que não funciona? A operação efetuada não produz o resultado esperado, e o usuário não entende o por quê (ou não se conforma com o fato). O sintoma é quando o usuário executa uma ação (ou sequência de ações), percebe que não obteve o resultado desejado e então repete sua ação na tentativa de identificar a causa de não ter atingido o efeito esperado e corrigi-la.</p>
<p>Socorro! O usuário não consegue realizar sua tarefa através da exploração da interface e recorre a signos de meta-comunicação para conseguir entender e dar continuidade à sua tarefa. O sintoma é recorrer aos sistemas de ajuda (e.g. agentes automáticos ao qual se pode fazer uma pergunta, ou funções de ajuda que apresentam explicações a elementos da interface em um contexto), documentação (eletrônica ou impressa), ou mesmo pedir explicação a outra pessoa.</p>

Vai de outro jeito. O usuário não consegue realizar a tarefa da forma prevista como preferencial pelo designer (seja porque não sabe que ela está disponível, ou não sabe como utilizá-la), e resolve seguir outro caminho, geralmente mais longo ou complicado. Cabe ao avaliador determinar, se possível junto ao designer, qual é a forma preferencial de execução da tarefa. Normalmente, as formas mais salientes na interface são as consideradas preferenciais. O sintoma é a tentativa frustrada de executar uma ação utilizando a forma preferencial, seguida da adoção de uma solução alternativa, ou mesmo a direta da solução alternativa, sem dar sinais de conhecimento da existência da forma preferencial.

Não, obrigado. O usuário conhece a solução preferencial do designer, mas opta explicitamente por uma outra forma de interação. O sintoma é o usuário utilizar a ação preferencial (ou demonstrar conhecê-la) e depois utilizar uma ou mais formas alternativas para se alcançar o mesmo resultado.

Para mim está bom... O usuário acha equivocadamente que concluiu uma tarefa com sucesso. O sintoma típico é encerrar a tarefa e indicar na entrevista ou no questionário pós-teste que a tarefa foi realizada com sucesso. O observador, no entanto, sabe que se trata de um engano, provavelmente causado por uma falha de resposta do sistema ou modo de visualização inadequado para a tarefa atual.

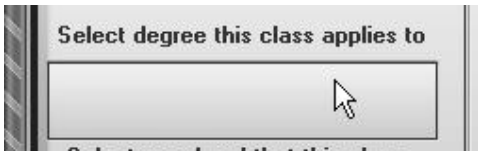
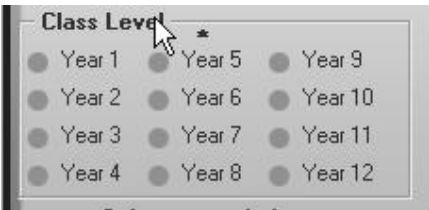



Desisto. O usuário não consegue fazer a tarefa e desiste. O sintoma é a interrupção prematura da tarefa. A causa pode ser falta de conhecimento, tempo, paciência, informação necessária, etc.

Para exemplificar o passo de etiquetagem, retomemos o exemplo apresentado na Figura 4, descrevendo as rupturas experimentadas por um aluno, e identifiquemos a expressão associada a cada uma. A Figura 10 apresenta a tela com a qual o aluno está interagindo, e a Tabela 2 descreve as ações que identificam a ruptura e a etiqueta associada.

Figura 10. Tela para incluir disciplina com a qual o aluno está interagindo.

Tabela 2. Exemplo de etiquetagem na tentativa de incluir uma disciplina no Student Life.

Descrição da sequência de ações	Etiqueta associada
---------------------------------	--------------------

<p>Aluno vê a lista de cursos vazia e tenta clicar no título da lista, ou com o botão direito do mouse sobre a lista para inserir um curso.</p> 	<p>Por que não funciona?</p>
<p>Aluno vê os anos a que o nível da disciplina pode ser associado e passa o cursor sobre o título <i>Class Level</i>, ou <i>Year</i> para ver se obtém alguma dica do que significa todos aqueles anos.</p> 	<p>O que é isto?</p>
<p>O aluno acessa o guia de usuário para tentar descobrir o que significam os 12 anos.</p> 	<p>Socorro!</p>
<p>O aluno tenta acionar o botão para salvar <i>Save and Exit</i>, mas vê que ele está desabilitado.</p> 	<p>Ué o que houve?</p>
<p>O aluno verifica os campos marcados como obrigatórios e percorre a tela tentando descobrir como abilitar o botão <i>Save and Exit</i>.</p>	<p>E agora?</p>
<p>O aluno não consegue descobrir e clica no botão <i>Cancel</i>.</p> 	<p>Assim não dá.</p>

Interpretação da Etiquetagem

Nesta etapa, o avaliador tabula os problemas identificados e sua interpretação, que depende da sua experiência e conhecimento em EngSem. Alguns aspectos devem ser considerados durante a interpretação para permitir ao avaliador identificar os principais problemas da meta-comunicação [de Souza 2005]: (1) classificação das expressões que caracterizam a ruptura quanto ao tipo de falha que representam na comunicação entre o sistema (enquanto preposto do designer) e usuário; (2) a frequência e contexto em que ocorrem as rupturas; (3) identificação de padrões de seqüências de expressões; (4) o nível da ação em que ocorre a ruptura.

Os tipos de falha são definidos em função da relação entre a intenção de uma comunicação e o efeito que ela causa. Quando a intenção de uma comunicação é consistente com o efeito que ela causa, então a comunicação é de sucesso. No entanto, se existem inconsistências, ela apresenta alguma falha, que pode ser ou não percebida pelos usuários. Estas falhas podem ser definidas como completas, parciais ou temporárias.

Falhas completas acontecem quando a intenção da comunicação e seu efeito são inconsistentes. **Falhas parciais** ocorrem quando parte do efeito pretendido da comunicação não é atingido. Finalmente, **falhas temporárias** são aquelas que ocorrem na expressão ou intenção de um ato comunicativo entre usuário e sistema, e que são percebidas pelo usuário que tenta então superá-las. É fácil ver que falhas completas são mais graves que as parciais ou temporárias, uma vez que representam o insucesso da comunicação designer-usuário. A Tabela 3 mostra a classificação das etiquetas em relação aos tipos de falhas na comunicação designer-usuário através da interface.

Tabela 3. Classificação de etiquetas em relação ao tipo de falhas

Tipo de Falha	Aspecto Semiótico	Característica Específica	Expressão
Completas		Usuário percebe	Desisto.
		Usuário não percebe	Para mim está bom...
Parciais		Usuário entende solução proposta	Não, obrigado.
		Usuário não entende solução proposta	Vai de outro jeito.
Temporárias	Semiose do usuário é interrompida temporariamente	(a) Não encontra expressão apropriada para sua intenção	Cadê?
		(b) Não percebe ou entende expressão do preposto	Ué o que houve?
		(c) Não consegue formular sua intenção	E agora?
	Usuário percebe que seu ato comunicativo não foi bem sucedido.	(a) "Dito" no contexto errado	Onde estou?
		(b) A expressão utilizada está errada	Epa!
		(c) Vários passos da comunicação não chegaram ao resultado desejado.	Assim não dá.
	Usuário procura esclarecer ato comunicativo	Através de metacomunicação implícita	O que é isso?
		Através de metacomunicação explícita	Socorro!

	feito pelo sistema	Através de repetidos testes de hipóteses sobre o significado da comunicação.	Por que não funciona?
--	--------------------	--	-----------------------

Além do tipo de falha a que cada etiqueta está associada, a frequência e contexto em que elas acontecem também são relevantes. Etiquetas que ocorrem com maior frequência apontam para um problema recorrente na metacomunicação designer-usuário. Por exemplo, uma grande incidência de “Epa!” pode indicar um excesso de ambigüidade na meta-mensagem, ou uma grande incidência de “O que é isto?” para uma baixa interseção entre o sistema de significação adotado pelo designer e o conhecido pelo usuário. O contexto em que as etiquetas acontecem fornece informações relevantes sobre a falha em indicar a inconsistência de determinados caminhos interpretativos (e.g. falha em indicar a diferença entre contextos ao usuário) ou mesmo apontar para o designer caminhos interpretativos que deveriam ter sido considerados (e.g. ações que poderiam ser consideradas válidas em um determinado contexto).

Padrões de seqüências de expressões que se repetem na etiquetagem podem fornecer informações relevantes sobre os caminhos interpretativos inconsistentes que o usuário tem seguido. Isto pode ser um indicador de falhas na expressão sobre caminhos interpretativos consistentes ou desejáveis. Por exemplo, padrões de “Cadê?” seguido por um “Vai de outro jeito.” apontam para a tentativa do usuário de comunicar algo ao sistema, e a adoção de um caminho que não é o ótimo. O fato de este padrão se repetir mostra que o usuário pode estar fazendo um uso sub-ótimo da solução como um todo.

O nível em que a ação do usuário se dá pode ser classificado como **operacional** – uma ação individual; **tático** – uma seqüência de ações necessárias para se atingir um objetivo; e **estratégico** – relativo à formulação do problema e de sua solução. As rupturas relativas a ações são normalmente mais fáceis de superar ou resolver do que aquelas nos níveis táticos e estratégicos. No nível tático, elas podem indicar problemas nos métodos codificados, ou seja, como o designer acredita que o usuário deveria resolver seu problema. Rupturas no nível estratégico são graves, pois indicam falhas completas na comunicação designer-usuário. Além disso, rupturas em um nível podem gerar rupturas em outro. Por exemplo, porque o usuário não entende que uma determinada seqüência de ações (nível tático), ele pode achar que a solução do designer não atende suas necessidades (nível estratégico), declinando-a como um todo. A análise das rupturas nos níveis operacional, tático e estratégico pode identificar também alguns aspectos relativos a aspectos da usabilidade do sistema [de Souza et al. 2000, de Souza 2005]. Embora usabilidade esteja fora do foco da EngSem, esta visão pode ser de interesse na prática e pode enriquecer a análise dos problemas de comunicabilidade. Nesta mesma direção, a análise pode ser enriquecida fazendo-se a associação das rupturas de comunicação com dimensões ou diretrizes propostas por outras teorias ou métodos [de Souza, 2005].

Ao final do passo de interpretação, espera-se que o avaliador tenha sido capaz de identificar os principais problemas na meta-comunicação designer-usuário. Em muitos casos a própria etiquetagem já pode oferecer ao designer indicadores também sobre uma possível solução.

Geração do Perfil Semiótico

O 3º e último passo da análise é a reconstrução da meta-comunicação transmitida do designer tal como potencialmente percebida pelo usuário. Para isso, o avaliador pode mais uma vez fazer uso do *template*:

*“Esta é a **minha interpretação** sobre **quem você é**, o que eu entendi que **você quer ou precisa fazer**, de que **formas prefere fazê-lo e por quê**. Eis, portanto, o sistema que conseqüentemente concebi para você, o qual você **pode ou deve usar assim**, a fim de realizar uma série de objetivos associados com esta (minha) visão.”*

À medida que ele preenche o *template*, ele deve endereçar os desencontros entre o que o designer pretendia dizer e as evidências de como os usuários estão interpretando o que ele diz. Tais desencontros terão sido identificados.

* * *

Esta seção apresentou os dois métodos fundamentados na EngSem para apreciação da comunicabilidade de um sistema interativo. Do ponto de vista teórico, a principal diferença da apreciação feita por eles é que o MIS se concentra na mensagem enviada pelo designer, enquanto que o MAC se concentra em como esta mensagem está sendo recebida e entendida pelo usuário. Em outras palavras, o MIS avalia a emissão da meta-mensagem designer-usuário e, o MAC, a sua recepção [de Souza et al. 2006].

6.4. Projeto de Interação

A seção anterior dedicou-se a métodos de avaliação da EngSem, uma das atividades essenciais no ciclo de vida de design da interação. Nesta seção, apresentamos um modelo simples de ciclo de vida de design da interação, bem como modelos e representações de design adaptados ou concebidos com base na EngSem. Vale observar que esses modelos também constituem ferramentas epistêmicas. Ainda em linha com Schön (1983), objetivamos aqui apoiar e motivar a reflexão do designer sobre a meta-mensagem sendo elaborada, durante todo o processo de sua elaboração.

Segundo Preece e co-autoras, o design da interação envolve quatro atividades principais: identificação de necessidades e estabelecimento de requisitos; desenvolvimento de designs alternativos que satisfaçam esses requisitos; construção de versões interativas; e avaliação [Preece et al. 2002] (Figura 11).

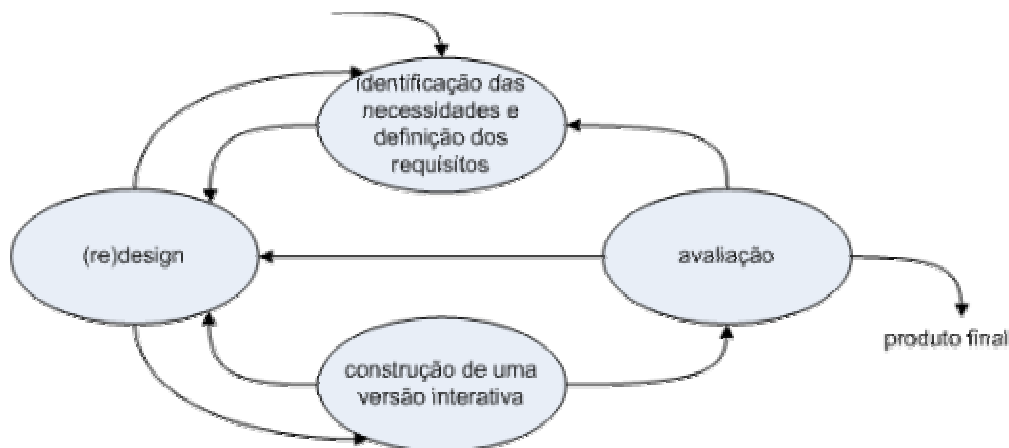


Figura 11. Modelo de ciclo de vida para o design da interação [Preece et al. 2002].

