

INTERAÇÃO HUMANO-COMPUTADOR



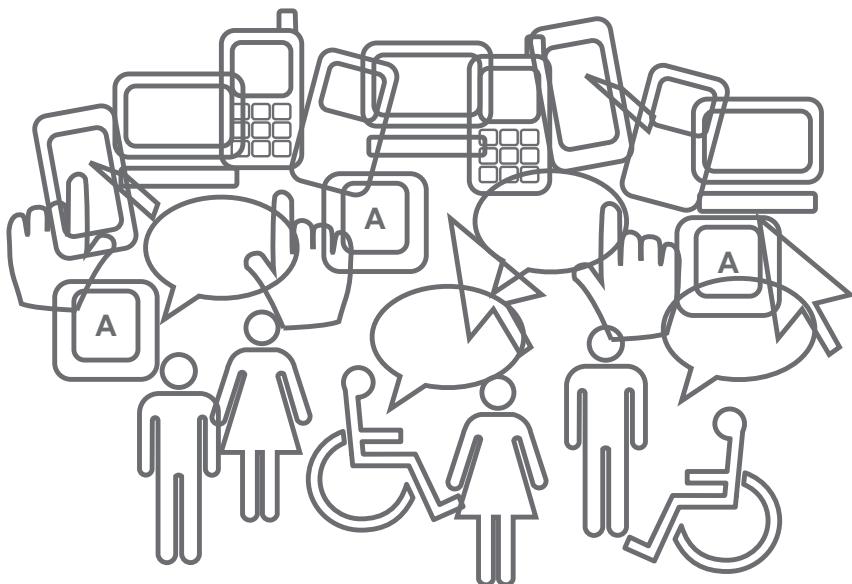
Preencha a **ficha de cadastro** no final deste livro
e receba gratuitamente informações
sobre os lançamentos e as promoções da
Elsevier Editora.

Consulte também nosso catálogo
completo e últimos lançamentos em
www.elsevier.com.br

Simone Diniz Junqueira Barbosa

Bruno Santana da Silva

INTERAÇÃO HUMANO-COMPUTADOR



© 2010, Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei no 9.610, de 19/02/1998.

Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Copidesque: Adriana Araújo Kramer

Revisão: Marco Antônio Corrêa

Elsevier Editora Ltda.

Conhecimento sem Fronteiras

Rua Sete de Setembro, 111 – 16º andar

20050-006 – Centro – Rio de Janeiro – RJ – Brasil

Rua Quintana, 753 – 8º andar

04569-011 – Brooklin – São Paulo – SP

Serviço de Atendimento ao Cliente

0800-0265340

sac@elsevier.com.br

ISBN 978-85-352-3418-3

Nota: Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação ao nosso Serviço de Atendimento ao Cliente, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso desta publicação.

CIP-BRASIL. CATALOGAÇÃO-NA-FONTE
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

B212i

Barbosa, Simone Diniz Junqueira

Interação humano-computador / Simone Diniz Junqueira Barbosa, Bruno Santana da Silva. –
Rio de Janeiro: Elsevier, 2010.

il. - (Série SBC, Sociedade Brasileira de Computação)

Inclui bibliografia

ISBN 978-85-352-3418-3

1. Interação homem-máquina. 2. Informática - Aspectos sociais. 3. Tecnologia - Aspectos sociais.
4. Comunicação e tecnologia. 5. Tecnologia da informação. I. Barbosa, Simone D. J. (Simone Diniz
Junqueira). II. Sociedade Brasileira de Computação. III. Título. IV. Série.

10-2633.

CDD: 004.019

CDU: 004.5

Aos meus filhos, Gabriel e Eduardo.

S. D. J. Barbosa

À minha família.

B. S. da Silva

Agradecimentos

Muitas pessoas contribuíram para tornar este livro uma realidade. Somos afortunados por termos ao nosso redor pesquisadores e profissionais altamente qualificados, alunos interessados e motivados, amigos e familiares queridos.

Este livro não existiria se não fosse nossa eterna professora e amiga, Clarisse Sieckenius de Souza, que nos despertou o interesse pela área de IHC e sempre nos apoia e nos motiva a aprender mais e a contribuir mais para a área. Interação Humano-Computador no Brasil não seria a mesma sem seu pioneirismo e seriedade.

Somos gratos a diversas pessoas com quem colaboramos ao longo de todos os nossos anos de atuação na área: a Carla Faria Leitão, Elton José Silva, Jair Cavalcanti Leite, Milene Selbach Silveira, Raquel Oliveira Prates, Sérgio Roberto Pereira da Silva e todos aqueles que passaram pelo *Semiotic Engineering Research Group* (SERG). Alunos que deveriam aprender conosco muito nos ensinaram. Gostaríamos de agradecer aos egressos do SERG e do Departamento de Informática da PUC-Rio: Adéle, Ana Carolina, André Luiz, Andréia, Ariane, Clarissa, Gilda, Gustavo, Hildebrando, José Eurico, Luciana, Maíra, Marcus, Otávio, Rafael, Rodrigo, Sandra, Sílvia, Thaís, Ugo, Vinícius e Viviane.

Agradecemos ao Departamento de Informática da PUC-Rio, que sempre apoiou a área de IHC, firmando seu pioneirismo ao oferecer várias disciplinas de graduação e pós-graduação há mais de uma década e ao reconhecer IHC como uma área de pesquisa legítima, ainda enquanto a área era pouco difundida no país.

Nossos pais, Marleny, Carlos, Lélia e Oriovaldo, são grandes responsáveis pelos nossos méritos e por tudo o que alcançamos. Durante os longos meses de pesquisa e redação deste livro, nossos familiares e amigos nos apoiaram incondicionalmente. A todo tempo sentimos sua carinhosa presença, apesar do isolamento e da privação do convívio que esse tipo de empreitada exige dos autores. Luís Fernando, Gabriel, Eduardo, Kamila: este livro também é de vocês.

Finalmente, gostaríamos de agradecer novamente a Clarisse Sieckenius de Souza e a Vinícius Costa Villas Bôas Segura pela revisão de partes deste livro, a Antonio Luz Furtado, pela sua colaboração no design da capa, e à equipe editorial da Campus/Elsevier, pela sua dedicação e apoio ao longo de toda essa jornada.

S.D.J. Barbosa & B.S. da Silva
Rio de Janeiro, junho de 2010

Prefácio

Em 2006, um Grupo de Trabalho se reuniu no Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, IHC 2006, em Natal, para discutir o currículo de Interação Humano–Computador (IHC) nas universidades brasileiras (Silveira e Prates, 2007). Visando apoiar aulas de graduação de IHC em cursos de Ciência da Computação, Sistemas de Informação e Engenharia de Computação, tomamos como base o currículo recomendado pelo GT para selecionar o conteúdo incluído neste livro, apoiados pela nossa própria experiência em lecionar disciplinas de IHC por diversos semestres.

Sabemos que não é viável abordar todos os aspectos de uma área de conhecimento ampla e multidisciplinar como IHC em um único livro. Além disso, reconhecemos que nem todas as universidades incluem mais de uma disciplina de IHC em suas grades curriculares. Objetivamos então elaborar um livro-texto como material didático de apoio a um curso de introdução à IHC com a duração de um semestre.

Além do apoio a aulas de graduação, incluímos material de cunho prático, a fim de que o livro possa ser utilizado também por profissionais que precisem incorporar atividades de IHC em sua prática de trabalho.

Os autores podem ser contatados pelo e-mail: livroihc@gmail.com.

1

Introdução

Objetivos do Capítulo

- Discutir a importância da área de Interação Humano-Computador (IHC) considerando o impacto das tecnologias de informação e comunicação no nosso cotidiano.
 - Apresentar diferentes visões da Computação sobre a construção de sistemas computacionais interativos.
 - Descrever os objetos de estudo de IHC.
 - Discutir a importância da multidisciplinaridade em IHC.
 - Apresentar alguns benefícios proporcionados por incorporar práticas de IHC no desenvolvimento de sistemas computacionais interativos.
-

As tecnologias de informação e comunicação (TICs) oferecem maneiras eficientes de processar e trocar informações com diversos objetivos. Elas permitem criar sistemas computacionais embutidos nos mais diferentes dispositivos eletrônicos, que combinam poder computacional e meios de comunicação (telefonia, rádio, TV, Internet etc.). Neste livro, nos concentramos nos sistemas computacionais *interativos* que compõem as TICs, isto é, sistemas computacionais compostos por hardware, software e meios de comunicação desenvolvidos para *interagirem com pessoas*.

1.1 O Impacto das Tecnologias de Informação e Comunicação no Cotidiano

As TICs estão se desenvolvendo em ritmo acelerado, e cada vez mais fazem parte das nossas vidas pessoais e profissionais. A evolução e a disseminação dessas tecnologias alcançaram um nível em que é difícil encontrar pessoas que ainda não tiveram direta ou indiretamente contato com elas, independente de classe social, do nível de escolaridade e do local onde moram. A mídia tem apresentado vários exemplos de indígenas brasileiros, comunidades distantes dos grandes centros e moradores de comunidades de baixa renda nas grandes cidades com acesso à Internet onde moram.

Você já parou para pensar sobre como as TICs estão presentes na sua vida e na nossa sociedade? Em que áreas elas estão presentes? Em que quantidade? Que importância elas adquiriram? O que significa ter tanta tecnologia na vida das pessoas? Quais são as consequências disso para as pessoas que utilizam e para as pessoas que desenvolvem essas tecnologias? Vamos começar analisando algumas áreas que empregam essas tecnologias atualmente.

Faz tempo que somente computadores pessoais ou de grande porte (como *mainframes* e servidores) eram capazes de processar informações. Já é possível encontrar vários produtos eletrônicos “inteligentes”, tais como máquinas fotográficas que se configuram automaticamente de acordo com a luminosidade do ambiente e que são capazes de reconhecer faces e expressões humanas como o sorriso; máquinas de lavar louças que detectam o nível de sujeira na água e indicam o melhor programa para a lavagem; ou ainda condicionadores de ar que regulam automaticamente a temperatura, velocidade e direção da ventilação de acordo com a temperatura ambiente.

As TICs estão revolucionando a área de entretenimento. Jogos eletrônicos estão ficando mais sofisticados, com enredos mais elaborados, melhores gráficos e com maior aplicação de inteligência artificial. Estão surgindo novos dispositivos de interface com o jogador: controles sem fio e com sensor de movimento (como os controles do Wii®), e câmeras que detectam os movimentos do jogador (projeto Natal para o Xbox®). Além disso, jogos em rede permitem a interação entre pessoas como parte do entretenimento. A TV digital interativa (TVDI) também está mudando a

forma de produzir e consumir conteúdos na TV. Na TV analógica, a emissora era capaz de transmitir apenas um conteúdo por vez para todos os telespectadores, e eles normalmente recebiam o conteúdo numa atitude passiva, sentados no sofá de casa. A tecnologia por trás da TVDI permite à emissora enviar mais de um conteúdo simultaneamente. Dessa forma, cabe ao telespectador escolher a qual desses conteúdos enviados ele irá assistir, no dispositivo e local desejados. A TVDI também permite ao usuário interagir com a emissora numa atitude mais ativa, por exemplo, emitindo alguma opinião ou realizando alguma votação (Soares e Barbosa, 2009). E ainda temos a rádio digital, que está começando a ser discutida no Brasil.

As TICs também transformaram a noção de distância e tempo na comunicação entre pessoas. E-mail, programas para troca de mensagens, como MSN e GTalk, e comunidades virtuais, como Orkut e Facebook, permitem que pessoas espalhadas geograficamente possam se comunicar usando texto, vídeo e som, de forma síncrona ou assíncrona. Essas tecnologias permitem trocar rapidamente arquivos de diversos formatos, como música, fotos, vídeos etc. O próprio telefone celular oferece um canal de comunicação individual disponível em praticamente qualquer lugar do mundo. Hoje, um grande número de pessoas carrega consigo um aparelho eletrônico que integra telefonia, câmera digital, acesso à Internet, jogos, reproduutor de música e de vídeos, GPS e TVDI.

O acesso a informação vem sofrendo grandes transformações com a evolução tecnológica. Na educação, por exemplo, um professor não pode mais considerar que ele e os livros são as únicas fontes de conhecimento disponíveis aos alunos. A Internet disponibiliza uma enorme quantidade de informação que os alunos podem acessar quando e onde desejarem. Diferentemente de outras mídias, como papel e quadro-negro, as TICs permitem criar materiais dinâmicos e interativos que podem favorecer o aprendizado, como vídeos, simulação de fenômenos naturais, exploração de realidades virtuais, comunicação e colaboração entre alunos e professores com apoio computacional, e assim por diante. Se unirmos a oferta de conteúdo didático em meio computacional a uma comunidade virtual dispersa geograficamente, podemos explorar o ensino a distância utilizando as TICs. Já existem diversos cursos a distância disponíveis no Brasil e no mundo, inclusive cursos de nível superior. A Organização das Nações Unidas (ONU) acaba de criar uma universidade on-line de ensino a distância em escala mundial: a *University of the People*.¹

No campo da política, as TICs também estão mudando a relação entre eleitores e políticos, sejam ainda candidatos ou aqueles que tenham assumido algum cargo

¹ <http://www.uopeople.org/>. Exceto quando indicado explicitamente, todos os Web sites mencionados neste livro foram acessados em junho de 2010.

público. Antes, a comunicação em larga escala dos políticos era geralmente restrita à propaganda política obrigatória com data marcada e horário limitado na TV e no rádio, que são canais de comunicação unidirecionais. A Internet oferece novos canais de comunicação bidirecionais, como Web sites dos partidos, blogs dos políticos e dos eleitores, vídeos no YouTube, dentre outros. Agora, um grande número de eleitores dispersos geograficamente pode consultar, opinar e questionar na Internet as propostas e ideias dos políticos quando e onde desejarem. Vários políticos no Brasil e no mundo mantêm contato com os eleitores pela Internet, como, por exemplo, alguns prefeitos de grandes cidades brasileiras. O próprio ato de votar nos representantes brasileiros mudou drasticamente com a utilização das urnas eletrônicas. Hoje o resultado de eleições federais pode ser apurado em horas.

Muitas relações do Estado com a população são atualmente mediadas pelas TICs, o que tem sido chamado de governo eletrônico (*e-gov*). No Brasil, matrículas em escolas públicas já são feitas exclusivamente pela Internet ou por telefone; a maior parte das declarações de imposto de renda é entregue pela Internet; e já é possível consultar informações sobre processos jurídicos também pela rede. Algumas prefeituras fazem uso de sistemas de informações geográficas na gestão de seus municípios, como o uso de imagens de satélites para verificar construções irregulares, para analisar o fluxo de veículos e reestruturar o tráfego nas vias.

As TICs estão afetando o comércio e a nossa relação com dinheiro e bancos. Boa parte das transações financeiras já não manipula mais papel-moeda. Cartões e operações on-line estão ganhando cada vez mais espaço no nosso cotidiano. Muitas operações bancárias podem ser realizadas através de praticamente qualquer dispositivo com acesso à Internet, onde e quando desejarmos, sem a necessidade de ir a uma agência bancária. O comércio também ganhou versão on-line. Sem sair de casa, as TICs nos permitem comprar o que desejarmos, comparar preços de produtos em diferentes lojas on-line com apoio computacional, e obter mais informações sobre os produtos desejados direto com os fabricantes, e ter acesso às opiniões e experiências de uso de outras pessoas sobre os produtos desejados.

Grande parte do transporte atual é controlada com ajuda das TICs: controle de tráfego aéreo, metrô, trens urbanos, o fluxo de ônibus em cada linha, e assim por diante. Até o próprio carro possui muita tecnologia embutida para evitar acidentes, usar o combustível de forma eficiente e ajudar a estacionar, por exemplo. As TICs já são capazes de conduzir veículos sem ajuda do ser humano no modo de piloto automático, em aviões e carros. Também podem ajudar a monitorar a emissão de gases poluentes, contribuindo para a preservação do meioambiente.

Na área da saúde, as TICs vêm se tornando fundamentais para o diagnóstico e tratamento de doenças. Muitos aparelhos utilizados em Medicina são controlados com ajuda da computação, tais como aparelhos de ressonância magnética, de tomografia computadorizada e de radioterapia. Existem robôs que realizam cirurgias sendo manipulados por médicos muito distantes do paciente. Já existe uma cápsula programada para liberar remédio dentro do corpo humano no local, na quantidade e no fluxo certos para tratar doenças de forma mais eficiente.² Além disso, as TICs permitem que o histórico de saúde de um paciente esteja on-line à disposição dos médicos, incluindo resultados de exames que acabaram de ficar prontos em um laboratório distante. E pesquisas em computação gráfica e realidade aumentada vêm contribuindo com novas formas de visualizar os resultados dos exames.

Ao analisarmos esses exemplos de diversas áreas, percebemos que as TICs estão ocupando espaço importante nas nossas vidas. Quando as incorporamos no nosso cotidiano, não estamos apenas trocando de instrumentos, como quem troca de garfo, caneta ou régua. As modificações são mais profundas e significativas, pois modificam também a nossa forma de trabalhar, de prestarmos serviços, de nos relacionarmos com outras pessoas e instituições, de ensinarmos e aprendermos, de participarmos da política, de lidarmos com o dinheiro, de cuidarmos da saúde, e assim por diante. É importante reconhecermos que as TICs estão modificando não apenas *o que* se faz e *como* se faz, mas também *quem* as faz, *quando*, *onde* e até mesmo *por quê*.

Tomando como exemplo a transição da votação em cédula de papel para a votação na urna eletrônica, a mudança foi além da forma *como* o eleitor manifesta seu voto. Quantas pessoas (*quem*) sabem votar nulo (*o que*) na urna eletrônica? Será que a motivação para o voto nulo (*por que votar nulo ou não*) foi modificada na transição da cédula de papel para a urna eletrônica (Figura 1.1)?



Figura 1.1 Urna eletrônica.³

² <http://uk.reuters.com/article/idUKTRE4AA53S20081111>.

³ http://www.tse.gov.br/internet/eleicoes/urna_eletronica/simulacao_votacao/2008/SimUrnaBR.html.

Boa parte das pessoas que não sabe votar nulo na urna eletrônica sabia votar nulo na cédula de papel. Não existe um botão exclusivo para votar nulo, semelhante ao que existe para votar em branco. Para votar nulo, o eleitor precisa digitar um número de candidato inválido e confirmar o seu voto. Observamos que a urna eletrônica foi projetada, intencionalmente ou não, para desestimular o voto nulo, dificultando uma atitude de protesto dos eleitores. Na cédula de papel, os eleitores podiam escrever o que quisessem, até votar no macaco Tião.⁴

Há outros exemplos de como a introdução de TICs afeta o comportamento humano. Os japoneses não costumam sorrir muito como os brasileiros. Essa característica cultural faz diferença no atendimento ao público. Para se tornarem mais simpáticos (*por que*), os funcionários do metrô de Tóquio (*quem*) estão sendo convidados a exercitar o sorriso (*o que*) diante de um sistema interativo capaz de identificar expressões faciais. O exercício é realizado antes do expediente (*quando*) como uma espécie de jogo, no qual quem sorrir melhor ganha mais pontos (*como*). Esse é um exemplo claro de como as TICs estão sendo utilizadas para intervir no comportamento das pessoas de forma bastante significativa, pois atuam sobre seu jeito de ser e sua cultura. Qualquer intervenção na cultura, nas habilidades e nos conhecimentos das pessoas deve ser realizada com cuidado e respeito às individualidades de cada uma, não importa quanta inovação e tecnologia estejam sendo utilizadas.

As TICs também vêm afetando a nossa vida pessoal. Por exemplo, João possui um *smartphone* que agrega um canal de comunicação de um telefone celular com alguns recursos computacionais de um notebook, em um dispositivo que cabe no bolso. Enquanto ele faz sua caminhada matinal (*onde e quando*), ele está acessível (*por que*) através de seu *smartphone* para receber notícias de casa (*o que*), como a notícia de que ele precisa comprar algo antes de voltar ou que seu filho está passando mal. Entretanto, esse mesmo dispositivo permite receber telefonemas sobre algo do trabalho no meio da caminhada, trazendo problemas para um momento de exercício e relaxamento. Nem sempre o que um sistema interativo permite fazer é desejável e bom. Por isso, também é importante pensarmos no mau uso da tecnologia. Por exemplo, se João souber quem está ligando antes de atender a ligação, ele pode escolher atender ou não. Se for um colega de trabalho que costuma ser inconveniente, ele pode ser ignorado naquele momento. Se for um colega que só liga antes do expediente caso seja um assunto sério e urgente, João pode escolher atender a ligação. Ali mesmo, ele também pode consultar e reajustar a sua agenda da semana no *smartphone* e enviar alguns e-mails para começar a resolver o problema urgente do trabalho.

⁴ Em 1988, o Macaco Tião do zoológico do Rio de Janeiro recebeu 400 mil votos para prefeito da cidade em sinal de protesto. http://veja.abril.com.br/200897/p_093b.html.

Diante disso, o desenvolvedor de TICs deve estar ciente de que o resultado do seu trabalho vai modificar a vida de muitas pessoas (inclusive a sua própria) de forma previsível e imprevisível. Sempre que possível, devemos tentar prever essas modificações e encaminhá-las da melhor forma possível. Para os casos em que não é possível prever os efeitos das novas tecnologias, é importante que o desenvolvedor também crie salvaguardas para os usuários, por exemplo, fornecendo maneiras fáceis de desfazer ações e maneiras alternativas de realizar as coisas sem depender da tecnologia desenvolvida. Quem desenvolve tecnologia precisa sempre se perguntar: o que acontece se o usuário errar, a tecnologia falhar ou permanecer indisponível por algum tempo? As salvaguardas serão desenvolvidas de acordo com as respostas a perguntas como essa.

1.2 Diferentes Visões sobre a Construção de Sistemas Interativos

Existem diversos atores envolvidos no desenvolvimento e uso dos sistemas computacionais interativos: fabricantes de hardware, de software, vendedores, profissionais de suporte e manutenção, provedores de acesso à Internet, produtores de conteúdo, usuários, organizações, dentre outros. Todas essas partes interessadas costumam ser denominadas *stakeholders*. Cada um enxerga a tecnologia sob um ponto de vista diferente, enfatizando alguns aspectos em detrimento de outros.

Por exemplo, um usuário costuma estar mais interessado no acesso à Internet que um dispositivo possibilita e como isso pode ser útil para ele do que interessado nas peças que compõem o dispositivo ou como elas foram montadas. Apesar de as duas perspectivas serem sobre o mesmo dispositivo, a segunda perspectiva é mais comum a um desenvolvedor de hardware e a um profissional de suporte.

Em outro exemplo, pense em uma organização que utiliza software como instrumento de trabalho. Para apoiar os processos de trabalho da organização, um gerente encomenda um sistema a uma empresa de desenvolvimento de software. Os desenvolvedores costumam se concentrar nas funcionalidades do software e em como ele é estruturado internamente. Já os funcionários da organização geralmente se preocupam em como vão aprender e utilizar o software para realizar o seu trabalho com eficiência. Existe uma diferença sutil, porém importante, entre o que um sistema interativo deve permitir fazer (visão do cliente, responsável pela aquisição do sistema), o que ele de fato permite fazer (visão de quem produz, focada nas funcionalidades do software) e a maneira como ele é utilizado (visão dos usuários, focada no impacto do software no seu trabalho ou na sua vida). A identificação dos diferentes atores envolvidos e a articulação dos seus interesses e pontos de vista são importantes desafios no desenvolvimento de tecnologia.

Se nas relações entre pessoas ainda encontramos tantos problemas (mal-entendidos, discórdias, brigas, guerras etc.) depois de milênios de experiência, imagine quantos problemas podemos encontrar nas interações entre pessoas e sistemas computacionais, considerando que a Computação ainda não completou um século. Além do pouco tempo de convívio, pessoas são bem diferentes dos sistemas computacionais que sabemos construir atualmente. Os sistemas computacionais são construídos para sempre executarem um conjunto predefinido de instruções. Tudo o que um sistema é capaz de fazer foi definido na sua construção. Consequentemente, os sistemas sempre “interpretam” as ações do usuário de uma forma predefinida. Isso traz grandes dificuldades para os sistemas lidarem com a criatividade e a reinterpretação das coisas pelas pessoas.

As diversas áreas de conhecimento possuem perspectivas distintas sobre o problema, com diferentes experiências, estratégias de solução e conhecimentos estabelecidos. Cada área analisa os sistemas interativos de acordo com critérios de qualidade particulares, cada qual assumindo diferentes graus de importância. Grande parte da Computação e, em particular, a subárea de Engenharia de Software, está interessada na construção de sistemas interativos mais eficientes, robustos, livres de erros, e de fácil manutenção. Por outro lado, a área de Interação Humano-Computador (IHC) está interessada na qualidade de uso desses sistemas e no seu impacto na vida dos seus usuários. Apesar de fortemente relacionados, a *construção* e o *uso* de um artefato ocorrem em contextos distintos e seguem lógicas diferentes, envolvendo pessoas diversas. Essas diferenças permitem que um sistema interativo com alta qualidade de construção possa ter baixa qualidade de uso, e vice-versa. Por exemplo, é possível que um sistema seja útil e agradável ao usuário, mas com manutenção bem difícil. Também é possível que um sistema seja robusto e livre de erros, mas difícil de ser compreendido pelo usuário e pouco útil para ele.

Essa dualidade entre construção e uso também ocorre em outras atividades envolvendo diferentes pessoas, como na produção e consumo, na escrita e leitura, e em outras áreas de conhecimento como, por exemplo, entre a Engenharia Civil e a Arquitetura. A primeira enfatiza a construção de ambientes físicos, focando aspectos como custo, durabilidade, estrutura e métodos de construção, enquanto a segunda enfatiza o uso destes ambientes, focando as pessoas e suas interações entre elas e com o ambiente.

Por ter a qualidade de construção como prioritária, grande parte da Computação costuma conceber um sistema interativo de “dentro para fora”, isto é, conceber primeiro (ou pelo menos com ênfase bem maior em) representações de dados, algoritmos que processam esses dados, arquitetura do sistema e tudo mais que permite

um sistema interativo funcionar (Figura 1.2a). Pouca ou nenhuma atenção é de fato dedicada ao que fica fora do sistema e a como ele será utilizado. Parece haver um pressuposto de que tudo o que for externo ao sistema vai, sem esforço, adaptar-se a ele e ser capaz de tirar proveito dele da melhor forma possível. Infelizmente, nem sempre o mundo fora de um sistema interativo se adapta a ele e o aproveita de maneira tão fácil, simples e rápida quanto alguns desenvolvedores gostariam que acontecesse. Se seguirmos uma abordagem de “dentro para fora”, corremos um grande risco de concebermos um sistema interativo inapropriado para o mundo que o cerca, pois a nossa compreensão do mundo pode ser equivocada.

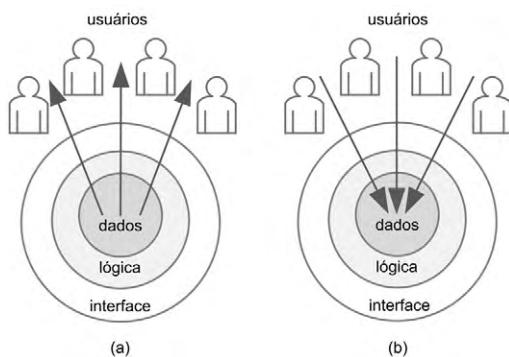


Figura 1.2 Abordagem de desenvolvimento (a) de “dentro para fora” e (b) de “fora para dentro”.

Para conceber um sistema interativo mais adequado ao mundo onde será inserido, a área de IHC (e, sob alguns aspectos, também a área de Engenharia de Requisitos) busca seguir uma abordagem de “fora para dentro” (Figura 1.2b). Nessa abordagem, o projeto de um sistema interativo começa investigando os atores envolvidos, seus interesses, objetivos, atividades, responsabilidades, motivações, os artefatos utilizados, o domínio, o contexto de uso, dentre outros, para depois identificar oportunidades de intervenção na situação atual, a forma que a intervenção tomará na interface com o usuário e, finalmente, como o sistema viabiliza essa forma de intervenção. Tomando como exemplo duas áreas da Computação, a área de Engenharia de Requisitos privilegia os critérios de qualidade da Engenharia de Software, enquanto a área de IHC privilegia a qualidade de uso dos sistemas interativos.

Dentro da Computação, outras áreas auxiliam a concepção de uma solução interativa com alta qualidade de uso, como a Engenharia de Software, Inteligência Artificial e Trabalho Cooperativo Apoiado por Computador (*Computer-Supported Cooperative Work*). Por exemplo, técnicas de inteligência artificial são utilizadas em interfaces em linguagem natural e também em adaptações da interface ao contexto de uso. Embora IHC utilize conhecimentos e técnicas de diferentes áreas dentro e

fora da Computação, IHC se distingue delas por focar o **uso** de sistemas interativos (Sharp *et al.*, 2007).

1.3 Objetos de Estudo em IHC

Afinal, a área de IHC trata de quais assuntos? Qual é o seu escopo? Quais são seus objetos de estudo? IHC é uma disciplina interessada no projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos relacionados a esse uso (Hewett *et al.*, 1992). De acordo com Hewett e seus colegas (1992), os objetos de estudo de IHC podem ser agrupados em cinco tópicos inter-relacionados: a natureza da interação humano-computador; o uso de sistemas interativos situado em contexto; características humanas; arquitetura de sistemas computacionais e da interface com usuários; e processos de desenvolvimento preocupados com uso (Figura 1.3).

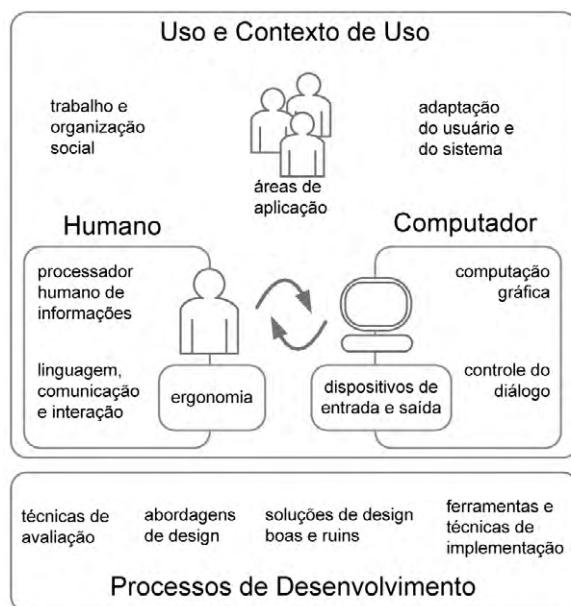


Figura 1.3 Objetos de estudo em IHC (adaptado de Hewett *et al.*, 1992).

Estudar a **natureza da interação** envolve investigar o que ocorre enquanto as pessoas utilizam sistemas interativos em suas atividades. É possível descrever, explicar e prever esse fenômeno e algumas de suas consequências na vida das pessoas.

O **contexto de uso** influencia a interação de pessoas com sistemas interativos, pois elas estão inseridas em determinada cultura, sociedade e organização, possuem modo próprio de realizar suas atividades, possuem conhecimentos e concepções pró-

prios e utilizam linguagem para interagir com as outras pessoas. É importante estarmos cientes de que o contexto de uso costuma ser diferente do contexto em que os desenvolvedores estão inseridos e com o qual estão acostumados. Daí a importância de investigarmos o contexto de uso com foco nos usuários e sob o seu ponto de vista. Isso nos permite avaliar o impacto dos diferentes aspectos do contexto sobre a interação humano-computador sendo concebida ou avaliada.

As **características humanas** também influenciam a participação das pessoas na interação com sistemas interativos. A interação com qualquer artefato novo, principalmente os sistemas computacionais interativos, que lidam com informações, requer capacidade cognitiva para processar informações e aprender a utilizá-los. A forma como as pessoas se comunicam e interagem, entre si e com outros artefatos, também influencia a interação humano-computador, pois elas tendem a continuar utilizando essas mesmas formas de interação quando lidam com um sistema computacional interativo (Reeves e Nass, 2003). Além disso, as características físicas dos seres humanos, como visão, audição, tato e capacidade de movimentar o corpo, são responsáveis pela sua capacidade de percepção do mundo ao seu redor e sua capacidade de atuar sobre ele. Conhecer as características humanas dos usuários nos permite aproveitar suas capacidades e, principalmente, respeitar suas limitações durante a interação com sistemas computacionais. O Capítulo 3 apresenta algumas teorias e trabalhos empíricos que investigam essas características humanas.