A atividade de **identificação das necessidades e definição dos requisitos** visa descobrir quem é o usuário, o que ele faz, como o faz (ou preferiria fazer), em que ambiente ou contexto de uso, e por quê. Isto é importante para que seja possível definir como o sistema deve apoiá-lo, o escopo do sistema e as condições e restrições que devem ser respeitadas. A análise em IHC difere fundamentalmente da análise de sistemas tradicional: se concentra em gerar insumos para o projeto da interação e da interface de usuário, em vez de modelos e estruturas de dados e funções. As representações mais utilizadas nesta etapa são cenários e modelos de tarefa. Na EngSem, nos concentramos, nesta atividade, em produzir insumos para a construção da meta-mensagem designer-usuário, que será elaborada e detalhada na atividade de design e redesign.

Na atividade de **design (ou redesign)**, o projetista elabora o design conceitual e o físico. O design conceitual define o que o produto deve fazer, como deve se comportar e se apresentar. Já o design da interface envolve o detalhamento de todos os elementos da interface do produto, incluindo cores, imagens, menus e ícones, bem como a organização desses elementos em cada tela ou unidade de apresentação. A todo momento são consideradas alternativas, tanto no design conceitual quanto no físico. Do ponto de vista da EngSem, nesta atividade o designer elabora a meta-mensagem a ser

transmitida aos usuários, projetando sistemas de significação que lhes forneçam pistas sobre como interpretá-la. Neste curso, nos concentramos no design conceitual, através de modelos de interação que representem a meta-mensagem designer-usuário. Em particular, utilizaremos a MoLIC, uma linguagem para modelagem de interação ancorada na EngSem.

Segundo Preece e co-autoras, a maneira mais razoável para os usuários avaliarem um design é interagindo com ele. Isto requer a **construção de uma versão interativa** dos designs, mas não significa necessariamente que uma versão em software seja necessária. Protótipos em papel são rápidos e baratos de se construir, e podem ser avaliados, por exemplo, através de *role playing*.

A atividade de **avaliação** visa determinar a qualidade do design ou produto. Pode-se utilizar uma variedade de critérios, como o número de erros cometidos, o quanto satisfaz os requisitos, e o quanto é atraente, entre outros. Como visto na seção 6.3, a EngSem se concentra na avaliação da comunicabilidade.

Nas subseções seguintes, apresentamos como cenários e modelos de tarefas podem ser utilizados (com algumas adaptações) em projetos de design pautados na EngSem, e introduzimos a modelagem da interação como uma conversa e a modelagem do sistema de ajuda, elaboradas como produto da teoria.

6.4.1 Cenários

Um **cenário** é uma narrativa, textual ou pictórica, concreta, rica em detalhes contextuais, de uma situação de uso da aplicação, envolvendo usuários, processos e dados reais ou potenciais. Um cenário típico é uma história sobre pessoas realizando uma atividade. Rosson & Carroll (2002) descrevem os seguintes elementos característicos de um cenário: ambiente/contexto; atores; objetivos; planos; avaliação; ações; e eventos. A identificação desses elementos nos cenários também contribui para torná-los mais completos e promove a reflexão do designer sobre a situação de uso sendo descrita. **O ambiente ou contexto** envolve detalhes circunstanciais que motivam ou explicam objetivos, ações e reações dos atores do cenário. Os **atores** são as pessoas que interagem com o computador ou com o contexto. A descrição de um ator no cenário deve incluir as características pessoais que forem relevantes ao cenário. Os **objetivos** são os efeitos na situação que motivam as ações que os atores realizam. Já um **plano** é uma atividade mental direcionada para converter um objetivo em comportamentos. Um comportamento observável é uma **ação**. Já um **evento** é uma ação ou reação externa produzida pelo computador ou outras características do ambiente; algumas dessas podem estar ocultadas dos atores, mas ser importantes para o cenário. Por sua vez, a **avaliação** é a atividade mental direcionada para interpretar características de uma situação.

Alguns dos objetivos do uso de cenários são esclarecer questões complexas e explorar decisões alternativas de projeto. Por serem ricos em contextualização, os cenários permitem explorar com detalhes os impactos da tecnologia a ser projetada nas atividades e nos processos de trabalho dos usuários. As técnicas baseadas em cenários vêm ganhando grande aceitação por parte dos projetistas e seus clientes. Isso se deve principalmente à natureza do problema de design de software, que não é uma forma fácil ou rotineira de resolução de problemas [Carroll 2000]. Os problemas de design costumam ser especificados de forma incompleta, as soluções não são conhecidas *a priori*, envol-

vem *trade-offs* entre diversos elementos interdependentes e requerem uma diversidade de conhecimento e habilidades. Além disso, o design de artefatos interativos causa transformações no mundo que alteram as possibilidades de atividade e experiência humanas, com frequência extrapolando as fronteiras das soluções de design originais pretendidas. O design de IHC baseado em cenários busca gerenciar a complexidade da resolução de problemas de design através de uma **concretização de uma situação de uso** [Carroll 2000]. Carroll acredita que descrever a solução desde o início utilizando diagramas e especificações é muito bom para documentar e transmitir a solução para quem vai implementá-la. No entanto, essas representações abstraem a riqueza do uso do sistema em situações reais, e podem se tornar um obstáculo tanto para uma exploração mais ampla do espaço de design quanto para o envolvimento e participação dos clientes e futuros usuários do sistema no processo de design.

Os cenários ajudam designers e analistas a se concentrarem nas suposições sobre pessoas e suas tarefas. Cenários podem ser representados como protótipos, através do uso de *storyboards*, vídeos, ou ferramentas de prototipação rápida. Eles podem ser utilizados em todas as fases do processo de desenvolvimento de sistemas interativos: durante a fase de elicitação e análise de requisitos, para capturar os requisitos e explorar a introdução de tecnologia; durante a fase de design e especificação, para explorar soluções alternativas e basear decisões de design; durante a implementação, para verificar se a implementação está de acordo com os cenários anteriores; e durante os testes, para apoiar os usuários na validação do sistema.

Apesar de ricos em detalhes e contextualizados, os cenários, quando usados na fase de análise de necessidades e definição de requisitos, não devem conter detalhes da **interface** propriamente dita, como textos e rótulos, tipos de *widgets* utilizados, etc. Pretende-se com isto evitar um comprometimento precoce dos *designers* com uma solução de interface a ser adotada, o que dificultaria a exploração de soluções alternativas que emergissem da modelagem de tarefas e do projeto cuidadoso da interação. A Figura 12 ilustra um cenário de uso hipotético do Student Life, elaborado pelo designer para explorar situações em que o objetivo do usuário é adiar um trabalho.

João, aluno de Serviço Social, está utilizando o Student Life (SL) pela primeira vez esse período [2]. Ele quer organizar melhor sua vida acadêmica, registrando as datas de prova e de entrega de trabalhos das disciplinas que está cursando [1]. Na segunda semana de aula, durante uma palestra sobre o uso do SL, ele cadastrou no seu notebook todas as disciplinas, uma a uma, juntamente com os dados do professor, de horários e salas de aula de cada uma, e para cada uma cadastrou as datas de provas e trabalhos [3]. No entanto, André, seu professor de Ética Profissional, resolveu adiar o seu trabalho por 2 semanas, atendendo aos pedidos dos alunos que tinham trabalhos de 2 disciplinas a serem entregues na mesma data. Ele aproveita um intervalo entre as aulas para utilizar seu notebook e entra no SL. Logo ativa o calendário [7], vai até o mês de abril para procurar o trabalho e o encontra na data em que estava cadastrado [6, 7]. João seleciona o trabalho, pede para modificar seus dados e estabelece a nova data: 3 de maio de 2007 [5]. Ao voltar ao calendário, percebe que já não está mais no mês de abril, e sim maio, e confere que o trabalho está marcado corretamente na nova data. Ainda um pouco inseguro, João volta para o mês de abril para se certificar de que o trabalho não está mais na data antiga.

Figura 12. Cenário de uso para adiamento da data de entrega de um trabalho.

Algumas críticas ao uso de cenários se referem à frequência com que ficam incompletos ou ambíguos.

Ainda com o objetivo de elaborar cenários mais completos, descobrindo informações que foram omitidas, Carroll et al. (1994) propõem a técnica de questionamento sistemático, onde são feitas perguntas sobre proposições extraídas dos cenários. Eles descrevem alguns tipos de questões e a razão por que formulá-las, mas uma taxonomia completa é deixada para um trabalho futuro.

Tabela 4. Questões para refinamento de cenários.

Questão	Conteúdo
Por que ...?	condições para realização de uma atividade, suas conseqüências e estados e eventos anteriores ou posteriores à atividade sob investigação
Como...?	detalhes sobre a seqüência de ações que compõem uma atividade, e com freqüência revelam também objetos que não constavam do cenário original
O que é...?	objetos e seus atributos, organizados em uma hierarquia
Questões de verificação: <Isto> pode ser feito <as-sim>?, <Isto> faz parte <da-quilo>?	respostas são sim ou não, servem para avaliar se uma ação ou atributo está bem definido e localizado no nível certo da hierarquia

No âmbito da EngSem, utilizar cenários durante a atividade de análise visa principalmente definir a primeira parte da meta-mensagem designer-usuário: “Esta é a **minha interpretação** sobre **quem você é**, o que eu entendi que **você quer ou precisa fazer**, de que **formas prefere fazê-lo e por quê**.” [de Souza 2005:84]. Isto servirá de base para a elaboração do restante da meta-mensagem (“Eis, portanto, o sistema que conseqüentemente concebi para você,...”).

Neste contexto, Paula (2003) propõe complementar os cenários com perguntas que ajudem o designer a construir a meta-mensagem. Essas perguntas podem revelar os intuídos do *designer* ao elaborar os cenários ou identificar os pontos que o *designer* almeja descobrir, explorar ou ratificar junto aos usuários. Além de apoiar o entendimento do designer e a sua reflexão sobre a meta-mensagem que será projetada, estas perguntas também podem evitar que os cenários fiquem incompletos ou ambíguos, ou até mesmo revelar novos elementos nos cenários. O *designer* pode gerar uma lista global de perguntas que seriam referenciadas nos cenários gerados (Figura 13).

Algumas perguntas exploradas nos cenários do sistema de exemplo:
1. Para que serve o SL?
2. Qual é o perfil de usuários do SL?
3. Que tipos de informação podem ser cadastradas no SL?
4. Como cadastrar uma disciplina no SL?
5. Como se pode cadastrar informações sobre provas e trabalhos no SL?
6. Como as informações sobre provas e trabalhos estão organizadas no SL?
7. Quais são as formas de consultar uma data de prova ou entrega de trabalho no SL?

Figura 13. Parte das perguntas exploradas no cenário do *Student Life*.

A referência pode ser feita incluindo-se o número da pergunta entre colchetes, no trecho do cenário onde se descreve o aspecto que a pergunta pretende abordar, como ilustrado na Figura 12. Como algumas questões são específicas a determinadas situações de uso, nem todas as questões serão respondidas em todos os cenários. Nesse exemplo, observa-se que a questão 4 não foi explorada nesse cenário, pois estava fora do seu escopo. Cabe

ao designer se certificar de que as questões estão sendo respondidas nos cenários relevantes.

6.4.2 Análise e Modelagem de Tarefas

Mesmo utilizando-se as estratégias descritas acima para a elaboração de cenários, ainda é possível que haja ambigüidades ou omissões nos cenários descritos. Para tentar reduzir ainda mais este problema e obter uma visão geral e estruturada das tarefas que se pretende apoiar, utilizam-se com frequência técnicas de análise e modelagem de tarefas¹⁵.

A **análise hierárquica de tarefas** (HTA – *Hierarchical Task Analysis*) foi um dos primeiros métodos de análise de tarefas desenvolvido [Annett & Duncan 1967, Diaper & Stanton 2004]. Ela foi criada pela necessidade de uma base racional para a compreensão das habilidades necessárias para tarefas complexas e não repetitivas, especialmente tarefas de controles de processos, que envolviam tanto atividades físicas quanto mentais. A análise de tarefa é uma abordagem *top-down*, que examina primeiro as metas de alto nível (ou objetivos), antes de considerar as ações através das quais a meta pode ser atingida. Essas metas são decompostas e definidas em termos de uma hierarquia de metas e submetas aninhadas, buscando identificar quais submetas são mais difíceis de atingir ou causam mais erros, ou seja, quais submetas podem limitar ou impedir o alcance da meta maior.

A HTA é um método sistemático de investigar problemas de desempenho humano, verificar se a tarefa é realmente atingida ou, caso contrário, por que não. O principal objetivo da HTA é poder relacionar o que os usuários fazem (ou o que se recomenda que façam), por que o fazem e as conseqüências de isso não ser feito corretamente. A HTA permite caracterizar a prática de trabalho, identificando, categorizando e decompondo tarefas. Um **objetivo** é o estado desejado do sistema; **tarefas** descrevem a maneira como o objetivo pode ser alcançado; **operações** são as menores e mais básicas unidades de comportamento; e **planos** especificam as condições sob as quais uma tarefa ou subtarefa deve ser realizada [Preece et al. 1994].

Inicialmente, deve-se especificar os objetivos principais que cobrem toda a área de interesse. Cada um deve ser dividido em subtarefas. Essas subtarefas devem ser dispostas como planos em camadas, certificando-se de que estejam lógica e tecnicamente corretas, e de que não haja omissões. O próximo passo é decidir o nível de detalhe necessário, para saber até onde deve-se decompor as tarefas. Prossegue-se a análise, decompondo as tarefas em profundidade ou em largura. Deve-se utilizar uma numeração que permita a identificação inequívoca de um plano, que consiste em um caminho desde a raiz da tarefa até as folhas correspondentes. Ao final da análise, deve-se verificar se todas as decomposições e numerações estão consistentes. Além do diagrama de tarefas, pode ser útil produzir um texto explicativo. Como boa prática, pode-se mostrar o diagrama para outra pessoa verificar sua consistência. Esta pessoa deve conhecer a tarefa, mas não deve ter participado do processo de modelagem.

¹⁵ Há pesquisadores e profissionais que dispensam o uso de cenários, e em vez deles utilizam modelos de tarefas em diversas etapas do processo de design. Este capítulo considera que cenários e modelos de tarefas podem ser utilizados de forma complementar.

No design de IHC pautado pela EngSem, os modelos de tarefas devem representar, além da estrutura hierárquica: o tipo de tarefa, subtarefa ou operação no que diz respeito ao seu conteúdo, forma e ilocução [de Souza 2005]. São representados: pré-condições (se houver); estruturas de seqüência e iteração; tarefas alternativas, opcionais e ubíquas; signos associados e mecanismos de prevenção e tratamento de erros ou rupturas de comunicação.

A representação utilizada é uma adaptação ao diagrama hierárquico de tarefas [Paula 2003, de Souza 2005]. O **objetivo** (meta de alto nível) é representado por um retângulo com bordas arredondadas. As **tarefas** que o compõem são representadas por retângulos, com marcações especiais para indicar a que tipo de estrutura estão associadas, como será visto adiante. A decomposição das tarefas em subtarefas deve parar antes que o modelo inclua detalhes operacionais de interface, tais como “digitar X”, “pressionar botão Y”, etc. Dependendo do ponto em que se pára a decomposição das tarefas, existem tarefas tão simples que poderão – em fases posteriores no processo de *design* – ser mapeadas diretamente em um elemento de interface ou de interação. Estas tarefas são representadas no modelo como **operadores**, os quais são representados por uma linha abaixo do retângulo. O modelo de tarefas deve refletir como o usuário trabalha, evitando se concentrar em um ambiente ou plataforma tecnológica específica. Esta consideração não apenas facilita o re-uso de modelos de tarefas, como também evita que decisões sobre a solução de interação ou de interface sejam tomadas prematuramente, dificultando a exploração de soluções alternativas por parte dos projetistas.

Por exemplo, na Figura 14, Adiar data de entrega de trabalho é o objetivo, Localizar trabalho é uma tarefa e Selecionar trabalho é um operador. O projetista optou por modelar a tarefa Ativar calendário como um operador. Deve-se observar que as tarefas que o usuário vai realizar de fato são as que estão representadas nas folhas (último nível) da estrutura hierárquica, muitas das quais são representadas por operadores.

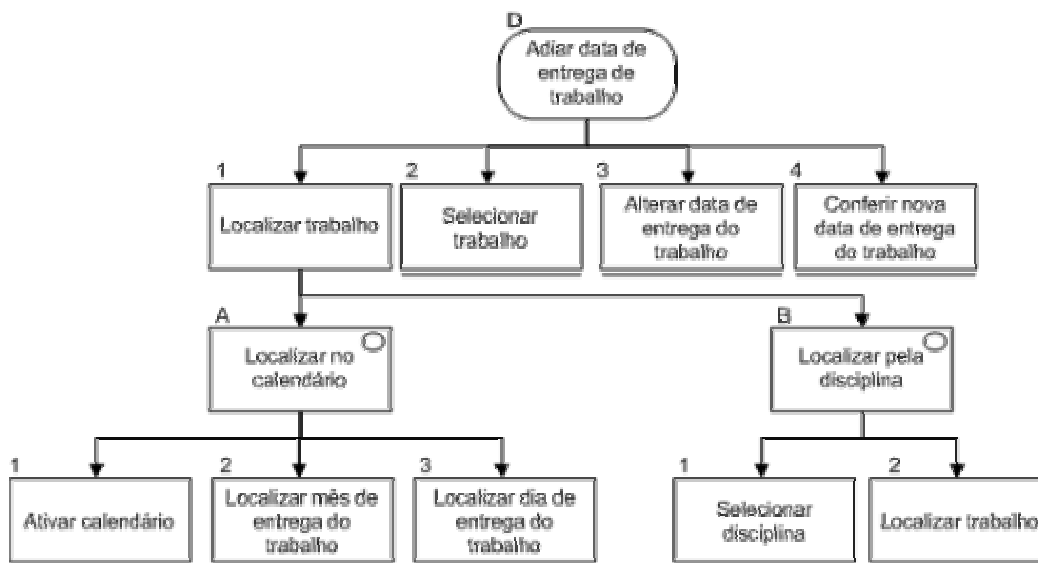


Figura 14. Modelo hierárquico de tarefas adaptado.

As tarefas podem ser organizadas nos seguintes tipos de estruturas: sequenciais, independentes de ordem, alternativas e iterativas. Em uma estrutura **sequencial**, existe uma ordem em que as tarefas devem necessariamente ser efetuadas pelo usuário. As tarefas, nesta estrutura, são representadas por retângulos contendo o nome da tarefa, expresso do ponto de vista do usuário, e um número indicando sua posição na sequência (Figura 15a).

Algumas tarefas podem ser realizadas em qualquer ordem. Uma estrutura de tarefas **independente de ordem** representa um conjunto (e não uma sequência) de tarefas a serem efetuadas pelo usuário. Tipicamente, o projetista sugere uma ordem de execução, mas é o usuário quem determina, de fato, em que ordem as tarefas serão efetuadas. Neste tipo de estrutura, as tarefas são representadas como as tarefas sequenciais, mas, como a ordem é apenas sugerida, incluímos um ponto de interrogação após o número que indica a posição relativa da tarefa na estrutura. No exemplo, identificamos algumas tarefas independentes de ordem (Figura 15b).

Para o alcance de uma meta, há momentos em que diversos cursos de ação são possíveis. Tais cursos de ação são representados por uma estrutura **alternativa**, onde o usuário deverá selecionar qual das tarefas da estrutura será efetuada. Nesta estrutura, utilizam-se pequenos círculos no canto superior direito do retângulo de cada tarefa alternativa, e letras como identificadores em vez de números (Figura 15c).

Quando uma tarefa pode ser realizada diversas vezes, utiliza-se uma estrutura **iterativa**. Um asterisco (*) no canto superior direito do retângulo é utilizado para indicar a iteração (Figura 15d). Geralmente, uma tarefa iterativa representa tarefas que podem ser efetuadas zero ou mais vezes. Caso seja necessário definir um número mínimo ou máximo de repetições, pode-se incluir, acima do retângulo e alinhada à direita, uma expressão que indica a cardinalidade da iteração. A expressão [n+] indica que a tarefa deve ser realizada pelo menos n vezes; [m..n] indica que a tarefa deve ser realizada no mínimo m e no máximo n vezes. Além disso, quando o usuário pode optar por realizar ou não uma tarefa, ela é dita **opcional**, e é representada com uma borda tracejada (Figura 15e).

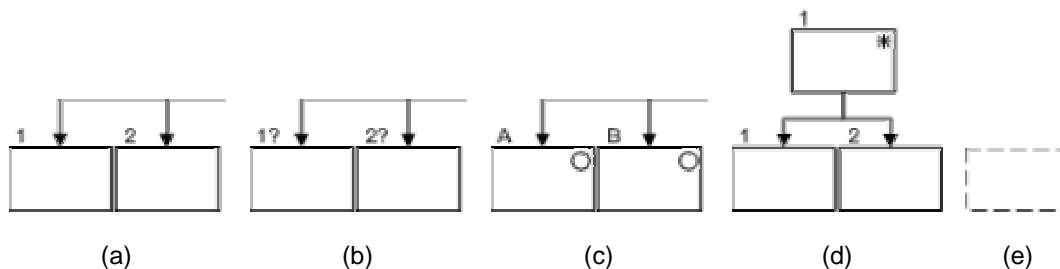


Figura 15. Representação de estrutura de tarefa (a) sequencial, (b) independente de ordem, (c) alternativa, (d) iterativa e (e) opcional.

A EngSem ressalta a importância de se representar, associadas a cada tarefa, os signos e as rupturas na conversa entre o preposto do *designer* e o usuário que possam ocorrer durante a realização da tarefa. Para isto, a representação diagramática das tarefas é complementada por uma especificação textual, como será visto adiante.

Signos

Através dos cenários, pode-se identificar os signos que farão parte das ilocuições do pre-posto do designer e do usuário, ou seja, que são apresentados ao usuário ou por ele manipulados. Examinando todos os signos presentes nos cenários, percebe-se que alguns deles podem ser agrupados e relacionados a conceitos ou entidades do domínio e/ou da própria aplicação. Estes são representados como signos compostos. A Tabela 5 mostra os signos extraídos do cenário apresentado na Figura 12. Os signos disciplina, título, enunciado, data de entrega e referências podem ser agrupados no signo composto trabalho. Observa-se que o signo disciplina é, ele próprio, um signo composto de outros signos (e.g. nome, código, professor). Em última análise, essas composições e relacionamentos podem auxiliar a definição de um modelo de entidades e relacionamentos. Neste momento do design de IHC, no entanto, o foco está na identificação de quais signos estão relacionados a quais tarefas, e ainda não no relacionamento entre os próprios signos.

Os signos da interface de uma aplicação podem ser classificados como signos de domínio, transformados ou de aplicação. Signos encontrados no mundo do usuário, independentemente da aplicação, são chamados de signos do **domínio** (por exemplo, nome e endereço). Signos originados no domínio, mas que sofrem alguma transformação ao serem incorporados à aplicação, como uma transformação resultante de analogias ou metáforas, são representados como signos **transformados** (por exemplo, pastas na área de trabalho do Windows). Por último, signos que só fazem sentido dentro da aplicação, e não têm significado prévio para os usuários, são chamados de signos da **aplicação** (por exemplo, login e senha).

Por que classificar os signos dessa forma? Porque tipos de signos diferentes requerem diferentes tomadas de decisão por parte do *designer*. Pode ser necessário projetar ilocuições que melhor apoiem a interpretação do signo pelo usuário, para que este, por sua vez, também seja capaz de enunciar ilocuições adequadas envolvendo o signo ou sobre ele. Para signos do domínio, tais ilocuições podem ser necessárias, caso haja limitações impostas pela própria aplicação, como uma restrição na sua expressão (formato do dado de entrada) ou conteúdo (valores válidos). Por exemplo, um signo de *data* pode requerer explicações sobre o formato esperado (dd/mm/aaaa) e sobre as datas permitidas (nos últimos 5 anos; somente dias úteis). Já para os signos transformados, o designer deve fornecer aos usuários informações sobre os limites da analogia ou metáfora realizada para transportá-los para a aplicação. Por exemplo, uma explicação sobre as pastas em um *desktop* seria “Estas pastas funcionam como no mundo real, exceto que nunca ficam cheias. Isto é, você pode manter e colocar muitas coisas nelas. Na realidade, o disco do seu computador é o real local onde as coisas estão sendo guardadas. Então, é ele que controla a quantidade de espaço que pode ser ocupado.” Finalmente, signos de aplicação, que podem ser totalmente desconhecidos pelos usuários, requerem uma explicação completa sobre o que significam e como são utilizados. Por exemplo, o signo *zoom* em uma aplicação gráfica. Alguns signos podem gerar dúvidas no momento de classificação em um dos tipos. Por exemplo, o signo senha pode ser classificado em aplicação, mas pode aparecer a seguinte dúvida “Não se pode considerar o signo senha nesta aplicação como uma analogia à assinatura, impressão digital ou algo que identifica uma única pessoa? Se sim, então também se pode classificá-lo como signo transformado.” Neste caso, fica a cargo do *designer* definir o tipo do signo, com base nas características dos usuários e suas tarefas, bem como a quantidade de informação a

ser fornecida. Em todo caso, a classificação auxilia o *designer* a refletir sobre a explicação a ser associada a cada signo.

Tabela 5. Signos simples e compostos definidos para o sistema de exemplo.

signo composto	signos que o compõem	tipo	descrição
trabalho		domínio	Trabalho de uma disciplina a ser entregue na data estipulada.
	disciplina (composto)	domínio	
	data de entrega	domínio	Quando o trabalho deve ser entregue.
	título	domínio	Identificação breve do trabalho.
	enunciado	domínio	Enunciado completo do trabalho.
	referências	domínio	Material de apoio à execução do trabalho.
	lembrete	transformado	Lembrete emitido alguns dias antes da data de entrega do trabalho
disciplina		domínio	Disciplina cursada pelo aluno.
	nome	domínio	Nome ou título da disciplina.
	código	domínio	Código da disciplina.
	professor (composto)	domínio	Professor que leciona a disciplina.
...			

Para se fazer referência a um dos signos que compõem um signo composto, deve-se utilizar o formato nome_signo_composto.nome_signo. Se, por outro lado, for necessário fazer referência a todos os signos de uma determinada composição, pode-se utilizar o formato nome_signo_composto.*. Por exemplo, o signo correspondente ao nome do professor pode ser expresso como professor.nome. Já em uma ilocução que envolve todos os dados de um professor, como nome, e-mail, horários de atendimento, o conjunto destes signos pode ser representado por professor.*. Cada signo que for enunciado pelo preposto, ou seja, apresentado ao usuário (considerado comumente como um dado de saída), é representado por seu nome seguido de ponto de exclamação (nome!). Já um signo que será enunciado pelo usuário, ou seja, correspondente a um dado que será fornecido ou manipulado pelo usuário (considerado um dado de entrada), é representado por seu nome seguido de ponto de interrogação (nome?). Caso haja diversas instâncias de um mesmo tipo, pode-se utilizar os construtos conjunto([signos], cardinalidade) ou seqüência ([signos], cardinalidade, ordenação) para representá-las. Na Figura 16, observam-se, para a tarefa A.2 Fornecer dados do trabalho, diversos signos que serão manipulados pelo usuário: disciplina, título, data de entrega, enunciado e referências. Para a tarefa B.2 Examinar trabalhos, é definida uma seqüência de signos que serão apresentados em ordem cronológica inversa de data de entrega. Já para a tarefa B.4 Examinar dados do trabalho, todos os signos relacionados ao signo composto trabalho serão apresentados.

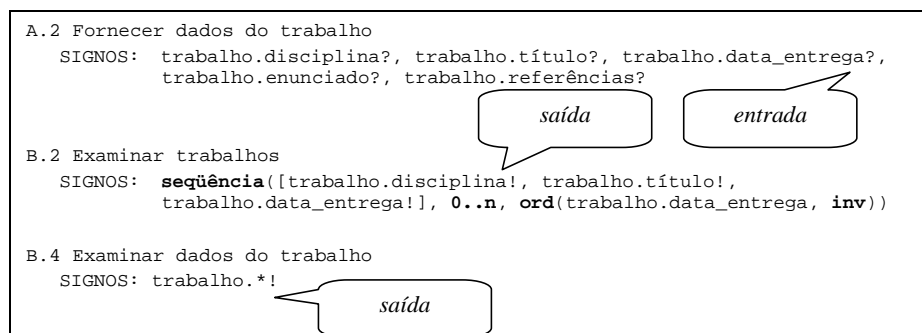


Figura 16: Identificação dos signos em um modelo de tarefas.