Existem estudos sobre a **arquitetura de sistemas computacionais** e interfaces com usuário buscando construir sistemas que favoreçam a experiência de uso (John *et al.*, 2004). Diversas tecnologias e dispositivos têm sido desenvolvidos para permitir e facilitar a interação com pessoas. Os dispositivos de entrada e saída são os meios físicos responsáveis por mediar o contato físico entre pessoas e sistemas computacionais. Esse contato ocorre de acordo com técnicas de diálogo, como preenchimento de formulários utilizando o teclado e seleção de menus utilizando o mouse, por exemplo. O projeto da interação costuma aproveitar modelos conceituais já conhecidos pelos usuários para facilitar a adoção e o aprendizado do sistema. Por fim, existem técnicas para construir a interface com usuário, desenvolvidas, por exemplo, na área de Computação Gráfica e em Inteligência Artificial. Conhecer essas tecnologias e dispositivos é fundamental para sermos capazes de propor, comparar, avaliar e tomar decisões sobre formas alternativas de interação com sistemas computacionais.

Finalmente, o **processo de desenvolvimento** de um sistema interativo influencia a qualidade do produto final. Por isso é importante conhecermos abordagens de design de IHC, métodos, técnicas e ferramentas de construção de interface com usuário e de avaliação de IHC. Também é importante conhecermos e analisarmos casos de

sucesso e de insucesso de interfaces com usuário, sempre buscando identificar os motivos que levaram a tal resultado. O Capítulo 4 apresenta alguns processos de desenvolvimento comumente utilizados em IHC.

1.4 IHC como Área Multidisciplinar

IHC se beneficia de conhecimentos e métodos de outras áreas fora da Computação para conhecer melhor os fenômenos envolvidos no uso de sistemas computacionais interativos. Áreas como Psicologia, Sociologia e Antropologia contribuem para aquisição de conhecimento sobre a cultura e o discurso dos usuários e sobre seus comportamentos no ambiente onde realizam suas atividades, sejam elas individuais ou em grupo. A definição da interface com usuário faz uso de conhecimentos e técnicas de áreas como: Design, Ergonomia, Linguística e Semiótica.

Alguns conhecimentos e técnicas importados de outras áreas além da Computação são adaptados às necessidades de IHC. Por exemplo, a Psicologia utiliza extensamente entrevistas para ter acesso às concepções, emoções e subjetividade das pessoas. Isso é muito mais profundo e complexo que a utilização mais frequente de entrevistas em IHC, através das quais normalmente investigamos a compreensão sobre um domínio, opiniões sobre certos sistemas interativos e o que ocorreu durante uma experiência de uso para avaliação da interface com usuário. Algumas técnicas de apresentação de conteúdo estático, como as utilizadas em jornais, revistas e livros, foram adaptadas em IHC para lidar com a dinâmica da interface, bem como conteúdos hipermídia.

Conforme discutido até aqui, percebemos que a área de IHC articula uma grande quantidade de conhecimentos oriundos de diversas áreas. Isso torna muito difícil que um único profissional tenha conhecimento profundo de todos os objetos de estudo de IHC. Se um único profissional dificilmente conhece em profundidade todos os assuntos relacionados com a interação entre pessoas e sistemas computacionais, como é possível cuidarmos das questões relacionadas a IHC de forma adequada? Idealmente, a responsabilidade de cuidar de IHC deve ser atribuída a uma **equipe multidisciplinar**. Dessa forma, profissionais com formações diferentes podem trabalhar em conjunto, concebendo e avaliando a interação de pessoas com sistemas computacionais.

Esse ambiente heterogêneo de profissionais com diferentes formações facilita o surgimento de ideias, a criatividade e a inovação, bem como auxilia a análise do problema e de alternativas de soluções sob pontos de vista bem variados, enriquecendo, assim, o resultado do trabalho. Muitos projetos de IHC são realizados por

uma equipe multidisciplinar, contando com engenheiros, designers,⁵ programadores, psicólogos, antropólogos, sociólogos, artistas, dentre outros (Sharp *et al.*, 2007). A decisão de quais profissionais devem fazer parte da equipe multidisciplinar de IHC precisa considerar vários fatores, como, por exemplo, o domínio e porte do sistema, e o orçamento disponível.

Entretanto, uma equipe multidisciplinar requer que profissionais com diferentes formações superem as dificuldades de trabalhar em conjunto. Cada profissional tem uma visão de mundo, uma forma particular de pensar e, muitas vezes, um vocabulário próprio. Para aproveitar as competências de cada profissional e evitar conflitos, é necessário facilitar a comunicação e a compreensão entre os membros da equipe multidisciplinar. Além disso, é importante criar um ambiente de respeito aos valores e às contribuições de cada profissional para que as discussões sejam proveitosas e cooperativas e, não se tornem uma luta entre posições individuais, opostas e intransigentes.

Se não for possível compor uma equipe multidisciplinar para cuidar de IHC, o resultado do trabalho de mais de uma pessoa com a mesma formação tende a ser melhor do que o resultado do trabalho de apenas uma pessoa. Cada um percebe as questões e reflete sobre elas de maneira diferente, o que lhes facilita propor um conjunto maior de ideias e compará-las sob diferentes aspectos.

1.5 Benefícios de IHC

Por que devemos estudar e cuidar da interação entre pessoas e sistemas computacionais? Quais são os benefícios? Começamos este capítulo analisando como as TICs estão presentes na vida das pessoas e concluímos que essas tecnologias afetam direta ou indiretamente o que as pessoas fazem, como, onde, quando e por quê. Estudar fenômenos de interação entre seres humanos e sistemas computacionais nos permite comprehendê-los para melhorarmos a concepção, construção e inserção das TICs na vida das pessoas, sempre buscando uma boa experiência de uso. Nesse sentido, devemos procurar aproveitar as características humanas e o poder computacional para desenvolvermos sistemas interativos que melhorem a vida das pessoas, trazendo bem-estar, aumentando sua produtividade, satisfazendo suas necessidades e desejos, e respeitando suas limitações e valores. Para isso, também devemos conhecer as capacidades e limitações das tecnologias disponíveis.

Aumentar a qualidade de uso de sistemas interativos apresenta vários benefícios para a experiência pessoal do usuário em decorrência do uso e, consequentemente,

⁵ Neste livro, o termo designer de IHC, ou simplesmente designer, é utilizado para referenciar a pessoa ou a equipe responsável pelo projeto de IHC.

para a sua vida (Norman, 1988; Rubin, 1994; Bias e Mayhew, 2005). Esse aumento da qualidade de uso contribui para:

- aumentar a produtividade dos usuários, pois, se a interação for eficiente, os usuários podem receber apoio computacional para alcançar seus objetivos mais rapidamente;
- reduzir o número e a gravidade dos erros cometidos pelos usuários, pois eles poderão prever as consequências de suas ações e compreender melhor as respostas do sistema e as oportunidades de interação;
- reduzir o custo de treinamento, pois os usuários poderão aprender durante o próprio uso e terão melhores condições de se sentirem mais seguros e motivados para explorar o sistema;
- reduzir o custo de suporte técnico, pois os usuários terão menos dificuldades para utilizar o sistema e, se cometerem algum erro, o próprio sistema oferecerá apoio para se recuperarem dos erros cometidos; e
- aumentar as vendas e a fidelidade do cliente, pois os clientes satisfeitos recomendam o sistema a seus colegas e amigos e voltam a comprar novas versões.

Além disso, cuidar desde o início da qualidade de uso contribui para reduzir o custo de desenvolvimento, já que as modificações que favorecem o uso ocorrerão mais cedo no processo de desenvolvimento. Dessa forma, a qualidade de uso está se estabelecendo como uma vantagem competitiva e adquirindo papel importante na percepção de valor do produto e da empresa, pois influencia a percepção do usuário sobre a qualidade do sistema. Bias e Mayhew (2005) apresentam estudos indicando retorno de investimento em qualidade de uso. É importante lembrar que, embora os custos de desenvolvimento possam aumentar ligeiramente, o investimento nessa área traz benefícios sempre que o sistema for utilizado e para todos os envolvidos com seu uso, seja direta ou indiretamente, ao longo de toda a vida útil do sistema.

Atividades

1. *Impacto das TICs no cotidiano dos usuários.* Analise o cenário a seguir, identificando qual é o impacto das TICs sobre o que o usuário faz, como, quando, onde e por quê:

Marcos costuma viajar a trabalho entre duas cidades distantes. Durante o trajeto, ele sempre procurava algo para se entreter. Com um celular capaz de exibir TV digital interativa, ele pôde assistir a seu jogo preferido durante a viagem. De repente acontece uma jogada polêmica contra o seu time. Foi pênalti ou não? Intrigado, ele revê a jogada em diferentes ângulos para conferir o ocorri-

do. Antes da TV digital interativa, não era possível escolher quais ângulos ele poderia rever a jogada para tirar suas próprias conclusões sobre o lance, muito menos remotamente durante uma viagem.

2. *Análise dos elementos envolvidos no processo de interação.* Analise o que muda nas seguintes situações de uso:

- uma pessoa que paga as suas contas pelo computador pessoal de casa ou em um caixa eletrônico;
- um adolescente com poucos compromissos usando um sistema de agenda no seu celular, ou um adulto com muitos compromissos administrando sua agenda no seu computador pessoal.

O que muda nessas situações em relação ao contexto de uso, aos objetivos dos usuários, à interface e à interação? Que considerações sobre IHC perderiam importância? Que outras ganhariam importância?

2

Conceitos Básicos

Objetivos do Capítulo

- Explicar os conceitos de interação, interface e *affordance*.
 - Descrever critérios de qualidade de uso utilizados em IHC: usabilidade, experiência do usuário, acessibilidade e comunicabilidade.
-

Para aumentarmos a qualidade de uso de sistemas interativos, devemos inicialmente identificar os elementos envolvidos na interação usuário–sistema. Este capítulo apresenta os conceitos de interação usuário–sistema, interface com usuário e *affordance*, e descreve critérios de qualidade comumente considerados em IHC: usabilidade, experiência do usuário, acessibilidade e comunicabilidade.

2.1 Interface, Interação e *Affordance*

A Figura 2.1 ilustra uma situação típica de uso: um usuário engajado num processo de interação com a interface de um sistema interativo, buscando alcançar um objetivo em determinado contexto de uso. O contexto de uso é caracterizado por toda situação do usuário relevante para a sua interação com o sistema (Dey, 2001), incluindo o momento de utilização do sistema (quando) e o ambiente físico, social e cultural em que ocorre a interação (onde).

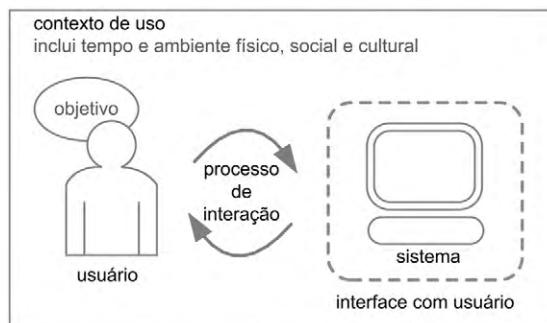


Figura 2.1 Elementos envolvidos no processo de interação.

O exemplo a seguir examina esses elementos através de um cenário de uso de uma aplicação de produção e apresentação de slides.

Exemplo 2.1 – Elementos envolvidos no processo de interação

Vejamos algumas situações de uso em que o professor Lucas cria, edita e visualiza slides. No conforto da sua casa (contexto de uso), Lucas (usuário) costuma usar o Impress® do BrOffice¹ (sistema) no seu computador pessoal de mesa (desktop) para preparar os slides que vai utilizar nas aulas (objetivo). Em alguns casos, ele começa a preparar sua aula a partir de um documento em branco (Figura 2.2). Ele escreve o título da aula, cria uma sequência de slides de acordo com os tópicos a serem abordados e conclui o conteúdo detalhando cada tópico (processo de interação). Depois de definido o conteúdo, ele cuida do *layout* dos slides, tal como cores, fonte dos textos, figuras etc. Sempre que possível, ele prefere elaborar as aulas em casa por dispor de um ambiente mais tranquilo, com menos interrupção e distrações.

Durante o processo de interação, Lucas manipula a interface gráfica do Impress® usando o teclado e o mouse para alcançar seu objetivo. O tamanho do monitor permite visualizar vários slides

1 <http://www.broffice.org>.

lado a lado, para analisar e organizar a estrutura da apresentação (Figura 2.3). O contexto de uso também é bastante propício para ele explorar ideias, seja pelo fato de haver menos interrupções, por ter livros e materiais didáticos à sua disposição ou simplesmente por ser um ambiente mais confortável para ele.

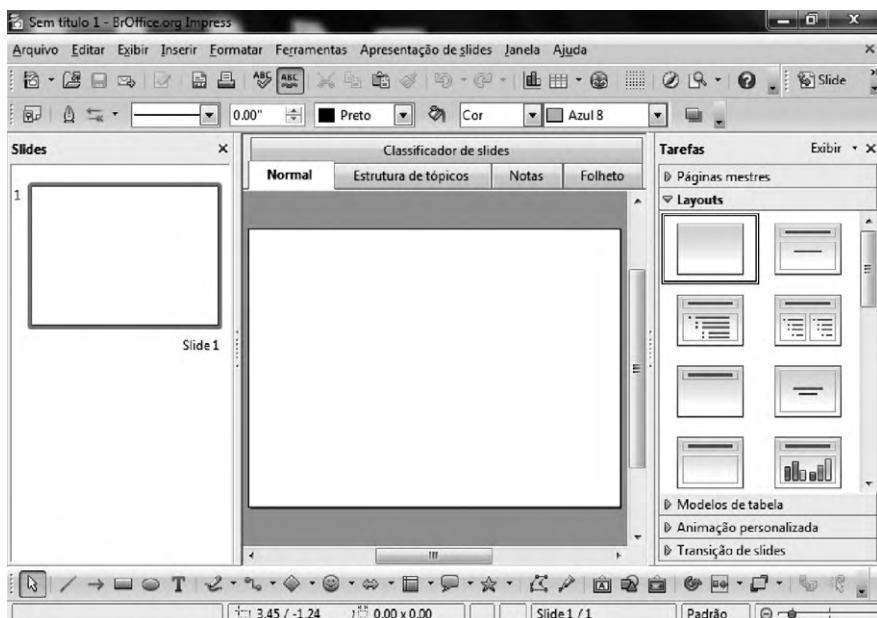


Figura 2.2 Tela inicial do Impress® do BrOffice.

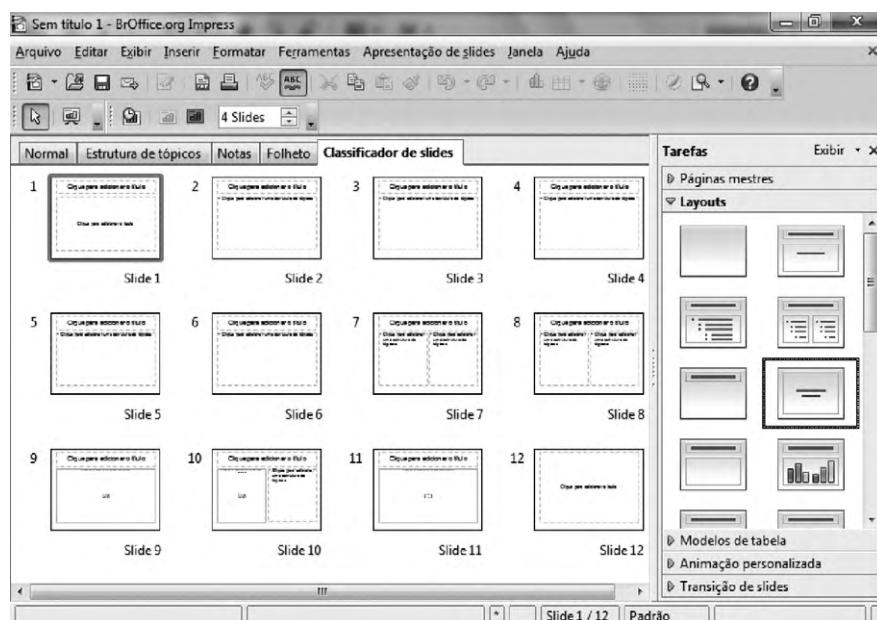


Figura 2.3 Visão geral dos slides no Impress® do BrOffice.

O que mudaria na situação de uso se Lucas, no aeroporto, precisasse rever os slides da sua próxima aula usando seu *smartphone* enquanto espera o avião de volta para sua cidade? Primeiro, trata-se de um dispositivo bem diferente, que impõe restrições importantes na entrada e saída de dados. A digitação e a manipulação do cursor costumam ser menos eficientes se comparados com as permitidas pelo teclado e mouse de desktop. A tela limita a quantidade de informação disponível a cada instante. O objetivo do usuário também mudou. Antes ele tinha interesse maior em criar e editar slides, agora o interesse maior é em visualizá-los. Além disso, o contexto de uso mudou significativamente. Passou de um ambiente confortável e com poucas interrupções para um ambiente com várias interrupções e que dispersa facilmente a atenção do usuário.

Quando Lucas chegar à sala de aula, o contexto de uso mudará novamente, e a relação com os alunos e a forma de apresentar o conteúdo vai influenciar a apresentação de slides. Na sala de aula, Lucas continua com o objetivo de exibir os slides como no aeroporto. Entretanto, o modo de exibi-los é fortemente influenciado pelo novo contexto. Por exemplo, o tempo gasto em cada slide será diferente, ou pode ser necessário voltar a slides anteriores porque um aluno ficou com alguma dúvida. Nesse caso, os dispositivos de entrada e saída serão os oferecidos por um notebook e um projetor de tela. Esses dispositivos de entrada e de saída não são muito diferentes daqueles oferecidos por um desktop, a não ser o projetor de tela, que nem sempre reproduz as cores conforme o esperado e é mais influenciado pela luminosidade do ambiente.

Se considerarmos Lucas como um usuário alvo de um novo sistema de edição e apresentação de slides, as diferenças nas situações de uso por ele vivenciadas no exemplo deveriam ser consideradas no design e avaliação desse novo sistema. Se isso não ocorrer, Lucas e outros usuários que vivenciem situações semelhantes poderão ter dificuldades ao utilizar o sistema.

2.1.1 Interação

A definição de **interação** usuário-sistema evoluiu ao longo do tempo. A princípio, tratava essencialmente de uma sequência de estímulos e respostas, como na interação de corpos físicos. Com o surgimento das pesquisas de base cognitiva, passou-se a enfatizar a interação como a *comunicação com máquinas*, em vez de a *operação de máquinas* (Card, Moran e Newell, 1983). Investigou-se também a interação como um processo através do qual o usuário formula uma intenção, planeja suas ações, atua sobre a interface, percebe e interpreta a resposta do sistema e avalia se seu objetivo foi alcançado (Norman, 1986). Em geral, a interação usuário-sistema pode ser considerada como tudo o que acontece quando uma pessoa e um sistema computacional se unem para realizar tarefas, visando a um objetivo (Hix e Hartson, 1993). Mais recentemente, enfatiza-se a interação usuário-sistema como processo de comunicação entre pessoas, mediada por sistemas computacionais (de Souza, 2005a). Sendo assim, podemos considerar a interação usuário-sistema como sendo um processo de manipulação, comunicação, conversa, troca, influência, e assim por diante. As abor-

dagens teóricas de IHC privilegiam diferentes definições do fenômeno de interação usuário-sistema.

Kammersgaard (1988) identificou quatro perspectivas de interação usuário-sistema: perspectiva de sistema, de parceiro de discurso, de ferramenta e de mídia. Cada uma atribui ao usuário e ao sistema determinado papel e caracteriza a interação sob um ponto de vista diferente, como ilustrado na Figura 2.4.

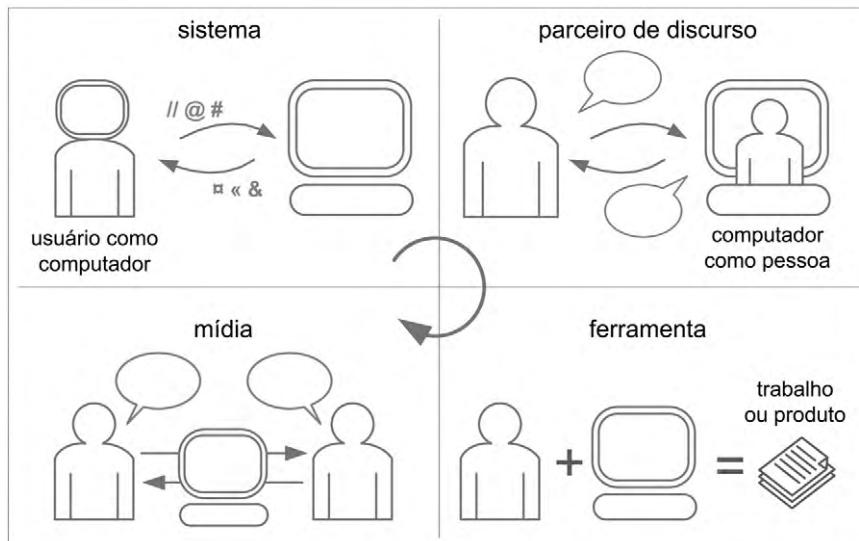


Figura 2.4 Perspectivas de interação humano-computador.

Na perspectiva de **sistema**, o usuário é considerado como um sistema computacional, e a interação humano-computador aproxima-se da interação entre sistemas computacionais, ou seja, é vista como uma mera transmissão de dados entre pessoas e sistemas computacionais, análoga à transmissão de dados entre sistemas. Desse modo, o usuário precisa se comportar como uma verdadeira máquina, aprendendo a interagir de forma bem disciplinada e restrita por formatos de entrada padronizados e rígidos. É comum a utilização de linguagem de comando ou de programação (como linguagens de script) nessa transmissão de dados. Quando se trabalha na perspectiva de sistema, o principal objetivo é aumentar a eficiência e a transmissão correta de dados, reduzindo o tempo de interação e o número de erros cometidos pelos usuários.

Um exemplo clássico do emprego dessa perspectiva é o terminal de comando de sistemas operacionais, tais como DOS e Linux (Figura 2.5).

```

Janela de comando
ubuntu@ubuntu:~$ ls
Desktop Documentos Imagens Modelos
ubuntu@ubuntu:~$ cd Desktop
ubuntu@ubuntu:~/Desktop$ ls
exemplos.desktop relatorio.txt tela001.jpg
tela002.jpg tela003.jpg tela004.jpg
ubuntu@ubuntu:~/Desktop$
```

Figura 2.5 Ilustração de um terminal do Linux, exemplificando a perspectiva de sistema.

Outro emprego comum da perspectiva de sistema é limitar aquilo que os usuários podem dizer, através de listas fechadas, controles de calendário e outros elementos de interface restritivos, como ocorre em sites de empresas aéreas (Figura 2.6).

RESERVAS DE VOO

ida e volta só ida múltiplo

de:

para:

busca por: data preço

DATAS

ida:

volta:

Figura 2.6 Fragmento de formulário ilustrando a perspectiva de sistema.

Combinações de teclas de atalho, tal como “Ctrl+C” para copiar e “Ctrl+V” para colar, também são exemplos de emprego da perspectiva de sistema. Elas são muito úteis e eficientes para usuários que possuem habilidade com o teclado e tenham tempo, disposição e capacidade cognitiva para aprender a sequência de teclas e os comandos associados.

A perspectiva de sistema pode ser inadequada para a realização de algumas atividades por certas classes de usuários, pois ela pode requerer algum tipo de treinamento e seu uso pode ser difícil e tedioso no início. Exemplo disso são os sistemas de linguagem de comando utilizados intensamente por funcionários de companhias aéreas, nos balcões dos aeroportos. Esses funcionários recebem extenso treinamento,

para que consigam utilizar o sistema. A vantagem é que, após o treinamento, utilizam o sistema com muita eficiência. Tais sistemas seriam inadequados a passageiros interessados em efetuar suas reservas ou seu check-in, pois seu uso esporádico não justificaria o investimento necessário em treinamento.

Em oposição à perspectiva de sistema, surgiu na área de Inteligência Artificial uma proposta de transformar o sistema interativo em **parceiro do discurso**. Nessa perspectiva, o sistema interativo deve participar da interação assumindo papel à altura de um ser humano, sendo capaz de raciocinar, fazer inferências, tomar decisões, adquirir informação, enfim, o sistema deve ser capaz de se comportar de forma semelhante aos seus usuários. Essa perspectiva visa tornar a interação humano-computador mais próxima de uma interação entre seres humanos. Nesse caso, a interação costuma ser compreendida como uma conversa. O objetivo do designer nessa perspectiva é cuidar da quantidade, conteúdo e sequência das falas durante a conversa usuário-sistema que auxilia o usuário a atingir seu objetivo. Geralmente, a perspectiva de parceiro de discurso faz uso da linguagem natural como meio de comunicação usuário-sistema. Por exemplo, poderíamos encontrar o diálogo do Exemplo 2.2 em um sistema de busca por produtos e serviços.

Exemplo 2.2 – Interação na perspectiva de parceiro de discurso.

Sistema: Em que posso ajudar?

Usuário: Quero procurar um presente para a minha tia.

Sistema: Do que sua tia gosta?

Usuário: Flores e bombons de chocolate com licor de cereja.

Sistema: Que tal um *bouquet* de rosas por R\$60,00 e uma caixa de bombons por R\$80,00?

Usuário: Acho melhor orquídeas.

Sistema: Que tal uma orquídea da família *Cattleya Trianae* por R\$250,00?

Usuário: É esta que eu quero.

Sistema: O telefone da floricultura é 5555-5555. E da loja de bombons é 5555-1234.

Usuário: Obrigado.

Sistema: De nada.

Construir um sistema parceiro de discurso não é algo trivial. Até hoje existem diversas pesquisas nessa linha, pois ainda temos grandes desafios no processamento em linguagem natural. Um bom exemplo desses desafios são os problemas que ocorrem em tradutores automáticos de texto.

Na perspectiva de **ferramenta**, o sistema interativo é considerado um instrumento que auxilia o usuário a realizar suas tarefas. Nessa perspectiva, a interação representa “um processo de aplicar uma ferramenta a algum material e avaliar o re-

sultado” (Kammersgaard, 1988, p. 353) durante a realização de uma atividade. O processo de interação é descrito principalmente pelo encadeamento de ações e reações empregando tal ferramenta (um sistema interativo). O sucesso da interação depende do conhecimento do usuário sobre a ferramenta e de sua capacidade de manipulá-la com destreza. Durante a interação, o usuário deve se concentrar no seu trabalho e manipular a ferramenta de forma automática, sem precisar pensar sobre essa manipulação; assim como um carpinteiro manipula um martelo enquanto fabrica móveis, ou um cozinheiro manipula talheres enquanto cozinha. Essa perspectiva é predominante nos sistemas de propósito geral e famílias de aplicações de escritório, como no Microsoft Office® e no OpenOffice®. Os fatores de qualidade mais evidentes nessa perspectiva são a relevância das funcionalidades oferecidas e a facilidade de uso da ferramenta.

A perspectiva de **mídia** vem ganhando cada vez mais espaço em sistemas interativos atuais, em particular sistemas que conectam pessoas através da Internet. Nessa perspectiva, o sistema interativo é visto como uma mídia (semelhante à imprensa, televisão, rádio e telefone) através da qual as pessoas se comunicam umas com as outras. Assim, a interação significa comunicação por meio da mídia num contexto coletivo. Além da comunicação entre usuários mediada por sistemas interativos, como ocorre em sistemas de e-mail, fórum, chats e redes sociais, também existe a comunicação unilateral dos designers do sistema para os usuários, explícita na ajuda on-line, nas instruções na interface e na documentação do sistema, ou implícita através da seleção e disposição dos elementos de interface em si. Nessa perspectiva, busca-se principalmente zelar pela qualidade da comunicação entre pessoas mediada por um sistema interativo e o seu entendimento mútuo.

A perspectiva de mídia e a perspectiva de parceiro de discurso são distintas. Enquanto a primeira vê a interação como uma conversa *usuário-sistema*, a segunda a vê como uma comunicação *entre pessoas* mediada por tecnologia. Apesar de essas duas perspectivas considerarem a interação como um processo de comunicação, a diferença entre elas aparece nos interlocutores. Na perspectiva de discurso, o sistema é um dos interlocutores buscando conversar como um ser humano. Já na perspectiva de mídia, o sistema é apenas um meio através do qual outros interlocutores (usuários e designer) podem se comunicar.

A Tabela 2.1 apresenta um resumo comparativo das perspectivas de interação, destacando os diferentes significados de interação e os principais critérios de qualidade de cada uma delas. É importante observar que mais de uma perspectiva pode coexistir em um único sistema interativo. Em sistemas de empresas aéreas, por exemplo, encontramos a perspectiva de sistema empregada na escolha dos destinos e origens,

e a perspectiva de mídia empregada em seções do tipo “fale conosco”. A escolha das perspectivas será feita de acordo com o perfil e as necessidades dos usuários, com o contexto de uso e com o apoio computacional que pretendemos lhes oferecer.

Tabela 2.1 Comparação das perspectivas de interação (Kammersgaard, 1988).

perspectiva	significado de interação	fatores de qualidade mais evidentes
sistema	transmissão de dados	eficiência (tal como indicado pelo tempo de uso e número de erros cometidos)
parceiro de discurso	conversa usuário-sistema	adequação da interpretação e geração de textos
ferramenta	manipulação de ferramenta	funcionalidades relevantes ao usuário, facilidade de uso
mídia	comunicação entre usuários e comunicação designer-usuário	qualidade da comunicação mediada e entendimento mútuo

2.1.2 Interface

Se a interação é um processo que ocorre durante o uso, o que é a interface de um sistema interativo? A **interface** de um sistema interativo comprehende toda a porção do sistema com a qual o usuário mantém contato físico (motor ou perceptivo) ou conceitual durante a interação (Moran, 1981). Ela é o único meio de contato entre o usuário e o sistema. Por isso, a grande maioria dos usuários acredita que o sistema é a interface com a qual entram em contato (Hix e Hartson, 1993).

O contato *físico* na interface ocorre através do hardware e do software utilizados durante a interação. Dispositivos de entrada, como teclado, mouse, joystick, microfone, caneta (que escreve sobre a tela) e câmera (webcam), permitem ao usuário agir sobre a interface do sistema e participar ativamente da interação. Já os dispositivos de saída, como monitor, impressora e alto-falante, permitem ao usuário perceber as reações do sistema e participar passivamente da interação.

Os dispositivos mecânicos tinham uma relação física aparente com seu comportamento quando eram programados apenas via hardware. Por exemplo, as teclas numéricas de um telefone representavam apenas os números que o usuário podia discar. Mais recentemente, as teclas numéricas ganharam novas funções, como a de servir de teclado alfanumérico e para qualquer comportamento possível de se programar em software. Com a incorporação, em diversos dispositivos, de teclas de propósito múltiplo ou configuráveis, bem como telas de apresentação de informação, o software também passou a ter grande importância na definição da interface com usuário. O software determina os efeitos no comportamento do sistema decorrentes

das ações do usuário sobre os dispositivos de entrada, bem como os efeitos nos dispositivos de saída decorrentes de um processamento realizado pelo sistema. Em interfaces gráficas, por exemplo, pode-se clicar com o mouse (hardware) em um botão com um [x] e obter como resultado o término da execução do programa (software).

O contato *conceptual* com a interface envolve a interpretação do usuário daquilo que ele percebe através do contato físico com os dispositivos de entrada e de saída durante o uso do sistema. Essa interpretação permite ao usuário compreender as respostas do sistema e planejar os próximos caminhos de interação.

A interface com usuário determina os processos de interação possíveis, à medida que determina o que ele pode falar ou fazer, de que maneira e em que ordem. Portanto, quando definimos como a interação deve ocorrer, estamos restringindo ou determinando algumas características da interface, e vice-versa. Por exemplo, se projetarmos um processo de interação para compra on-line em três passos — escolher produtos, informar endereço de entrega e comunicar forma de pagamento — a interface deve permitir que o usuário percorra esses passos mantendo-o informado sobre a evolução do processo de compra. Outro exemplo nesse domínio seria a disposição das informações sobre produtos (modelo, preço, fabricante, especificações técnicas etc.) na interface, que pode facilitar ou dificultar a interação do usuário com o sistema para comparação de produtos.

Todos os elementos envolvidos no processo de interação estão fortemente relacionados. O contexto de uso influencia a forma como os usuários percebem e interpretam a interface, e também seus objetivos. Por exemplo, uma resposta sonora é pouco útil em um ambiente de uso barulhento porque pode passar despercebida. Os objetivos de um professor usando um editor de slides em casa e na sala de aula costumam ser diferentes. Em casa, o foco costuma ser a criação e edição de slides, enquanto, em sala de aula, o foco costuma ser a sua apresentação. As características físicas e cognitivas dos usuários também influenciam a definição de uma interface apropriada. Por exemplo, pessoas daltônicas podem não diferenciar informações expressas por determinadas cores na interface. Muitas delas não diferenciam o verde do vermelho. A formação, o conhecimento e as experiências do usuário também não podem ser ignorados na definição da interface. Por exemplo, não podemos esperar que um analfabeto aprenda a usar a interface lendo instruções na tela.

2.1.3 *Affordance*

As características físicas de um artefato evidenciam o que é possível fazer com ele e as maneiras de utilizá-lo. O mesmo ocorre com a interface com usuário. O conjunto de características do hardware e do software perceptíveis pelo usuário aponta para

um conjunto de operações que podem ser realizadas com o sistema interativo, bem como para as formas de realizá-las manipulando os elementos da interface. Existe um termo técnico para esse conjunto de características: *affordance*.

Gibson (1977, 1979) definiu o termo *affordance* na Psicologia, que mais tarde foi adaptado para IHC por Norman. Em IHC, a *affordance* de um objeto corresponde ao conjunto das características de um objeto capazes de revelar aos seus usuários as operações e manipulações que eles podem fazer com ele (Norman, 1988). Em uma interface gráfica, por exemplo, a *affordance* de um botão de comando diz respeito à possibilidade de pressioná-lo usando o mouse ou o teclado e, assim, acionar uma operação do sistema.

As *affordances* da interface de um sistema interativo são importantes para guiar o usuário sobre o que o sistema é capaz de fazer e como ele pode manipular a interface para fazê-lo. Os designers devem tomar cuidado para não criarem falsas *affordances*, pois os efeitos colaterais são inconvenientes. As falsas *affordances* podem dar a impressão de que a interface funciona de determinada maneira, quando na verdade funciona de outra forma. Uma falsa *affordance* pode ser criada, por exemplo, quando um desenvolvedor utiliza uma caixa de texto ou um botão de comando apenas para apresentar uma mensagem ou conteúdo não editável (Figura 2.7).

Resultado: 357 itens processados.

Resultado: 357 itens processados.

Resultado: 357 itens processados.

Figura 2.7 Exemplos de diferentes *affordances* de alguns elementos de interface (*widgets*).

Na caixa de texto, o usuário pode acreditar que é possível editar o texto da mensagem. No botão de comando, ele pode acreditar que existe um comando associado ao evento de pressionar o botão. Desses elementos, apenas o rótulo apresenta uma *affordance* adequada à apresentação de dados e mensagens ao usuário.

2.2 Qualidade em IHC

Usar um sistema interativo significa interagir com sua interface para alcançar objetivos em determinado contexto de uso. A interação e a interface devem ser adequadas para que os usuários possam aproveitar ao máximo o apoio computacional oferecido pelo sistema. Que características a interação e a interface devem ter para serem consideradas adequadas?

Os critérios de qualidade de uso enfatizam certas características da interação e da interface que as tornam adequadas aos efeitos esperados do uso do sistema. Os critérios de qualidade de uso descritos neste livro são: usabilidade, experiência do usuário, acessibilidade e comunicabilidade. A usabilidade é o critério de qualidade de uso mais conhecido e, por conseguinte, o mais frequentemente considerado. Para muitas pessoas, inclusive, qualidade de uso chega a ser sinônimo de usabilidade.

A **usabilidade** está relacionada com a facilidade de aprendizado e uso da interface, bem como a satisfação do usuário em decorrência desse uso (Nielsen, 1993). Tradicionalmente, a usabilidade enfoca a maneira como o uso de um sistema interativo no ambiente de trabalho é afetado por características do usuário (sua cognição, sua capacidade de agir sobre a interface e sua capacidade de perceber as respostas do sistema). Com a disseminação dos sistemas computacionais interativos em ambientes diferentes do trabalho, a usabilidade passou a englobar também as emoções e os sentimentos dos usuários. Por vezes essa qualidade relacionada com os sentimentos e emoções dos usuários é denominada de **experiência do usuário** (Sharp *et al.*, 2007).

Para um usuário tirar proveito do apoio computacional oferecido pelo sistema, não podem existir barreiras que o impeçam de interagir com sua interface. O critério de **acessibilidade** está relacionado à remoção das barreiras que impedem mais usuários de serem capazes de acessar a interface do sistema e interagirem com ele. Cuidar da acessibilidade significa permitir que mais pessoas possam interagir com o sistema, tenham elas alguma deficiência ou não. A intenção é incluir, não excluir.

O critério de **comunicabilidade** chama atenção para a responsabilidade de o designer comunicar ao usuário suas intenções de design e a lógica que rege o comportamento da interface. Esse critério se pauta no pressuposto de que, se o usuário tiver acesso à lógica de design, ele terá melhor condição de fazer um uso produtivo e criativo do apoio computacional oferecido pelo sistema.

A seguir, analisamos cada um desses critérios em mais detalhes.

2.2.1 Usabilidade e Experiência de Usuário

Ao definir os critérios de qualidade de software, a norma ISO/IEC 9126 (1991) define **usabilidade** como sendo:

Um conjunto de atributos relacionados com o esforço necessário para o uso de um sistema interativo, e relacionados com a avaliação individual de tal uso, por um conjunto específico de usuários.

E a norma sobre requisitos de ergonomia, ISO 9241-11 (1998), define usabilidade como sendo:

O grau em que um produto é usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico.

Segundo essa norma, **eficácia** está relacionada com a capacidade de os usuários interagirem com o sistema para alcançar seus objetivos *corretamente*, conforme o esperado. A **eficiência** está relacionada com os *recursos necessários* para os usuários interagirem com o sistema e alcançarem seus objetivos. Normalmente, os recursos necessários são tempo, mão de obra e materiais envolvidos. A norma também destaca a importância de considerarmos o grau de **satisfação** dos usuários com a experiência de usar o sistema interativo no contexto de uso para o qual foi projetado.

Nielsen (1993) define o critério de usabilidade como um conjunto de fatores que qualificam quão bem uma pessoa pode interagir com um sistema interativo. Esses critérios estão relacionados com a facilidade e o esforço necessários para os usuários aprenderem e utilizarem um sistema. Desse modo, a usabilidade endereça principalmente a capacidade cognitiva, perceptiva e motora dos usuários empregada durante a interação. Os fatores de usabilidade por ele considerados são:

- facilidade de aprendizado (*learnability*)
- facilidade de recordação (*memorability*)
- eficiência (*efficiency*)
- segurança no uso (*safety*)
- satisfação do usuário (*satisfaction*)

Cada sistema interativo possui características e peculiaridades que o tornam único e distinto dos demais. Logo, a interação com cada sistema é um processo particular que exige do usuário certo grau de aprendizado. Ele precisa dispor de tempo e interesse para se empenhar em aprender a utilizar um sistema interativo e ser capaz de usufruir de suas funcionalidades. É possível estabelecer níveis de aprendizado do uso do sistema. Por exemplo, podemos definir os conhecimentos e as habilidades necessárias para usufruir das funcionalidades do sistema num nível simples, intermediário e avançado. A **facilidade de aprendizado** se refere ao tempo e esforço necessários para que o usuário aprenda a utilizar o sistema com determinado nível de competência e desempenho.

As pessoas esperam que o apoio computacional oferecido por um sistema interativo seja tão simples, fácil e rápido de aprender quanto possível. Afinal, empregar tecnologias de informação e comunicação no nosso cotidiano se justifica para facilitar a realização das nossas atividades, e não torná-las mais difíceis e complexas. Isso vale tanto para um sistema de uso cotidiano, como correio eletrônico, quanto para um sis-

tema utilizado raramente, como o sistema de declaração anual de imposto de renda. Em atividades mais complexas, temos uma tolerância maior em relação ao esforço e tempo necessários para aprendermos a utilizar um sistema interativo. Portanto, cuidar da facilidade de aprendizado significa equilibrar (1) a complexidade da atividade sendo apoiada e o conjunto de funcionalidades oferecido como apoio, e (2) o tempo e o esforço necessários para aprender a utilizar o sistema em cada nível de competência e desempenho estabelecidos como meta. É possível avaliar o tempo e o esforço necessários para a transição entre diferentes níveis de competência e desempenho de uso. Por exemplo, podemos avaliar quanto tempo um usuário leva para aprender a realizar as atividades principais e quanto tempo ele leva para aprender a realizar um conjunto mais amplo de atividades.

O ser humano é capaz de aprender, mas também esquece o que aprendeu. Diante de um esquecimento, pistas sobre o que foi esquecido são muito úteis para resgatarmos da memória o que aprendemos no passado. Se a interface com usuário possuir elementos obscuros, mal organizados, sem sentido para o usuário, com passos mal encadeados ou muito diferentes do que ele espera, muito provavelmente o usuário terá dificuldade para lembrar como utilizar o sistema (Sharp *et al.*, 2007). A **facilidade de recordação** diz respeito ao esforço cognitivo do usuário necessário para lembrar como interagir com a interface do sistema interativo, conforme aprendido anteriormente.

Um sistema de fácil recordação auxilia o usuário a se lembrar de como utilizá-lo, evitando que ele cometa erros ao utilizar partes do sistema que já tenha utilizado anteriormente. Por exemplo, o usuário pode não se lembrar do nome de um item de menu, mas pode lembrar que ele faz parte de uma determinada categoria. Desse modo, a organização e descrição dos itens de menu ajudam o usuário a lembrar da opção desejada. Em outro exemplo, a interface pode revelar pistas sobre a sequência de operações durante a execução de uma tarefa através de ícones, nomes de comandos e opções de menus bem projetados. Sistemas de comércio eletrônico costumam guiar o usuário pelos passos necessários até a conclusão da compra, sempre indicando qual o passo atual em relação à sequência de passos necessários. A facilidade de recordação é especialmente importante quando existem operações ou sistemas com baixa frequência de uso, como, por exemplo, efetuar a matrícula numa universidade, a cada seis meses.

A maneira como um sistema interativo apoia a realização das atividades do usuário influencia o tempo necessário para realizá-las e, portanto, influencia a produtividade do usuário. A **eficiência** de um sistema interativo diz respeito ao tempo necessário para conclusão de uma atividade com apoio computacional. Esse tempo

é determinado pela maneira como o usuário interage com a interface do sistema. A eficiência de um sistema interativo se torna importante quando desejamos manter alta a produtividade do usuário, depois de ele ter aprendido a utilizar o sistema.

Errar faz parte do processo de aprendizado. Se uma pessoa se sente segura para tentar fazer algo sem medo de errar, ela possui melhores condições para experimentar fazer coisas novas e explorar novos caminhos. Sendo assim, é muito interessante que os sistemas interativos ofereçam segurança ao usuário durante o uso para motivá-lo a aprender a usar o software explorando suas funcionalidades. A **segurança no uso** se refere ao grau de proteção de um sistema contra condições desfavoráveis ou até mesmo perigosas para os usuários. Existem duas formas para alcançarmos a segurança no uso: buscando *evitar* problemas e auxiliando o usuário a *se recuperar* de uma situação problemática. Uma forma de evitar problemas é reduzir a possibilidade de acionar por engano teclas, botões e comandos indesejados. Um exemplo disso seria não colocar botões “perigosos” como “remover tudo” muito próximos a botões de “gravar” (Sharp *et al.*, 2007). Mecanismos para desfazer e refazer facilmente uma ação (*undo* e *redo*) e mecanismos para cancelar ou interromper operações demoradas são formas eficientes de recuperação de erros ou equívocos do usuário e, portanto, favorecem a exploração das funcionalidades do sistema (veja Seção 8.2).

A **satisfação do usuário** é o fator de usabilidade relacionado com uma avaliação subjetiva que expressa o efeito do uso do sistema sobre as emoções e os sentimentos do usuário. Até pouco tempo, sistemas interativos eram utilizados principalmente em atividades relacionadas ao trabalho. Por isso a satisfação do usuário costumava receber menor atenção do que outros critérios mais relevantes a essas atividades, como a eficiência, por exemplo.

Recentemente, os sistemas interativos deixaram de ser utilizados apenas no trabalho e passaram a estar presentes em muitas atividades humanas (entretenimento, educação, saúde, política etc.) e em diversos locais (no trabalho, em casa, na escola, em trânsito, no hospital, no museu, no shopping etc.). Essas novas atividades aumentaram a necessidade de considerarmos a forma como o uso de um sistema interativo afeta os sentimentos e as emoções do usuário. Alguns interpretam a preocupação com emoções e sentimentos dos usuários como uma atenção maior à satisfação do usuário como parte do critério de usabilidade (Bevan, 2009). Outros, no entanto, consideram essa preocupação como um critério de qualidade distinto, chamado de experiência do usuário (*user experience* — Sharp *et al.*, 2007).

Além da satisfação do usuário, tornou-se importante investigar outros aspectos da sua subjetividade, caracterizando seus sentimentos, estado de espírito, emoções e sensações decorrentes da interação com um sistema interativo em determinado

contexto de uso. Podemos investigar vários aspectos positivos e negativos dessa subjetividade, como, por exemplo (Sharp *et al.*, 2007): satisfação, prazer, diversão, entretenimento, interesse, atração, motivação, estética, criatividade, provocação, surpresa, desafio, cansaço, frustração e ofensa.

É claro que não podemos prever completamente nem controlar a experiência de cada usuário durante a interação. A experiência de uso é algo subjetivo, pessoal. Entretanto, podemos projetar sistemas interativos visando promover uma boa experiência de uso, incorporando características que promovam boas emoções nos usuários e que evitem provocar sensações desagradáveis, sempre respeitando as limitações dos usuários.

Existem alguns aspectos importantes para experiência do usuário a serem considerados durante o (re)projeto de um sistema interativo, como, por exemplo, atenção, ritmo, divertimento, interatividade, controle consciente e inconsciente, envolvimento e estilo de narrativa (Sharp *et al.*, 2007). Um bom envolvimento emocional dos usuários durante a interação agrega valor ao sistema interativo. Cabe ao designer decidir quais aspectos subjetivos devem ser promovidos durante a interação e articular isso com os demais critérios de qualidade de uso.

Dificilmente um único sistema será muito bom em todos os critérios de usabilidade, porque não é fácil articular esses critérios sem que haja perdas em um ou mais deles. Por exemplo, um sistema pode ser eficiente com muitas teclas de atalho, mas elas podem ser mais difíceis de serem lembradas por usuários ocasionais. Já um sistema com muitas explicações e tutoriais pode ser de fácil aprendizado, mas pode não satisfazer um usuário experiente, que privilegie a eficiência. É importante conhecermos as necessidades dos usuários e estabelecermos quais critérios de usabilidade devem ser priorizados no sistema em questão.

2.2.2 Acessibilidade

Durante a interação, o usuário emprega (1) sua habilidade motora para agir sobre os dispositivos de entrada, (2) seus sentidos (visão, audição e tato) e capacidade de percepção para identificar as respostas do sistema emitidas pelos dispositivos de saída, e (3) sua capacidade cognitiva, de interpretação e de raciocínio para compreender as respostas do sistema e planejar os próximos passos da interação. Se a interface impuser alguma barreira ao usuário durante o processo de interação, ele não será capaz de aproveitar o apoio computacional oferecido pelo sistema.

O critério de **acessibilidade** está relacionado com a capacidade de o usuário acessar o sistema para interagir com ele, sem que a interface imponha obstáculos. Melo e Baranauskas (2005, p. 1505) definem acessibilidade como sendo “a flexibili-

dade proporcionada para o acesso à informação e à interação, de maneira que usuários com diferentes necessidades possam acessar e usar esses sistemas”. Uma interface com usuário acessível não pode impor barreiras para interação e para o acesso à informação, nem no hardware e nem no software do sistema interativo.

A acessibilidade atribui *igual importância* a pessoas *com e sem* limitações na capacidade de movimento, de percepção, de cognição e de aprendizado. Cuidar da acessibilidade significa permitir que *mais pessoas* possam perceber, compreender e utilizar o sistema para usufruir do apoio computacional oferecido por ele (WAI, online). Isso não significa que o sistema deve ser desenvolvido para atender exclusivamente a uma classe especial de usuários. A intenção é incluir pessoas com limitações ou deficiências no grupo de usuários-alvo, e não excluir desse grupo as pessoas sem limitações ou deficiências.

Um usuário que possui limitações físicas (e.g., deficiência visual, auditiva e motora), mentais ou de aprendizado (e.g., analfabetos plenos e analfabetos funcionais) tem mais chances de encontrar barreiras que o dificultam ou impedem de interagir com o sistema. Essas limitações podem ser temporárias, como aquelas causadas por acidentes e superadas algum tempo depois, ou limitações persistentes por toda a vida, como cegueira e paralisia causadas por deficiência congênita ou por alguma doença grave. A idade dos usuários também influencia suas capacidades físicas, mentais e de aprendizado, seja quando criança, porque seu corpo ainda não amadureceu, ou na terceira idade, quando algumas de suas capacidades são afetadas pelo envelhecimento. Vejamos alguns casos em que usuários com limitações podem encontrar dificuldades para interagir com sistemas computacionais (Exemplo 2.3).

Exemplo 2.3 – Cenários evidenciando a importância da acessibilidade

Deficiência auditiva

Paulo é um deficiente auditivo que acessa a Internet frequentemente sem grandes dificuldades. A sua conexão com a Internet parou de funcionar em casa e ele precisa entrar em contato com seu provedor de acesso. Como ele se sentiria ao descobrir que é obrigado a utilizar um sistema interativo por telefone para ter acesso ao suporte do seu provedor de Internet? Todo o seu esforço para aprender o Português, além da Língua Brasileira de Sinais (Libras), não seria útil nesse caso.

Deficiência motora

João maneja bem o teclado e o mouse. Entretanto, no último mês ele descobriu uma tendinite crônica nas mãos e sente muitas dores ao manipular esses dois dispositivos de entrada. Certamente ele ficaria feliz se pelo menos alguns comandos pudessem ser ativados via voz até que sua dor diminuísse.

Deficiência visual

Joana é uma jovem brasileira deficiente visual interessada em continuar estudando. Ela ouviu no noticiário da TV que o vestibular de várias universidades públicas levará em conta a nota no Enem (Exame Nacional do Ensino Médio). Utilizando um leitor de telas, ela conseguiu acessar o site de inscrição do Enem (Figura 2.8) para obter informações a respeito do exame. No Web site ela descobriu que precisava do número de identidade e CPF, mas não conseguiu encontrar um link para iniciar a inscrição, nem percebeu que o período de inscrição terminou. Por que ela não percebeu essas informações? Se analisarmos a figura, vamos perceber que o link para iniciar a inscrição era uma figura, e a informação de que o período de inscrição terminou se encontrava dentro dessa figura. Nenhuma dessas informações pôde ser lida pelo leitor de tela, e ela não teve acesso a informações sobre um serviço que o Estado deveria oferecer para toda a população brasileira.

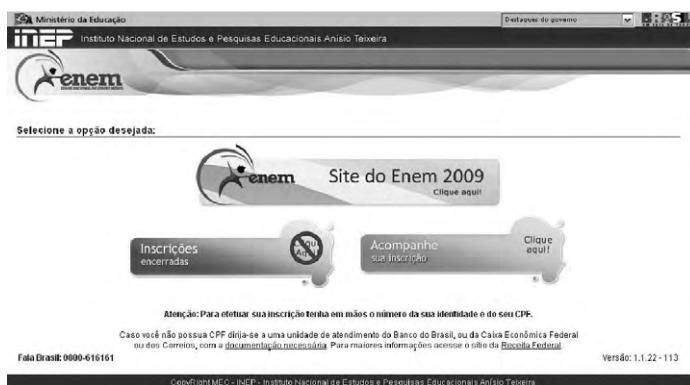


Figura 2.8 Site de inscrição no Enem em julho de 2009.²

Nesses exemplos, as limitações físicas dos usuários dificultaram ou impossibilitaram o acesso aos sistemas interativos. A interação tornou-se pouco produtiva ou impossível devido a dificuldades para agir sobre o sistema através dos dispositivos de entrada, e para perceber e interpretar os resultados emitidos pelos dispositivos de saída.

Um bom exemplo de adequação às limitações físicas e cognitivas do usuário são os dispositivos GPS (Sistema de Posicionamento Global) para guiar o motorista em trânsito utilizando mapas digitais. Enquanto dirige, o motorista não pode utilizar as mãos para agir sobre o dispositivo, nem ler instruções na tela. Desse modo, enquanto está parado, o motorista informa ao navegador GPS onde ele pretende ir. Durante o trajeto, o sistema vai lhe orientando sobre o caminho que deve seguir, via respostas sonoras. Repare que, nesse caso, o sistema precisou ser adequado a limitações temporárias impostas pelo contexto de uso. Podemos observar que nem sempre a acessibilidade está relacionada com deficiências persistentes ou com características de um grupo específico de usuários.

² <http://sistemasenem2.inep.gov.br/Enem2009>, último acesso em julho de 2009.

O governo brasileiro fornece vários serviços aos cidadãos por meio de sistemas computacionais, principalmente via Internet. Por exemplo, existem vários serviços do INSS e da Receita Federal disponíveis on-line; em alguns estados as matrículas em escolas públicas são realizadas on-line; e em alguns municípios é possível obter segunda via do IPTU no site da prefeitura. O governo deve servir igualmente a todos os cidadãos do país, sem discriminação e respeitando as limitações e diferenças de cada um. Por isso devemos permitir que pessoas com limitações físicas, mentais e de aprendizado tenham acesso aos serviços oferecidos via tecnologias de informação e comunicação. Essa preocupação se manifesta no decreto presidencial número 5.296, de 2 de dezembro de 2004, que regulamenta as leis nº 10.048, de 8 de novembro de 2000, e nº 10.098, de 19 de dezembro de 2000.³ Esse decreto torna obrigatória a acessibilidade em sites do governo. No texto do decreto, podemos destacar:

Art. 8º Para os fins de acessibilidade, considera-se:

I - acessibilidade: condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida;

II - barreiras: qualquer entrave ou obstáculo que limite ou impeça o acesso, a liberdade de movimento, a circulação com segurança e a possibilidade de as pessoas se comunicarem ou terem acesso à informação, classificadas em:

(...)

d) barreiras nas comunicações e informações: qualquer entrave ou obstáculo que dificulte ou impossibilite a expressão ou o recebimento de mensagens por intermédio dos dispositivos, meios ou sistemas de comunicação, sejam ou não de massa, bem como aqueles que dificultem ou impossibilitem o acesso à informação;

(...)

Art. 47. No prazo de até doze meses a contar da data de publicação deste Decreto, **será obrigatória a acessibilidade nos portais e sítios eletrônicos da administração pública** na rede mundial de computadores (Internet), para o uso das pessoas portadoras de deficiência visual, garantindo-lhes o pleno acesso às informações disponíveis.

As limitações físicas, mentais e de aprendizado dos usuários não podem ser desprezadas, sejam elas limitações permanentes, temporárias ou circunstanciais. É desejável que um sistema interativo seja acessível a qualquer pessoa, mas a acessibilidade depende das características dos usuários que pretendemos atender e dos contextos de

³ http://www.planalto.gov.br/CCIVIL/_Ato2004-2006/2004/Decreto/D5296.htm.

uso pretendidos. Cada tipo de limitação ou deficiência requer um cuidado específico para criarmos interfaces acessíveis. Por exemplo, uma deficiência visual requer cuidados bem diferentes de uma deficiência auditiva. Portanto, o zelo com a acessibilidade também requer conhecimento sobre as capacidades e limitações dos usuários e sobre os diferentes contextos de uso (Stephanidis, 2001; Melo e Baranauskas, 2006; Lazar, 2007).

2.2.3 Comunicabilidade

Um sistema interativo é resultado de um processo de design no qual um designer estabelece uma visão (interpretação) sobre os usuários, seus objetivos, o domínio e o contexto de uso e toma decisões sobre como apoiá-los. Para o usuário usufruir melhor do apoio computacional, é desejável que o designer remova as barreiras da interface que impedem o usuário de interagir (acessibilidade), torne o uso fácil (usabilidade) e comunique ao usuário as suas concepções e intenções ao conceber o sistema interativo. Mas por que o usuário precisaria saber disso? Vamos analisar duas situações bem simples e comuns no uso de TICs (Exemplo 2.4).

Exemplo 2.4 – Importância da comunicabilidade

Cópia de arquivos

Maria gosta de música e está interessada em utilizar o computador para organizar e ouvir seus arquivos de música. Ela comprou seu primeiro computador recentemente e ainda não sabe utilizar os sistemas interativos disponíveis.

Maria decide colocar alguns arquivos de música no seu *pen drive* para poder ouvir em outro lugar. Depois de algum tempo copiando os arquivos, mas antes de concluir a cópia, ela decide parar a operação em andamento porque está atrasada para sair de casa. O que acontece se ela cancelar a operação não concluída? Os arquivos já copiados permanecem no *pen drive* ou serão removidos? Como Maria pode aprender o significado de cancelar a operação de cópia em andamento? A Figura 2.9 apresenta a interface do Windows® XP, que permite a Maria acompanhar a operação de cópia de arquivos.

Não há nessa interface uma explicação do que significa para o sistema (conforme concebido pelo designer) cancelar a cópia em andamento. Existe mais de uma interpretação aceitável para o comando cancelar: (1) apenas a operação de cópia é interrompida; e (2) a operação de cópia é interrompida e seus resultados parciais são desfeitos (isto é, os arquivos já copiados são apagados do *pen drive*). Por não conhecer qual o significado do comando cancelar nessa interface, Maria se sente insegura sobre o comportamento do sistema. Para compreender o funcionamento do sistema nesse caso, ela precisa arriscar cancelar a cópia e verificar se alguns arquivos copiados ainda permanecem no seu *pen drive*. Infelizmente, nem sempre é simples verificar o funcionamento do sistema. Seria muito mais fácil e adequado o próprio designer comunicar ao usuário (por exemplo, através de dicas, instruções ou mensagens associadas ao botão cancelar) qual foi o significado que ele atribuiu a esse comando, ou ainda oferecer diferentes comandos para os possíveis comportamentos identificados, cada qual indicando o significado correspondente.

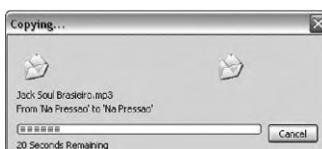


Figura 2.9 Copiando arquivos para outro diretório no Windows XP®.

Reprodução de música

Usando a interface do reproduutor de música Songbird, Maria também fica insegura sobre seu comportamento. Ela quer ouvir as músicas de um CD, exceto uma que lembra seu namorado porque brigou com ele há poucos dias. Ela então decide remover a música da lista presente na interface do Songbird. Conforme apresentado na Figura 2.10, ela ativa o menu pop-up, e decide clicar em Remover. Qual será o efeito de clicar nesse item de menu? A música será removida da lista de reprodução, será removida da biblioteca do Songbird ou o arquivo da música será removido do computador? Novamente, a interface do sistema não comunica ao usuário o significado atribuído pelo designer a um comando, e Maria volta a ficar insegura. Nesse caso, não compreender corretamente o significado do comando remover pode trazer consequências indesejadas e difíceis ou impossíveis de serem revertidas, pois Maria pode perder o arquivo da música que lembra seu namorado. O objetivo dela no momento não é apagar o arquivo, mas ouvir apenas as outras músicas do CD agora. Essa dúvida e insegurança não aconteceriam se o designer deixasse claro o significado do item Remover.



Figura 2.10 Removendo arquivo de música no Songbird.

Problemas na comunicação das concepções e intenções do designer para o usuário se tornam mais significativos quando tratamos de estratégias de uso da interface para alcançar diferentes objetivos. É mais difícil o usuário aprender estratégias de uso concebidas pelo designer sem que elas lhe sejam bem comunicadas. Por exemplo, é difícil os usuários perceberem e aproveitarem as formas mais eficientes de organizar e-mails em sistemas como Outlook® e Thunderbird® sem que exista uma comunicação do designer explícita e eficiente nesse sentido. Para evitar que o sistema seja subutilizado, de Souza (2005b) propõe que o designer, além de *produzir* sistemas interativos, também deve *apresentá-los* adequadamente ao usuário durante a interação. Nesse ponto de vista, a interação humano-computador envolve a comunicação dos passos necessários para alcançar um objetivo, e também do valor de estratégias inovadoras para realizar atividades e solucionar problemas com apoio computacional.

O conceito de comunicabilidade foi proposto pela engenharia semiótica (de Souza, 2005a), teoria de IHC discutida na Seção 3.8. A **comunicabilidade** diz respeito à capacidade da interface de comunicar ao usuário a *lógica do design*: as intenções do designer e os princípios de interação resultantes das decisões tomadas durante todo o processo de design (Prates *et al.*, 2000a; de Souza, 2005a; de Souza e Leitão, 2009). Acreditamos que, se um usuário for capaz de compreender a lógica utilizada na concepção do sistema interativo, terá maiores chances de fazer um uso criativo, eficiente e produtivo dele (Prates e Barbosa, 2007; 2003).

É importante observar que compreender a *lógica de design* não implica adquirir conhecimentos técnicos de design de um sistema interativo, mas sim obter uma compreensão pragmática e utilitária das relações de causa e efeito que determinam seu comportamento (de Souza, comunicação pessoal). O entendimento dessa lógica de design permite que os usuários tirem melhor proveito da tecnologia e sigam estratégias adequadas a cada situação de uso. Por exemplo, não precisamos conhecer a mecânica de um automóvel em profundidade para dirigi-lo. Mas fazemos melhor uso do automóvel se entendemos os riscos e as consequências de utilizá-lo com pouca gasolina, com nível de óleo inadequado, de dirigir em alta velocidade em pistas escorregadias, de dirigir muito próximos do carro à nossa frente etc. De modo análogo, não precisamos saber como funcionam os recursos de estilos de formatação ou numeração automática de um editor de texto para utilizá-lo, mas de posse desse conhecimento podemos fazer uso mais eficiente dele e menos propenso a erros.

A lógica do design comunicada ao usuário deve refletir as decisões tomadas sobre: a quem se destina o sistema, para que ele serve, qual a vantagem de utilizá-lo, como ele funciona e quais são os princípios gerais de interação com o sistema (Prates *et al.*, 2000a; de Souza, 2005a; de Souza e Leitão, 2009). Essas questões normalmente fazem parte da atividade de design de um sistema interativo, porém nem sempre o designer se preocupa em comunicá-las adequadamente através da interface com usuário. Como vimos nos exemplos de cópia de arquivos no Windows® XP e de remoção de uma música no Songbird, se os usuários não compreenderem a lógica de design, a interação frequentemente se torna um processo de tentativa e erro, tedioso, ineficiente ou até mesmo arriscado.

A analogia é um recurso de comunicação utilizado para facilitar e aumentar a comunicabilidade. Esse recurso permite ao usuário formular hipóteses sobre a interação com sistemas interativos tendo como base suas experiências de interação anteriores com artefatos semelhantes. O uso de analogias deve contribuir para que as hipóteses do usuário sobre como interagir sejam compatíveis com aquelas pretendidas

pelo designer. Contudo, é importante deixar claros os limites das analogias para não induzir o usuário a criar hipóteses erradas (Exemplo 2.5).

Exemplo 2.5 – Uso de analogias

Analice rapidamente a interface dos dois sistemas na Figura 2.11, sem se preocupar em ler o conteúdo de seus elementos textuais. O que esses sistemas fazem? Eles são reprodutores de música? Como você chegou à sua conclusão?

Esses dois sistemas possuem botões, imagens e características semelhantes a um CD Player físico, no qual o usuário deve apertar botões para controlar a reprodução (*play, pause, stop, next e previous*), girar um botão para controlar o volume, pressionar o botão de *eject* para abrir o compartimento de CDs, e assim por diante. Sem dúvida, o uso dessa analogia com CD Players favorece a comunicabilidade de sistemas reprodutores de áudio e vídeo. Entretanto, o que podemos dizer sobre a comunicabilidade quando essa mesma analogia é utilizada em um programa antivírus? O Avast (o programa na parte inferior da Figura 2.11) é um programa antivírus, não um reprodutor de música, como facilmente poderíamos interpretar analisando a interface. O designer da versão 4.7 do Avast não foi muito feliz na escolha da analogia da interface com um CD Player, pois cria expectativas que não pode atender e induz os usuários a criarem várias hipóteses falsas sobre como interagir com o sistema e o que ele é capaz de fazer. Felizmente essa analogia com o CD Player já foi abandonada em versões posteriores do Avast.

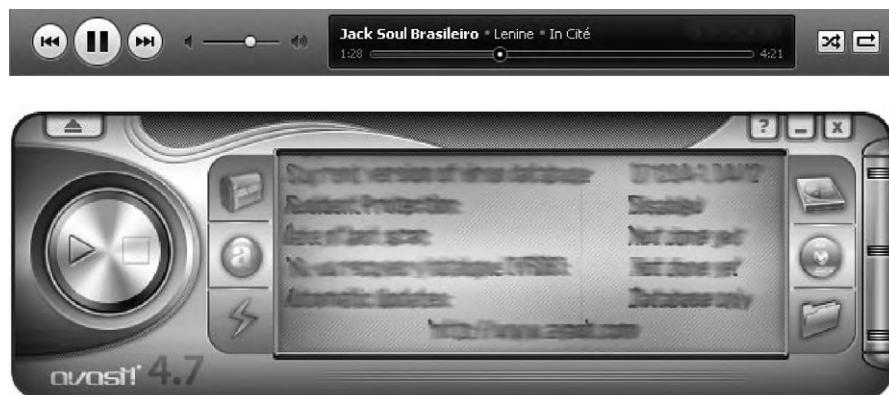


Figura 2.11 Interfaces do Songbird (em cima) e do Avast (embaixo, cujo texto foi propositalmente desfocado).