Prevenção e Tratamento de Rupturas

Como dito anteriormente, a EngSem ressalta a importância de se representar os signos e rupturas que possam ocorrer durante a realização de uma tarefa. No momento de representação de uma tarefa, já é possível prever algumas possíveis rupturas (*break-downs*) na comunicação entre o preposto do *designer* e o usuário que podem ocorrer durante a sua realização. Nestes casos, o designer deve representar, para cada tarefa, os tipos de apoio à prevenção e tratamento de ruptura que pretende oferecer, classificando-os em uma das seguintes categorias [Barbosa & Paula 2003]:

- **prevenção passiva (PP):** Rupturas que devem ser evitadas por ilocuições do preposto do designer, como documentação ou instruções *online* (por exemplo, uma dica de formato como “(dd/mm/aaaa)” ao lado de um campo de data; ou uma instrução explícita como “* indica campo obrigatório”).
- **prevenção ativa (PA):** Rupturas que devem ser evitadas ativamente pelo sistema, que restringe a possibilidade de expressão do usuário a ilocuições válidas. No modelo de interface, por exemplo, isto poderá ser concretizado através de: habilitar ou desabilitar botões de acordo com o *status* atual da aplicação; impedir que o usuário digite letras ou símbolos em campos numéricos; um controle de calendário que impede que o usuário indique uma data inválida; e assim por diante.
- **prevenção apoiada (alerta - AL):** Situações que o sistema detecta como sendo rupturas em potencial, mas cuja decisão recai sobre o usuário. Cabe ao preposto descrever adequadamente a situação e solicitar que o usuário tome uma decisão informada sobre os rumos da interação. Isto ocorre, por exemplo, quando o usuário expressa a intenção de gravar um arquivo com um nome diferente (“Salvar como...”), mas em seguida informa o nome de um arquivo existente. Geralmente esse mecanismo é concretizado na interface por mensagens de confirmação (por exemplo, “Arquivo já existe, deseja sobrescrevê-lo?”; “Foram feitas alterações no trabalho. Deseja armazená-las?”).
- **tratamento apoiado (TA):** Rupturas que ocorreram e devem ser tratadas pelo usuário com apoio do preposto do *designer*, ou seja, a partir de ilocuições do preposto, o usuário enuncia novas ilocuições que lhe permitam prosseguir com a sua tarefa (por exemplo, o preposto apresenta uma mensagem descrevendo o problema e dá uma oportunidade para o usuário corrigi-lo).

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

Os tipos de prevenção e tratamento de ruptura são apresentados de forma textual no modelo de tarefas, logo após a indicação dos signos (Figura 17).

A.2 Fornecer dados do trabalho
SIGNOS: trabalho.disciplina?, trabalho.título?, trabalho.data_entrega?, trabalho.enunciado?, trabalho.referências?
PREVENÇÃO PASSIVA e TRATAMENTO APOIADO: data_entrega e título são campos obrigatórios

Figura 17. Representação de alguns tipos de prevenção e tratamento de ruptura.

6.4.3 Modelagem de Interação

A partir da caracterização dos usuários, seu ambiente de trabalho, e suas tarefas, podemos prosseguir para a modelagem da interação¹⁶. O objetivo da **modelagem de interação** é definir as conversas que o usuário poderá travar com o preposto do designer (e quando, e sob que condições), ou seja, especificar as ilocuições que esses interlocutores poderão enunciar para o usuário atingir seus objetivos. É responsabilidade do designer “contar” para o usuário sua visão de design e dar-lhe melhores condições de entender e aprender sobre o sistema projetado e as formas de utilizá-lo. Portanto, precisamos estabelecer **sobre o que** se está conversando e **de que forma** esta conversa ocorre. A modelagem da interação pode então ser considerada como a especificação de **todas as possíveis conversas** que os usuários poderão travar com o preposto do designer. Em linhas gerais, trata-se de projetar os **caminhos de interação** que o usuário pode percorrer, incluindo caminhos alternativos e de exceção ou ruptura. Cada passo de interação ocorre dentro de uma **cena**, na qual são travados **diálogos** entre usuário e sistema, guiados por um ou mais objetivos comunicativos. Um diálogo tem tópico e foco. O **tópico** é o tema central do diálogo, e o **foco** consiste em um trecho de conversa sobre algum aspecto específico do tópico. A Tabela 6 apresenta um exemplo de interação como uma conversa na marcação de uma reunião de grupo de trabalho.

Tabela 6. Exemplo de interação como uma conversa.

U: Preciso marcar uma reunião do grupo do trabalho de IHC.	Tópico: marcar uma reunião
U: Deixe-me ver os compromissos dessa semana. S: Ei-los.	Foco: compromissos da semana
U: A reunião deve durar umas 3 horas. Não tem nenhum horário disponível para ela.	Foco: horários disponíveis com duração determinada
U: Deixe-me ver a próxima semana. S: Aqui está. Você tem aulas todos os dias de manhã, e as tardes de 2ª, 4ª e 6ª estão ocupadas.	Foco: compromissos da próxima semana
U: Tem horário disponível na 3ª feira às 14h-17h.	Foco: horários disponíveis com duração determinada

¹⁶ Esta subseção apresenta uma síntese da monografia [Silva & Barbosa 2007].

U: Vou marcar a reunião nesse dia e horário	Foco: novo compromisso no dia e horário selecionados
S: Que tipo de compromisso? Com quem e onde? U: Reunião com o grupo de trabalho, nas salas de estudo.	Foco: dados da reunião
S: Marcado.	Foco: verificação dos dados da reunião

Paula (2003) propôs, no âmbito da EngSem, uma linguagem para a modelagem da interação humano-computador como uma conversa, MoLIC (*Modeling Language for Interaction as Conversation*). MoLIC foi projetada para apoiar os designers no planejamento da interação, motivando sua reflexão sobre as estratégias de resolução de problemas dos usuários que deveriam ser apoiadas pelo sistema interativo. Desde sua proposta, diversas aplicações vêm sendo modeladas com a MoLIC, para diferentes domínios e plataformas. Algumas extensões foram propostas e dispararam um extenso esforço de revisão, que resultou na 2ª edição da MoLIC [Silva 2005]. Esta seção apresenta a versão diagramática dessa segunda edição.

MoLIC foi projetada para representar a interação humano-computador como o conjunto de conversas que os usuários podem (ou devem) travar com o sistema (mais precisamente, com o preposto do designer) para atingir seus objetivos. Nessas conversas, para os usuários entenderem melhor seu interlocutor, o preposto do designer precisa comunicar-lhes adequadamente: o que o sistema fez (ou não fez), o que está fazendo (ou não está fazendo), o que ele permite ou proíbe os usuários de fazer, como e por quê. Essa comunicação é particularmente importante quando uma situação inesperada ocorre, como uma ruptura na comunicação. MoLIC foi projetada para ser uma ferramenta epistêmica, ou seja, utilizada para aumentar a compreensão dos designers sobre o problema sendo resolvido e o artefato sendo projetado, e não para oferecer soluções e respostas diretas para os problemas e questões em pauta. É importante observar que a MoLIC foi proposta para uso humano e, portanto, não representa um modelo formal processável por computador. MoLIC não foi concebida para substituir representações existentes, mas sim complementá-las. Por exemplo, seqüências de eventos descritas em cenários e estruturadas em modelos de tarefas podem ser articuladas num diagrama MoLIC, motivando a representação de relações e interseções entre objetivos, cenários ou tarefas.

MoLIC é composta atualmente de quatro artefatos: um diagrama de metas, um esquema conceitual de signos e um diagrama de interação complementado por uma especificação textual. O diagrama de metas indica **o que** os usuários podem fazer com a aplicação, ou seja, quais objetivos podem atingir. O esquema conceitual de signos define e organiza os **conceitos** envolvidos no sistema, em particular aqueles que emergem na interface de usuário. Inclui informações envolvidas em cada ação do usuário ou ação externa que afete a interação usuário-sistema. O diagrama de interação representa **como** as metas podem ser atingidas durante a interação, e a especificação textual **detalha** o conteúdo do diagrama de interação para servir como ponte entre a modelagem da interação e o projeto da interface propriamente dita. Nesta seção, nos concentramos na construção dos diagramas de metas e de interação. As representações detalhadas (esquema conceitual de signos e especificação textual da interação) não são endereçadas neste documento.

Diagrama de Metas

O diagrama de metas é utilizado para representar as metas dos usuários que tenham sido identificadas na etapa de análise. Um diagrama de metas é diferente de um modelo de tarefas: é utilizado apenas para definir **o que** o usuário deseja realizar, sem considerar **como** ele o fará. Para construir um diagrama de metas, o designer precisa inicialmente listar as metas identificadas na etapa de análise. Essas metas correspondem aos objetivos descritos nos cenários e estruturados nos modelos de tarefas.

As metas podem ser classificadas em finais e instrumentais. As **metas finais** podem geralmente ser formuladas como: “*Eu (usuário no papel <Papel>) quero utilizar o sistema para <atingir metaFinal>.*” Já as **metas instrumentais** são utilizadas como facilitadoras para as metas finais. Além disso, as metas instrumentais podem ser planejadas ou oportunistas. As **metas instrumentais planejadas** podem ser formuladas como: “*Quero <atingir metaInstrumental> para <atingir metaFinal> [de forma mais eficiente/fácil/...]*”. Já as **metas oportunistas** surgem durante a interação, e podem ser formuladas como: “*Aqui de onde estou no sistema, vou <atingir metaInstrumental> para <atingir metaFinal> [de forma mais eficiente/fácil/...]*”. No diagrama de metas, são representadas apenas metas finais e instrumentais planejadas.

Através do diagrama de metas, visa-se organizar e anotar as metas dos usuários de acordo com algumas dimensões de interesse do designer. Essas dimensões variam com o tipo de projeto. Alguns elementos que podem ser utilizados para organizar as metas são:

- papéis de usuários (i.e. papéis que podem atingir cada meta, tais como aluno/professor);
- tipo de meta (meta final ou instrumental planejada);
- “entidade” principal envolvida na meta (e.g. professor, disciplina, prova etc.)

Metas concretas, que os usuários poderão atingir utilizando a aplicação, podem ser agrupadas em metas mais abstratas para facilitar a visualização. O diagrama de metas é representado por uma estrutura hierárquica anotada, indicando grupos de metas e papéis de usuários. No diagrama, nós-filhos herdam os papéis do nó-pai, isto é, se um nó A possui um nó B como filho, então todos os papéis que podem atingir a meta A também podem atingir a meta B. No caso do sistema de exemplo, algumas metas são:

- cadastrar, visualizar, modificar, remover disciplinas;
- cadastrar, visualizar, modificar, remover provas e trabalhos; e
- descobrir quais são as atividades nos próximos dias.

A Figura 18 apresenta um diagrama de metas ilustrando o que pode ser feito com o sistema de exemplo. Esse diagrama é bem simples, pois há somente um papel de usuário: aluno. Nele, pode-se observar que as metas finais e instrumentais estão separadas: as metas finais são aquelas relacionadas diretamente com as atividades dos alunos, ao passo que as instrumentais (planejadas) são aquelas relacionadas à configuração do sistema. Além disso, algumas metas abstratas são definidas para organizar as metas concretas de acordo com os signos sendo manipulados. Estas são indicadas por linhas pontilhadas no diagrama.

Cada meta pode estar associada a um ou mais cenários. Numa abordagem baseada exclusivamente em cenários, algumas relações entre as metas podem ser deixadas sem representação. Considerando as metas Visualizar objeto e Remover objeto ilustrados na figura, pode acontecer de um cenário para Remover um objeto definir uma forma diferente de localizar um trabalho do cenário Visualizar objeto. Isso pode causar inconsistências e dificuldades na interação usuário-sistema. Uma situação extrema aconteceria se o usuário visualizasse um trabalho e, percebendo que este deveria ser removido, ele não pudesse fazê-lo naquele momento. Em vez disso, teria que abandonar a “atividade de visualização”, selecionar uma opção para remover o trabalho, localizar novamente o trabalho e só então removê-lo. Esse exemplo de problema é muito trivial e portanto raro, mas ilustra uma possível inconsistência ou ineficiência que pode não ser detectada utilizando-se somente cenários.

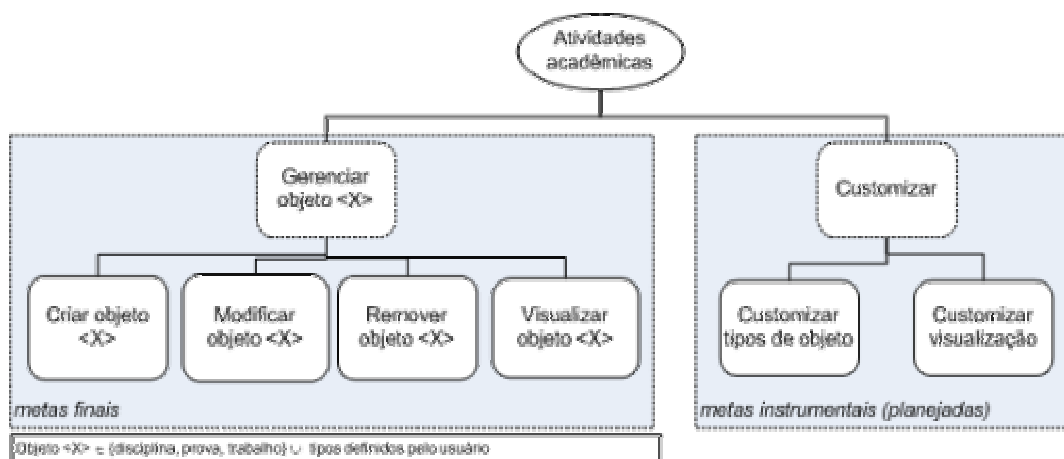


Figura 18. Diagrama de metas para o sistema de exemplos (adaptado de [Araujo 2005]).

Diagrama de Interação

O diagrama de interação da MoLIC permite aos designers representar todas as possíveis conversas interativas que os usuários podem travar com o preposto do designer para atingir suas metas [Paula 2003, Silva 2005]. Ele foi concebido para motivar os designers a refletir sobre a metacomunicação, permitindo-lhes especificar conversas alternativas para o atingimento de um mesmo objetivo, e analisar o relacionamento e interferências entre metas (e, portanto, especificar as conversas relativas a metas instrumentais oportunistas). A representação diagramática foi escolhida para promover uma visão global da aplicação tal como o preposto vai apresentá-la para o usuário¹⁷. Isso ajuda a evitar o tipo de problema descrito acima.