Outro recurso de comunicação que favorece a comunicabilidade é oferecer mais informações sobre a lógica de design conforme a demanda dos usuários. Um exemplo interessante de melhoria na comunicabilidade é percebido quando comparamos as dicas de botões da barra de ferramentas no Microsoft Office®, versões XP e 2007. A Figura 2.12 apresenta a dica sobre o botão Pincel nas duas versões.

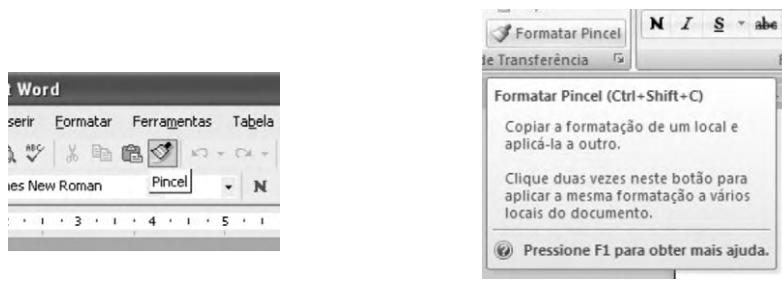


Figura 2.12 Dica da ferramenta de pincel no Microsoft Office® (a) XP e no (b) 2007.

Enquanto a versão XP apresenta apenas o nome do comando associado ao botão, a versão 2007 apresenta também o significado do comando, as teclas de atalho a ele associadas, uma estratégia de uso para aplicá-lo em múltiplos locais do documento e informações sobre como obter mais ajuda. Observamos uma grande melhoria na qualidade da informação sobre a lógica de design e, consequentemente, da comunicabilidade do sistema. Essa melhoria contribui também para a usabilidade do sistema, pois ela facilita o aprendizado sobre a cópia de formato e revela as teclas de atalho e os efeitos do duplo clique que tornam seu uso mais eficiente. Sendo assim, um sistema com alta comunicabilidade é com frequência um sistema com alta usabilidade.

Apesar de distintos, os diversos critérios de qualidade de uso estão fortemente interligados, influenciando uns aos outros. Por exemplo, quando uma pessoa com deficiência visual consegue navegar sem grandes barreiras (acessibilidade) por sites na Web e obter as informações desejadas, sua motivação e satisfação (experiência do usuário ou usabilidade) tendem a aumentar, porque ela se torna capaz de realizar atividades sozinha e adquire certa independência. Em contrapartida, mesmo que uma interface seja acessível, se há ambiguidade ou falta de clareza no significado dos elementos de interface (baixa comunicabilidade), como ocorre no comando Cancelar da cópia de arquivos no XP e no comando Remover no Songbird, a eficiência do usuário e a facilidade de aprendizado do sistema tendem a diminuir. Além disso, a incerteza sobre o efeito de uma ação pode causar angústia e insatisfação aos usuários.

Em geral, quando um usuário consegue compreender como o sistema funciona porque o designer se expressou adequadamente através da interface (comunicabilidade), torna-se mais fácil aprender a utilizá-lo (usabilidade).

Ao projetarmos um sistema interativo, nem sempre é possível satisfazer igualmente todos os critérios e aspectos envolvidos na qualidade de uso, seja por questões de tempo, orçamento ou mesmo incompatibilidade entre critérios. Sendo assim, é importante definir quais são os critérios prioritários no sistema em questão para poder privilegiá-los no projeto de interação. A prioridade dos critérios de qualidade de uso

deve ser definida com base no conhecimento sobre os usuários (limitações, necessidades, motivações etc.), suas atividades e objetivos, e contextos de uso.

Atividades

1. *Fatores de Usabilidade.* Identifique quais fatores de usabilidade deveriam ser privilegiados nos seguintes casos:
 - um sistema para gestão dos documentos produzidos e consumidos por uma organização;
 - um quiosque de informações em uma livraria;
 - um caixa eletrônico;
 - um sistema Web para fornecer os resultados de exames de saúde a pacientes e seus médicos;
 - um jogo educacional de simulação de fenômenos físicos (e.g., deslocamento, aceleração e atrito).
2. *Acessibilidade.* Cite exemplos de sistemas interativos para os quais a acessibilidade beneficiaria seus usuários em certas situações. Discuta os benefícios da acessibilidade nesses sistemas para os usuários e para a organização responsável pelo sistema.
3. *Comunicabilidade.* Na interface do Microsoft Powerpoint® 2007 ou posterior, analise os signos correspondentes ao uso da caneta (*ink*). Tente identificar a visão do designer sobre para que serve esse recurso e como ele deve ser utilizado. Compare a edição de ilustrações utilizando a caneta e utilizando formas geométricas predefinidas (e.g., criação, modificação, seleção, agrupamento e deslocamento das ilustrações).
4. *Critérios de qualidade de uso.* Escolha alguns sistemas interativos a que você tenha acesso e que possa utilizar. Inspecione sua interface para analisar usabilidade, experiência do usuário, acessibilidade e comunicabilidade, considerando diferentes perfis de usuário:
 - um usuário que está utilizando o sistema pela primeira vez;
 - um usuário que utiliza o sistema diariamente;
 - um usuário que enxerga com dificuldade;
 - um usuário com baixo grau de instrução ou analfabeto funcional;
 - um usuário que tem baixo poder de concentração;
 - um usuário que realiza diversas atividades ao mesmo tempo e é interrompido com frequência;
 - um usuário que realiza uma tarefa longa, que precisa ser suspensa no final do dia e retomada no dia seguinte.

3

Abordagens Teóricas em IHC

Objetivos do Capítulo

- Apresentar fundamentos teóricos de base psicológica, etnográfica e semiótica: leis de Hick-Hyman e de Fitts, psicologia aplicada, princípios da Gestalt, engenharia cognitiva, ações situadas, teoria da atividade e engenharia semiótica.
 - Discutir como os fundamentos teóricos influenciam métodos e modelos utilizados no projeto e avaliação da interação humano-computador.
-

Embora IHC seja uma área de cunho bastante prático, muitos dos métodos, modelos e técnicas utilizados em IHC se baseiam em teorias, em particular teorias de base psicológica (principalmente cognitiva), etnográfica e semiótica. Conhecer essas teorias é fundamental, não apenas para melhor entender os métodos, modelos e técnicas apresentados na literatura de IHC, mas também para saber quando utilizá-los e identificar a necessidade de adaptá-los em projetos de design particulares, seja em domínios complexos ou envolvendo tecnologias inovadoras.

3.1 Introdução

Antes de apresentar os processos, métodos, técnicas, modelos e representações, é importante introduzir seus embasamentos teóricos. Este capítulo apresenta abordagens que têm feito grandes contribuições para a área de IHC: abordagens ancoradas na psicologia, na etnografia e na semiótica.

As primeiras abordagens teóricas utilizadas para investigar fenômenos de interação humano-computador nasceram na psicologia. Nos anos 50, com ênfase na psicologia experimental, diversos modelos de informação dos processos psicológicos surgiram para mensurar e modelar o comportamento humano (MacKenzie, 1991). Em IHC, o interesse nesses modelos se deve ao fato de permitirem modelar e prever o desempenho humano. Dentre os modelos propostos, os que mais utilizamos em IHC são a lei de Hick-Hyman para o tempo de reação de escolha (Hick, 1952; Hyman, 1953) e a lei de Fitts, para a capacidade de processamento de informação do sistema motor humano (Fitts, 1954).

Com base principalmente na psicologia cognitiva, no início dos anos 80, a atenção voltou-se para os aspectos cognitivos da interação humano-computador. Dessa época destacam-se o modelo de processador humano de informações (Card *et al.*, 1983) e a engenharia cognitiva (Norman, 1986).

No final da década, Suchman (1987) desafiou as abordagens de base cognitiva e trouxe para o estudo dos fenômenos de IHC o conceito de ação situada e práticas da etnometodologia. Dando sequência à investigação da atividade humana em contexto, surgiram trabalhos ancorados na teoria da atividade (Bødker, 1996) e trabalhos que ampliam a noção de cognição, como a cognição distribuída (Hollan *et al.*, 2000).

Mais recentemente, e com base na semiótica, a engenharia semiótica firmou-se como uma teoria de IHC centrada nos processos de significação e comunicação que envolvem designers, usuários e sistemas interativos (de Souza, 2005a).

3.2 Psicologia Experimental

3.2.1 Lei de Hick-Hyman

A lei de Hick-Hyman relaciona o tempo que leva para uma pessoa tomar uma decisão com o número de possíveis escolhas que ela possui (Hick, 1952; Hyman, 1953). Essa lei define que o tempo médio, T, necessário para escolher dentre N opções pode ser calculado aproximadamente pelas seguintes fórmulas, onde k é empiricamente determinado. Em geral, assumimos que $k \sim 150\text{ ms}$:

$$T = k \times \log_2(N+1), \text{ caso as opções tenham igual probabilidade;}$$

ou

$$T = k \times \sum p_i \log_2(1 + 1/p_i), \text{ onde } p_i \text{ é a probabilidade da alternativa } i, \text{ caso as } N \text{ opções tenham probabilidades diferentes.}$$

Em linhas gerais, a lei de Hick-Hyman indica que uma pessoa subdivide o conjunto total de opções em categorias, eliminando aproximadamente metade das opções a cada passo, em vez de considerar todas as escolhas uma a uma, o que requereria tempo linear. Essa lei pode ser utilizada para fazer uma estimativa de quanto tempo uma pessoa levará para encontrar uma dentre diversas opções disponíveis numa interface, como, por exemplo, os itens de uma lista de opções em ordem alfabética. No entanto, caso não haja um princípio de organização das opções que permita ao usuário eliminar metade delas rapidamente, essa lei não se aplica, pois a busca binária não pode ser realizada.

3.2.2 Lei de Fitts

Originada na psicologia experimental, a lei de Fitts relaciona o tempo (T) que uma pessoa leva para apontar para algo com o tamanho (S) do objeto-alvo e com a distância (D) entre a mão da pessoa e esse objeto-alvo (Fitts, 1954; Figura 3.1).

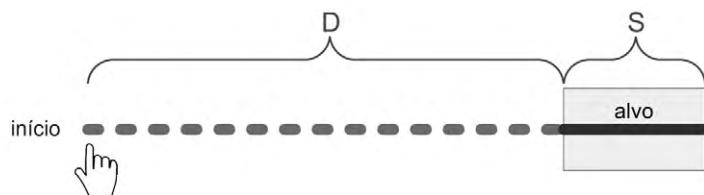


Figura 3.1 Ilustração da lei de Fitts.

Segundo Fitts, o tempo médio para apontar para um alvo pode ser calculado através de uma fórmula como a seguir:

$$T = k \times \log_2(D/S + 0.5), \text{ onde a constante } k \sim 100\text{ms} \text{ é determinada empiricamente e pode variar conforme o tipo de dispositivo utilizado.}$$

Variações dessa lei¹ são utilizadas para modelar o tempo que leva para um mouse ou outro dispositivo de entrada semelhante atingir um objeto numa tela.

Importante para aplicações em que o desempenho é crítico, a lei de Fitts ajuda os designers a decidirem sobre o tamanho e a localização de elementos de interface com os quais o usuário precisa interagir. Essa lei pode ser considerada em diversas situações de design (Tognazzini, 1999).

Um botão de acionamento de operação pode possuir ambos, imagem e rótulo. Quando o usuário já conhece o botão, o rótulo poderia ser dispensado. Porém, sua presença torna o botão maior e, portanto, seu acesso mais rápido (Figura 3.2).

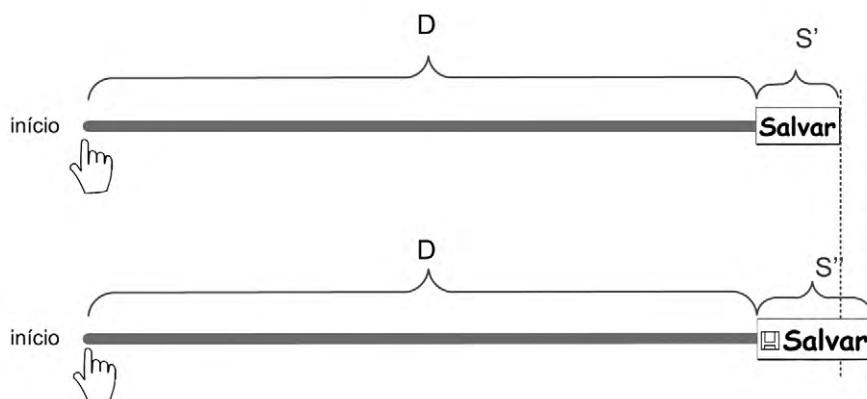


Figura 3.2 Botões com somente rótulo ou com rótulo e ícone.

Uma palheta de ferramentas deve ser posicionada ao longo de um lado da tela. Tal posicionamento permite que um deslocamento indefinidamente longo naquela direção acerte o alvo. Quando há uma palheta com poucas ferramentas, a lei de Fitts indica que é melhor organizá-las em uma única coluna ou linha, ao longo de um lado da tela, do que distribuí-las em duas ou mais colunas ou linhas (Figura 3.3).

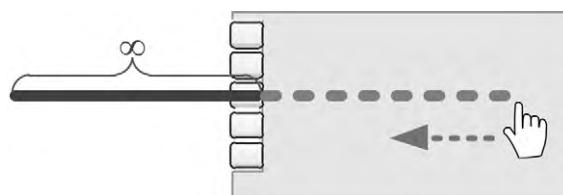


Figura 3.3 Posicionamento de palhetas de ferramentas num lado da tela.

No sistema operacional Mac OS®, o menu de uma aplicação fica sempre no topo da tela, e não no topo de cada janela, como no Microsoft Windows®. O acesso ao menu

¹ Algumas variações desta fórmula encontradas na literatura são: $T = a + b \log_2(2D/S)$ e $T = a + b \log_2(D/S + 1)$, onde a e b são constantes determinadas empiricamente.

no topo da tela é, em média, em torno de cinco vezes mais rápido do que um menu semelhante em uma aplicação Windows (Figura 3.4).

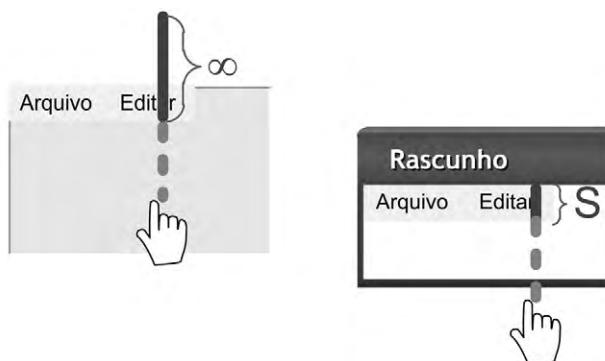


Figura 3.4 Posicionamento do menu no topo da tela e no topo da janela.

Um menu pop-up circular (*pie menu*) tem como vantagem sobre um menu pop-up horizontal o fato de que todas as opções estão equidistantes e próximas do ponto em que o menu foi acionado (Figura 3.5).



Figura 3.5 Menu pop-up circular.

3.3 Psicologia Cognitiva Aplicada

Card, Moran e Newell (1983) propuseram uma psicologia aplicada de processamento de informação. Segundo eles, a interação humano-computador consiste em o usuário e o computador se engajarem num diálogo comunicativo com o objetivo de realizar alguma tarefa. E todos os mecanismos utilizados nesse diálogo constituem a interface: os dispositivos físicos, como os teclados e as telas, assim como os programas computacionais que controlam a interação. Seu objetivo era criar uma psicologia baseada em análise de tarefas, cálculos e aproximações, para que o designer do sistema pudesse alcançar um equilíbrio entre parâmetros computacionais de desempenho humano e outras variáveis de engenharia.

Para eles, a análise da estrutura da tarefa oferece grande parte do conteúdo preditivo da psicologia. Uma vez que conheçamos os objetivos das pessoas, e consideran-

do suas limitações de percepção e de processamento de informação, devemos poder fornecer respostas a perguntas do tipo: aproximadamente quanto tempo leva para uma pessoa realizar as tarefas físicas predefinidas que lhe permitem alcançar seus objetivos?

3.3.1 Processador Humano de Informação

Com base na psicologia de processamento de informações, Card, Moran e Newell (1983) propuseram o “processador humano modelo” de informações (*Model Human Processor*, MHP). Segundo eles, o uso de modelos que veem o ser humano como um processador de informações fornece um arcabouço comum nos quais modelos de memória, de resolução de problemas, de percepção e de comportamento podem ser integrados uns com os outros. Considerando a mente humana como um sistema de processamento de informações, é possível fazer previsões aproximadas de parte do comportamento humano.

O MHP é composto de três subsistemas, cada qual com suas próprias memórias e processadores, juntamente com alguns princípios de operação (Figura 3.6): o perceptivo, o motor e o cognitivo.

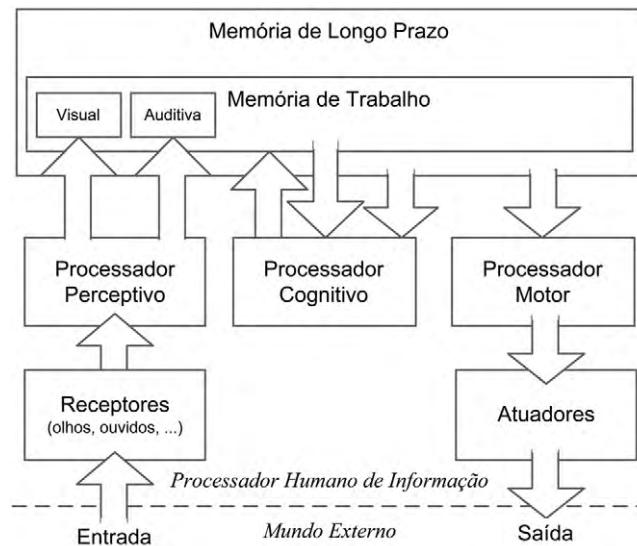


Figura 3.6 Modelo do Processador Humano de Informações (adaptado de Card *et al.*, 1983).

O sistema **perceptivo** transmite as sensações do mundo físico detectadas pelos sistemas sensoriais do corpo (visão, audição, tato, olfato, paladar) para representações mentais internas. A visão central, a visão periférica, os movimentos dos olhos e os movimentos da cabeça operam como um sistema integrado para nos fornecer uma

representação contínua da cena visual de interesse. Essas sensações são armazenadas temporariamente em áreas de memória sensorial (principalmente nas memórias visual e auditiva), ainda codificadas fisicamente e com um tempo de decaimento (ou esquecimento) rápido, conforme a intensidade do estímulo. Em seguida, algumas dessas sensações são codificadas simbolicamente e armazenadas na memória de trabalho.

O processador cognitivo pode especificar quais conteúdos das memórias sensoriais devem ser codificadas simbolicamente e armazenadas na memória de trabalho, principalmente quando o conteúdo da memória perceptiva for complexo ou percebido apenas por um período de tempo muito curto. O **sistema cognitivo** recebe a informação codificada simbolicamente dos armazenamentos sensoriais na sua memória de trabalho e utiliza informações previamente armazenadas na memória de longo prazo para tomar decisões sobre como responder aos estímulos recebidos.

Nosso pensamento é traduzido em ação através da ativação de padrões de músculos voluntários, em uma série de micromovimentos discretos realizados pelo **sistema motor**.

Com relação às memórias, os parâmetros a serem considerados são: a capacidade de armazenamento em número de itens, o tempo de decaimento (no caso, o tempo para o esquecimento) de um item e o tipo de código principal (físico, acústico, visual ou semântico). Já com relação aos processadores, o parâmetro mais importante é o tempo do ciclo.

Para algumas tarefas, uma pessoa precisa se comportar como um processador serial, ou seja, realizar uma tarefa de cada vez. Um exemplo desse tipo de tarefa é pressionar uma tecla em resposta a um estímulo visual, tal como o acendimento de uma lâmpada. Para outras tarefas, é possível que haja uma operação integrada e paralela dos três subsistemas, com informação fluindo continuamente da entrada à saída, em um período de tempo tão curto que todos os processadores trabalham simultaneamente. Por exemplo, digitação, leitura e tradução simultânea são tarefas desse tipo (Card *et al.*, 1983).

Nas tarefas mais simples, o sistema cognitivo serve apenas para conectar as entradas do sistema perceptual às saídas adequadas do sistema motor. Mas a maioria das tarefas é complexa e envolve aprendizado, recuperação de fatos ou solução de problemas. A memória de trabalho retém informações em uso, e a memória de longo prazo armazena conhecimento para uso futuro. Os elementos ativados da memória de longo prazo consistem em símbolos ou grupos de símbolos, chamados de *chunks* (elementos de informação). O conteúdo de um *chunk* depende da tarefa, da pessoa e do conteúdo da sua memória de longo prazo. Quando um *chunk* na memória de lon-

go prazo é ativado, essa ativação se espalha para *chunks* relacionados (e.g., de um livro para a livraria onde foi comprado, para a viagem em que foi comprado, para as fotos que foram tiradas etc.). Miller (1956) mostrou que a capacidade da memória de trabalho é limitada a 7 ± 2 *chunks*. Como ela é limitada, à medida que novos *chunks* são ativados, outros *chunks* que se encontram na memória de trabalho tornam-se menos acessíveis. Os itens armazenados têm uma determinada probabilidade de poderem ser recuperados mais tarde da memória de longo prazo. Quanto mais associações um item tiver ao ser armazenado, maior é a probabilidade de ele ser recuperado. Se uma pessoa quer poder se lembrar de alguma coisa mais tarde, a melhor estratégia é tentar associá-la a itens que já estão na memória de longo prazo.

Card e coautores (1983) calcularam o período de tempo aproximado do ciclo t para diversas tarefas comuns realizadas por pessoas com diferentes níveis de habilidades. Eles mostraram como sua abordagem permite calcular a taxa de quadros em uma animação necessária para criar a ilusão de movimento, a taxa máxima de recepção de código Morse para permitir a sua decodificação por uma pessoa, o tempo entre dois eventos para manter uma ilusão de causalidade e o tempo que uma pessoa leva para ler um texto. Ainda com base no MHP, eles elaboraram um modelo de decomposição de tarefas chamado GOMS, amplamente utilizado até hoje em análise e design da interação humano-computador. O modelo GOMS é descrito na Seção 6.4.2.

3.3.2 Princípios de Gestalt

Segundo Ware (2003), muito da nossa inteligência pode ser caracterizada pela nossa capacidade de identificar padrões, e o sistema visual é o nosso mecanismo de reconhecimento de padrões mais sofisticado. Ware destaca que um objetivo primário do design de representações visuais deve ser mapear dados numa forma visual compatível com as nossas capacidades perceptivas.

A escola de psicologia gestáltica foi fundada em 1912, e dentre seus principais pesquisadores encontram-se Wesheimer, Koffka e Kohler (Ware, 2003). Eles produziram um conjunto de leis de percepção de padrões, denominadas leis gestálticas ou simplesmente de Gestalt (Figura 3.7):

- **proximidade:** as entidades visuais que estão próximasumas das outras são percebidas como um grupo ou unidade;
- **boa continuidade:** traços contínuos são percebidos mais prontamente do que contornos que mudem de direção rapidamente;
- **simetria:** objetos simétricos são mais prontamente percebidos do que objetos assimétricos;

- **similaridade:** objetos semelhantes são percebidos como um grupo;
- **destino comum:** objetos com a mesma direção de movimento são percebidos como um grupo;
- **fecho:** a mente tende a fechar contornos para completar figuras regulares, “completando as falhas” e aumentando a regularidade.

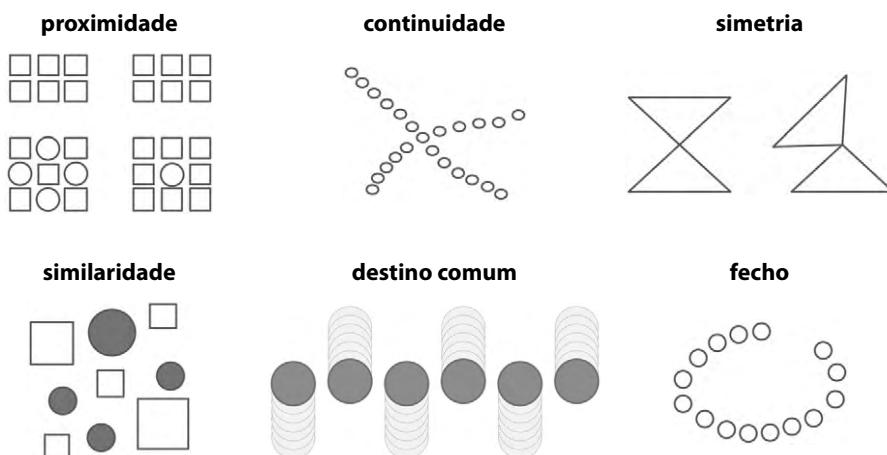


Figura 3.7 Ilustração de princípios gestálticos comumente considerados.

De acordo com Ware (2003), podemos considerar como adições recentes às leis de Gestalt as seguintes (Figura 3.8):

- **região comum:** objetos dentro de uma região espacial confinada são percebidos como um grupo (Palmer, 1992);
- **conectividade:** objetos conectados por traços contínuos são percebidos como relacionados (Palmer e Rock, 1994).

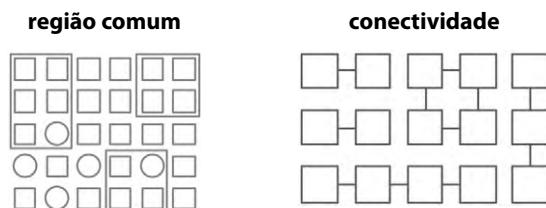


Figura 3.8 Ilustração dos princípios gestálticos de região comum e conectividade.

3.3.3 Percepção de Cores

Estudos sobre a percepção de cores e luminância resultaram em diversas diretrizes de design que podem ser utilizadas no projeto de interfaces com usuário.

Com relação à percepção de luminância, que, *grosso modo*, é a nossa capacidade de perceber padrões de tons de cinza, aprendemos que o contraste ideal para texto deve respeitar uma razão de 10:1 entre claro e escuro. O conceito de cores opostas explica por que as cores vermelho, verde, amarelo, azul, preto e branco são especiais em todas as sociedades investigadas. Isso significa que, caso seja necessário utilizar códigos de cores para categorizar informações visuais, essas cores devem ser utilizadas em primeiro lugar. Entretanto, a semântica atribuída a uma determinada cor varia amplamente. Por exemplo, “vermelho” pode significar um alerta de perigo ou boa sorte (Ware, 2003).

Ware ressalta que algumas características visuais são prontamente entendidas sem treinamento prévio. Se pedirmos para diversas pessoas ordenarem um conjunto de amostras de diferentes tons de cinza, todas utilizarão a mesma ordenação (ou a ordenação contrária), sem qualquer instrução adicional (Figura 3.9).



Figura 3.9 Ordenação de valores caracterizada por diferentes tons de cinza.

Cor, forma, movimentos simples e profundidade estereoscópica são características pré-atencionais, ou seja, características processadas antes que uma pessoa volte sua atenção a elas, antes que se tornem conscientes. Essas características são processadas simultaneamente, fazendo com que alguns elementos visuais se destaquem imediatamente de sua vizinhança. Como há limitações sobre o que pode ser percebido de modo pré-atencional, é importante que um símbolo que deva ser destacado tenha algum atributo básico distinto dos seus símbolos vizinhos (Ware, 2003). Considerando as leis de Gestalt, Ware sugere que a apresentação de dados seja elaborada de modo a tornar padrões fáceis de se perceber, para facilitar a resolução de problemas. No entanto, ele observa que, além dos mecanismos perceptivos inatos aos seres humanos, existem fatores culturais que influenciam a percepção de elementos visuais. Além disso, diferentes representações podem enviesar a interpretação das pessoas na direção de certas soluções para um problema, em detrimento a outras. Por exemplo, uma tabela, um gráfico e um mapa enfocam diferentes aspectos de uma região geográfica.

3.4 Engenharia Cognitiva

A engenharia cognitiva foi concebida por Donald Norman em 1986 como uma tentativa de aplicar conhecimentos de ciência cognitiva, psicologia cognitiva e fatores humanos ao design e construção de sistemas computacionais. Os principais objetivos de Norman eram:

- entender os princípios fundamentais da ação e desempenho humano relevantes para o desenvolvimento de princípios de design;
- elaborar sistemas que sejam agradáveis de usar e que engajem os usuários até de forma prazerosa.

Em outras palavras, Norman visava a entender as questões envolvidas no design de sistemas computacionais, mostrar como fazer melhores escolhas de design e mostrar quais são os *tradeoffs* quando uma melhoria em um aspecto leva a uma piora em outro.

Na base da engenharia cognitiva está a discrepância entre os objetivos expressos *psicologicamente* e os controles e variáveis *físicos* de uma tarefa. Uma pessoa inicia com objetivos e intenções, que são **variáveis psicológicas**, pois existem apenas na mente da pessoa e se relacionam diretamente às suas necessidades e à sua situação atual. Entretanto, a tarefa deve ser realizada em um sistema *físico*, com controles físicos a serem manipulados, resultando em mudanças nas **variáveis físicas** e no **estado do sistema**. Assim, uma pessoa precisa interpretar as variáveis físicas em termos relevantes aos objetivos psicológicos, e precisa traduzir as suas intenções psicológicas em ações físicas sobre os controles e mecanismos do sistema. Isso significa que deve haver um estágio de interpretação que relaciona as variáveis físicas e psicológicas, assim como funções que relacionem a manipulação das variáveis físicas e a mudança resultante no estado físico (Figura 3.10).

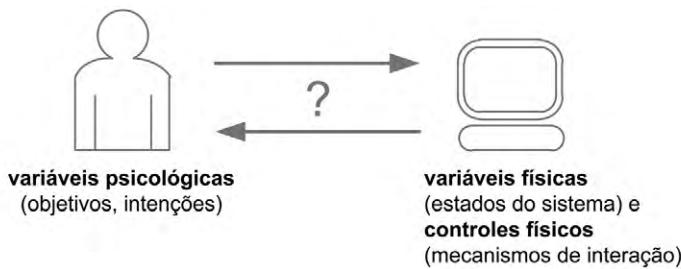


Figura 3.10 Discrepância entre o mundo psicológico e o mundo físico.

Em muitas situações, as variáveis que podem ser facilmente controladas não são aquelas pelas quais a pessoa se interessa. Por exemplo, numa torneira convencional,

as variáveis físicas que podem ser manipuladas são: fluxo de água fria e fluxo de água quente. No entanto, o usuário quer controlar duas variáveis distintas: o fluxo total de água e a temperatura da água. Nesse caso, podemos levantar as seguintes questões:

- **problemas de mapeamento** (Figura 3.11a): Qual é o controle de água quente e qual é o de água fria? De que maneira cada controle deve ser girado para aumentar ou reduzir o fluxo da água?
- **dificuldade de controle** (Figura 3.11b): Para aumentar a temperatura da água mantendo o fluxo constante, é necessário manipular simultaneamente as duas torneiras.
- **dificuldade de avaliação** (Figura 3.11c): Quando há dois bicos de torneira, às vezes se torna difícil avaliar se o resultado desejado foi alcançado.

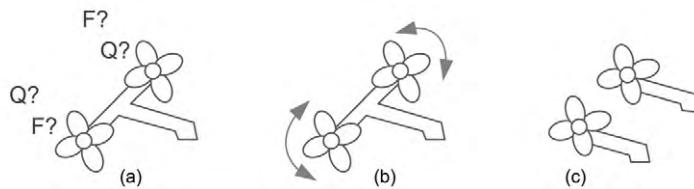


Figura 3.11 Ilustrações de diferentes torneiras para exemplificar problemas de mapeamento, dificuldade de controle e dificuldade de avaliação.

Hoje em dia, há torneiras com um controle único, em que uma dimensão de movimento controla o fluxo total da água e outra dimensão (ortogonal) controla a sua temperatura (Figura 3.12). Apesar de o mapeamento não ser óbvio — é necessário ainda aprender qual dimensão controla qual variável —, é uma solução melhor, pois as variáveis sendo manipuladas fisicamente são as mesmas variáveis de interesse.

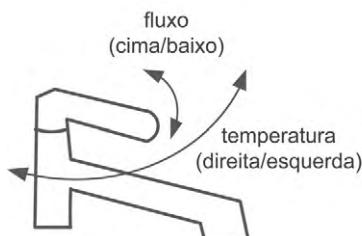


Figura 3.12 Ilustração de uma torneira com monocomando.

Exemplo 3.1 – Mapeamento, controle e avaliação em diálogos para seleção de cor.