A construção de diagramas MoLIC é realizada em dois passos. Primeiro, os designers definem os tópicos de todas as possíveis conversas usuário-sistema e as trocas de turno entre usuários e o preposto que encadearão os tópicos dessas conversas. Esse nível de abstração promove uma reflexão, análise e discussões sobre a interação por

¹⁷ Quando há múltiplos papéis de usuários, a interação usuário-sistema de cada papel deve ser representada em seu próprio diagrama MoLIC. Isso significa que, quando dizemos “visão global”, nos referimos à perspectiva de um único usuário da aplicação (i.e., todas as partes da aplicação às quais ele tem acesso), e não à aplicação inteira transpassando múltiplos papéis, tal como visto pelo designer.

uma equipe multidisciplinar cedo no processo de design [Paula et al. 2005]. Ao final dessas discussões, ou mesmo durante, caso necessário, os tópicos são detalhados, e os designers definem os signos envolvidos nas trocas comunicativas que correspondem a cada tópico. O diagrama MoLIC detalhado é um recurso importante para o projeto da interface de usuário concreta nas etapas posteriores do processo de desenvolvimento, mas não será apresentado neste capítulo por restrições de espaço.

Deve haver um diagrama MoLIC para cada papel de usuário. Cada diagrama representa a visão completa que um usuário poderá ter do sistema. Para o usuário atingir um objetivo, ele deve “conversar” com o preposto do designer sobre o que ele quer realizar (e como o preposto permite/recomenda/requer que ele o faça). É importante observar que essa perspectiva comunicativa não implica que a interface de usuário concreta será uma interface no estilo conversacional. Significa apenas que as questões comunicativas envolvidas na interação usuário-sistema são trazidas para o primeiro plano e explicitamente durante o design da interação.

No primeiro passo da construção de diagramas MoLIC, os designers especificam todas as conversas entre usuário e preposto do designer. O foco dessa etapa é apoiar os designers na reflexão de questões gerais de interação, sem detalhar cada passo de interação em estruturas de elementos atômicos. Algumas dessas questões gerais são:

- troca de turnos entre usuário e sistema para atingir uma meta;
- conversas (caminhos de interação) alternativas para os usuários atingirem uma mesma meta (possivelmente endereçando as necessidades e preferências de diferentes perfis de usuários);
- conversas relativas a metas instrumentais oportunistas;
- conversas para a recuperação de rupturas, i.e., mecanismos para os usuários se recuperarem de rupturas;
- a consistência (ou não) entre caminhos de interação semelhantes ou análogos.

No restante desta subseção, vamos utilizar como exemplo a situação em que o aluno quer cadastrar uma disciplina no sistema StudentLife.

Como a conversa é aberta? E como é fechada?

Um **ponto de abertura** é onde a conversa usuário-sistema se inicia. Na maioria dos ambientes, é o momento em que a aplicação é ativada no sistema operacional. Num navegador, é o momento em que uma URL válida é digitada ou um link é seguido para a aplicação Web. Em aplicações baseadas em documentos, por outro lado, geralmente há dois pontos de entrada: um acessado ativando-se a aplicação, e outro acessado quando um documento produzido por aquela aplicação é ativado. Em cada caso, a conversa pode iniciar de forma diferente: abrindo um documento em branco ou o documento acessado. Já um **ponto de fechamento** pode ser utilizado para indicar o encerramento da conversa (i.e. término da interação usuário-sistema). Isso geralmente indica momentos específicos da interação de onde o usuário pode sair da aplicação. Os pontos de saída geralmente são representados quando uma conversa de encerramento pode ocorrer antes do final real da interação. Por exemplo, quando é necessário armazenar mudanças realizadas antes de sair da aplicação.

A Figura 19a ilustra dois pontos de entrada para um editor de texto, representados por círculos preenchidos na cor preta, e a Figura 19b ilustra a representação de um ponto de saída para um editor de texto, representado por um círculo negro circunscrito a um círculo branco.

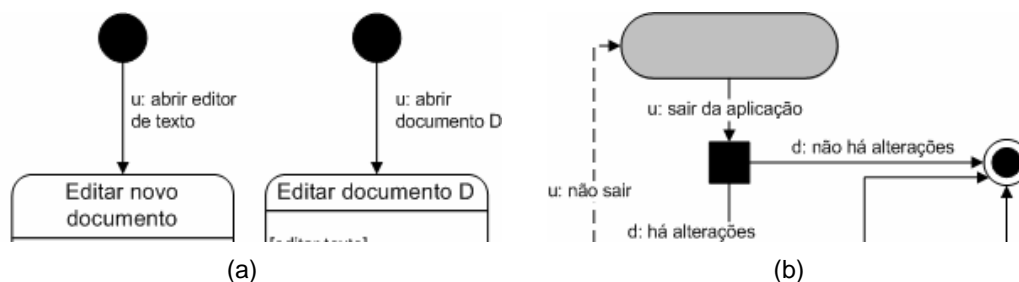


Figura 19. Pontos de a) abertura e b) fechamento para um editor de texto.

Sobre o quê o usuário poderá falar?

A forma mais simples de representar isso na MoLIC é utilizar uma cena. As **cenas** representam conversas sobre um determinado tópico, culminando na vez de o usuário dizer algo para concluir a conversa, suspendê-la, desviar do tópico ou mesmo abandoná-la. Uma cena pode ser vista como uma cena real numa peça teatral, onde as trocas comunicativas entre usuário e sistema ocorrem. Na notação da MoLIC, uma cena é representada por um retângulo com bordas arredondadas. No início da construção de uma cena, somente o tópico da conversa precisa ser representado. Essa representação “mínima” de uma cena é representada na Figura 20a. O tópico de uma cena pode ser lido, do ponto de vista do preposto¹⁸, como “Neste momento, você (usuário) pode (ou deve) <tópico>¹⁹”.

A conversa numa cena pode ser composta de diversos diálogos, que por sua vez são compostos de pares conversacionais. Os **diálogos** de uma cena são representados num segundo compartimento da cena, entre colchetes. Os diálogos indicam os subtópicos da cena, isto é, o conteúdo das ilocuções que o usuário e o preposto podem enunciar sobre o tópico da cena naquele momento. A Figura 20b ilustra a cena Cadastrar disciplina com um diálogo, indicando ainda os signos envolvidos nas ilocuções correspondentes.



Figura 20. Representação da cena Cadastrar disciplina: a) mínima; b) com diálogos.

¹⁸ Um diagrama MoLIC pode ser lido do ponto de vista do preposto do designer ou do ponto de vista do usuário, dependendo do objetivo do leitor.

¹⁹ No momento, não há uma diferenciação, na MoLIC, entre “pode” e “deve”. Essa distinção está implícita na indicação dos signos obrigatórios que compõem os diálogos da cena. Estamos investigando, no entanto, uma forma de tornar essa diferenciação mais explícita, além de acrescentar uma outra modalidade: “eu [preposto] recomendo a você [usuário]...”, que contribuirá ainda mais para o mapeamento da MoLIC para a interface concreta.

Numa cena, alguns diálogos podem ocorrer que não são parte da aplicação sendo projetada, mas sim do sistema operacional ou ambiente computacional. Por exemplo, quando um usuário manipula uma página Web utilizando uma barra de rolamento, ele não está trocando falas com a aplicação projetada, e sim com o navegador. Em geral, o designer conhece esses tipos de diálogos, mas tem pouco ou nenhum controle sobre eles. Por isto, esses diálogos não são representados no diagrama de interação. Somente nos casos em que o designer quer que seu preposto “fale” sobre esses interlocutores “externos” é que ele deve representar os diálogos e signos correspondentes.

Quando vista de forma isolada, uma cena não informa muita coisa aos designers. Quando é que um usuário pode “falar sobre” aquele tópico (i.e. cadastrar uma disciplina)? Quando e como o sistema pode responder ao que o usuário disse sobre o tópico? Qual é o resultado (i.e. as alterações no sistema ou no “mundo real”) desse trecho de conversa (i.e. o que o usuário realiza)? Pode algo dar errado? Se der, o quê, e como o usuário pode se recuperar disso e prosseguir em direção ao alcance de sua meta? Qual é a correspondência entre um diagrama MoLIC e o diagrama de metas? Os elementos da MoLIC que permitem aos designers endereçar essas questões são vistos a seguir.

Quando e como o preposto pode responder ao que o usuário disse?

Para o sistema responder a tudo o que o usuário falou sobre o tópico da cena, o usuário precisa passar o turno para o sistema, que “pensará sobre” (i.e. processará) a comunicação do usuário e responderá de acordo. Dois elementos da MoLIC são utilizados aqui: uma fala de transição (ou fala de troca de turnos) e um processo de sistema.

Uma fala de transição representa uma troca de turno com mudança de tópico conversacional²⁰. Pode ser causada por uma fala do usuário ou do preposto do designer. Tal fala é representada por uma linha direcionada, indicando pelo menos o enunciador da fala (“u” para usuário e “d” para o preposto do designer) e o seu conteúdo. A Figura 21a ilustra uma fala de transição do usuário a partir da cena Cadastrar disciplina:

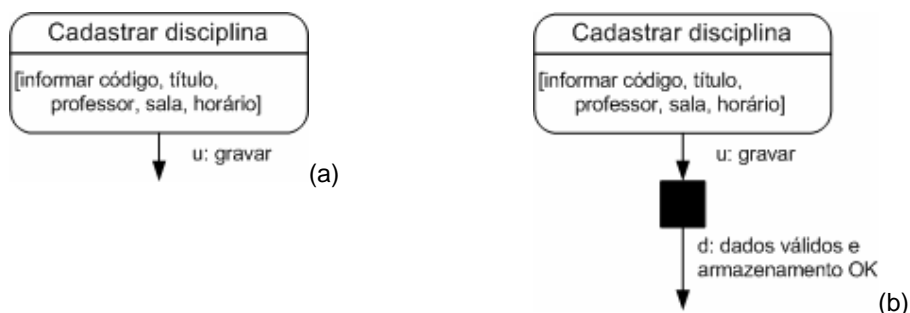


Figura 21. a) Fala de transição do usuário; b) Processamento do sistema como uma caixa-preta, seguida de fala de transição do preposto.

Em alguns casos, as falas de transição se referem a algum signo manipulado na cena de origem. Por exemplo, quando o usuário examina um conjunto de disciplinas, pode

²⁰ Vale observar que trocas de turno podem ocorrer dentro de uma cena, tal como expresso nos diálogos. No entanto, optamos por representar explicitamente como fala de transição somente as trocas que envolvem mudança de tópico.

selecionar uma delas para modificar ou remover. Isso poderia ser representado por uma fala de transição [u: modifica disciplina D], onde D representa uma disciplina. O signo pode ser indicado através de uma seleção de usuário, por exemplo, mas as decisões sobre a expressão dos signos são deixadas para mais tarde, na etapa de modelagem da interface.

O que houve? Qual é o resultado (i.e. as alterações no sistema ou no “mundo real”) desse trecho de conversa (i.e. o que o usuário realiza) concluído pela fala de transição?

Um processo de sistema ocorre quando é a vez de o sistema “pensar” e decidir qual será o próximo tópico da conversa, geralmente como resposta a uma fala de transição do usuário. Enquanto o sistema está “pensando”, o usuário não sabe o que está acontecendo exceto pelas falas do preposto do designer sobre o que está pensando/fazendo, tanto durante quanto após o processamento. Portanto, é importante motivar os designers a pensarem sobre como comunicar aos usuários sobre o progresso e os resultados do processamento do sistema, e sobre para onde a conversa vai dali, e por quê.

Um processo de sistema é representado num diagrama MoLIC por uma “caixa preta” (retângulo com fundo preto). Essa representação foi escolhida para reforçar o fato de que os usuários não podem olhar para dentro da “caixa” para saber o que está acontecendo durante o processamento. Eles realmente precisam aprender isso pelas falas do preposto do designer, que precisam ser cuidadosamente projetadas como a única forma de comunicar ao usuário o que aconteceu (ou está acontecendo), como e por quê. A Figura 21b ilustra a troca de turnos usuário-sistema, indicada pela sequência [cena, fala de transição do usuário, processo do sistema, fala de transição do preposto do designer].

Como resultado do processamento, o preposto do designer pode levar a conversa para uma cena ou, caso não haja mais nada para o usuário fazer para atingir a meta correspondente, pode encerrar a conversa com um monólogo, que deve ser apenas percebido e interpretado pelo usuário. Um monólogo é representado por uma caixa branca e seu conteúdo é redigido entre colchetes duplos, como em: << conteúdo >>²¹. A Figura 22a ilustra um monólogo do preposto do designer como resultado de um processamento de sistema sobre o cadastramento da disciplina²².

Cenas sem diálogos

Há cenas em que o preposto do designer fala sobre um ou mais signos, para que o usuário possa examiná-los e decidir como a conversa deve prosseguir dali. Nesse caso, o segundo compartimento contém apenas os signos, sem uma indicação do diálogo correspondente. A Figura 22b ilustra uma cena para gerenciar disciplinas, a partir da qual o

²¹ É importante observar que o texto entre colchetes não expressa as palavras exatas que o preposto do designer utilizará, apenas o conteúdo relevante a ser comunicado naquele momento da interação.

²² Geralmente, esse tipo de operação levaria a uma nova cena do tipo Visualizar disciplinas, onde o usuário veria todas as disciplinas cadastradas, ou ainda a uma cena Visualizar disciplina, onde o usuário veria os detalhes da disciplina que acabou de cadastrar e poderia iniciar outras conversas sobre a disciplina, tal como Cadastrar trabalho, por exemplo. O monólogo foi utilizado aqui para introduzir esse elemento, apesar de não constituir necessariamente uma boa solução de design.

usuário pode prosseguir para criar uma nova disciplina, modificar ou remover uma disciplina existente.

Nesse caso, pode-se considerar que o usuário está envolvido numa conversa no sentido atribuído por Schön (1983) à expressão “conversa com materiais”. Podemos considerar que esse tipo de conversa é composto de falas epistêmicas, enunciadas (ou simplesmente pensadas) pelo usuário para seu próprio entendimento do que lhe foi (ou está sendo) comunicado pelo seu interlocutor sobre os signos representados na cena. Vale observar que isso é diferente do monólogo do preposto do designer descrito acima.