Em sistemas computacionais, temos problemas semelhantes. Suponha que queiramos escolher uma cor de fundo para uma ilustração, e que o diálogo apresentado seja o da Figura 3.13a. Para definir uma cor de fundo, é necessário indicar os valores das componentes vermelha (*red* – R), verde

(green – G) e azul (blue – B). No entanto, a Figura 3.13a não deixa claro qual controle está associado a qual componente, o que consiste num *problema de mapeamento*. Além disso, geralmente estamos interessados na matiz da cor (*hue* – H), na sua saturação (*saturation* – S, grau de mesclagem da matiz com a cor branca, também denominado grau de pureza) e na sua luminosidade (*luminance* – L, fração da cor que vai do completamente escuro ao completamente claro). Como não podemos definir valores para essas propriedades, identificamos também um problema de dificuldade de *controle*. Finalmente, não há uma resposta visual da cor resultante, o que dificulta a *avaliação* do resultado.

A Figura 3.13b apresenta o diálogo modificado. Nele, observamos claramente o mapeamento, por proximidade, das componentes R, G e B aos controles correspondentes, assim como uma indicação visual da cor resultante, reduzindo então os problemas de mapeamento e avaliação. A dificuldade de controle das variáveis de interesse (H, S, L), no entanto, permanece.

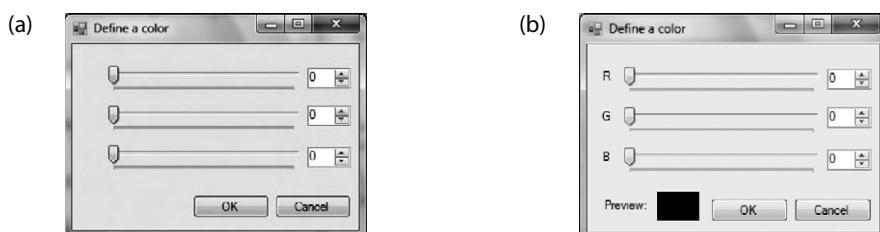


Figura 3.13 Diálogos para escolha de cores ilustrando problemas de mapeamento, controle e avaliação.

A Figura 3.14 apresenta o diálogo padrão da ferramenta Microsoft Visual Studio® para a escolha de cores. Podemos observar que esse diálogo permite a definição de cores utilizando tanto as componentes R, G e B, quanto as componentes H, S e L, reduzindo o problema de mapeamento. Além disso, é possível selecionar diretamente nos quadros de cores as componentes H (através do deslocamento horizontal no quadro maior), S (através do deslocamento vertical no quadro maior) e L (através do deslocamento na barra vertical), reduzindo assim a dificuldade de controle.

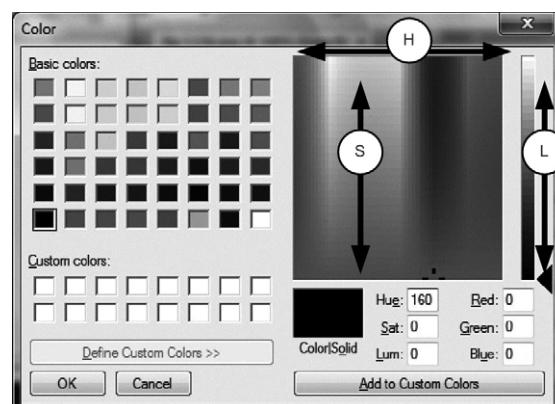


Figura 3.14 Diálogo padrão da ferramenta Microsoft Visual Studio® para a escolha de cores.

Para melhor caracterizar o papel das questões de mapeamento, controle e avaliação na interação humano-computador, Norman elaborou uma teoria da ação, descrita a seguir.

Teoria da Ação

A abordagem de projeto centrado no usuário estuda os fenômenos que ocorrem durante a interação de um usuário com um artefato cognitivo (Norman, 1991). Um **artefato cognitivo** é um dispositivo artificial projetado para manter, apresentar ou manipular informação. Um aspecto importante de um artefato cognitivo se refere ao quanto a interação é direta e envolvente. Todo artefato atua como um mediador entre as pessoas e o mundo.

Norman propôs uma teoria da ação que distingue diversos estágios de atividade ocorridos durante a interação usuário-sistema. No âmbito da engenharia cognitiva, a principal questão é a discrepância entre as variáveis psicológicas (objetivos das pessoas) e os controles e variáveis físicos (mecanismos de interação e estados do sistema). Norman representa essa discrepancia através de dois golpos que precisam ser superados ou “atravessados”: o golfo de execução e o golfo de avaliação, conforme ilustrado pela Figura 3.15. Em outras palavras, o processo de interação com um artefato pode ser visto como ciclos de ação envolvendo fases de execução e de avaliação, alternadamente.

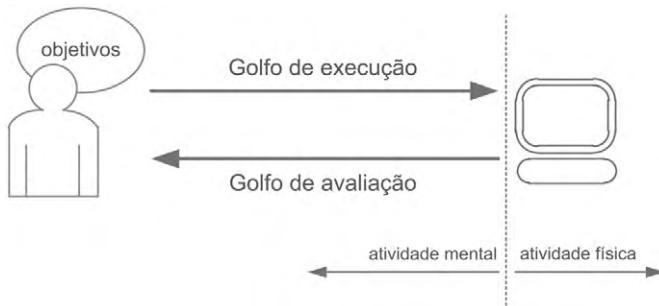


Figura 3.15 Golpos de execução e de avaliação que o usuário precisa atravessar ao interagir com um sistema físico.

Segundo Norman, o **golfo de execução** se refere à dificuldade de atuar sobre o ambiente e ao grau de sucesso com que o artefato apoia essas ações. O **golfo de avaliação**, por sua vez, se refere à dificuldade de avaliar o estado do ambiente e ao grau de sucesso com que o artefato apoia a detecção e interpretação desse estado. Tais golpos podem ser reduzidos através de um projeto adequado do artefato ou através

de treinamento e esforço mental por parte de seus usuários. A Figura 3.16 ilustra os processos físicos e cognitivos que ocorrem na travessia de cada golfo.

Segundo Norman, o ciclo se inicia na fase de execução, quando o usuário estabelece um **objetivo** de alto nível, ou seja, um estado do mundo que ele deseja alcançar através da interação com o sistema (e.g., produzir um documento esteticamente agradável). Uma vez estabelecido um objetivo, o usuário precisa formular sua **intenção**, que é a decisão de agir em direção ao objetivo, estabelecendo um subobjetivo que ele poderá alcançar diretamente através do uso do sistema. Ao formular uma intenção, o usuário escolhe uma estratégia para alcançar seu objetivo, influenciado não apenas pelo próprio objetivo, mas também pela sua experiência com aquele sistema e com outros sistemas computacionais em geral (e.g., definir uma cor específica para uma forma geométrica, em vez de selecionar uma das cores padrão).

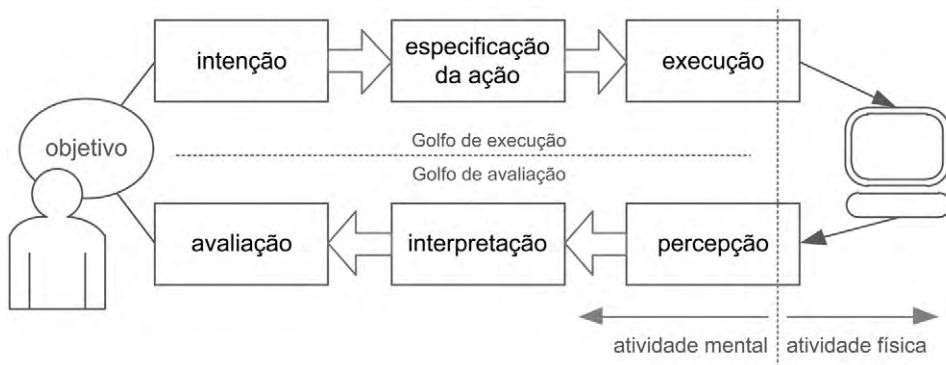


Figura 3.16 Estágios de atividade do usuário na travessia dos golfos de execução e de avaliação (adaptado de Norman, 1986, p. 42).

A partir da intenção formulada, o usuário deve **especificar as ações** a serem realizadas, ou seja, determinar quais configurações das variáveis do sistema correspondem ao estado desejado (e.g., cor verde oliva definida pelas variáveis H=58, S=99, L=77) e quais mecanismos de controle levam a esse estado (e.g., qual diálogo acionar, quais campos preencher, quais controles manipular [e como], em quais botões clicar). Especificar as ações envolve um exercício de planejamento do usuário cujo resultado é uma representação mental de quais ações devem ser executadas sobre a interface, e em que ordem.

De posse dessa especificação, o usuário deve **executar** as ações planejadas, seguindo a ordem especificada. Isso significa manipular dispositivos de entrada da interface (e.g., levar o cursor do mouse para a forma geométrica desejada, clicar com o botão da direita para acionar o menu pop-up, levar o cursor do mouse para o item

“cor de fundo” do menu pop-up, clicar com o botão esquerdo do mouse sobre esse item, levar o mouse para a caixa de texto H, clicar sobre essa caixa de texto, digitar 58, pressionar a tecla Tab para levar o foco da interação para a caixa de texto S, e assim por diante).

Vale observar que a escolha dos dispositivos de entrada a serem utilizados determina não apenas quais são as ações físicas necessárias, mas também influencia a qualidade de uso de acordo com a habilidade do usuário em manipular aquele dispositivo para executar as ações. Por exemplo, uma ação que exija do usuário pressionar duas teclas enquanto ele move o mouse mantendo simultaneamente o botão direito do mouse pressionado pode ser difícil para muitos usuários.

A cada ação executada, o sistema modifica seu estado e atualiza sua interface, apresentada através dos dispositivos de saída, para refletir o novo estado. Nesse momento o usuário começa a fase de avaliação. Ela se inicia pela **percepção**, por parte do usuário, da mudança de estado da interface (e.g., a cor da imagem de pré-visualização é alterada). Caso o sistema não realize nenhuma mudança na interface, faça uma mudança imperceptível ao usuário, ou demore muito, o que o usuário percebe é uma ausência de resposta do sistema, que influenciará negativamente sua interpretação.

Após perceber o novo estado da interface, o usuário inicia uma atividade de **interpretação**, na qual busca atribuir um significado ao novo estado do sistema tal como percebido através dos seus dispositivos de saída (e.g., a nova cor da imagem de pré-visualização reflete o novo valor de H informado pelo usuário). Essa interpretação é guiada pelos mapeamentos que o usuário tenha feito entre as variáveis (mentais) de interesse e as variáveis (físicas) do sistema. Uma ausência de resposta geralmente é interpretada como “nada aconteceu”, como se as ações não tivessem sido de fato executadas.

Finalmente, o ciclo se fecha com a **avaliação** do novo estado do sistema, tal como interpretado, comparando-o com o estado desejado (correspondente à intenção formulada e ao objetivo almejado). O resultado da avaliação determina se as ações realizadas contribuíram para o usuário se aproximar do seu objetivo ou não. Caso o resultado da avaliação determine que o estado interpretado corresponde ao estado desejado, o usuário atingiu seu objetivo. Caso contrário, o usuário precisaria percorrer novamente o ciclo, retificando uma ou mais das atividades realizadas, a fim de atingir seu objetivo original. É possível que, após percorrer o ciclo uma ou mais vezes, obtendo avaliações malsucedidas, o usuário considere que não há como atingir o objetivo original e passe a considerar um outro objetivo, ou até mesmo desista de atingir seus objetivos naquele momento.

Nem sempre a travessia dos golfos é iniciada pelo golfo de execução. Um usuário cuja atividade envolva monitorar alguma operação fica observando a saída do sistema até perceber que houve uma mudança. Quando alguma mudança ocorrer, o usuário deve diagnosticá-la e tomar as providências necessárias, percorrendo os golfos de execução e avaliação. Nesse caso, a avaliação inclui não apenas verificar se as ações desejadas foram executadas adequadamente e as intenções satisfeitas, mas se o diagnóstico original foi adequado.

Exemplo 3.2 – Travessia dos golfos de execução e avaliação

Considerando o diálogo apresentado na Figura 3.14, a travessia dos golfos de execução e de avaliação para o exemplo de mudança de cor de fundo de um objeto selecionado pode ser ilustrada pelos passos a seguir:

- estabelecimento do objetivo: mudar a cor de fundo do retângulo selecionado.
- formulação da intenção: definir uma cor verde oliva com os valores H=58, S=99, L=77.
- especificação das ações:
 1. acionar o item de menu Formatar > Cor de fundo;
 2. informar o valor 58 para a componente H;
 3. informar o valor 99 para a componente S;
 4. informar o valor 77 para a componente L;
 5. confirmar a cor definida pelos valores informados.
- execução (ação nº 1): aciono o item de menu Formatar > Cor de fundo utilizando o mouse.
- percepção: observei que apareceu uma janela de diálogo.
- interpretação: o título da janela de diálogo é “Selecionar cor”, e há controles de definição de cada componente de cor individual.
- avaliação: me aproximei do meu objetivo. A especificação de ações parece correta e, portanto, posso prosseguir para o próximo passo..
- execução (ação nº 2): informo o valor 58 para a componente H, digitando esse valor na caixa de texto correspondente.
- percepção: o valor na caixa de texto correspondente à componente H mudou, assim como a cor da imagem de pré-visualização.
- interpretação: o novo valor corresponde ao valor digitado.
- avaliação: me aproximei do meu objetivo. A especificação de ações parece correta e, portanto, posso prosseguir para o próximo passo.
- execução (ação nº 3): informo o valor 99 para a componente S, digitando esse valor na caixa de texto correspondente.
- percepção: o valor na caixa de texto correspondente à componente S mudou, assim como a cor da imagem de pré-visualização.
- interpretação: o novo valor corresponde ao valor digitado.
- avaliação: me aproximei do meu objetivo. A especificação de ações parece correta e, portanto, posso prosseguir para o próximo passo.
- execução (ação nº 4): informo o valor 77 para a componente L, digitando esse valor na caixa de texto correspondente.
- percepção: o valor na caixa de texto correspondente à componente L mudou, assim como a cor da imagem de pré-visualização.

- interpretação: o novo valor corresponde ao valor digitado e a cor da imagem de pré-visualização corresponde à cor desejada.
- avaliação: me aproximei do meu objetivo. A especificação de ações parece correta e, portanto, posso prosseguir para o próximo passo.
- execução (ação nº 5): confirmo a cor definida pelos valores informados, clicando em OK.
- percepção: a janela de diálogo foi ocultada; a cor do retângulo mudou.
- interpretação: a nova cor do retângulo é verde oliva.
- avaliação: alcancei meu objetivo.

O designer do sistema deve tentar abreviar os golfos de execução e de avaliação que precisam ser atravessados pelo usuário a fim de reduzir os problemas que ocorrem durante a interação. O mapeamento adequado das variáveis de interesse envolvidas na tarefa do usuário para variáveis físicas do sistema contribui para a travessia de ambos os golfos. Mecanismos e controles de interação (elementos de interface) para manipular dados de entrada e a representação desses dados contribuem para abreviar o golfo de execução. De modo semelhante, a representação dos dados de saída e as mensagens de resposta do sistema (*feedback*) contribuem para abreviar o golfo de avaliação. Outra maneira de auxiliar o usuário a atravessar os golfos é fornecer-lhe treinamento e oportunidades de adquirir experiência no uso de um sistema. Entretanto, cabe ao designer tentar reduzir essa necessidade de treinamento tanto quanto possível.

A engenharia cognitiva considera três modelos, dois mentais e um físico (Figura 3.17): o modelo de design, a imagem do sistema, e o modelo do usuário.



Figura 3.17 Modelos considerados pela engenharia cognitiva (adaptado de Norman, 1986, p. 46).

O **modelo de design** é o modelo conceitual do sistema tal como concebido pelo designer. Ele descreve a lógica de funcionamento do sistema que será construído. O modelo de design deve se basear em tarefas, requisitos, capacidades e experiência do usuário. Deve considerar também as capacidades e limitações dos mecanismos de processamento de informação do usuário, em particular limitações nos recursos de processamento e de memória de curto prazo.

A **imagem do sistema** corresponde ao sistema executável, isto é, o modelo físico construído com base no modelo conceitual de design, e a partir do qual os usuários elaboram seus modelos conceituais (modelo do usuário).

O **modelo do usuário** é o modelo conceitual construído por ele durante sua interação com o sistema,² resultando assim da sua interpretação da imagem do sistema. Tudo o que o designer construir na imagem do sistema pode auxiliar ou prejudicar essa interpretação, tal como: elementos de interface (*widgets*) para entrada e saída de dados; documentação, instruções, ajuda on-line e mensagens de erro.

O objetivo do designer é que o usuário seja capaz de elaborar um modelo conceitual compatível com o modelo de design através da sua interação com a imagem do sistema. Para isso, o designer deverá produzir uma imagem de sistema explícita, inteligível e consistente.

Podemos observar que, como o foco da engenharia cognitiva está nos processos psicológicos do usuário, o trabalho do designer se torna tão mais desafiador quanto mais heterogênea for a população de usuários-alvo em termos de suas características, necessidades e atividades. Por exemplo, um usuário com pouca experiência se beneficia de uma interface de assistente (*wizard*), ao passo que um usuário especializado requer uma interface mais eficiente. Cabe ao designer tentar elaborar uma interface que concilie essas diferentes necessidades.

3.5 Abordagens Etnometodológicas

Suchman (1987) foi pioneira ao trazer para a pesquisa em IHC a visão da antropologia etnográfica de que o significado e o valor da ação humana são situados, ou seja, têm uma relação essencial com as suas circunstâncias concretas particulares e com suas interações dinâmicas com o mundo material e social. Ao fazer isso, ela deslocou o foco do *usuário individual* para o *contexto social* do uso do computador e desafiou a visão de ações intencionais dominante na época: a de ação planejada.

Para a visão de **ação planejada**, a ação humana pode ser completamente caracterizada em termos de seus objetivos, intenções e planos. Para entender como as pessoas agem, bastaria entender como elas seguem um plano predefinido. Em geral, os trabalhos de análise do desempenho dos usuários que enfocam a estrutura de suas tarefas (i.e., a partir da decomposição de objetivos em tarefas e operações) se encaixam nessa visão. Esta favorece o pensamento analítico abstrato, no qual a organiza-

2 A expressão “modelo de usuário” possui vários significados. Diversos trabalhos utilizam essa expressão como significando uma representação do perfil e características do usuário, utilizadas em tempo de design ou embutidas no sistema para realizar adaptações durante a interação.

ção, o significado e o valor da ação humana estão nos planos subjacentes, definidos *a priori*.

Um **plano** é uma sequência de ações projetada para alcançar algum objetivo. Dado um objetivo e uma situação inicial, uma pessoa constrói um plano e então realiza as ações definidas nesse plano. As ações são descritas em detalhes pelas suas precondições ou pré-requisitos (aquilo que precisa ser verdadeiro para que a ação seja possível) e consequências ou efeitos (o que precisa ser verdadeiro após a ação ser executada). Nessa visão, as condições para a execução de uma ação são definidas *a priori*, e a atividade humana é considerada uma forma de resolução de problemas, na qual o ator deve encontrar um caminho de algum estado inicial para algum estado final desejado, dadas algumas condições ao longo do caminho (cf. Newell e Simon, 1972). Qualquer condição imprevista, ou seja, que não esteja de acordo com o plano, requer um replanejamento.

Já na visão de **ação situada**, defendida por Suchman, a cada instante é feita uma avaliação das circunstâncias concretas particulares e do valor das ações mediante a essas contingências. É à luz dessas contingências que as pessoas constroem e se engajam nas suas ações sociais e em suas interações umas com as outras. Essas contingências não podem ser completamente previstas com antecedência nem se mantêm estáveis ao longo do tempo. Portanto, não é possível projetar em detalhes como um comportamento vai se desdobrar antes que os participantes se engajem nas suas interações sociais (Button, 2003).

Em outras palavras, um plano não pode determinar o curso de ações de uma pessoa, como propõe a visão de ação planejada. Todo curso de ação depende das circunstâncias materiais e sociais em que ocorre. Em vez de tentar abstrair a ação das suas circunstâncias e representá-la como um plano racional, a abordagem de ações situadas consiste em estudar como as pessoas usam suas circunstâncias para atingir seus objetivos.

A etnometodologia foi proposta por Harold Garfinkel na década de 1960. Ela considera que o significado, o valor e a forma de se compreenderem as ações não se encontram de forma isolada, nem no que é estritamente observável do comportamento, nem num estado mental prévio do ator, mas sim numa relação construída de forma contingencial entre o *comportamento observável*, as *circunstâncias* em que ele ocorre, as *intenções* dos atores presentes nas situações observadas e os *relacionamentos* entre esses atores. Em outras palavras, a etnometodologia examina processos interacionais (de comunicação entre as pessoas) e circunstanciais (Garfinkel, 1967).

Suchman busca identificar os recursos (cognitivos e de interação) que possibilitam a comunicação humana bem-sucedida. Para ela, o fato de as pessoas consegu-

rem se entender (i.e., alcançar a inteligibilidade mútua) em suas interações cotidianas é sempre o produto de trabalho colaborativo situado, às vezes sem esforço aparente, outras a partir de trabalho evidente. Ela destaca que a comunicação face a face inclui naturalmente recursos para detectar e remediar problemas no entendimento.

A comunicação humana segue a máxima geral de que as falas devem ser projetadas especificamente para os seus receptores e para a ocasião em que são emitidas. Ela deve ser sensível às circunstâncias e aos recursos locais para a remediação de problemas no entendimento que inevitavelmente surgem. Nesse sentido, a comunicação humana difere de um manual instrucional impresso, no qual há uma desassociação entre a ocasião de sua produção e a ocasião do seu uso. O manual impresso não é voltado a cada receptor específico. Já um sistema computacional interativo tem potencial de se afastar do design instrucional do manual impresso e de se aproximar do instrutor humano, que utiliza recursos possibilitados pela interação presencial (Suchman, 1987).

A etnometodologia explorou a produção situada da ordem social através de dois domínios de interesse: análise da conversação e estudos etnometodológicos do trabalho. As próximas subseções examinam conceitos de análise da conversação e indicam como aspectos da comunicação humana podem ser utilizados para descrever e analisar a comunicação usuário-sistema.

3.5.1 Análise da Conversação

A análise da conversação descreve a forma como uma conversa é organizada pelos participantes a cada momento, durante o desdobramento de cada turno de fala (Schegloff, 1972; Schegloff e Sacks, 1973).

A **conversação** é caracterizada por mecanismos projetados para apoiar o controle local sobre o desenrolar de tópicos ou atividades, maximizar a acomodação de circunstâncias imprevistas que venham a ocorrer e identificar e remediar eventuais problemas na comunicação. Dessa forma, a análise da conversação também enfatiza a natureza situada das trocas conversacionais.

O controle local está relacionado à distribuição de turnos de fala e à direção do assunto abordado. Isso significa que durante a conversação os participantes decidem quem fala sobre o que e quando, construindo colaborativamente a conversa. Sacks, Schegloff e Jefferson (1974) delinearam um conjunto de convenções ou “regras” sobre a troca de turnos que descrevem práticas comuns observadas por analistas da conversação. No curso de uma conversa, quando uma fala puder ser considerada concluída, ocorre um dos seguintes eventos:

- o falante atual seleciona o próximo falante (e.g., direcionando uma pergunta ou outra fala a um ouvinte particular);
- um outro participante se autosseleciona para começar a falar;
- o falante atual continua.

O falante atual deve deixar claro para os ouvintes em que ponto está seu turno: se ele está no meio do turno, ou se o turno se encerrou. No entanto, o falante não define o turno unilateralmente. A conclusão de um turno representa tanto uma inclinação do ouvinte em responder quanto a disposição do falante em ceder o turno. As fronteiras de um turno são mutáveis, e a estrutura da conversação é elaborada localmente pelos falantes e ouvintes. Em outras palavras, o turno é essencialmente determinado pela interação entre os participantes ao longo da conversação. Por exemplo, o silêncio numa fala pode ser considerado uma simples pausa no meio de um turno, no qual o falante quer prosseguir falando, ou uma conclusão do turno, quando um ouvinte crê que pode tomar o turno para si.

Segundo Schegloff (1972), geralmente uma conversa coerente é aquela em que cada coisa dita pode ser tida como relevante, considerando o que veio antes. Isso significa que a relevância de um turno é condicionada pelo turno que imediatamente o precedeu. Duas falas numa relação de relevância condicional constituem um **par adjacente** (Schegloff e Sacks, 1973). Uma fala que seja considerada como a primeira parte de um par adjacente estabelece uma expectativa com relação ao que deve vir em seguida, e orienta a forma como a fala seguinte é ouvida. Tanto a presença como a ausência de uma segunda parte esperada são significativas (Exemplo 3.3).

Exemplo 3.3 – Pares adjacentes em uma conversa

Considere um diálogo entre um vendedor (V) e um comprador (C), numa livraria, como a seguir:

V: Bom dia! Como posso ajudá-lo?
C: Estou procurando o novo livro da série “Harry Potter”.

A fala do vendedor pode ser considerada uma primeira parte de um par adjacente, que cria a expectativa de que o comprador responderá com alguma informação sobre um produto de seu interesse. Como o comprador responde com uma fala do tipo esperado, a conversa é tida como coerente e bem-sucedida. Já no diálogo a seguir, isso não acontece, e a conversa é tida como incoerente (a incoerência está marcada por um asterisco):

C: Estou procurando o novo livro da série “Harry Potter”.
* V: Semana passada eu fui à praia e o mar estava ótimo!

Em interfaces com usuário, quando o usuário aciona um item de menu Salvar como..., ele espera que o sistema lhe pergunte com que nome e onde deve salvar o arquivo. Caso algo diferente ocorra, há uma ruptura na comunicação.

Quando o ouvinte não entende uma fala, ele pode iniciar uma **sequência colateral**, uma troca de falas em que o ouvinte busca esclarecimentos sobre o que foi dito anteriormente. Essa solicitação explícita de esclarecimentos também cria expectativas sobre o que vem a seguir na conversa, como ocorre com os pares adjacentes.

3.5.2 Comunicação Usuário-Sistema

Segundo Suchman (1987), a descrição de artefatos computacionais como interativos é apoiada pelas suas propriedades *reativas*, *linguísticas* e internamente *opacas*. Ela propõe ainda que essas propriedades nos levam a enxergar esses artefatos como interativos e a atribuir explicações intencionais ao seu comportamento. Na prática, isso sugere que, como um ator humano, o computador seja capaz de se expressar, ou expressar a intenção por trás de suas ações, para o usuário.

Suchman ressalta que a forma de controlar as máquinas computacionais e o comportamento resultante são cada vez mais *linguísticos*, em vez de mecânicos. A operação da máquina se torna menos uma questão de pressionar botões ou puxar alavancas com algum resultado físico, e mais uma questão de especificar operações e avaliar seus efeitos através do uso de linguagem. Isso contribui para a tendência dos designers em descreverem o que ocorre entre as pessoas e máquinas utilizando termos emprestados da descrição da interação humana (e.g., diálogo, conversação). Esses termos trazem um conjunto de intuições sobre propriedades comuns à comunicação humana e ao uso de artefatos de base computacional. Ela observa ainda que, num sistema computacional, os momentos de troca de turnos são predeterminados. Ao estabelecer uma relação determinada entre ações detectáveis dos usuários e respostas da máquina, o designer controla unilateralmente a interação, mas de forma condicional às ações do usuário.

Para comparar as visões da interação do usuário e do sistema, Suchman (1987) utiliza um framework analítico simples:

- Do ponto de vista do usuário, separa os eventos em ações dos usuários que não estão disponíveis ao sistema (por exemplo, conversas entre os usuários) e que estão disponíveis ao sistema (por exemplo, ações sobre os elementos da interface do sistema).
- Do ponto de vista do sistema, separa os efeitos disponíveis ao usuário (por exemplo, instruções apresentadas na tela) e o *design rationale*, ou seja, presunções e planos do designer que foram embutidos no sistema.

Suchman observa que o sucesso da interação assume que o usuário interpreta as instruções e as respostas do sistema da forma como o designer pretendia. Para transmitir a intenção do design ao usuário, e fazê-lo interativamente, o designer se apoia ta-

citamente em certas convenções da conversação humana. O problema prático com o qual o designer de um sistema interativo precisa lidar é como assegurar que o sistema responda de forma apropriada às ações do usuário. Devemos levar em consideração que a interação é um processo altamente contingente, no qual toda ação envolve não apenas a intenção do ator, mas também o trabalho interpretativo do seu interlocutor. Este, por sua vez, deve determinar o significado e o valor da ação para então responder adequadamente, tomando uma nova ação. De modo geral, o designer e o usuário compartilham a expectativa de que a relevância de cada fala está condicionada à fala mais recente e que, dada uma ação por um interlocutor que pede uma resposta, a próxima ação do outro *será* uma resposta. Essa expectativa não assegura que qualquer próxima ação de fato será uma resposta à última, mas significa que, sempre que possível, o usuário vai buscar uma interpretação da próxima ação como se fosse essa resposta.

Já a expectativa do usuário é de que toda resposta do sistema indique, implícita ou explicitamente, uma avaliação da última ação que o usuário tomou e uma recomendação sobre o que ele pode ou deve fazer em seguida. Mais especificamente, toda vez que age sobre a interface, o usuário tem as seguintes expectativas com relação à resposta do sistema (Suchman, 1987):

- Se o sistema responde com uma nova instrução, a ação anterior do usuário foi confirmada pelo sistema. Ao realizarmos um procedimento passo a passo, temos uma expectativa geral de que completar uma ação permite progredir para uma nova instrução e uma próxima ação. Esse tipo de resposta ocorre, por exemplo, após submeter um formulário de busca, quando o sistema responde com alguns documentos encontrados e instruções sobre como acessá-los.
- Se o sistema não responde, a ação anterior do usuário de algum modo estava incompleta, e deve haver mais alguma ação para o usuário tomar de forma a completá-la. A falta de resposta do sistema traz informações sobre a última ação do usuário, indicando que o turno de fato não mudou. Por exemplo, caso o usuário preencha um formulário de busca mas não ative a busca de fato, o sistema fica aguardando uma próxima ação do usuário.
- Se a resposta do sistema for repetir a instrução, a repetição implica que (1) a ação prévia do usuário deve ser repetida (i.e., que o procedimento é iterativo); (2) houve erro na ação prévia e o sistema retorna ao estado anterior à instrução, desfazendo a ação (isso não ocorre na interação humana, e os usuários frequentemente não reconhecem isso); ou (3) a ação do usuário falhou em satisfazer a intenção da instrução do sistema e precisa ser remediada. Por exemplo, quando o usuário submete um formulário de busca e o

sistema lhe reapresenta o mesmo formulário, em geral isso indica que houve uma falha na tentativa anterior (e.g., o usuário não definiu nenhum termo antes de acionar a busca), ou seja, que o usuário não seguiu um curso de ação esperado e deve tentar remediar o problema naquele ponto.

Assim, a interação entre pessoas e máquinas requer essencialmente o mesmo trabalho de interpretação que caracteriza a interação entre pessoas, mas com recursos fundamentalmente diferentes disponíveis aos participantes. Em particular, as pessoas fazem uso de uma gama rica de recursos linguísticos, não verbais e inferenciais ao tentar compreender ações e eventos, ao tornar suas próprias ações razoáveis, e ao gerenciar os problemas de entendimento que inevitavelmente surgem. Muitos sistemas computacionais, por outro lado, se apoiam numa gama fixa de entradas sensoriais, mapeadas a um conjunto predefinido de estados internos e respostas. O resultado é uma assimetria que limita substancialmente o escopo da interação entre pessoas e sistemas computacionais. Essa assimetria traz ao menos três desafios para o design de sistemas computacionais interativos (Suchman, 1987):

- como reduzir a assimetria, aumentando o acesso do sistema às ações e circunstâncias do usuário;
- como tornar claros ao usuário os limites do acesso do sistema a esses recursos de interação básicos;
- como encontrar maneiras de compensar a falta de acesso do sistema à situação do usuário com alternativas computacionais disponíveis.

Assim como a comunicação humana, a comunicação usuário-sistema não é livre de problemas. Concepções erradas do usuário podem levá-lo a encontrar evidências para um erro em suas ações onde não há nenhum, ou podem levá-lo a um erro nas suas ações que não possa ser detectado pelo sistema.

Suchman argumenta que a interação usuário-sistema é geralmente limitada às intenções dos designers e à sua capacidade de prever e restringir as ações do usuário. Cabe ao designer entender essas limitações e tentar estender, através de um design cuidadoso, a gama de comportamentos úteis do sistema.

3.5.3 Estudos Etnometodológicos de IHC

Estudos fundamentados em etnometodologia têm sido aplicados em IHC de diversas maneiras (Button, 2003):

- para analisar o impacto que um sistema teve no trabalho realizado no ambiente em que o sistema é introduzido;

- para analisar princípios e métodos organizacionais subjacentes a um domínio de trabalho;
- para analisar os impactos de um sistema sobre esses métodos;
- para criticar o design do sistema quando entra em conflito com esses métodos.