Pode algo dar errado? Se der, o quê, e como o usuário pode se recuperar disso e prosseguir em direção ao atingimento de sua meta?

Em algumas situações, rupturas comunicativas podem ocorrer. Quando resultam de um problema ou falha no processamento do sistema (e.g. “Ué, o que houve?” ou “Por que não funciona?”), a fala do preposto do designer pode representar um caminho de recuperação. Falas de transição de recuperação de ruptura são representadas na MoLIC por linhas direcionadas tracejadas. A Figura 22c ilustra uma fala desse tipo quando o aluno tenta cadastrar uma disciplina com um código que já tinha sido cadastrado previamente.

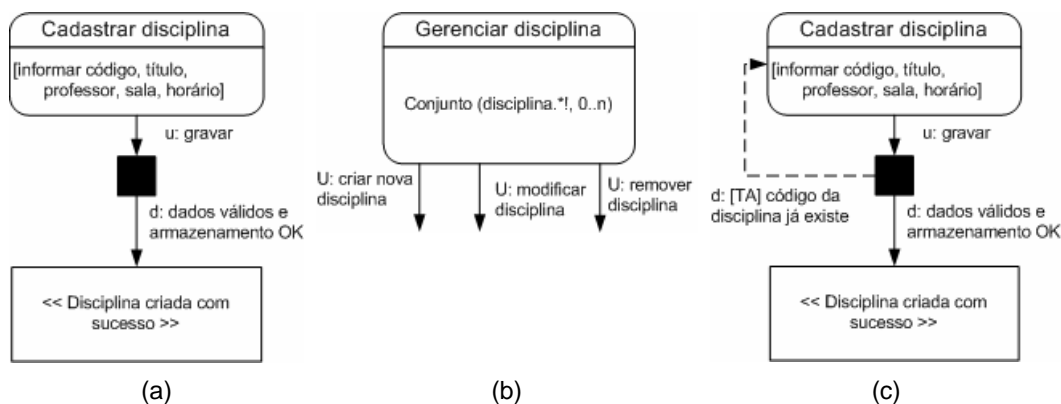


Figura 22. a) Monólogo do preposto como resultado de um processamento; b) Cena sem diálogos; c) Fala de transição de recuperação de ruptura.

Dizemos que uma ruptura ocorre quando a expressão da intenção do usuário não o leva a atingir o efeito pretendido com essa intenção (tal como presumido pelo preposto do designer), ou seja, a *perlocução* é inconsistente com a *ilocução*. Nesse caso, o usuário precisa desviar da interação na direção da meta a ser atingida e buscar um entendimento da própria interação. Em outras palavras, o usuário deve aprender, com o preposto, como expressar sua intenção de forma que o preposto possa “entendê-la” (i.e., processá-la para a realização bem-sucedida da meta tal como projetada, para satisfazer a intenção presumida do usuário). Como é natural e freqüente que mal-entendidos ocorram numa conversa, a EngSem destaca a importância de representar as rupturas comunicativas que podem ocorrer durante a interação. Mais do que tentar antecipar as rupturas possíveis, é necessário definir a forma como o preposto comunicará ao usuário que uma ruptura ocorreu e apoiá-lo na recuperação do problema, i.e., como o usuário deverá prosseguir com a conversa para atingir seus objetivos. Portanto, para cada momento de interação

no qual uma ruptura pode ser antecipada, o designer deve definir os mecanismos e falas de recuperação apresentadas aos usuários.

Pode-se observar o prefixo [TA] na Figura 22c. Isso indica o tipo de mecanismo oferecido aos usuários para se recuperarem da ruptura, tal como descrito na subseção sobre modelagem de tarefas. Nesse caso, o mecanismo oferecido é o de tratamento apoiado. É importante observar que nem todas as falas de recuperação de ruptura são enunciadas pelo preposto do designer. Quando uma fala de recuperação de ruptura é enunciada por um usuário, representa uma oportunidade explicitamente projetada para o usuário se recuperar de um caminho de interação acidental (não-intencional). Tais falas podem ser lidas como “Epa! (Não era isso que eu queria fazer)”.

Nos processos de sistema vistos até agora, o resultado do processamento só é comunicado após o processo ter sido concluído, através de uma fala de transição do preposto do designer. Para alguns processamentos, o designer pode querer comunicar ao usuário também o progresso do processamento ou seus estados intermediários. Essa **comunicação síncrona** se torna mais importante na medida em que a duração do processamento aumenta, como por exemplo durante o *download* de um arquivo ou a realização de uma busca. Nesses casos o preposto do designer pode emitir diversas falas durante o processamento. Para representar esse tipo de comunicação, representamos uma caixa branca acoplada à caixa preta, que representa falas sincronizadas com o processamento do sistema e sobre ele. A Figura 23 ilustra esse elemento.

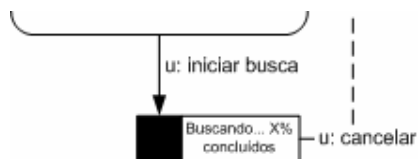


Figura 23. Comunicação síncrona sobre o progresso de um processamento de sistema em andamento.

É importante assegurar que a comunicação sobre um processamento seja uma indicação correta do estado desse processamento. Isso significa que deve haver uma relação causal entre o conteúdo que está sendo comunicado e a semântica do processamento. Além disso, ao informar o usuário sobre o progresso de um processamento em andamento, o designer agora deve ser capaz de oferecer ao usuário algum controle sobre esse processamento, tal como suspendê-lo, cancelá-lo ou ajustá-lo. Portanto, pode haver falas de transição de usuário saindo da caixa branca, como ilustrado na figura.

Quando um usuário pode “falar sobre” um tópico (i.e. travar uma conversa sobre um tópico)?

Para completar nosso exemplo sobre o cadastramento de uma disciplina, precisamos representar o momento de início da conversa sobre esse tópico específico. A abertura de conversas sobre tópicos específicos é representada por **acessos ubíquos**, falas de transição de usuário a partir de cenas anônimas com fundo cinza. Um acesso ubíquo significa que, a partir de qualquer cena, e em qualquer momento da aplicação, tal conversa pode ser iniciada. A Figura 24a ilustra um acesso ubíquo para a cena Cadastrar disciplina.

Examinando um diagrama, o designer pode perceber que não faria sentido para o usuário travar uma determinada conversa. Por exemplo, pode não fazer sentido tentar

agendar uma reunião de estudos para uma prova que já passou²³. Para restringir os momentos na interação onde o usuário pode travar certas conversas, uma fala de transição pode ser representada com um ou mais **pressupostos** (pré-condições). Esses são representados também como texto na fala de transição, precedidos pela palavra-chave *pre*. Eles podem ser redigidos em linguagem natural ou numa linguagem artificial escolhida pelo designer. Por exemplo:

pre: data de prova anterior à data do sistema
[u: agendar reunião de estudos para prova]

Qual é a correspondência entre um diagrama MoLIC e o diagrama de metas?

Até agora, não mostramos como o diagrama MoLIC está relacionado com as metas representadas no diagrama de metas, ou seja, como as seqüências de ilocuções estão associadas às perlocuções globais. Apesar de Cadastrar disciplina ser o mesmo texto que identifica uma meta e um tópico de cena, não está claro quando a meta é considerada atingida ou qual é o caminho completo de interação correspondente à meta. No diagrama MoLIC, representamos as metas utilizando uma forma geométrica com fundo cinza claro envolvendo os elementos da MoLIC que correspondem à meta, com uma identificação da meta em um de seus lados (Figura 24b).

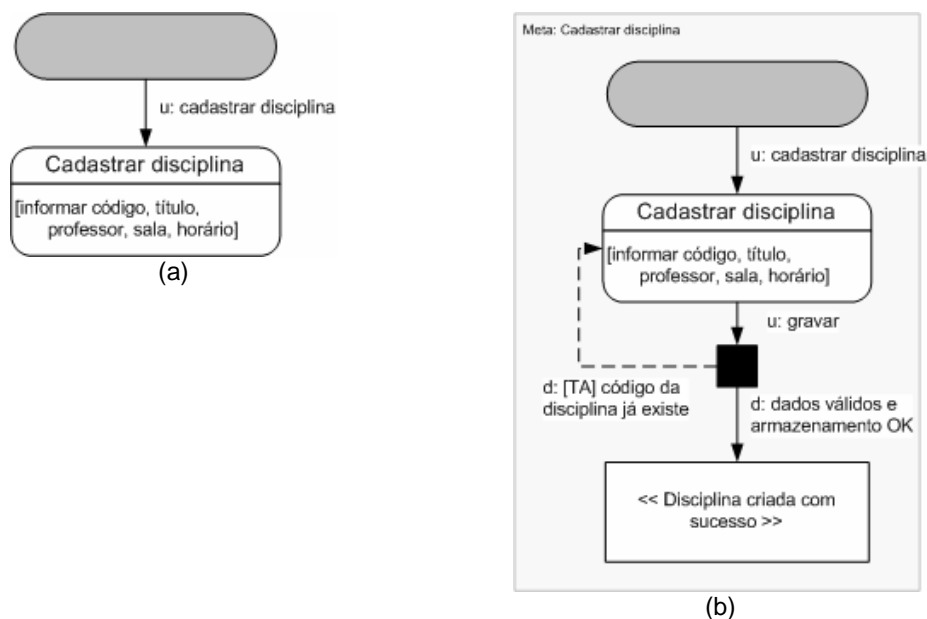


Figura 24. a) Acesso ubíquo para a cena Cadastrar disciplina; b) Correspondência entre um trecho de diagrama MoLIC e o diagrama de metas.

A identificação das metas no diagrama MoLIC aumenta a rastreabilidade entre os modelos. Isso facilita a avaliação do impacto de correções e revisões que podem

²³ Não existe ainda um método de avaliação formativa de diagramas de interação, que organize e sistematize as reflexões relevantes sobre a solução de interação sendo projetada. No ano de publicação deste capítulo, uma proposta de método para isto está sendo elaborada por membros do grupo de pesquisa em EngSem, o SERG.

ocorrer, assim como a manutenção de consistência entre os modelos, a cada nova versão.

Pontos de contato

Alguns diálogos durante a interação usuário-sistema têm como interlocutor um ator externo ao contexto imediato da interação. Quando isso ocorre, o designer deve representar as influências entre o usuário e esses atores externos [Silva & Barbosa 2004]. Após construir os diagramas de interação para cada papel, representamos **pontos de contato** entre os diagramas para indicar a influência de um ator A1 com o sistema sobre a interação de um outro ator A2 com o sistema. As falas chegando a ou partindo de um ponto de contato podem ser originadas ou direcionadas a qualquer ponto no diagrama, seja processamento ou cena. Cada ponto de contato é representado graficamente por um círculo com um rótulo indicando o papel correspondente no diagrama alvo. A Figura 25 ilustra os pontos de contato de uma aplicação de publicação eletrônica, na qual um autor redige um texto e o submete para o editor aprová-lo ou enviar de volta ao autor para revisão.

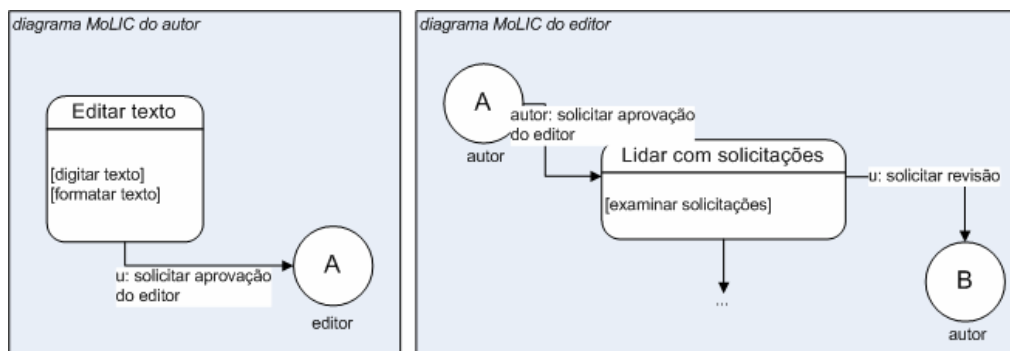


Figura 25. Ponto de contato entre dois diagramas MoLIC.

Apoiando decisões de design

Esta subseção apresenta brevemente dois exemplos do uso da MoLIC para ajudar a descrever algumas decisões de design que a MoLIC objetiva apoiar.

Exemplo1: solicitar ou não confirmação. A partir de uma cena onde o preposto do designer pede ao usuário para informar alguns dados, o designer pode considerar as seguintes opções (Figura 26):

- armazenar os valores fornecidos pelo usuário e prosseguir com a interação (supondo improvável que o usuário informe um valor incorreto);
- solicitar uma confirmação do usuário, levando-o a verificar os dados que forneceu e somente após a confirmação armazená-los; ou
- armazenar os dados, mas, em vez de prosseguir com a interação, mostrar os dados armazenados para dar ao usuário uma chance de verificar e corrigir os valores, se necessário.

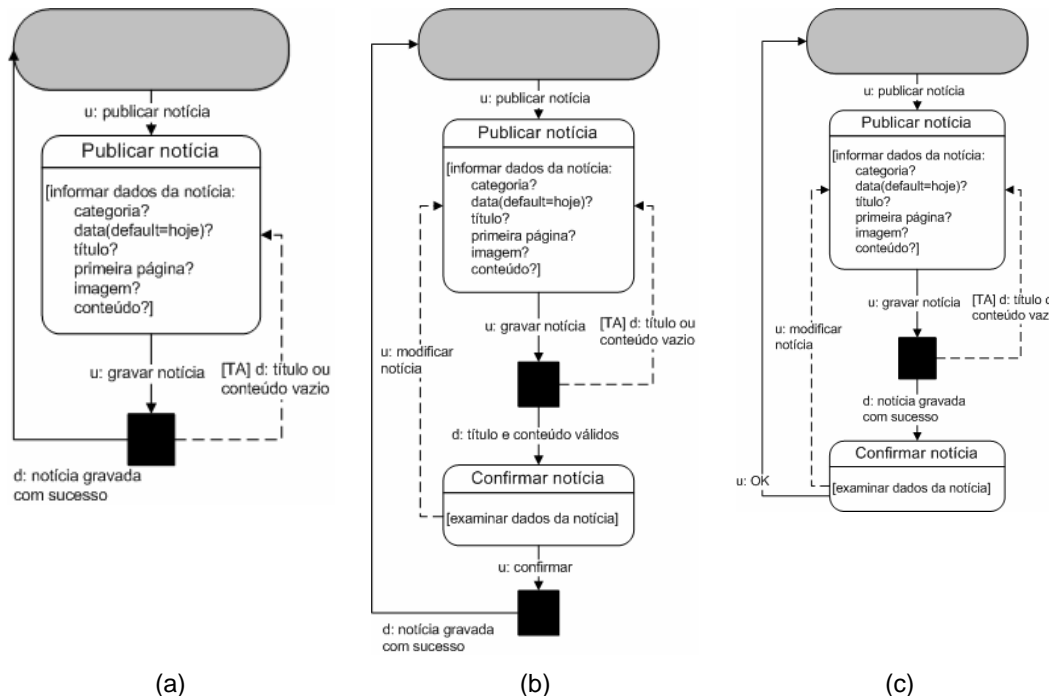


Figura 26. Opções para uma meta de "publicar notícia ": a) grava sem pedir confirmação; b) pede confirmação antes de gravar; c) grava e permite verificação.