Numa abordagem de estudo de campo fundamentada em etnometodologia, a atividade humana é observada enquanto se desdobra nas circunstâncias reais em que ocorre. O observador-participante atua como uma “sombra” de um indivíduo particular e testemunha muitas circunstâncias em que ele está envolvido durante um dia de trabalho. A relação entre as interpretações da ação e as circunstâncias da ação devem ser investigadas. O ponto de partida para o estudo é a suposição de que não temos uma descrição *a priori* da estrutura da ação situada. Não queremos *pressupor* quais são as condições relevantes ou sua relação com a estrutura da ação. Precisamos capturar tanto quanto possível do fenômeno e pressupor tão pouco quanto possível (Suchman, 1987). Essa atitude se contrapõe à análise de exemplos artificiais, observações ou relatos de entrevistas, que se fiam em circunstâncias imaginadas ou lembradas (Button, 2003).

Segundo Button, os estudos sobre a forma como as pessoas têm de trabalhar com um sistema e frequentemente contorná-lo têm sido utilizados para *criticar* as metodologias de projeto que apoiam diversos designs de sistemas baseados em entendimentos abstratos e formais do trabalho. Sistemas assim projetados podem encontrar dificuldades quando são implantados em ambientes de trabalho reais por causa da natureza situada da organização do trabalho, que costuma ser um fenômeno muito mais flexível, envolvendo práticas *ad hoc* e operações de contingência na sua realização, em vez de regras ou normas prescritivas.

Em IHC, estudos do trabalho também vêm sendo utilizados para *avaliar* designs tecnológicos particulares. Ao conduzir estudos sobre o uso real de um sistema no local de trabalho, torna-se possível coletar dados detalhados sobre a aplicação de tecnologia que podem ser utilizados na sua avaliação e no redesign subsequente.

A expressão *ação reportável (accountable action)* chama atenção para o fato de que as ações sociais não são apenas realizadas, mas são feitas de forma que possam ser reconhecidas como tal, isto é, de forma que seja possível fazer um relato delas. Utilizando a ideia de que a ação humana é reportável (feita para que possa ser reconhecida), Dourish e Button (1998) sugerem que abstrações podem ser desenvolvidas de modo que atuem como uma forma de visualizar os mecanismos operantes quando alguma tarefa está ocorrendo. Por exemplo, uma operação de cópia no computador é feita para tornar o que está sendo feito reconhecível. Dessa forma, o sistema con-

segue fornecer um relato do que está sendo feito à medida que o faz, o que permite aos usuários determinarem melhor qualquer ação (remediadora) que seja necessária (Button, 2003).

3.6 Teoria da Atividade

A teoria da atividade teve origem no início do século XX como uma psicologia materialista dialética elaborada por Vygotsky e seus alunos. Vygotsky argumentava contra separações artificiais entre mente e comportamento, e entre mente e sociedade. Ele advogava pela unidade da percepção, fala e ação. Além disso, enfatizava a centralidade dos dispositivos mediadores, como linguagens e outros símbolos ou ferramentas, no desenvolvimento da mente e do pensamento. A ênfase no significado através da ação, a conexão entre o individual e o social, e o papel das ferramentas mediadoras são o cerne em torno do qual a teoria da atividade se desenvolveu (Gay e Hembrooke, 2004).

Segundo Vygotsky (1978), a atividade humana possui três características fundamentais:

- é *dirigida a um objeto* material ou ideal;
- é *mediada* por artefatos;
- é *socialmente constituída* dentro de uma cultura.

A teoria da atividade rejeita o ser humano isolado como uma unidade de análise adequada, e insiste na **mediação cultural e técnica** da atividade humana. A unidade de análise inclui os artefatos técnicos e a organização cultural, que tanto determinam o ser humano como são criados por ele. Essa teoria entende o comportamento humano como ancorado em práticas coletivas compartilhadas. Não considera um ser humano “genérico”, e endereça mais do que o conhecimento, as habilidades e o julgamento individual. Permite analisar a adequação de uma ferramenta para uma prática, bem como estudar de que maneira a introdução de um artefato particular modifica a prática e como a prática pode modificar o uso do artefato. O fato de a atividade humana ser mediada por **artefatos socialmente construídos** (e.g., ferramentas, linguagens e representações) significa que, na sua relação imediata com seu ambiente, uma pessoa se estende com artefatos externos a ela (Bertelsen e Bødker, 2003).

Segundo Leontiev (1978), a atividade humana pode ser analisada numa hierarquia de atividade, ação e operação. A **atividade** é realizada através de ações conscientes direcionadas a objetivos do sujeito. As **ações** são realizadas através de **operações** inconscientes, disparadas pela estrutura da atividade e as condições do ambiente. A

atividade busca satisfazer uma necessidade do sujeito através de um objeto material ou ideal (Bertelsen e Bødker, 2003).

É importante observar que os níveis de atividade não são fixos (Figura 3.18). Uma ação se torna uma operação quando a orientação para um ato é transformada da interação consciente com objetos externos em um plano de ação interno e inconsciente, num processo de aprendizado. Dizemos então que ocorreu uma **internalização**. As operações podem se desenvolver natural, histórica ou culturalmente. Podem resultar de padrões inatos à espécie, do uso apropriado de ferramentas e da relação com outras pessoas, dentre outros fatores. De forma semelhante, em situações problemáticas, uma operação pode se tornar uma ação. A **externalização** ocorre em situações que precisam de reparo, que não podem ser resolvidas apenas internamente (e.g., contas com números muito grandes) ou quando duas ou mais pessoas trabalham juntas.



Figura 3.18 Relacionamento dinâmico entre níveis de atividade (figura adaptada de Bertelsen e Bødker, 2003).

A teoria da atividade utiliza como unidade de análise básica e irredutível a **atividade motivada**. A atividade pode ser entendida como uma estrutura sistêmica. É o engajamento de um sujeito (ou coletivo) direcionado a um objeto. Esse engajamento é socialmente mediado pela comunidade em que a atividade se constitui. A noção de atividade humana de Leontiev pode ser ilustrada através de triângulos aninhados (Figura 3.19).



Figura 3.19 Teoria de atividade humana de Leontiev (Engeström, 1987).

Na figura, identificamos o Sujeito e Objeto da atividade, mediados pela Comunidade e pelo Instrumento, que pode ser tanto um instrumento técnico (ferramenta) como um instrumento psicológico (signo). Da mediação Sujeito–Comunidade emergem

Regras e rituais, ao passo que da mediação Comunidade–Objeto surge a Divisão de trabalho. E, quando há diversas atividades interligadas, temos redes de atividade.

Perguntas do tipo *por que*, *o que* e *como* ajudam a entender melhor a atividade (Bertelsen e Bødker, 2003):

- Perguntas “*por quê?*” revelam o motivo da atividade, o significado social e pessoal da atividade e a sua relação com motivos e necessidades.
- Perguntas “*o quê?*” revelam possíveis objetivos, objetivos críticos e subobjetivos particularmente relevantes.
- Perguntas “*como?*” revelam operações, formas concretas de executar uma ação de acordo com condições específicas em torno do objetivo da atividade.

Por exemplo, numa atividade relacionada ao uso de um dispositivo de reprodução de música, o motivo poderia ser identificado como “relaxar”, um objetivo poderia ser “ouvir músicas preferidas”, e a forma concreta de realizar uma ação em direção ao objetivo poderia ser a sequência “examinar listas de músicas” e “ativar lista de músicas denominada ‘favoritas’”.

3.6.1 Princípios da Teoria da Atividade

Os princípios da teoria da atividade comumente citados são: mediação, orientação a objetos e perturbação (*disturbance*).

Conforme pode ser visto na Figura 3.19, a relação de um indivíduo com um objetivo é **mediada** por instrumentos que são utilizados para atingir o objetivo, pela comunidade que participa da atividade e pela divisão de trabalho que existe nessa comunidade. Kaptelinin (1996) endereça especificamente os efeitos mediadores da atividade computacional na consciência, no aprendizado e no desenvolvimento humano. Para ele, tecnologias computacionais possibilitam e transformam atividades através de ações, objetivos e relações sociais de agentes individuais. Segundo Bertelsen e Bødker (2003), tomando a atividade motivada como a unidade básica de análise, precisamos estudar o que acontece quando usuários se concentram no seu trabalho ou qualquer outro ato intencional enquanto utilizam o artefato computacional. Com base na estrutura hierárquica da atividade, isso significa que a situação tende a ser rotineira quando o objeto da ação *consciente* do usuário é o mesmo objeto do trabalho e as operações *inconscientes* do usuário são dirigidas ao artefato mediador. Nesse caso, o artefato computacional se torna uma ferramenta transparente. Por exemplo, quando uma pessoa utiliza frequentemente um editor de texto como ferramenta ou

instrumento, o objeto da sua ação consciente se torna o documento que está sendo elaborado, e não mais a aplicação de editor de texto em si.

O próximo passo é olhar para como os próprios objetos (coisas ou pessoas) que são o foco desse trabalho estão visíveis dentro ou fora do computador. Esses objetos “reais” de interesse da nossa atividade (também denominados objetos do domínio) constituem a base para a análise futura. Gay e Hembrooke (2004) enfatizam duas visões sobre mediação: a bidirecionalidade dos efeitos (das percepções, motivações, cultura e ações que moldam a ferramenta e que são moldadas por ela) e a necessidade de estudos longitudinais sustentados para revelar como essas relações mediadoras se desenvolvem e se modificam ao longo do tempo.

Na teoria da atividade, a **orientação a objetos** se refere ao engajamento das pessoas com objetos e objetivos (Kaptelinin, 1996). Recebem status de objeto os fenômenos físicos, sociais e culturais, incluindo fenômenos não materiais como expectativas e afinidades. O propósito, a intenção ou a motivação para agir sobre um objeto ou trabalhar em direção a um objetivo são os fundamentos do sistema de atividade, e atuar sobre um objeto é o espaço de orientação da ação. Gay e Hembrooke (2004) identificam dois aspectos importantes do conceito de orientação a objetos: (1) objetos psicológicos e sociais podem ter o mesmo nível de importância que objetos físicos; (2) artefatos (instrumentos) podem ser transpostos a objetos e vice-versa. Por exemplo, um artefato ou ferramenta no framework de um sistema de atividade principal pode ao mesmo tempo ser um objeto num outro sistema. Em contrapartida, quando ocorre um problema na utilização de um artefato computacional, ele geralmente se torna o objeto de uma nova atividade de resolução de problemas.

A **perturbação** se refere ao fato de que as relações entre os diversos elementos do modelo da teoria da atividade são flexíveis e estão sempre mudando. À medida que perturbações se tornam evidentes dentro de um sistema de atividade ou entre sistemas de atividade, os participantes podem começar a endereçar as questões subjacentes e modificar suas situações, atividades, ou a si próprios. Para Gay e Hembrooke (2004), as perturbações podem ser informativas no processo de design como sinais para descobrir por que a perturbação se materializou, por que ela não existia até um certo ponto no tempo, quais efeitos ela pode ter e como pode ser resolvida.

3.6.2 Contradição e Aprendizado

Sistemas de atividade são fundamentalmente marcados por contradições. Engeström (1987) classifica as **contradições** em um sistema de atividade e entre sistemas de atividades como as forças motrizes do aprendizado e desenvolvimento humano. Contradições podem resultar das relações entre o uso e o valor obtido (e.g., a tensão

entre a melhor solução possível e o que pode ser projetado com o tempo e recursos disponíveis), entre a atividade em estudo e as atividades vizinhas, entre a atividade considerada e a atividade desejada (potencial). Essas contradições podem ser geradas deliberadamente, através da elaboração de exemplos e visões em um processo de desenvolvimento de uma comunidade de prática para lançá-la a um novo estágio.

O **aprendizado** não se trata apenas de como o indivíduo adapta o artefato ou se adapta a ele. Também trata de como a prática coletiva se desenvolve. Projetar um artefato não significa apenas projetar uma “coisa” ou dispositivo que pode ser utilizado pelas pessoas como artefatos num tipo específico de atividade. À medida que o uso de artefatos faz parte da atividade social, projetamos novas condições para essa atividade coletiva, tal como uma nova divisão do trabalho e novas formas de coordenação e comunicação. No uso real, artefatos frequentemente mediam diversas atividades de trabalho, e as contradições e os conflitos resultantes dessa miríade de usos são essenciais para a análise e o projeto do artefato no âmbito da teoria da atividade (Bertelsen e Bødker, 2003).

A teoria da atividade entende que os seres humanos não estão apenas selecionando ações dentre as oferecidas pelo ambiente. Segundo a teoria, as pessoas estão ativa e constantemente recriando seu próprio ambiente, como resultado de contradições, instabilidades e do surgimento de novas necessidades.

3.6.3 Teoria da Atividade em IHC

Em IHC, a teoria da atividade tem se concentrado principalmente em quatro pontos (Bertelsen e Bødker, 2003):

- análise e design de uma prática de trabalho específica, considerando as qualificações, o ambiente de trabalho, a divisão de trabalho e assim por diante;
- análise e design com foco no uso real e na complexidade da atividade multiusuário e, em particular, na noção essencial do artefato como mediador da atividade humana;
- o desenvolvimento da experiência e do uso em geral;
- a participação ativa do usuário no design, e com foco no uso como parte do design.

Segundo Bertelsen e Bødker (2003), a teoria da atividade permite estudar diversos níveis de atividade combinados: desde a atividade de uso estrito de um artefato computacional até o contexto mais amplo de uso e design. Também permite modificar a escala e estudar as conexões em múltiplos níveis de atividades em que artefatos com-

putacionais são utilizados e projetados, sem estabelecer uma hierarquia permanente na análise.

Segundo Gay e Hembrooke (2004), o potencial explicativo da teoria da atividade está na atenção que ela dá a múltiplas dimensões de engajamento humano com o mundo e no framework que fornece para configurar essas dimensões e processos numa “atividade” coerente. Em IHC, o papel mediador exercido por artefatos e ferramentas culturais e seu poder transformador são essenciais para o entendimento desses processos de engajamento durante o uso.

Para projetar uma aplicação computacional situada no uso, é necessário (Bødker, 1996): enquadrar historicamente o trabalho e a aplicação computacional; situar a aplicação computacional numa rede de atividades em que ela é utilizada; caracterizar o uso da ferramenta; considerar o apoio necessário às diversas atividades que ocorrem em torno da aplicação computacional; identificar os objetos que são trabalhados na ou através da aplicação computacional; considerar a rede de atividades e contradições dentro de uma atividade e entre atividades. Para cada atividade específica, Bødker propõe perguntar:

- Qual é o objetivo da atividade e das ações para o usuário?
- Qual objeto é focado pelo usuário? Onde esse objeto se encontra (dentro, fora ou através da aplicação computacional)?
- Qual é o instrumento? Onde ele se encontra (dentro, fora ou através da aplicação computacional)?
- Quando há mais usuários cooperando, devemos perguntar: os objetivos, objetos e instrumentos estão alinhados ou conflitantes (entre indivíduos e entre o grupo e seus membros)?
- Para cada mudança de foco, devemos perguntar: de qual foco/objeto para qual outro? A mudança foi uma ruptura ou uma mudança deliberada? O que causou a mudança? Ela se originou dentro ou fora da aplicação?

Focando os principais constituintes de uma atividade considerada como central, Korpela (1999) levanta uma série de questões:

- *resultado*: que serviços ou produtos são produzidos?
- *objeto e processo*: com que materiais brutos ou pré-requisitos uma atividade se inicia? Como são produzidos os serviços e produtos com as entradas que se tem?
- *instrumentos*: que tipos de ferramenta física, conhecimento e habilidade são necessários para esse trabalho?

- *sujeitos*: quem são? Que tipos diferentes de pessoas são necessários para produzir esses serviços e produtos?
- *relações e meios sociais*: quando se trabalha para produzir esses serviços e produtos, que tipos de regras, divisão de trabalho e comunicação se aplicam entre as pessoas?

Já no caso de uma rede de atividades, eles incluem as seguintes questões:

- *resultado*: quem precisa dos serviços ou produtos? Por que eles são necessários — para produzir serviços ou produtos para outros?
- *objeto*: de quem se obtém o “material bruto”? Como se produz aquilo que é necessário?
- *instrumentos*: de quem se obtém as ferramentas e conhecimentos necessários? Como eles são produzidos?
- *sujeitos*: de onde eles vêm? Quem educa as pessoas envolvidas nas atividades? Como isso ocorre?
- *relações e meios sociais*: quem estabelece as regras para as atividades? Como elas são geradas?

3.7 Cognição Distribuída

A teoria da cognição distribuída, assim como qualquer outra teoria cognitiva, busca entender a *organização de sistemas cognitivos*. Diferente das teorias cognitivas tradicionais, no entanto, a cognição distribuída amplia a semântica de cognitivo para abranger as interações entre pessoas, recursos e materiais no ambiente (Hollan *et al.*, 2000).

Segundo Perry (2003), a cognição distribuída surgiu de uma necessidade de entender o trabalho que extrapola o indivíduo, entender como o processamento de informação e a resolução de problemas incorporam o uso de ferramentas e envolvem outras pessoas. Em outras palavras, os designers precisam entender como os grupos coordenam o comportamento de seus membros, colaboram e resolvem problemas.

Em cognição distribuída, a unidade de análise para a cognição é o **processo cognitivo**, delimitado pelos relacionamentos funcionais dos elementos que dele participam, independentemente de onde estejam. Os mecanismos que participam dos processos cognitivos extrapolam a manipulação de símbolos na mente de atores individuais. Eles incluem o mundo material, que fornece oportunidades para reorganizar o sistema cognitivo distribuído de modo a fazer uso de um conjunto diferente de processos internos e externos (Hollan *et al.*, 2000).

A cognição distribuída considera que um sistema pode se configurar dinamicamente para coordenar subsistemas que realizam diversas funções. Os processos cognitivos podem ser distribuídos entre os membros de um grupo social, envolver a coordenação entre estruturas internas (símbolos e modelos mentais) e externas (materiais ou do ambiente) e estar distribuídos no tempo, de forma que os produtos de acontecimentos passados possam transformar a natureza de acontecimentos posteriores. Na cognição distribuída há três crenças básicas (Hollan *et al.*, 2000):

- a organização social é uma forma de **arquitetura cognitiva**. A organização social, juntamente com a estrutura fornecida pelo contexto da atividade, determina em grande parte a forma como a informação flui através de um grupo. Padrões dessas trajetórias de informação refletem uma arquitetura cognitiva subjacente. Essa perspectiva traz três questões fundamentais sobre interações sociais: (1) como os processos cognitivos geralmente associados a uma mente individual são implementados em um grupo de indivíduos; (2) como as propriedades cognitivas de grupos diferem das propriedades cognitivas das pessoas que atuam nesses grupos; e (3) como as propriedades cognitivas de mentes individuais são afetadas pela sua participação em atividades em grupo.
- a cognição é **materializada**. As relações entre processos internos e externos envolvem a coordenação ao longo do tempo entre recursos internos (memória, atenção e função) e externos (objetos, artefatos e materiais disponíveis no ambiente que nos cerca).
- o estudo da cognição não pode ser separado do **estudo da cultura**. Por um lado, a cultura *emerge* da atividade de agentes humanos em seus contextos históricos à medida que estruturas mentais, materiais e sociais interagem. Por outro, na forma de uma história de artefatos materiais e práticas sociais, a cultura *molda* os processos cognitivos, atuando como recurso para o aprendizado, a resolução de problemas e o raciocínio.

As pessoas estabelecem e coordenam diferentes tipos de estrutura em seu ambiente, externalizando informações (i.e., utilizando e criando representações externas) de modo a apoiar o seu trabalho individual e coletivo. Representações externas reduzem a carga cognitiva das pessoas, incluindo memória (por exemplo, utilizando lembretes) e processamento (por exemplo, traçando um gráfico à medida que dados se tornam disponíveis, buscando tendências). A cognição distribuída destaca a importância de se investigar como projetar representações que facilitem seu uso flexível, assim como representações mais ativas que ajudem os usuários a enxergarem o que é mais relevante e decidirem o que fazer a cada momento.

Sob uma perspectiva teórica, para resolver problemas, um sistema inteligente percorre um “espaço de problema”, passando por diversos estados transitórios em direção a um objetivo. Esses estados de problema são de natureza representacional, e uma análise deve focar esses estados e as transformações entre eles. A resolução cooperativa de problemas envolve uma unidade maior que o indivíduo, que se torna um componente dos recursos de resolução de problemas do grupo (Perry, 2003).

Em termos práticos, Perry afirma que uma análise de cognição distribuída envolve (Figura 3.20):

- descrever o contexto da atividade, os objetivos do sistema funcional e seus recursos disponíveis;
- identificar as entradas e saídas do sistema funcional;
- identificar as representações e os processos disponíveis;
- identificar as atividades de transformação que ocorrem durante a resolução de problemas para atingir o objetivo do sistema funcional.

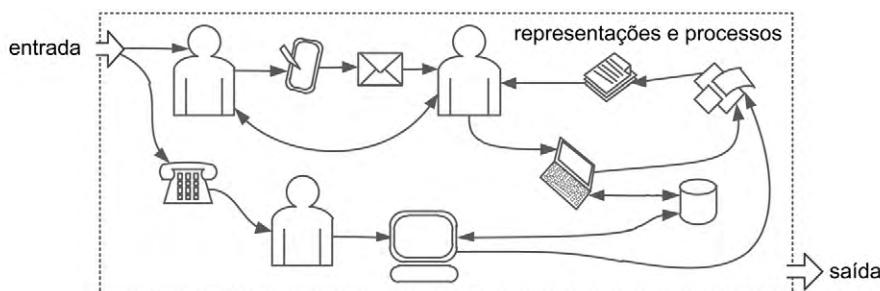


Figura 3.20 Elementos de análise da cognição distribuída (figura adaptada de Perry, 2003).

3.8 Engenharia Semiótica

A engenharia semiótica é uma teoria de IHC centrada na comunicação. Ela caracteriza a interação humano-computador como um caso particular de *comunicação humana mediada por sistemas computacionais* (de Souza, 2005a). Seu foco de investigação é a comunicação entre designers, usuários e sistemas. Os processos de comunicação investigados são realizados em dois níveis distintos: a comunicação direta usuário-sistema e a metacomunicação (i.e., comunicação sobre uma comunicação) do designer para o usuário mediada pelo sistema, através da sua interface (Figura 3.21).



Figura 3.21 Metacomunicação designer–usuário e comunicação usuário–sistema.

A engenharia semiótica caracteriza aplicações computacionais como **artefatos de metacomunicação**, ou seja, artefatos que comunicam uma mensagem do designer para os usuários sobre a comunicação usuário–sistema, sobre como eles podem e devem utilizar o sistema, por que e com que efeitos (de Souza, 2005a; de Souza, 2005b; de Souza e Leitão, 2009). O conteúdo dessa *mensagem de metacomunicação*, ou *metamensagem*, pode ser parafraseado no seguinte modelo genérico (de Souza, 2005a, p. 25):

Este é o meu entendimento, como designer, de quem você, usuário, é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão.

Em tempo de design, o designer estuda os usuários, suas atividades e seu ambiente e, a partir desse estudo, concebe sua visão sobre como contemplar o que os usuários desejam ou necessitam e sobre como os usuários, suas atividades e seu ambiente podem ou devem mudar com a introdução do sistema sendo projetado. Ele então expressa essa sua visão na forma de tecnologia computacional, elaborando a metamensagem e codificando-a em palavras, gráficos, comportamento, ajuda on-line e explicações. Com isso, ele visa atingir a sua *intenção comunicativa*: que os usuários interpretem adequadamente, gostem e se beneficiem do produto. Como os designers não estarão fisicamente presentes durante a interação para que os usuários possam falar com ele, dizemos que a metamensagem é única e unidirecional (do inglês, *one-shot message*). Tudo o que o preposto do designer precisa comunicar deve ser planejado em tempo de design e implementado na forma de um programa computacional nos estágios seguintes de desenvolvimento (de Souza, 2005a, p. 24).

Em tempo de interação, os usuários decodificam e interpretam gradualmente a metamensagem do designer, buscando atribuir sentido aos significados nela codificados e respondendo de forma apropriada. Assim, designers, sistemas e usuários

são *interlocutores* igualmente envolvidos nesse processo comunicativo que constitui a interação humano-computador (de Souza, 2005a; 2005b).

Para que a metacomunicação seja bem-sucedida, o designer “deve se tornar um interlocutor legítimo na interação humano–computador (...), deve falar *através* do sistema, que se torna então o **preposto do designer**” (de Souza, 2005b). O preposto do designer (*designer's deputy*) é responsável por comunicar ao usuário, durante a interação, a metamensagem do designer, que “contém todos os significados e possibilita todas as manipulações de significados que os designers escolheram incorporar no sistema a fim de que ele fizesse aquilo para o que foi projetado” (de Souza, 2005a, p. 24).

Conforme visto na Seção 2.2.3, **comunicabilidade** é um conceito de qualidade dos sistemas computacionais interativos que comunicam de forma eficiente e efetiva aos usuários as intenções comunicativas do designer, a lógica e os princípios de interação subjacentes (Prates et al., 2000a; de Souza, 2005a). A comunicabilidade pode ser definida tecnicamente como a “capacidade do preposto do designer de alcançar a metacomunicação completa, comunicando aos usuários a essência da mensagem original do designer” (de Souza, 2005a, p. 114), permitindo, portanto, que os usuários gerem significados compatíveis com aqueles codificados pelo designer.

Como apresentado no Capítulo 2, um sistema com alta comunicabilidade auxilia os usuários a interpretarem e atribuírem sentido à metamensagem do designer, sentindo esse compatível com o que o designer pretendia comunicar e, portanto, codificou na interface. “A competência comunicativa (ou discursiva) do preposto do designer deve ser analisada em termos de *o que* ele pode comunicar e *como* o comunica” (de Souza, 2005a, p. 90). Para avaliar a comunicabilidade de um sistema computacional interativo, a engenharia semiótica oferece o método de inspeção semiótica e o método de avaliação de comunicabilidade (Prates et al., 2000a; de Souza, 2005a; de Souza et al., 2006; Prates e Barbosa, 2007; de Souza e Leitão, 2009). Esses métodos são apresentados nas Seções 10.1.3 e 10.2.2, respectivamente.

A ontologia da engenharia semiótica compreende (de Souza, 2005a, p. 95):

- *processos de significação*, que envolvem signos e semiose;
- *processos de comunicação*, que envolvem intenção, conteúdo e expressão nos dois níveis de comunicação investigados (a comunicação direta usuário–sistema e a metacomunicação designer–usuário mediada pelo sistema, através da sua interface);
- os *interlocutores* envolvidos nos processos de significação e comunicação: designers, sistemas (prepostos dos designers em tempo de interação) e usuários;

- o espaço de design de IHC, baseado no modelo do espaço de comunicação de Jakobson (1960), que caracteriza a comunicação em termos de emissores, receptores, contextos, códigos, canais e mensagens.

A seguir examinamos esses conceitos.

3.8.1 Semiótica: Signo, Significação, Comunicação e Semiose

A Semiótica estuda signos, processos de significação e processos de comunicação (Eco, 1976). Peirce define signo como “uma coisa que serve para veicular conhecimento de uma outra coisa (o *objeto* do signo), que ele *representa*. A ideia na mente que o signo motiva, e que é um signo mental do mesmo objeto, é chamada de *interpretante* do signo” (Peirce, 1992–1998, vol.2, p. 13). Em outras palavras, o signo é algo que representa alguma coisa para alguém. São signos: “toda imagem, diagrama, apontar de dedo, piscar de olhos, nó no lenço de alguém, memória, sonho, desejo, conceito, indicação, *token*, sintoma, letra, número, palavra, sentença, capítulo, livro, biblioteca, (...)” (Peirce, 1992–1998, vol.2, p. 326).

Nem toda representação é signo. Para ser um **signo**, uma representação deve possuir uma relação triádica com seu objeto e com o seu interpretante, conforme ilustrado pela Figura 3.22a. A fruta maçã (objeto) pode ser representada por uma ilustração (representação) e evocar na mente de alguém (intérprete) a ideia de maçã (interpretante). Nesse caso, dizemos que a representação (ilustração) é um signo de maçã (fruta). E o “interpretante é a **significação** do conceito” veiculado pelo signo (Peirce, 1992–1998 , vol.2, p. 497).

Segundo Eco, sempre que há convenções sociais ou culturais que nos permitem interpretar signos, temos um **sistema de significação** e, portanto, um **código** (Eco, 1976, p. 4). Segundo de Souza (2005a), em um **processo de significação**, “conteúdos são associados sistematicamente a expressões” (p. 98), “estabelecendo sistemas de signos com base em convenções sociais e culturais adotadas pelas pessoas que interpretam e produzem tais signos” (p. 26).

Já em um **processo de comunicação**, produtores de signos utilizam sistemas de significação para escolher formas de representar (*expressão*) seus significados pretendidos (*conteúdo*) de modo a alcançar uma variedade de objetivos (*intenção*). Para isso, os produtores de signos podem utilizar signos conhecidos (culturalmente convencionados) de formas convencionais, utilizar signos conhecidos de formas criativas ou até mesmo inventar signos (de Souza, 2005a, p. 98).

Um signo de interface é então codificado pelo designer visando comunicar sua intenção de design aos usuários. Por exemplo, ao representar a operação de “salvar o documento” por um botão com o rótulo Salvar e um ícone de um disquete, o designer

espera que os usuários interpretem esse signo como “Clicando nesse botão, eu consigo salvar o documento” (Figura 3.22b).

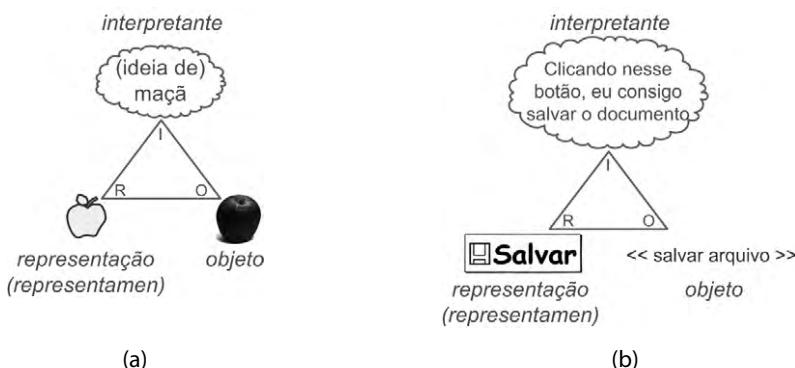


Figura 3.22 Exemplos de signos ilustrando a relação triádica do signo com seu objeto e seu interpretante: (a) signo que representa um objeto físico e (b) signo de interface.

Para Peirce, o interpretante de um signo é, ele próprio, outro signo. Sendo assim, é possível de ser, ele próprio, interpretado, gerando outro interpretante, e assim sucessivamente. Esse processo interpretativo que nos leva a associar cadeias de significados (interpretantes) a um signo é denominado **semiose** (Peirce, 1992–1998; Eco, 1976). Trata-se de um processo potencialmente ilimitado. Segundo Santaella (2000), “o signo, por sua própria constituição, está fadado a germinar, crescer, desenvolver-se num interpretante (outro signo) que se desenvolverá em outro e assim indefinidamente. Evidencia-se aí a natureza inevitavelmente incompleta de qualquer signo” (p. 29). Esse processo interpretativo humano em constante evolução, indefinidamente longo e imprevisível é denominado **semiose ilimitada**.

Na prática, a semiose é interrompida quando o intérprete fica satisfeito com o interpretante gerado (i.e., o significado temporariamente atribuído ao signo) ou não tem mais tempo ou outro recurso necessário para continuar gerando novos significados. É importante observar que essa interrupção deve ser considerada temporária, pois a qualquer momento podem surgir novos fatos ou circunstâncias que levem o intérprete a retomar o processo, prosseguindo no mesmo caminho interpretativo ou iniciando um novo caminho distinto do anterior. A qualquer momento, um significado pode ser revisto e corrigido, por exemplo, devido ao surgimento de evidências conflitantes ou contrárias.

A Figura 3.23 ilustra um processo de semiose associado a um signo de interface.

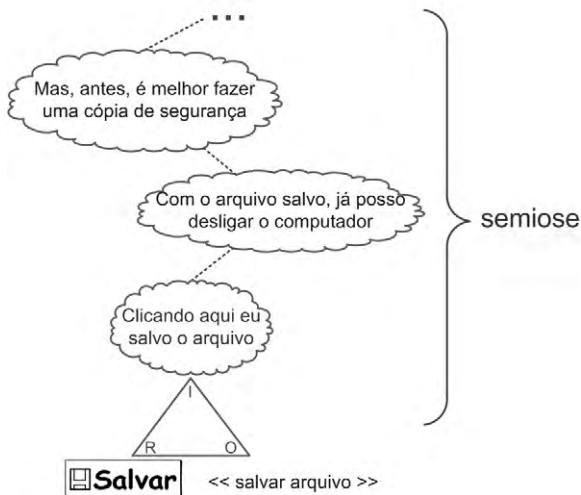


Figura 3.23 Um processo de semióse associado a um signo de interface.