Exemplo 2: Articulando a conversa. Quando um tópico envolve subtópicos, há pelo menos duas opções (Figura 27):

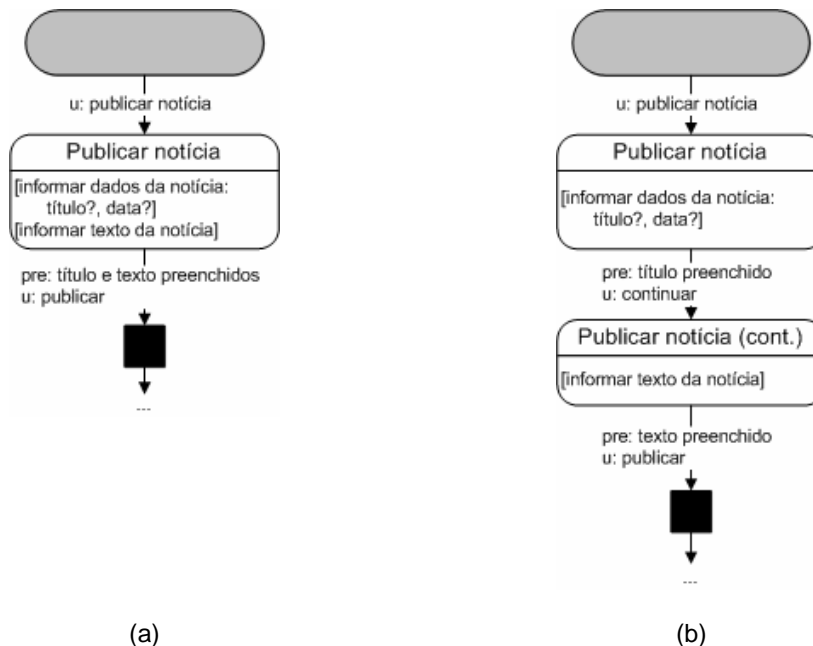


Figura 27. Disposição de diálogos a) numa única cena ou b) em diferentes cenas.

- a) oferecer múltiplos diálogos numa mesma cena, cada qual correspondendo a um subtópico diferente; ou
- b) projetar múltiplas cenas, ligadas por falas de transição de usuário, tal como num assistente ou *wizard*.

Essa situação também pode ocorrer quando o objetivo do usuário é simplesmente explorar signos de diferentes formas, envolvendo operações de navegação, sem processamento.

6.4.4 Modelagem do Sistema de Ajuda

O sistema de ajuda é na EngSem uma forma de comunicação privilegiada entre designer e usuários, uma vez que é uma comunicação direta [Silveira et al. 2004, Silveira 2002]. Portanto, é fundamental que se faça o projeto do sistema de ajuda para que o usuário possa entender melhor a solução do designer, se recuperar de problemas que porventura venham a ocorrer e fazer melhor uso do sistema. O modelo fundamentado em EngSem propõe que o sistema de ajuda seja apresentado de forma contextual, em camadas e sob demanda. O acesso ao sistema de ajuda é feito através de perguntas, dentre elas algumas do conjunto proposto no método de comunicabilidade (e.g. “O que é isto?”, “Cadê?”), adicionado de outras específicas para o entendimento da comunicação designer-usuário (e.g. Para que serve isto? Por que tenho que fazer isto?).

Como complemento aos modelos de tarefa e de interação, devemos representar diversas outras informações necessárias à elaboração do conteúdo de ajuda a ser disponibilizado no sistema. Em estudos realizados sobre sistemas de ajuda, foram compiladas dúvidas freqüentes dos usuários ao utilizarem sistemas interativos, classificadas de acordo com o tipo de dúvida [Baecker et al. 1995, Sellen & Nicol 1990] (Tabela 7).

Tabela 7. Tipos de dúvidas freqüentes dos usuários.

Informativas	O que posso fazer com este programa?
Descritivas	O que é isto? O que isto faz?
Procedimentais	Como eu faço isto?
De escolha	O que posso fazer agora?
Sugestivas	O que devo fazer agora?
Investigativas	O que mais devo fazer? Esqueci algo?
Interpretativas	O que está acontecendo agora? Por que isto aconteceu?
Navegacionais	Onde estou? De onde vim?
Históricas	O que eu já fiz?
De motivação	Por que devo usar este programa? Como ele irá me beneficiar?

Algumas destas perguntas podem (e devem!) ser respondidas durante as etapas iniciais de análise e modelagem de usuários e tarefas. Além disto, respostas a algumas destas perguntas podem ser extraídas diretamente dos modelos já construídos, mas outras devem ser explicitamente elaboradas pelos projetistas. Nesta seção, analisaremos as perguntas de acordo com os modelos que especificam a resposta, os elementos da aplicação a que estão relacionadas, ou com a etapa de desenvolvimento em que se pode respondê-las.

A pergunta **O que posso fazer com este programa?** pode ser respondida relacionando-se as metas dos usuários identificadas nos cenários e organizadas no diagrama de metas.

Questões descritivas, do tipo **O que é isto?**, **O que isto faz?** e **O que se faz com isto?** podem ser respondidas em dois níveis: um nível conceitual, relacionado ao domínio, e um nível operacional, relacionado ao modelo de interface. O nível conceitual pode ser descrito no início do processo de desenvolvimento, enquanto o nível operacional só poderá ser definido junto com os modelos de interação e de interface.

As perguntas procedimentais, de escolha, sugestivas e investigativas estão vinculadas principalmente ao modelo de tarefas. A estrutura hierárquica de uma tarefa representa uma resposta de alto nível à pergunta **Como eu faço isto?**. Uma resposta mais detalhada pode ser obtida nos modelos de interação e de interface, descrevendo os caminhos percorridos pelo usuário e os *widgets* ou elementos de interface que devem ser manipulados a cada instante. Para responder a pergunta **O que posso fazer agora?**, o sistema deve verificar quais são as tarefas cujas pré-condições estão satisfeitas naquele momento. Por outro lado, as perguntas **O que devo fazer agora?**, **O que mais devo fazer?** e **Esqueci algo?** requerem que o sistema mapeie as últimas ações do usuário nas estruturas de tarefas conhecidas, e procure identificar em que tarefa o usuário se encontra no momento e quais ações ou subtarefas faltam para completá-la.

As perguntas interpretativas, navegacionais e históricas, por sua vez, estão relacionadas principalmente ao modelo de interação, que representa os caminhos que o usuário pode seguir e que indicações o sistema fornece sobre onde ele se encontra a cada instante. A pergunta **O que está acontecendo agora?** está relacionada a mecanismo de *feedback*. Já a pergunta **Por que isto aconteceu?** deve ser respondida pela descrição do resultado esperado da ação que acaba de ser realizada, juntamente com os possíveis problemas que venham a ocorrer. As perguntas **Onde estou?**, **De onde vim?** e **O que eu fiz?** requerem um acompanhamento do histórico de interação do usuário.

Finalmente, as perguntas relacionadas à motivação **Por que devo usar este programa?** e **Como ele irá me beneficiar?** não podem ser derivadas diretamente dos modelos e, portanto, precisam ser explicitamente respondidas pelos projetistas. Estas perguntas dizem respeito à aplicação como um todo, e não a uma tarefa específica. Elas devem ser respondidas no início do processo de desenvolvimento, como registro da intenção de *design*, e depois revisadas para verificar se todos os objetivos iniciais foram atingidos.

6.5. Considerações Finais

Neste capítulo, apresentamos a teoria da EngSem e vimos métodos de avaliação e modelos de design de IHC que nela se baseiam. Nesta seção, descrevemos brevemente algumas questões de pesquisa da EngSem em diferentes domínios, como programação feita por usuário final, sistemas colaborativos, sistemas educacionais e sistemas de informação geográfica.

6.5.1 EngSem e Programação Feita por Usuários Finais

Este capítulo apresentou técnicas, modelos e representações desenvolvidos ou adaptados com o objetivo de oferecer sistemas interativos de alta qualidade de uso, que satisfaçam as necessidades dos usuários, respeitando seus valores e sua cultura. Até aqui trabalhamos com uma abordagem que instrumenta o designer para capturar essas necessidades e valores e codificá-los numa tecnologia que os apóie na realização de

suas atividades e no alcance de seus objetivos. No entanto, é fato que os designers não podem antecipar todas as possíveis atividades que um usuário precisará realizar com o apoio dos sistemas interativos que eles projetam. Para oferecer uma correspondência duradoura entre as intenções e necessidades dos usuários e as possibilidades tecnológicas é preciso dar mais poder aos usuários para que eles próprios possam customizar e estender os sistemas interativos de forma que contemplem suas necessidades ao longo do tempo.

Os computadores são máquinas de processamento simbólico nas dimensões léxica, sintática e semântica. Os seres humanos, por sua vez, para se comunicar uns com os outros, manipulam signos nas dimensões de intenção, conteúdo e expressão. De acordo com de Souza & Barbosa (2006), não existe uma correspondência das dimensões nessas diferentes perspectivas. A grosso modo, as dimensões de expressão humana estão relacionadas às dimensões léxica e sintática dos computadores, ao passo que a dimensão semântica do processamento computacional está relacionada às dimensões de conteúdo e intenção da comunicação humana. Para projetar sistemas computacionais com alta usabilidade, os designers precisam [de Souza 2005]: i) sintetizar um sistema de significação para apoiar a IHC; ii) comunicar sua visão de design através desse sistema de significação; iii) comunicar as regras e princípios de acordo com os quais certas expressões são sistematicamente associadas a certos conteúdos para atingir uma gama específica de intenções; iv) comunicar se e como tais princípios e regras podem ser modificados; e v) comunicar como os significados modificados podem ser utilizados de forma eficiente na interação com a aplicação.

Para se beneficiarem das qualidades desse tipo de design, os usuários precisam [de Souza & Barbosa 2006]: i) entender o sistema de significação projetado; ii) formular uma hipótese satisfatória sobre como os significados são codificados nesse sistema; iii) dominar seu uso para comunicar as intenções ao sistema e atingir uma variedade de objetivos com isso; iv) formular uma hipótese satisfatória sobre quais novos significados (ou modificações de significados) podem ser codificados e como; e v) codificar tais significados no sistema e incorporá-los às variedades possíveis de discurso interativo com a aplicação.

6.5.2 EngSem e Sistemas Colaborativos

Com a popularização da Internet, houve também um grande crescimento da oferta e uso de ambientes que promovem interação entre pessoas, sejam eles para fins sociais, de ensino ou de trabalho. Nestes ambientes, o usuário deve interagir não apenas com o sistema, mas com outros usuários através do sistema. Assim, os projetos destes sistemas e de suas interfaces lidam com novas questões relativas à comunicação, coordenação e colaboração. Como tratar estas questões no projeto e avaliação de interface é um desafio sendo tratado tanto na área de IHC, quanto de CSCW (*Computer Supported Collaborative Work*, ou sistemas colaborativos).

Na EngSem, a interface de um sistema colaborativo continua sendo uma mensagem do projetista para os usuários. No entanto, neste caso o projetista não está se comunicando apenas com um usuário, mas com todo um grupo. Assim além de sua mensagem ser capaz de transmitir aos usuários quem o projetista acredita que eles sejam, os seus propósitos para utilizar o sistema e como interagir com este sistema, ela deve também transmitir aspectos da solução relacionados com o grupo. A mensagem para o

grupo deve comunicar quem é o grupo, e os membros que o formam, quais as suas necessidades, com quem podem se comunicar através do sistema e através de que linguagem e protocolos o fazem [Prates 1998, de Souza 2005, Barbosa 2006].

As ferramentas epistêmicas para projetistas de ambientes multi-usuário devem apoiá-los na reflexão sobre a sua meta-comunicação com os usuários membros do grupo. A EngSem atualmente conta com diferentes tipos de ferramentas epistêmicas específicas para sistemas colaborativos. A primeira delas é a descrição de três metáforas que podem ser utilizadas com base na apresentação de um sistema colaborativo [de Souza 2005]. A segunda, um conjunto de questões a serem propostas para guiarem a avaliação por inspeção de um sistema colaborativo, mas que também poderia ser utilizado para apoiar o designer nas suas decisões de projeto [de Souza 2005]. Finalmente, modelos com o objetivo de apoiar o projetista na definição da sua solução para o projeto da interação entre membros de um sistema colaborativo [Prates 1998, Barbosa 2006].

Dentre estas ferramentas, damos destaque à Manas, uma ferramenta capaz de apoiar o designer na descrição de sua solução [Barbosa 2006]. O modelo de arquitetura proposto é composto por uma linguagem de design, utilizada para descrever a metacomunicação prevista; um interpretador para esta linguagem; e uma base de conhecimento para armazenar as decisões do designer sobre sua solução. A Manas propõe determinadas dimensões de análise para motivar a reflexão do designer enquanto ele descreve a sua solução. Desta forma, fornece indicadores qualitativos sobre as decisões tomadas, em particular no que tange a potenciais impactos sociais que essas decisões possam ter no grupo de usuários.