A natureza potencialmente ilimitada da semióse humana indica que não podemos falar em “o” significado de um signo, mas sim em “um” significado de um signo. Mas, se não podemos capturar de modo definitivo o significado de um signo atribuído por uma pessoa, como conseguimos nos comunicar com sucesso? Voltando nossa atenção para a comunicação usuário-sistema, “o que o usuário (como emissor) *quer dizer* com uma expressão para o preposto e o que o usuário (como receptor) entende que uma expressão do preposto *quer dizer* são contingentes à situação comunicativa em que a expressão surge” (de Souza, 2005a, p. 100). Se cada signo significa alguma coisa para os designers, possivelmente alguma outra coisa para os usuários, e esses significados podem mudar a qualquer momento, como conseguimos nos comunicar (interagir) com sistemas computacionais interativos?

Todo processo de semióse é fortemente influenciado pelo conhecimento prévio, hábitos e experiência pessoal do intérprete, pela cultura em que ele se insere e pelo contexto em que o signo é interpretado (de Souza, 2005a). Segundo Danesi e Perron (1999) a cultura auxilia a comunicação humana, funcionando como um “contêiner de signos e significados” que “convergem de formas previsíveis em padrões de representação que indivíduos e grupos podem utilizar para produzir ou trocar mensagens” (p. 67). Daí a importância de os designers — produtores de signos de interface — estudarem os usuários — consumidores desses signos —, suas atividades e experiências, seus valores e expectativas, sua cultura e os ambientes em que eles vão utilizar o sistema computacional interativo sendo projetado. Desse modo, os designers se capacitam a expressar adequadamente sua intenção comunicativa nos signos de in-

terface elaborados e codificados no sistema, tendo em vista o que ele aprendeu sobre as características e a cultura dos usuários.

3.8.2 Sistema Computacional Interativo como Artefato Intelectual

Um **artefato** é “algo criado pelo ser humano, geralmente para um propósito prático; (...) algo característico ou resultante de uma instituição, período, tendência ou indivíduo particular”.³ Trata-se de um produto artificial resultante da engenhosidade humana através de um processo de design, que geralmente envolve atividades de análise, síntese e avaliação (veja o Capítulo 4).

De Souza (2005a) define um sistema computacional interativo como um **artefato intelectual**. Ele resulta de atividades de análise, codificando um entendimento ou interpretação particular do seu produtor sobre uma situação-problema, e síntese, codificando um conjunto particular de soluções para a situação-problema analisada. A natureza *intelectual* desse artefato se deve principalmente ao fato de que (1) “a *codificação* da situação-problema e das soluções correspondentes é fundamentalmente *linguística* (i.e., baseada em um sistema de símbolos — verbais, visuais, sonoros e outros — que podem ser interpretados por regras semânticas consistentes)” e (2) “o propósito final do artefato só pode ser completamente alcançado por seus usuários se eles conseguem formulá-lo dentro do sistema linguístico no qual o artefato é codificado (i.e., os usuários devem ser capazes de entender e utilizar um *sistema de codificação linguística* particular para explorar e realizar as soluções possibilitadas pelo artefato)” (de Souza, 2005a, p. 10).

Em outras palavras, designers, preposto e usuários de um sistema computacional interativo devem utilizar uma mesma **linguagem**, um sistema de signos composto de vocabulário, gramática e regras semânticas, a fim de se comunicarem através do sistema. Essa **linguagem de interface** resulta de decisões do designer sobre as estratégias de atuação e de resolução de problemas dos usuários que pretende apoiar com o sistema por ele projetado (de Souza, 2005a).

Como visto, o processo de semiose ilimitada não pode ser completamente *determinado*; pode ser apenas *motivado* pela escolha dos sistemas de significação utilizados. Em contrapartida, sistemas computacionais geram significados através processamentos simbólicos determinados causalmente por algoritmos e codificados pelos seus desenvolvedores utilizando alguma linguagem de programação. Ao contrário dos significados humanos, os significados computacionais podem ser previstos e completamente inspecionados (de Souza, 2005a, p. 98; p. 250).

Em interação humano-computador, o que isso significa é que “o preposto do designer só pode reproduzir um segmento limitado (e governado por regras) da semio-

3 <http://www.merriam-webster.com/dictionary/artifact>.

se do designer. Portanto, no processo de comunicação usuário-sistema, o preposto do designer conta para o usuário apenas uma versão processável por máquina do que o designer realmente queria dizer. E diz a mesma coisa repetidas vezes, independentemente de como a semiose do usuário esteja evoluindo em torno de signos produzidos nessa comunicação” (de Souza, 2005a, p. 98). Da mesma maneira, o usuário só pode dizer aquilo que o preposto estiver preparado para “ouvir” e interpretar.

A limitação da “semiose” computacional pode introduzir rupturas na comunicação usuário-sistema que não existiriam em circunstâncias análogas de comunicação humana (de Souza e Leitão, 2009, p. 21). Torna-se então primordial “avaliar como *um* significado do usuário ou do designer, contingente à situação em que é investigado, se compara a *o* significado implementado do signo de interface correspondente” (de Souza, 2005a, p. 87), a fim de projetar signos que motivem no usuário uma semiose produtiva do usuário, ou seja, compatível com a intenção comunicativa do designer.

3.8.3 Espaço de Design de IHC

A engenharia semiótica é uma teoria *explicativa* de IHC. Ela não deve ser utilizada como uma teoria preditiva, mas sim para explicar fenômenos de IHC observáveis. Além disso, deve “fornecer meios para formular problemas e questões de IHC e para elaborar as soluções e respostas correspondentes” (de Souza, 2005a, p. 104).

Para organizar o espaço de design de IHC, a engenharia semiótica utiliza o modelo de espaço de comunicação proposto por Jakobson (1960), estruturado em termos de: contexto, emissor, receptor, mensagem, código e canal. “Um *emissor* transmite uma *mensagem* a um *receptor* através de um *canal*. A mensagem é expressa em um *código* e se refere a um *contexto*. Na comunicação, os interlocutores exercem alternadamente os papéis de emissor e receptor” (de Souza, 2005a, p. 65; Figura 3.24).

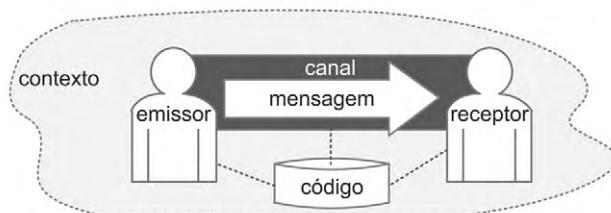


Figura 3.24 Modelo do espaço de comunicação de Jakobson (1960), adotado pela engenharia semiótica para estruturar o espaço de design de IHC (de Souza, 2005a).

A comunicação é guiada por uma *intenção*, os efeitos que o emissor quer provocar ao transmitir o *conteúdo* da sua mensagem ao receptor. Para que a comunicação seja bem-sucedida, o emissor deve escolher cuidadosamente uma *expressão* para o conte-

údo que deseja comunicar, utilizando um código que o receptor seja capaz de interpretar e, no caso de IHC, que o sistema computacional seja capaz de processar.

Sendo assim, ao projetar sua metamensagem, o designer de IHC precisa tomar decisões sobre cada elemento do espaço de design de IHC, respondendo as seguintes perguntas (de Souza, 2005a, p. 87):

- **quem é o emissor (designer)?** Que aspectos das limitações, motivações, crenças e preferências do designer devem ser comunicados ao usuário para o benefício da metacomunicação;
- **quem é o receptor (usuários)?** Que aspectos das limitações, motivações, crenças e preferências do usuário, tal como interpretado pelo designer, devem ser comunicados aos usuários reais para que eles assumam seu papel como interlocutores do sistema;
- **qual é o contexto da comunicação?** Que elementos do contexto de interação — psicológico, sociocultural, tecnológico, físico etc. — devem ser processados pelo sistema, e como;
- **qual é o código da comunicação?** Que códigos computáveis podem ou devem ser utilizados para apoiar a metacomunicação eficiente, ou seja, qual deve ser a *linguagem de interface*;
- **qual é o canal?** Quais canais de comunicação estão disponíveis para a metacomunicação designer-usuário, e como eles podem ou devem ser utilizados;
- **qual é a mensagem?** O que o designer quer contar aos usuários, e com que efeito, ou seja, qual é a intenção comunicativa do designer.

Sobre o Código Utilizado na Metacomunicação

Com relação ao código da comunicação, “mesmo que as interfaces de sistemas compartilhem diversos padrões interativos, cada sistema possui uma **linguagem interativa única**, cuja semântica é determinada pelo modelo semântico único do sistema (de Souza e Leitão, 2009, p. 9; de Souza, 2005a). É necessário que os designers construam essa linguagem, definindo os códigos expressivos que os usuários deverão utilizar para se comunicarem com o sistema.

A engenharia semiótica classifica os signos utilizados em uma linguagem de interface em três tipos (de Souza *et al.*, 2006; de Souza e Leitão, 2009, p. 19):

- **signos estáticos:** signos que expressam o estado do sistema e cujo significado é interpretado independentemente de relações causais e temporais da interface. Eles podem ser interpretados a partir de um retrato da interface num momento do tempo. São exemplos de signos estáticos: o *layout* geral

geral e a disposição de elementos em uma tela, os itens de menu, os botões de uma barra de ferramentas, os campos e botões de um formulário e o conteúdo expresso em um texto, lista, tabela, árvore ou outra forma de visualização que não inclua animações.

- **signos dinâmicos:** signos que expressam o comportamento do sistema, envolvendo aspectos temporais e causais da interface. Estão vinculados à própria interação e devem ser interpretados fazendo referência a ela. São exemplos de signos dinâmicos: a associação causal entre a escolha de um item de menu e a exibição do diálogo, a possibilidade de arrastar itens de uma área da tela para outra, o deslocamento do foco da entrada de dados durante o preenchimento de um formulário, a ativação e desativação de um botão de comando e o surgimento de uma dica sobre um elemento de interface ao ser sobreposto pelo cursor do mouse.
- **signos metalingüísticos:** signos principalmente verbais e que se referem a outros signos de interface, sejam eles estáticos, dinâmicos ou mesmo metalingüísticos. Em geral, ocorrem na forma de mensagens de ajuda e de erro, alertas, diálogos de esclarecimento, dicas e assemelhados. Através de signos metalingüísticos, os designers podem explicitamente comunicar aos usuários os significados codificados no sistema e como eles podem ser utilizados.

A Seção 10.1.3 descreve o papel desses tipos de signo no método de inspeção semiótica.

Sobre o Papel do Designer na Engenharia Semiótica

Ao incluir o designer no modelo do espaço de design, a engenharia semiótica ressalta a importância do seu papel ativo na interação. O designer deve se posicionar como um interlocutor engajado em ajudar os usuários a entenderem a metamensagem, a sua visão sobre o que os usuários querem ou precisam fazer utilizando o sistema e por que essa visão faz sentido para ele, bem como questões de design relevantes para esse entendimento. Para isso, o designer deve refletir sobre os tipos de estratégias comunicativas que ele pode utilizar, os tipos de signos que ele pode projetar na linguagem de interface e as consequências que as limitações dos significados computacionais trazem para a interação (de Souza e Leitão, 2009).

Devido à natureza evolutiva e imprevisível da semiose humana, o designer não tem como determinar de que maneira os usuários interpretarão os signos da interface, nem mesmo garantir que eles utilizarão o código correto para essa interpretação. Além disso, de Souza (2005a, p. 21; 2005b) ressalta que os usos mais sofisticados de sistemas computacionais interativos com frequência não são autoevidentes. Por

exemplo, não há uma correspondência óbvia de estilos de formatação em um editor de texto com algum meio de formatar textos no “mundo real”, e a mera existência de elementos de interface relacionados ao uso desses estilos não é suficiente para o usuário interpretá-los adequadamente. Portanto, é necessário que o designer tenha como objetivo *introduzir* aos usuários um sistema computacional interativo, e não apenas *produzi-lo* (de Souza, 2005a, p. 22; 2005b). Essa mudança de foco visa considerar em primeiro lugar os aspectos *estratégicos* da tecnologia (i.e., que valor a tecnologia agrupa às suas atividades e como tirar melhor partido dela, escolhendo qual dos caminhos de interação disponíveis seguir em determinada situação), deixando para um segundo momento seus aspectos *operacionais* (i.e., como utilizá-la, incluindo os tipos de ações que os usuários devem realizar e os recursos de que precisam para isso). Vale ainda observar que boa parte das diretrizes de design em IHC se referem a aspectos *operacionais* da tecnologia.

Como visto na Seção 2.2.3, a engenharia semiótica considera a comunicabilidade de um sistema interativo fundamental para que os usuários façam um bom uso da tecnologia. O conhecimento dos aspectos estratégicos da tecnologia se tornam ainda mais importantes em circunstâncias imprevistas ou infrequentes, em que não basta saber *como* usar o sistema, mas também é necessário saber *por que se pode ou deve* utilizar o sistema *de uma determinada maneira em uma determinada situação*.

A engenharia semiótica pode ser articulada com a perspectiva de **reflexão em ação** estabelecida por Schön para caracterizar a prática profissional (Schön, 1983; de Souza, 2005a; Seção 4.2). Schön vê cada problema de design como um **problema único**, cuja definição requer um estudo cuidadoso de situações mal definidas, caracterizadas pela sua “complexidade, incerteza, instabilidade, unicidade e conflito de valores” (Schön, 1983, p. 39). De Souza sumariza a *epistemologia da prática* de Schön indicando que “pesquisadores e designers sempre participam de cinco atividades principais: aproximação do problema, formulação do problema, geração de soluções candidatas, avaliação de soluções candidatas e reorganização do conhecimento” (de Souza, 2005a, p. 106). Para apoiar essas atividades centradas no conhecimento, a engenharia semiótica propôs um conjunto de *ferramentas epistêmicas* para avaliação e design de IHC.

Uma **ferramenta epistêmica** não gera diretamente uma resposta ou solução para o problema. Em vez disso, apoia o designer na exploração do espaço e da natureza do problema, bem como das restrições sobre soluções candidatas (de Souza, 2005a, p. 33). Dentre as ferramentas epistêmicas propostas pela engenharia semiótica, o Capítulo 7 apresenta modelos de design da interação e de ajuda, e o Capítulo

10 apresenta os métodos de inspeção semiótica e de avaliação de comunicabilidade (Seções 10.1.3 e 10.2.2, respectivamente).

3.8.4 Comparação com o Design Centrado no Usuário

Como visto na Seção 3.4, o objetivo do designer na engenharia cognitiva é que o usuário seja capaz de, através da interação com a imagem do sistema, construir um modelo conceitual compatível com o modelo de design. Nas abordagens de cunho cognitivo, o *aprendizado* dos usuários é um importante objeto de investigação. No entanto, o foco na usabilidade da imagem do sistema promove principalmente a consideração dos aspectos *operacionais* da interação usuário-sistema, em detrimento a seus aspectos *estratégicos*. Dessa maneira, pode alienar os usuários e impedir o alcance do alfabetismo computacional pleno (de Souza e Leitão, 2009, p. 8).

Diferentemente do design centrado no usuário adotado pela engenharia cognitiva, na engenharia semiótica os designers não tentam apenas construir a imagem do sistema, ou seja, *produzir* tecnologia, mas também *introduzir* a tecnologia criada (de Souza, 2005a; 2005b). Seu principal objeto de investigação é a *comunicação*, e não o aprendizado. Como visto, na engenharia semiótica, o designer (na forma de seu preposto) é um interlocutor presente no momento da interação, e a sua comunicação com o usuário sobre as estratégias de comunicação e atuação codificadas no sistema permite que o usuário utilize o sistema de maneira mais adequada à situação em que se encontra. Além disso, no caso de ocorrer uma situação inesperada, o usuário com conhecimento estratégico sobre o sistema terá melhores condições de interagir com ele de forma criativa e produtiva. A fim de contribuir com o alfabetismo computacional, a engenharia semiótica tem como foco a comunicação não apenas dos aspectos operacionais e táticos da metacomunicação, mas também dos seus aspectos estratégicos.

É importante deixar claro que o fato de a engenharia semiótica privilegiar a comunicação da visão e intenção de design não significa que os usuários sejam menos importantes que os designers. Todo esforço de design de sistemas computacionais interativos visa melhorar a vida das pessoas que os utilizam, satisfazendo suas necessidades e expectativas (de Souza e Leitão, 2009, p. 16).

Atividades

1. *Aplicação da lei de Hick-Hyman.* Calcule o tempo médio necessário para achar um livro em uma lista de 25, 50 e 100 livros numa página. Assuma que os itens estão em ordem alfabética e que não há necessidade de rolagem.

2. *Aplicação da lei de Fitts.* Considerando estritamente o desempenho humano, que cuidados podem ser tomados para que os itens de um menu pop-up vertical sejam igualmente acessíveis, sem modificar a orientação vertical do menu?
3. *O número mágico 7 ± 2 .* Explique por que o estudo de Miller não corrobora a seguinte afirmação: “O número de opções apresentadas para o usuário em cada tela deve ser 7 ± 2 ”.
4. *Princípios de Gestalt.* Escolha algumas telas complexas de uma aplicação que você utilize com frequência e verifique se os princípios de Gestalt estão sendo bem utilizados. Caso contrário, reprojete a tela para fazer melhor uso desses princípios.
5. *Planos e ações situadas.* Examine manuais de instruções de diferentes dispositivos de base computacional (e.g., aparelhos de telefone celular, máquinas fotográficas digitais) e sistemas interativos (e.g., editores de texto, planilhas eletrônicas). Você consegue imaginar uma situação em que seguir as instruções do fabricante não leva ao resultado desejado? Por que você acredita que isso aconteça? Utilizando as tecnologias de informação e comunicação disponíveis atualmente, como você reduziria ou resolveria esse problema?
6. *Mapeamento entre variáveis psicológicas e físicas.* Examine alguns aparelhos eletrodomésticos na sua residência (fogão, torradeira, geladeira, máquina de lavar roupa, televisor, telefone) e analise o mapeamento entre as variáveis psicológicas e físicas envolvidas no uso desses aparelhos. Faça o mesmo para o editor de texto e para o editor de planilhas da sua preferência.
7. *Golfos de execução e avaliação.* Escolha um modelo de aparelho de telefone celular e descreva os passos dos golfos de execução e avaliação percorridos por um usuário com o objetivo de inserir um novo número de telefone na agenda do aparelho. De que maneira os passos seriam diferentes para usuários com características distintas (e.g., um adolescente com muita familiaridade com seu aparelho e um senhor de terceira idade que acaba de adquirir seu primeiro aparelho de telefone celular)?
8. *Elementos de uma atividade.* Observando um dia típico de trabalho seu ou de um colega, defina os elementos envolvidos nas diversas atividades realizadas. Avalie a forma como os diversos artefatos utilizados mediam essas atividades.
9. *Cognição distribuída.* Faça uma análise da cognição distribuída, conforme sugerido por Perry (2003), de um sistema de publicação eletrônica de um jornal, levando em consideração pelo menos os papéis de jornalista, fotógrafo e editor.

10. *Signos de interface.* Examine uma aplicação que você não esteja acostumado a utilizar e procure identificar e classificar os diversos signos de interface que ela apresenta.
11. *Artefatos de metacomunicação.* Examine uma aplicação disponível em diferentes dispositivos de base computacional e estilos de interação (e.g., calendário num ambiente gráfico de janelas, na Web e em telefone celular). Com base nos elementos do espaço de design e nas perguntas propostas pela engenharia semiótica, caracterize a metacomunicação, implícita ou explícita, nesses diferentes artefatos. A partir dessa caracterização, analise a facilidade ou dificuldade de um usuário acostumado com uma dessas aplicações passar a utilizar uma outra, ou ainda ficar alternando entre elas.

4

Processos de Design de IHC

Objetivos do Capítulo

- Discutir as atividades envolvidas no design em geral e no design de um artefato computacional interativo em particular.
 - Descrever os fenômenos de IHC sob diferentes perspectivas.
 - Apresentar processos de design de IHC propostos na literatura: modelo de ciclo de vida simplificado, ciclo de vida em estrela, engenharia de usabilidade, design baseado em cenários, dirigido por objetivos e centrado na comunicação.
 - Discutir formas de integrar atividades de IHC e engenharia de software, incluindo métodos ágeis de desenvolvimento de software.
-

Desde sua concepção e durante todo o seu desenvolvimento, um sistema interativo deve ter o propósito de apoiar os usuários a alcançarem seus objetivos. O projeto de um sistema interativo é um processo iterativo de análise, síntese e avaliação, no qual artefatos são coletados e produzidos visando não apenas à construção do sistema, mas também à promoção de uma boa experiência de uso desse sistema. Neste capítulo, primeiro discutimos sobre a atividade de design e em seguida descrevemos alguns processos de design de IHC propostos na literatura.

4.1 O Que é Design?

Em nosso cotidiano, com frequência lidamos com diversos artefatos. Artefatos são produtos artificiais, fruto da inteligência e do trabalho humano, construídos com um determinado propósito em mente. São artefatos, por exemplo: um copo, um pente, um sofá, um carro, uma música e uma receita de bolo. Um artefato não surge espontaneamente na natureza. Alguém decide sua função, forma, estrutura e qualidade, e o constrói com seu trabalho (Löwgren e Stolterman, 2004). A inserção de um artefato numa situação do cotidiano representa uma intervenção sobre ela, em alguma medida, e a própria situação influencia a forma como o artefato é utilizado. Por exemplo, ter ou não uma geladeira em casa modifica significativamente a forma como os alimentos são conservados. Por sua vez, a conservação de alimentos influencia a escolha e a quantidade de produtos comprados, bem como a frequência de compra.

A introdução de um artefato pode trazer consequências positivas e negativas para a situação atual. Por exemplo, adquirir uma bicicleta pode significar: mudar o meio de transporte utilizado para várias atividades, como ir ao trabalho, à padaria ou fazer um passeio; a economia de dinheiro gasto em outros meios de transporte; a prática de esportes para uma vida mais saudável; e ajuda na preservação do meio ambiente. Nesse sentido, adquirir uma bicicleta é uma intervenção positiva. No entanto, o dono da bicicleta agora precisa reservar um local em sua casa para armazená-la, e comprar um capacete e outros equipamentos para utilizar a bicicleta com segurança. Essas podem ser consideradas consequências negativas da intervenção realizada.

Quando analisamos uma situação, considerando as pessoas, artefatos, processos, demais elementos e relações entre eles, podemos encontrar problemas, características desagradáveis ou algo que podemos melhorar.

É comum tomarmos atitudes para resolver os problemas, diminuir as características desagradáveis e melhorar o que mais for possível. Essas atitudes envolvem atividades de design, conscientes ou inconscientes, realizadas com um cuidado maior ou menor. Em termos bem gerais (Lawson, 2006; Löwgren e Stolterman, 2004), podemos caracterizar a atividade de **design** como sendo (Figura 4.1):

- a **análise** da situação atual: estudar e interpretar a situação atual;
- a **síntese** de uma intervenção: planejar e executar uma intervenção na situação atual;
- a **avaliação** da nova situação: verificar o efeito da intervenção, comparando a situação analisada anteriormente com a nova situação, atingida após a intervenção.

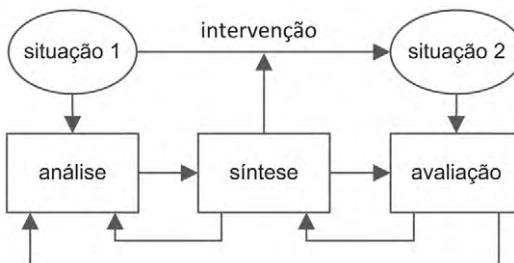


Figura 4.1 Atividades de design envolvidas na intervenção para transformar a situação atual (situação 1) em uma situação desejada (situação 2).

Na **análise da situação atual**, buscamos conhecer os elementos envolvidos e as relações entre eles. Dentre os diversos elementos, geralmente são analisados: pessoas, artefatos, e processos. Como resultado, obtemos uma interpretação da realidade estudada, através de um enquadramento e um recorte particular dela. A complexidade e a dinâmica da realidade fazem com que alguns (aspectos dos) elementos e suas relações sejam enquadrados e outros não. O foco da análise da situação atual depende de vários fatores, como, por exemplo, os assuntos sendo tratados (o domínio), os objetivos das pessoas envolvidas (usuários e demais *stakeholders*), tempo, orçamento e mão-de-obra disponíveis, e até mesmo a filosofia de trabalho. Diferentes focos de análise contribuem para diferentes interpretações da situação atual. Por exemplo, se o objetivo de uma organização for aumentar a produtividade dos seus funcionários, a análise da situação atual pode concentrar esforços na investigação de atividades inefficientes. De outra forma, se o objetivo da organização for compreender os motivos pelos quais seus funcionários não utilizam como esperado os sistemas computacionais oferecidos, a análise da situação atual pode concentrar esforços em compreender a opinião dos funcionários sobre esses sistemas.

A análise da situação atual é frequentemente denominada de análise do *problema*. Entretanto, a análise nem sempre aborda uma situação problemática. Mesmo que a situação atual seja satisfatória, pode surgir uma oportunidade, por conta de uma nova tecnologia, por exemplo, de atingirmos uma situação ainda mais desejável. Nesse caso, a análise da situação atual é importante para identificar as condições em

que uma nova tecnologia pode ser empregada para melhorar o que já é satisfatório. O “problema” a ser resolvido é *encontrar uma boa forma de melhorar* uma ou mais características da situação atual. Em outras palavras, resolver um problema de design significa responder a pergunta: “Como melhorar a situação atual?”

Quando se trata de sistemas computacionais, é comum investigarmos todos os elementos envolvidos durante o uso: os usuários com suas características, necessidades e preferências; as atividades e os objetivos em questão, considerando os artefatos e sistemas computacionais utilizados; e o contexto físico, social e cultural de uso ao longo do tempo (Hackos e Redish, 1998; Sharp *et al.*, 2007). Além desses elementos diretamente envolvidos no uso atual desses artefatos, também é preciso conhecer e articular os interesses das pessoas indiretamente envolvidas, como aquela que paga pelo sistema, geralmente conhecida como cliente, e aquela que constrói o sistema, normalmente chamada de desenvolvedor.

A análise da situação atual aponta as necessidades e oportunidades de melhoria para as quais será projetada uma intervenção. Essas razões para a intervenção costumam ser representadas por metas de design. Em IHC, metas de design referem-se não apenas aos objetivos dos usuários que precisamos ou desejamos apoiar, mas também aos critérios de qualidade de uso descritos na Seção 2.2. Por exemplo, durante a análise de uma situação é possível perceber que os usuários gastam muito tempo processando informações que um sistema computacional poderia fazer mais rapidamente. Nesse caso, uma meta de design poderia ser: aumentar a eficiência das atividades (ou tarefas) do usuário, que é um dos fatores de usabilidade. Já a análise de uma outra situação identifica que vários usuários encontram dificuldades para usar sistemas semelhantes porque não compreendem como funcionam. Nesse caso, uma meta de design poderia ser: comunicar adequadamente através da interface a visão do designer sobre as operações que o usuário pode realizar com o sistema, ou seja, aumentar a comunicabilidade de um artefato computacional interativo. Os Capítulos 5 e 6 apresentam técnicas e representações utilizadas na atividade de análise.

A diferença entre a situação atual e uma situação desejada é a motivação principal para projetarmos e sintetizarmos uma **intervenção**. Frequentemente, uma intervenção é denominada de *solução*, pois responde a pergunta que define um problema a ser resolvido: “Como melhorar esta situação?”. Em alguns casos, uma intervenção adequada é inserir na situação atual um novo sistema interativo ou uma nova versão de um sistema existente, com o intuito de transformá-la em uma situação desejada. Em outros casos, pode ser necessária apenas uma mudança em processos, sem alteração dos sistemas utilizados.

Quando a intervenção envolve o desenvolvimento de sistemas interativos, ela deve articular os interesses dos *stakeholders* com: (1) o conhecimento adquirido na análise da situação atual; (2) o conhecimento sobre intervenções bem e mal avaliadas em casos semelhantes; e (3) o conhecimento sobre as possibilidades e limitações das tecnologias disponíveis. Em particular, é fundamental tomar o devido cuidado com a experiência que pretendemos oferecer aos usuários durante o uso do sistema projetado. Dessa forma, o projeto de um sistema interativo deve definir uma solução de IHC com alta qualidade de uso para impactar a situação atual e a vida dos usuários conforme pretendido. Como visto na Seção 2.1, uma solução de IHC compreende tanto a *interface* com usuário como os *processos de interação* concebidos para apoiarem-no a alcançar seus objetivos. Os Capítulos 7 e 8 discutem a atividade de design (síntese) de uma solução de IHC, bem como as representações, diretrizes e padrões utilizados.

Uma vez definida uma intervenção, é preciso avaliar se ela modifica a situação atual da forma desejada. A **avaliação de uma intervenção** pode ocorrer em vários pontos do processo de desenvolvimento:

- durante a concepção e o desenvolvimento da intervenção, para tentar prever seus possíveis impactos na situação atual (por exemplo, inspecionando as telas produzidas durante o projeto da interface com usuário);
- logo antes da introdução da intervenção, para identificar consequências negativas ou problemas que possam ser evitados (por exemplo, fazendo testes com usuários e produzindo material de treinamento a partir dos seus resultados); ou
- depois da intervenção ter sido aplicada, para verificar os impactos ocorridos (por exemplo, avaliando como os usuários se apropriaram do sistema interativo desenvolvido e quais mudanças ocorreram na sua prática de trabalho).

Quando a intervenção envolve um sistema interativo, existem vários aspectos a serem avaliados, alguns relacionados com a construção do sistema, como a facilidade de manutenção e robustez (Pressman, 2002), outros com o seu uso, como a usabilidade e acessibilidade. Em IHC, os esforços de avaliação se concentram na experiência vivenciada pelos usuários durante a interação com a interface de um sistema interativo, ou seja, durante o uso do sistema. Uma avaliação de IHC deve verificar se a interação e a interface atendem aos critérios de qualidade de uso definidos como prioritários pela análise da situação atual. A avaliação de IHC pode ser feita ao longo do processo de design ou depois do produto pronto. Sempre que possível, devemos avaliar a qualidade de uso desde o início do processo de design dos sistemas interativos, pois o custo de correção de eventuais problemas será menor. Os Capítulos 9 e 10 apresentam conceitos e métodos de avaliação de IHC em detalhes.

4.2 Perspectivas de Design

Existem diferentes interpretações para a atividade de design. Simon (1981) interpreta o design sob uma perspectiva de **racionalismo técnico** (*technical rationality*). Nessa perspectiva, o designer pressupõe que, para um determinado problema, há soluções conhecidas ou métodos bem definidos e precisos para gerá-las. Isso significa que as soluções esperadas certamente serão produzidas se esses métodos forem seguidos. Essas soluções e métodos empregam leis, princípios, normas e valores, geralmente estabelecidos pela natureza, com base em disciplinas de ciências naturais e exatas, como a Física, a Química e a Matemática. Desse modo, a atividade de design se resume a enquadrar uma situação num tipo geral de problema cuja forma de solução seja conhecida. Nessa perspectiva, não existe espaço para o designer questionar ou mudar as verdades estabelecidas pelas relações de causa e consequência (“se eu fizer isto vai acontecer aquilo”).

Em oposição à perspectiva de racionalismo técnico, Schön (1983) propôs uma perspectiva de **reflexão em ação** (*reflection-in-action*) para a atividade de design, como visto na Seção 3.8.3. Nessa perspectiva, uma situação do cotidiano pode estar associada a um problema, que é considerado único. Cada caso é diferente do outro. Consequentemente, o processo de design e a solução encontrada também são únicos. Desse modo, o designer não “está procurando descobrir dicas [da situação atual que apontam] para uma solução padrão. Em vez disso, procura descobrir as características particulares de sua situação tida como problemática, e a partir dessa descoberta gradual, projeta uma intervenção” (Schön, 1983 p.129). Portanto, o processo de concepção de uma solução —o design— é semelhante ao processo de pesquisa científica, no qual a construção de uma hipótese (a partir de uma interpretação da situação atual), a experimentação (proposta de intervenções) e a avaliação são fundamentais.

Na perspectiva de reflexão em ação (Figura 4.2), o designer inicia seu trabalho identificando e interpretando os elementos envolvidos na situação atual, os interesses dos envolvidos direta e indiretamente com o sistema (*stakeholders*) e as possibilidades e limitações das tecnologias disponíveis. Isso lhe permite formular um problema único a ser resolvido. Sendo assim, ele pode resolver tal problema único explorando livremente sua criatividade e utilizar ou não soluções e métodos de resolução de problemas conhecidos, conforme julgar apropriado.

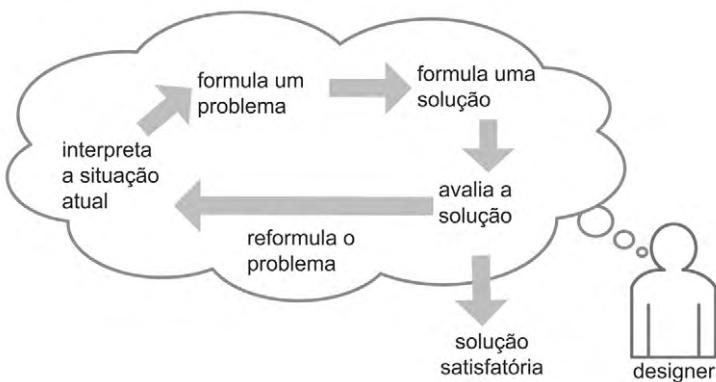


Figura 4.2 Processo de reflexão em ação.