6.5.3 EngSem e Sistemas Educacionais

Um importante aspecto no projeto de interfaces de apoio ao aprendizado é que várias das decisões relativas a estratégias de ensino e planos pedagógicos cabem ao educador, e não ao projetista. Assim, idealmente, o projeto é feito por uma equipe composta (no mínimo) pelo educador e pelo projetista. A implicação disto para a teoria da EngSem [Prates et al. 2003, Prates 2004] é que neste caso a interface é uma meta-mensagem conjunta do educador e do projetista para os alunos. Conseqüentemente, o preposto representa dois emissores distintos, cada um transmitindo a sua solução para um determinado aspecto do problema, e estas integradas em um único discurso.

Para a EngSem, esta co-autoria do projetista e do educador tem uma consequência sobre os aspectos em que as ferramentas epistêmicas devem apoiar os emissores da mensagem. Os problemas e espaços de solução desses emissores são distintos, embora fortemente relacionados. Por isto, essas ferramentas devem também apoiar a comunicação entre educador e projetista sobre suas mensagens, a sua integração, e a reflexão sobre a mensagem como um todo.

6.5.4 EngSem e Sistemas de Informação Geográfica

Sistemas de Informação Geográfica (SIGs) apresentam desafios interessantes para a EngSem. São sistemas que manipulam dados e objetos geográficos, ou seja, dados e objetos cujas características inerentes incluem a sua localização geográfica. Uma característica distintiva dos SIGs de especial interesse para a EngSem é o uso de interfaces baseadas em mapas, parcial ou completamente co-referentes, e em várias escalas

[Seixas 2004]. Seixas endereça características específicas desse tipo de sistema, como “a percepção de orientação, de navegação, a identificação de objetos e a interpretação da simbologia”. Ela parte da hipótese que “rupturas ou falhas na semiose do usuário, decorrentes de apresentações não relacionadas com tarefas e contextos, correspondem a rupturas no contínuo semiótico entre as linguagens de representação ou descrição da interface”. Seixas explora questões interessantes de escalas e abstração, *zoom* inteligente, metáforas, e múltiplas representações. Ela endereça questões como: “Como manter o foco da tarefa durante a interação com mapas? Quais elementos precisam ser mantidos visíveis para que o usuário não se perca? Como modificar a apresentação dos objetos de modo que o usuário consiga manter a referência do objeto?”. Seu trabalho aponta ainda para a necessidade de se considerar novas interpretações para expressões que representam rupturas na comunicação usuário-sistema de aplicações tradicionais, e que devem ser especializadas no contexto de domínios específicos como interfaces baseadas em mapas.

6.5.5 A EngSem no Cenário de IHC

Recentemente, a área de IHC vem ganhando teorias inspiradas na antropologia e na psicologia social, como por exemplo a teoria da atividade [Kaptelinin, Nardi & Macauley 1999] e cognição distribuída [Hollan, Hutchins & Kirsh 2000]. De acordo com de Souza (2005), essas teorias expandiram consideravelmente o escopo de análise apoiado pelas teorias cognitivas tradicionais. Nas palavras de de Souza, elas “ofereceram um rico relato da experiência do usuário *no mundo*, onde a tecnologia computacional está presente, e expandiram os horizontes dos designers com relação a para quem eles estavam projetando e por quê. No entanto, nenhuma dessas teorias foi suficientemente clara sobre como esses horizontes expandidos se traduzem em produtos de design. O preposto do designer lhes oferece uma ponte para a natureza de artefatos de base computacional e uma oportunidade de explorar o conteúdo e a forma do seu discurso interativo para incluir as dimensões que cada teoria promove como distintiva” [ibid.:91-2].

No cenário internacional, a EngSem teve seu reconhecimento com a publicação do livro sobre a teoria pela MIT Press [de Souza 2005] e alguns de seus métodos sendo publicados em livros básicos de IHC no exterior [Preece et al. 2007]. No Brasil, a inclusão da EngSem como um tópico recomendado para o ensino de IHC na graduação dos cursos de computação também demonstra sua aceitação pela comunidade acadêmico-científica nacional. Para saber mais sobre a EngSem, recomendamos acessar o site do grupo de pesquisa em EngSem, SERG, em <http://www.serg.inf.puc-rio.br>.

Referências Bibliográficas

- Adler, P. & Winograd, T. (eds., 1992) *Usability: Turning Technologies into Tools*. Oxford University Press. New York, NY.
- Annett, J., & Duncan, K. D. (1967). “Task analysis and training design”. *Journal of Occupational Psychology*, 41, 211-221.
- Araujo, A. C. I. C. (2005) Mecanismos de Customização no Ambiente de Discussão OriOn. Trabalho Final. Departamento de Informática, PUC-Rio.
- Baecker, R.M. et al. (1995) *Readings in Human-Computer Interaction: toward the year 2000*. San Francisco: Morgan Kaufmann Publishers, Inc.

- Barbosa, S.D.J.; Paula, M.G. (2003) Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering. In *10th International Workshop on Design, Specification and Verification of Interactive Systems, DSV-IS 2003*, Madeira Island, Junho 11-13, 2003, LNCS 2844, pp. 16–33.
- Barbosa, C. M. A. (2006) *Uma Ferramenta Epistêmica de Apoio ao Projeto da Comunicação em Sistemas Colaborativos*. Tese de doutorado. Departamento de Informática. PUC-Rio. Rio de Janeiro.
- Borman, L. (1996) “SIGCHI: The Early Years”. *SIGCHI Bulletin*. Vol.28 No.1, January 1996. <http://sigchi.org/bulletin/1996.1/borman.html>
- Carroll, J.M. (2000) *Making use: scenario-based design of human-computer interactions*. Cambridge, MA: The MIT Press.
- Carroll, J. M. (2003) *HCI Models, Theories, and Frameworks: Toward a multidisciplinary science*. San Francisco Morgan Kaufmann Publishers.
- Carroll, J.M.; Mack, R.L.; Robertson, S.P.; Rosson, M.B. (1994) “Binding Objects to Scenarios of Use”, *International Journal of Human-Computer Studies* **41**:243-276. Academic Press.
- CFCS (2004) – Committee on the Fundamentals of Computer Science: Challenges and Opportunities, National Research Council. *Computer Science: Reflections on the Field, Reflections from the Field*, 208 pages, National Academies Press, <http://www.nap.edu/catalog/11106.html> [última visita: fev/2007]
- CNS - Conselho Nacional de Saúde. Resolução 196/96. Disponível em <http://conselho.saude.gov.br/resolucoes/1996/Reso196.doc> [último acesso: fev/2007].
- de Souza, C. S.; Prates, R. O.; Carey, T. (2000) Missing and declining affordances: Are these appropriate concepts?. *Journal of the Brazilian Computer Society*. 1(7): 26-34.
- de Souza, C. S. (2005) *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, MA: The MIT Press.
- de Souza, C.S. & Barbosa, S.D.J. (2006) “A Semiotic Framing for End-User Development”. In: Henry Lieberman et al. (eds.) *End User Development*, 401-426. The Netherlands: Springer.
- de Souza, C. S., Leitão, C. F., Prates, R. O., da Silva, E. J. (2006) The Semiotic Inspection Method. *VII Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2006*.
- Decreto-Lei 5296 – Lei de Acessibilidade. Disponível em: <http://www.acessobrasil.org.br/index.php?itemid=43>. [Última visita: fev/2007].
- Diaper, D. & Stanton, N. (eds., 2004) *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Dourish, P. (2001). *Where the action is*. Cambridge, MA: The MIT Press.
- Eco, U. (1976). *The Theory of Signs*. Bloomington: Indiana University Press.
- Gould, J., Lewis, C. (1985) Designing for usability: key principles and what designers think. *Communications of the ACM*, Volume 28 , Issue 3 , pp 300 – 311
- Grice, H.P. (1975) Logic and Conversation. In Cole & Morgan (eds.) *Syntax and Semantics, Speech Acts*, vol. 3, 41-58. New York, NY: Academic Press.

- Hollan, J.E.; Hutchins, D.; Kirsh, D. (2000) Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction* 7(2): 174-196.
- Houser, N. & Kloesel, C. (1992-1998) *The Essential Peirce, Vols. I & II*. Bloomington, IN. Indiana University Press.
- Jakobson, R. (1960) Linguistics and Poetics. In: Sebeok (ed.) *Style in Language*, 350-377. Cambridge, MA: The MIT Press.
- Kaptelinin, V.; Nardi, B. & Macauley, C. (1999) The activity checklist: A tool for representing the “space” of context. *Interactions* 6(4): 27-39.
- Konstan, J (2007). Presidente do Special Interest Group in Computer Human Interaction (SIGCHI) da Association of Computing Machinery (ACM). Comunicação pessoal.
- Leech, G. (1983) *The Principles of Pragmatics*. London: Longman.
- Leitão, C, F. e Dias-Romão, D. (2003). Pesquisas em IHC: um debate interdisciplinar sobre a ética. Em Nicolaci-da-Costa, A. M. E Leite, J. C. *Atas do Workshop sobre Interdisciplinaridade em IHC, CLIHC 2003*.
- Moran, T. (1981) “The Command Language Grammars: a representation for the user interface of interactive computer systems. In: *International Journal of Man-Machine Studies* 15:3-50, Academic Press.
- Nielsen, J. (1993) *Usability Engineering*. Academic Press.
- Nielsen, J. (2000) *Test with 5 Users*, Alertbox. Disponível online em <http://www.useit.com/alertbox/20000319.html> [último acesso: fevereiro/2007].
- Paula, M. G.; Silva, B. S.; Barbosa, S. D. J. (2005) Using an Interaction Model as a Resource for Communication in Design. *Proceedings of CHI 2005, Extended abstracts Volume*. Portland, OR, USA.
- Paula, M.G. (2003) *Projeto da Interação Humano-Computador Baseado em Modelos Fundamentados na EngSem: Construção de um Modelo de Interação*. Dissertação de Mestrado. Departamento de Informática. PUC-Rio.
- Prates, R. O. (1998) *A EngSem de linguagens de interfaces multi-usuário*. Tese de doutorado. Departamento de Informática. PUC-Rio. Rio de Janeiro.
- Prates, R. O.; de Souza, C. S.; Barbosa, S. D. J. (2000) A method for evaluating the communicability of user interfaces. *Interactions*. 7(1): 31-38
- Prates, R. O., Barbosa, S. D. J. (2003) *Avaliação de Interfaces de Usuário - Conceitos e Métodos*. Jornada de Atualização em Informática, SBC.
- Prates, R. O.; Figueiredo, R. M. V. de; Bach, C. F. (2003) Um Modelo de Apoio ao Projeto de Design de Interfaces de Ambientes de Aprendizado. *Anais do WIE, SBC, 2003*. V.541- 552.
- Prates, R. O. (2004) A EngSem para o Domínio Educacional, *Atas da Oficina “Design e Avaliação de Interfaces para Ambientes Educacionais”* organizado no IHC 2004. (Disponível em <http://www.ime.uerj.br/~raquel/WIEd>).
- Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S.; Carey, T. (1994) *Human-computer interaction*. Reading, MA. Addison-Wesley.
- Preece, J.; Rogers, Y.; Sharp, E. (2002) *Interaction Design: Beyond Human-computer Interaction*. New York, NY: John Wiley & Sons, 1st edition.

- Preece, J.; Rogers, Y.; Sharp, E. (2007) Case Studies in Website material for *Interaction Design: Beyond Human-computer Interaction Website*. New York, NY: John Wiley & Sons, 2nd edition. Disponível em: http://www.id-book.com/casestudy_14-1.htm [última visita em fevereiro de 2007].
- Preece, J.; Rogers, Y.; Sharp, E.; Benyon, D.; Holland, S.; Carey, T. (1994) *Human-Computer Interaction*. Harlow, England: Addison-Wesley.
- Rosson, M.B. & Carroll, J.M. (2002) *Usability Engineering: scenario-based development of human-computer interaction*. San Francisco, CA: Morgan Kaufmann.
- Schön, D.A. (1983) *The Reflective Practitioner*. New York, NY: Basic Books.
- Schön, D. and Bennett, J. (1996). Reflective conversation with materials. In Winograd, T. (ed.) *Bringing design to software*. New York, NY. Addison-Wesley. Pp. 171-184.
- Seixas, M. L. A. (2004) *Um Método de Avaliação para Interfaces Baseadas em Mapas*. Tese de Doutorado. Departamento de Informática. PUC-Rio. 2004.
- Sellen, A.; Nicol, A. (1990) Building User-Centered On-line Help. In Laurel, B. *The Art of Human-Computer Interface Design*. Reading: Addison-Wesley.
- Shneiderman, B., Plaisant, C. (2005). Chapter 1: Usability of Interactive Systems. In *Designing the User Interface*. Addison Wesley.
- Silva, B.S. (2005) *MoLIC Segunda Edição: revisão de uma linguagem para modelagem da interação humano-computador*. Dissertação de Mestrado. Departamento de Informática. PUC-Rio. 2005.
- Silva, B.S. & Barbosa, S.D.J. (2007) *Designing Human-Computer Interaction With MoLIC Diagrams – A Practical Guide*. In C.J.P. Lucena (ed.) *Monografias em Ciência da Computação*. Departamento de Informática. PUC-Rio. 2007.
- Silva, B. S.; Barbosa, S. D. J. (2004) Modelando a Interação do Nita: um estudo de caso e extensões ao MoLIC. *Anais do VI Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2004*. Curitiba, outubro de 2004.
- Silveira, M.S. (2002) *Metacomunicação Designer-Usuário na Interação Humano-Computador*. Tese de Doutorado. Departamento de Informática. PUC-Rio.
- Silveira, M.S. (2007) Relatório do Grupo de Discussões sobre Currículo de IHC. *VII Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2006*. Disponível em: <http://www.sbc.org.br>.
- Silveira, M.S.; Barbosa, S.D.J.; de Souza, C.S. (2004) “Designing online help systems for reflective users”. In: *Journal of the Brazilian Computer Society, JBCS*, vol. 9, No. 3, abril de 2004. pp.25-38.
- Tesoro Software Student Life Product. Student Life v. 3.0 Disponível em: <http://www.tesorosoft.com/studentlife.htm> [última visita: fev/2007]
- Vanderheiden, G. (2000) Fundamental Principles and Priority Setting for Universal Usability. *Conference on Universal Usability*. Pages 32-38.
- W3C Web Acessibility Initiative. Disponível em: <http://www.w3.org/WAI> [última visita: fev/2007]