Durante a concepção de uma solução, é muito comum o designer explorar diferentes ideias para poder compará-las e aproveitar o melhor de cada uma delas (Buxton, 2007). As alternativas de solução geradas durante a exploração de ideias costumam ser representadas por algum desenho, maquete ou modelo. A representação de uma alternativa de solução tem um papel de manifestar (parte de) as ideias do designer para outras pessoas envolvidas no projeto. Dessa forma, o cliente, o usuário e o desenvolvedor podem compreender e opinar sobre o projeto do sistema, antes mesmo de ele ser construído e inserido no cotidiano das pessoas.

Segundo Schön (1983), a manifestação das ideias do designer em alguma representação possui outro benefício importante. Enquanto o designer expressa suas ideias, ele acaba “conversando” com a representação, um fenômeno por ele denominado **conversa com os materiais** (*conversation with materials*). Ora o designer “fala” com a representação, expressando suas ideias para a solução sendo concebida, como, por exemplo, “E se eu definir isso deste jeito?”, “Posso utilizar essa mesma ideia em outro lugar” ou “O que acontece se eu modificar isso aqui?”, ora a representação “fala” com o designer no momento em que ele percebe o que concebeu e o avalia, se questionando, por exemplo, sobre “O que é isso que eu representei?” e descobrindo que “Eu não entendi isso direito”, “Isso é diferente do que eu pensei que seria, mas é interessante!”, “Isso está desajeitado, isso não”, “Aquiló não parece bom para mim” ou “Isso não funciona” (Schön e Bennett, 1996).

Depois de expressar suas ideias em alguma representação, o designer tem uma condição melhor de avaliá-las para verificar se a solução proposta satisfaz seus objetivos. Caso não encontre uma solução satisfatória, o designer critica não apenas as soluções que se apresentam, mas também sua própria formulação do problema, e experimenta reformulá-lo. Essa reformulação pode modificar os elementos envolvidos na situação analisada e seus significados de uma maneira que não havia sido prevista

pelo designer. Dessa forma, o designer precisa continuar descobrindo e refletindo sobre quais são as consequências dessa reformulação para a próxima tentativa de resolver o problema. Ele pode se perguntar, por exemplo: “As consequências dessa reformulação são desejáveis?”, “Que novos (potenciais) problemas foram criados?”, “O que melhorou com essa nova reformulação?”, e assim por diante. Nesse processo reflexivo, “o esforço do designer para resolver o problema reformulado produz novas descobertas que estimulam novas reflexões em ação” (Schön, 1983, p. 132), até que uma ação do designer dê origem a uma solução satisfatória.

Resumindo, a reflexão em ação ocorre enquanto o designer conversa com (age sobre) a representação, refletindo, avaliando e aprendendo sobre o que está fazendo enquanto o faz, de forma que essas reflexões influenciem ações futuras em direção à concepção da solução. Segundo o próprio Schön:

Refletir em ação é “interagir com o modelo, obter resultados surpreendentes, tentar interpretá-los, e então inventar novas estratégias de ação com base nas novas interpretações” (Schön e Bennett, 1996, p.181).

Algumas pessoas, na tentativa de “resolver um problema” em IHC, buscam aplicar diretamente padrões de interface no projeto de sistemas interativos, numa perspectiva de racionalismo técnico. Entretanto, esses padrões não fornecem soluções prontas para um problema genérico, mas sim um repertório de soluções conhecidas para problemas específicos e contextualizados. Como descrito na Seção 8.3, padrões de design de interface devem ser cuidadosamente analisados e adaptados a cada projeto (Tidwell, 2005; Borchers, 2001). Em outras palavras, devem ser utilizados numa perspectiva reflexiva alinhada à epistemologia da prática de Schön (1983).

Conceber uma solução adequada ao problema não é uma tarefa simples, e geralmente requer uma equipe multidisciplinar de design. Ela exige do designer as seguintes habilidades e conhecimentos (Löwgren e Stolterman, 2004, p.45):

- criatividade e capacidade de análise para criar e modelar ideias;
- capacidade de crítica e julgamento para decidir;
- capacidade de comunicação e negociação para trabalhar com clientes, usuários e desenvolvedores;
- conhecimento sobre as tecnologias disponíveis para projetar qualidades estruturais e funcionais;
- conhecimento sobre valores e ideais dos envolvidos para projetar qualidades éticas;
- capacidade de apreciar e compor coisas agradáveis aos sentidos para projetar qualidades estéticas.

4.3 Processos de Design de IHC

Como visto na seção anterior, o design é um processo que envolve as seguintes atividades básicas: a análise da situação atual (identificação do problema), a síntese de uma intervenção e a avaliação dessa intervenção projetada ou já aplicada à situação atual. Cada processo de design detalha essas atividades básicas de uma forma particular, definindo: como executar cada atividade; a sequência em que elas devem ser executadas; quais atividades podem se repetir, e por quais motivos; e os artefatos consumidos e produzidos em cada uma delas.

Uma característica básica dos processos de design de IHC é a execução das atividades de forma **iterativa**, permitindo refinamentos sucessivos da análise da situação atual e da proposta de intervenção. Dessa forma, o designer tem boas oportunidades de aprender mais e melhor tanto sobre o problema a ser resolvido quanto sobre a solução sendo concebida.

Geralmente os processos de design de IHC começam analisando a situação atual. Quando o designer considera ter adquirido conhecimento suficiente sobre essa situação e identificado as necessidades e oportunidades de melhoria, ele prossegue seu trabalho sintetizando (concebendo, modelando e construindo) uma intervenção. Enquanto ele projeta uma intervenção, pode sentir necessidade de *conhecer mais* ou *rever uma interpretação anterior* sobre a sua análise da situação atual. Assim, ele volta à atividade anterior para ampliar, refinar ou reformular sua interpretação, numa nova iteração do processo de design. Com a revisão da análise, o designer amplia, refina ou reformula a sua proposta de intervenção.

Tendo uma proposta de intervenção em mãos, o designer passa a avaliá-la para julgar se ela é satisfatória. Durante essa avaliação, o designer pode perceber que ainda precisa rever sua análise ou sua proposta de intervenção. Esse processo iterativo se repete quantas vezes forem necessárias, até o designer obter uma intervenção satisfatória. O início do processo de design tende a ter uma iteração maior entre a análise da situação atual e a síntese da intervenção, enquanto o final do processo tende a ter uma iteração mais acentuada entre a síntese e a avaliação da intervenção.

Mesmo executando essas três atividades básicas do processo de design de forma iterativa, é possível empregar quantidade de tempo e esforço diferente em cada uma delas. Por exemplo, podemos ter um design dirigido pelo problema ou dirigido pela solução. O *design dirigido pelo problema* despende mais tempo analisando a situação atual, as necessidades e as oportunidades de melhoria (o problema), e menos tempo explorando possíveis intervenções (as soluções). O *design dirigido pela solução* faz exatamente o contrário, emprega pouco tempo analisando a situação atual, e mais tempo explorando possíveis intervenções. Kruger e Cross (2006) realizaram um ex-

perimento com designers experientes desempenhando a mesma tarefa para comparar as estratégias de design seguidas por eles (dirigidas pelo problema ou pela solução) e a qualidade das soluções obtidas. Eles concluíram que a qualidade geral das soluções não estava diretamente relacionada com a estratégia seguida. Observaram ainda que os designers que atuaram dirigidos pela solução produziram, em média, resultados mais criativos, mas com menos preocupação com os aspectos estéticos, ergonômicos, técnicos e comerciais. A criatividade parece ter sido favorecida pela exploração de mais ideias para possíveis soluções.

Alguns processos de design de IHC prescrevem qual deve ser a primeira atividade a ser realizada, bem como a sequência de transições entre elas. Lawson (2006) questiona essa sequência estrita de atividades de design. Para ele, é possível iniciar o processo de design em qualquer atividade e realizar qualquer transição durante o processo quantas vezes forem necessárias (Figura 4.3). Dessa forma, cabe ao designer decidir qual será a primeira atividade a ser executada e as transições entre atividades que ele vai realizar. O que realmente importa, segundo Lawson, é partirmos de um problema (as necessidades e oportunidades de melhoria na situação atual), realizarmos o processo de design (atividades de análise, síntese e avaliação) e no final chegarmos a uma solução (uma proposta de intervenção).

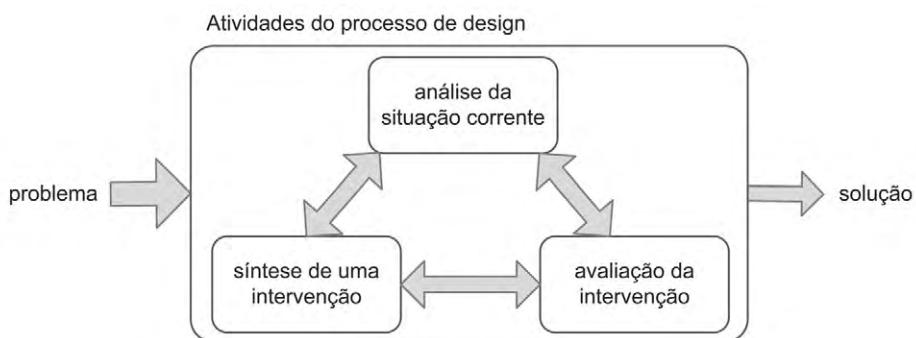


Figura 4.3 Sequência genérica de atividades durante o processo de design (adaptado de Lawson, 2006).

Os processos de design de IHC buscam atender e servir em primeiro lugar aos usuários e aos demais envolvidos (*stakeholders*), e não às tecnologias. Boa parte desses processos é **centrada no usuário**, isto é, seguem estes princípios (Gould e Lewis, 1985, p. 300):

- *foco no usuário*: o designer deve projetar a interação e a interface de um sistema interativo para atender às necessidades dos usuários e ajudá-los a alcançarem seus objetivos. Sendo assim, o designer deve estudar quem serão os usuários do sistema, seus objetivos, suas características físicas (capacida-

- de de movimentos, visão, audição etc.), cognitivas e comportamentais, sua formação educacional (capacidade de leitura e escrita, conhecimentos adquiridos etc.), e o que eles costumam fazer para alcançar seus objetivos;
- *métricas observáveis*: o processo de design deve permitir a realização de experimentos (estudos empíricos) em que representantes dos usuários usem simulações ou protótipos do sistema para realizarem suas atividades e alcançarem seus objetivos. Durante o experimento, a performance e as reações dos usuários devem ser observadas, registradas e analisadas;
 - *design iterativo*: quando problemas forem encontrados durante os experimentos com usuários, eles deverão ser corrigidos. Isso significa que as atividades do processo de design devem ser iterativas, ou seja, o ciclo de projeto, avaliação com medições empíricas e reprojeto deve se repetir quantas vezes forem necessárias.

Os processos de design de IHC destacam a importância de envolver os usuários durante suas atividades para dar-lhes oportunidade de participar, direta ou indiretamente, nas decisões tomadas. Quanto mais cedo os usuários forem envolvidos no processo de design, mais cedo será possível aprender sobre suas necessidades e assim influenciar positivamente a síntese da solução, bem como identificar e corrigir eventuais problemas. Isso nos permite desenvolver um sistema interativo mais interessante para os usuários e com maior qualidade de uso, porque temos acesso às interpretações e opiniões dos usuários sobre o resultado do design.

Como discutido no Capítulo 1, uma equipe multidisciplinar contribui para o design de IHC. Cada profissional observa e interpreta a situação atual de um ponto de vista particular, que contribui para enriquecer a identificação das necessidades e oportunidades de melhoria (a identificação e análise do problema). Uma equipe de pessoas formadas em diferentes áreas de conhecimento favorece a síntese (projeto) de uma intervenção, pois facilita o surgimento e a exploração de ideias diversas. Além disso, uma equipe multidisciplinar também permite uma avaliação mais rica e diferenciada das intervenções (soluções) propostas. Sempre que houver recursos disponíveis, devemos investir em uma equipe multidisciplinar de design de IHC.

Preece, Sharp e Rogers (Preece *et al.*, 2002; Sharp *et al.*, 2007) organizaram as atividades de design de IHC em um modelo de processo de design simples, ilustrado na Figura 4.4. Esse processo de design de IHC destaca a importância do design centrado no usuário, de avaliações da proposta de solução usando versões interativas e da iteração entre as atividades.

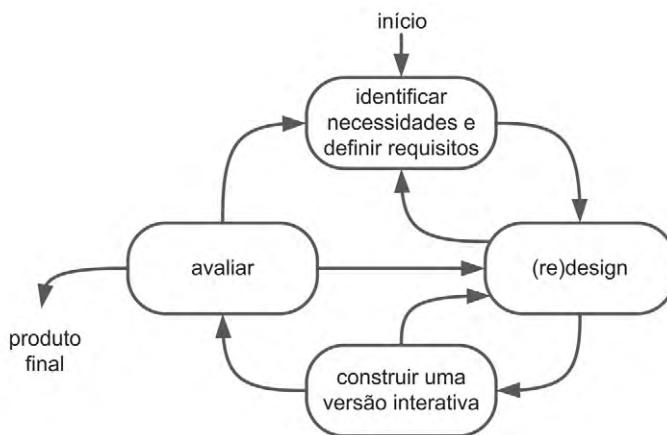


Figura 4.4 Modelo simples de processo de design de IHC (adaptado de Sharp *et al.*, 2007).

Considerando a sequência genérica de atividades de design identificada por Lawson (2006), observamos que Preece e coautoras segmentam a atividade de síntese em: design (ou redesign) conceitual e na construção de uma versão interativa.

Durante o **(re)design** da interação e da interface, o designer explora diferentes ideias em alternativas de design para elaborar uma solução adequada às necessidades e aos requisitos definidos na atividade de análise. O resultado dessa atividade de design pode ser registrado em descrições textuais da interação (cenários), esboços de interface (desenhos de tela) ou em qualquer modelo ou representação da interface e da interação usuário-sistema.

Para melhor avaliar o design resultante, o designer constrói **versões interativas** das propostas de solução que simulem o funcionamento da interface e deixem clara a interação projetada. Isso facilita a participação dos usuários durante a avaliação de IHC.

Assim como a maior parte dos processos de design, esse modelo de processo também é bastante iterativo. Cada atividade pode revelar a necessidade de retornar a uma atividade anterior para ampliar, refinar ou retificar algum entendimento ou artefato produzido. A iteração entre as atividades ocorre quantas vezes forem necessárias, limitada apenas pelo orçamento, tempo e recursos disponíveis. Idealmente o processo de design é concluído com uma avaliação de que a solução de IHC atende às necessidades e aos requisitos identificados. O produto final é uma especificação da interação e da interface com usuário que servirá de insumo nas fases seguintes de desenvolvimento do sistema iterativo.

Existem várias outras propostas de processos de design de IHC. Cada uma privilegia uma forma de pensar, uma sequência de atividades ou o emprego de certos

artefatos. Dentre as propostas existentes este livro apresenta brevemente o ciclo de vida em estrela (Hix e Hartson, 1993), dois ciclos de vida para Engenharia de Usabilidade (Nielsen, 1993; Mayhew, 1999), o design contextual (Beyer e Holtzblatt, 1998), o design baseado em cenários (Rosson e Carroll, 2002), o design dirigido por objetivos (Cooper *et al.*, 2007) e o design centrado na comunicação (Barbosa *et al.*, 2004).

4.3.1 Ciclo de Vida em Estrela

O ciclo de vida em estrela foi desenvolvido por Hix e Hartson no início da década de 1990 (Hix e Hartson, 1993) e foi um dos primeiros ciclos de vida voltados para IHC amplamente difundidos. Esse processo é composto por seis atividades, conforme ilustrado na Figura 4.5.

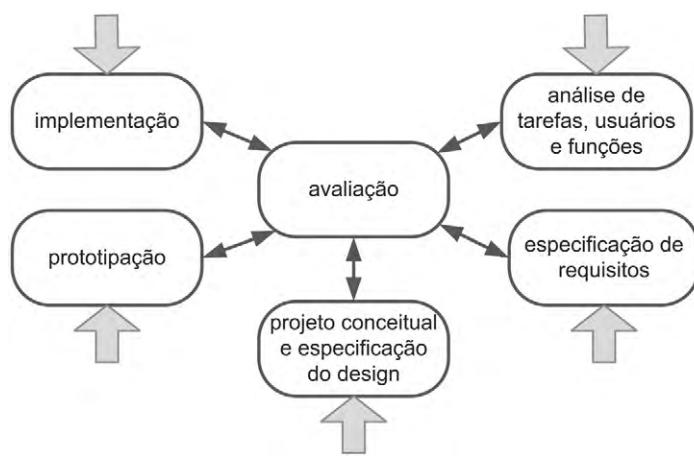


Figura 4.5 Ciclo de vida em estrela.

A **análise de tarefas, de usuário e funções** é a atividade responsável pelo aprendizado da situação atual e pelo levantamento das necessidades e oportunidades de melhoria. A atividade de **especificação de requisitos** de IHC consolida uma interpretação da análise, definindo os problemas que devem ser resolvidos com o projeto de uma solução de IHC.

A atividade geral de síntese é segmentada em três atividades: **projeto conceitual e especificação do design**, na qual a solução de IHC é concebida; **prototipação**, na qual versões interativas das propostas de solução são elaboradas para serem avaliadas; e **implementação**, na qual o sistema interativo final é desenvolvido. Essa última atividade está fora do escopo deste livro, pois já é abordada extensamente na literatura de Engenharia de Software.

A atividade de **avaliação** aparece no modelo como central, e é de fato desdobrada na avaliação dos resultados de cada uma das demais atividades. A avaliação deve

verificar se os dados coletados na atividade de análise e os requisitos especificados estão de acordo com a realidade e atendem às necessidades dos usuários. Deve também detectar problemas de usabilidade (e dos demais critérios de qualidade de uso discutidos na Seção 2.2) nas representações de design, nos protótipos e no sistema final. Hix e Hartson ressaltam que é importante avaliar o que foi produzido desde o início, enquanto ainda há tempo para corrigir os erros com um custo reduzido.

No ciclo de vida em estrela, cabe ao designer decidir qual atividade deve ser realizada primeiro, dependendo do que estiver disponível quando iniciar o processo. Por exemplo, se a intenção for projetar uma nova versão do sistema, o designer pode começar o projeto da nova versão pela avaliação da versão atual. Em outro caso, pode ser necessário implementar o mesmo sistema em outra plataforma semelhante, como outro sistema operacional, por exemplo. Sendo assim, o designer pode optar por começar pela implementação do sistema, aproveitando o projeto que havia sido feito para a plataforma anterior. Para projetar um novo sistema, é comum começar pela análise de tarefas, usuários e funcional e seguir no sentido horário pelas atividades na Figura 4.5.

Assim como na sequência genérica de atividades de design identificada por Lawson (2006), o ciclo de vida em estrela também é iterativo e não prescreve a sequência das atividades. A única exigência aqui é que, após concluir cada atividade, o designer avalie os resultados obtidos para verificar se ele encontrou ou está no caminho de encontrar uma solução satisfatória. Todas as atividades do ciclo de vida em estrela estão interligadas pela atividade de avaliação, ou seja, sempre é preciso passar por uma avaliação ao concluir uma atividade e antes de iniciar outra.

4.3.2 Engenharia de Usabilidade de Nielsen

Jakob Nielsen (1993) definiu **engenharia de usabilidade** como um conjunto de atividades que devem ocorrer durante todo o ciclo de vida do produto, ressaltando que muitas delas ocorrem nos estágios iniciais do projeto, antes que a interface com usuário em si seja projetada. Nielsen propõe o seguinte conjunto de atividades em seu ciclo de vida:

1. Conheça seu usuário
2. Realize uma análise competitiva
3. Defina as metas de usabilidade
4. Faça designs paralelos
5. Adote o design participativo
6. Faça o design coordenado da interface como um todo

7. Aplique diretrizes e análise heurística
8. Faça protótipos
9. Realize testes empíricos
10. Pratique design iterativo

Nesse ciclo de vida, o primeiro passo consiste em **estudar os usuários e os usos** pretendidos do produto. Segundo Nielsen, as características de usuários individuais e a variabilidade nas tarefas são os fatores de maior impacto na usabilidade. Ele utiliza o termo “usuário” de forma ampla, incluindo todos aqueles cujo trabalho é afetado de algum modo pelo produto, isto é, usuários diretos e demais *stakeholders*.

Essa atividade envolve conhecer as características individuais dos usuários e do seu ambiente físico e social de trabalho (veja Seção 5.2), suas atividades (veja Seção 6.4) e as formas como lidam com circunstâncias excepcionais e emergenciais. Nielsen sugere procurar usuários especialmente eficientes e que desenvolveram suas próprias estratégias para *contornar* as limitações dos sistemas existentes. Durante esse estudo, é fundamental ir além das atividades dos usuários tal como são realizadas atualmente, buscando identificar a *razão funcional* subjacente a cada atividade: o que realmente precisa ser feito, ou seja, os objetivos finais dos usuários e demais *stakeholders*.

Nielsen alerta para o fato de que os usuários não serão os mesmos após a introdução do sistema. O sistema modifica os usuários, e à medida que isso ocorre eles usarão o sistema de novas formas, num fenômeno denominado por Carroll e Rosson (1991) “*coevolução de tarefas e artefatos*”. Embora seja impossível prever todas as mudanças, um design mais flexível, adaptável ou extensível tem mais chances de apoiar esses novos usos (de Souza, 2005a; de Souza e Barbosa, 2005).

A **análise competitiva** consiste em examinar produtos com funcionalidades semelhantes ou complementares. Como esses produtos já estão prontos, podem ser testados com mais facilidade e realismo do que protótipos. Eles podem ser inspecionados e testados visando avaliar tanto as funcionalidades que apoiam como as questões de IHC tidas como relevantes para o projeto. Como resultado, o designer pode obter um conjunto de informações sobre o que funciona e o que não funciona naquele domínio, o que pode ser aperfeiçoado, e por quê. Nielsen ressalta que a análise competitiva envolve não apenas sistemas computacionais interativos, mas também qualquer outra forma de atividade de usuários com objetivos semelhantes. Por exemplo, antes de projetar uma agenda eletrônica, devemos investigar não apenas as agendas eletrônicas disponíveis, mas também a forma como as pessoas utilizam agendas em papel.

A **definição das metas de usabilidade** envolve definir os fatores de qualidade de uso que devem ser priorizados no projeto, como serão avaliados ao longo do proces-

so de design, e quais faixas de valores são inaceitáveis, aceitáveis e ideais para cada indicador de interesse. Com frequência, essa priorização se baseia nos indicadores atuais de desempenho dos usuários ao utilizarem o sistema. O Exemplo 4.1 ilustra uma definição de metas de usabilidade para um sistema de busca de um livro num quiosque de livraria.

Exemplo 4.1 – Metas de usabilidade para um sistema de busca de livros em uma livraria

Considere um sistema de quiosque de livraria pouco utilizado, em que 50% dos usuários desistem de fazer uma busca por um livro antes de concluir-la. Podemos estabelecer como metas que mais pessoas utilizem o sistema e que somente 30% dos usuários abandonem a tarefa de busca. As metas de usabilidade para esse projeto podem ser: aumentar a *facilidade de aprendizado* e a *eficiência do sistema*. Os indicadores correspondentes poderiam ser: número de usuários que acessam o sistema em diferentes dias da semana; proporção de usuários que completam/abandonam a tarefa de busca; tempo que cada usuário leva para concluir a tarefa com sucesso; tempo que cada usuário despende antes de abandonar a tarefa; número de erros cometidos.

Para avaliar melhor essas metas, podemos conduzir um estudo para avaliar qual é o problema principal. Vamos supor que o estudo revele que os principais problemas enfrentados pelos usuários são a dificuldade de uso (e.g., o usuário não sabe o que fazer num determinado momento por falta de instruções e controles claros na interface de usuário) e a ineficiência no uso do sistema (e.g., o usuário desiste quando descobre que há passos intermediários aparentemente desnecessários no processo de busca). Com base nesses resultados e nos dados quantitativos coletados, podemos então estabelecer as faixas de valores aceitáveis e ideais para cada indicador, conforme ilustrado pela Figura 4.6.

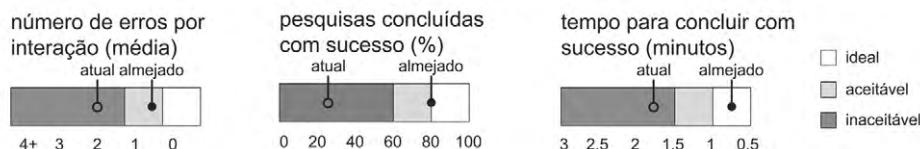


Figura 4.6 Faixas de valores para indicadores de usabilidade.

Durante a definição das metas de usabilidade, podemos estabelecer também metas de **retorno de investimento**, através de uma análise de custo e benefício envolvendo o sistema ou a prática atual, as despesas com o projeto do novo sistema e a economia que o seu uso deve proporcionar. Bias e Mayhew (2005) discutem diversas formas de avaliar o retorno de investimento. Podemos, por exemplo, calcular o tempo que um funcionário leva para realizar o seu trabalho antes e depois da introdução do novo sistema, e computar o ganho monetário com base no salário desse funcionário e o tempo economizado.

Nielsen advoga o **design paralelo**, que consiste em elaborar diferentes alternativas de design, de preferência por três ou quatro designers trabalhando de forma in-

dependente, para então selecionar as que vão ser detalhadas nas atividades seguintes do processo. Nessa etapa, cada designer deve empregar pouco tempo (desde algumas horas até dois dias) para elaborar seus designs iniciais e, portanto, trata-se de uma forma bastante barata de explorar o espaço de solução. Para motivar soluções bem diferentes, podemos solicitar a cada designer que explore um aspecto diferente do problema, tais como: usuários novatos *vs.* experientes; computador desktop *vs.* dispositivo móvel; interface gráfica *vs.* verbal *vs.* por caneta *vs.* por toque. Ao final dessa etapa, as soluções alternativas são analisadas e um design consolidado é elaborado, geralmente combinando elementos de mais de uma alternativa.

O **design participativo**, também advogado por Nielsen, consiste em a equipe de design ter acesso permanente a um conjunto de usuários tidos como representativos da população-alvo de usuários. Isso é importante porque, mesmo após as atividades iniciais de investigação, invariavelmente surgem questões ao longo do processo de design que requerem novas consultas aos usuários. Nielsen alerta, no entanto, que os usuários não são designers, então não podemos esperar que eles produzam designs ou entendam especificações produzidas utilizando notações que eles desconhecem, nem mesmo que saibam definir com clareza o que querem ou precisam. Em vez disso, ele chama atenção para a necessidade de produzirmos representações dos designs propostos que eles entendam facilmente, como protótipos, maquetes ou esboços de tela, para que eles possam reagir às propostas, fornecer *feedback* informativo, levantar novas questões e participar ativamente das discussões acerca das soluções propostas. Além disso, devemos envolver diferentes usuários ao longo do projeto, para que as decisões de design não se baseiem em idiossincrasias de uma ou duas pessoas, e para que tenhamos sempre um olhar novo sobre os designs propostos.

Para evitar inconsistências na interface com usuário projetada, é importante haver um responsável pelo **design coordenado da interface**, ou seja, da interface como um todo. Isso inclui não apenas os elementos de interface propriamente ditos, mas também toda a documentação, o sistema de ajuda e tutoriais produzidos sobre o sistema. Caso o produto deva ser introduzido como parte de uma família de produtos, devemos considerar a consistência entre eles de forma holística, mas sempre tomando o cuidado para que a consistência não adquira uma importância desmedida a despeito das metas de usabilidade prioritárias para o projeto.

Nielsen sugere que a equipe de design siga **diretrizes**, princípios bem conhecidos para o design da interface com usuário. À medida que a interface for projetada, deve ser feita uma avaliação heurística para avaliar se as diretrizes não estão sendo violadas (veja Seção 10.1.1). As diretrizes podem ser *gerais*, aplicáveis a todas as interfaces com usuário; *específicas a uma categoria* ou plataforma computacional

(e.g., interfaces gráficas baseadas em janelas para computadores desktop; interfaces gestuais para grandes monitores; interfaces verbais para atendimento automático pelo telefone; interfaces de toque para dispositivos móveis); ou ainda específicas a um *produto* individual (e.g., planilha eletrônica, editor de texto, editor gráfico). Por exemplo, “forneça *feedback*” é uma diretriz geral, à qual podemos relacionar diretrizes específicas a certas categorias (e.g., para uma interface gráfica: “certifique-se de que os principais objetos de interesse estejam visíveis na tela e que revelem seus principais atributos”; para um sistema Web: “certifique-se de que o usuário sabe onde ele está, de onde veio e para onde pode ir”) e produtos (e.g., “o ícone de uma lixeira deve estar visível e indicar se ela contém ou não algum item”). A Seção 8.2 apresenta algumas diretrizes comumente encontradas na literatura de IHC.

Antes de iniciar os esforços de implementação da interface com usuário, Nielsen recomenda fazer **protótipos** dos sistemas finais, que podem ser desenvolvidos rapidamente e a um custo baixo, para que sejam avaliados junto a usuários e modificados à medida que a equipe de design adquire um melhor entendimento dos problemas, visando oferecer uma solução mais adequada. Para que essa atividade possua custo baixo, apenas parte do sistema é prototipada. Podemos desenvolver um **protótipo horizontal**, que visa apresentar o sistema em abrangência mas com pouca profundidade (i.e., a aparência da interface e navegação entre telas, mas sem a funcionalidade subjacente, como algoritmos e acessos a bases de dados), ou um **protótipo vertical**, no qual pouca funcionalidade é explorada em profundidade para que seja testada em circunstâncias realistas. Nielsen enumera diversas estratégias para produzir protótipos mais rapidamente:

- não se importar muito com a eficiência da implementação (e.g., utilização de espaço em disco e tempo de processamento), desde que não seja essencial para a avaliação junto ao usuário;
- aceitar código de qualidade mais baixa ou pouco confiável;
- utilizar algoritmos simplificados e que não conseguem lidar com todos os casos específicos;
- fazer uma simulação com uma pessoa atuando como o computador “por trás da cortina”, numa técnica denominada *Wizard of Oz* (mágico de Oz), na qual o usuário, sem saber, interage com uma pessoa que determina as respostas do protótipo, e não com um processamento computacional;
- utilizar uma plataforma computacional diferente da almejada para facilitar o desenvolvimento do protótipo (e.g., utilizar uma ferramenta de autoria em vez da linguagem de programação que será utilizada para construir o produto final);

- utilizar protótipos de baixa fidelidade, mas que representem a essência da interação;
- utilizar dados falsos e conteúdos fictícios, desde que não atrapalhem a avaliação junto aos usuários;
- utilizar maquetes em papel em vez de um sistema funcional;
- utilizar um protótipo verbal, no qual o avaliador descreve oralmente uma interface possível e explora uma série de perguntas do tipo “E se...?”;
- utilizar cenários (veja Seção 4.3.5).

A partir dos protótipos, os designers devem fazer **testes empíricos**, que consistem principalmente na observação dos usuários ao utilizarem os protótipos para realizar certas tarefas. A Seção 10.2.1 descreve a avaliação através de testes de usabilidade.

Com base nos problemas de usabilidade e nas oportunidades reveladas pelos testes empíricos, os designers produzem uma nova versão da interface, e repassam pelas atividades do processo, num **design iterativo**. A cada iteração de design e avaliação, alguns problemas são corrigidos (e infelizmente outros podem ser introduzidos), e o processo deve ser repetir até que as metas de usabilidade tenham sido alcançadas. Nesse processo iterativo, é importante tornar as decisões de design explícitas e registrá-las para referência futura, ou seja, capturar o **design rationale** (Moran e Carroll, 1996). É importante registrar as decisões tomadas para que, no futuro, não sejam tomadas decisões que sacrificuem metas de usabilidade importantes ou que introduzam inconsistências por falta de informação sobre o histórico do projeto.

Finalmente, após a introdução de um produto, devemos **coletar dados de uso**, não apenas para avaliar o retorno de investimento, mas também para planejar a próxima versão do produto.

4.3.3 Engenharia de Usabilidade de Mayhew

Deborah Mayhew (1999) propôs um ciclo de vida para a engenharia de usabilidade. Com uma visão holística, esse processo de design reúne e organiza diferentes atividades propostas na área de IHC para orientar o trabalho do designer em direção a uma boa solução interativa. A Figura 4.7 apresenta as três fases desse processo iterativo: análise de requisitos, design/avaliação/desenvolvimento e instalação.

Na fase de **análise de requisitos** são definidas as metas de usabilidade com base no perfil dos usuários, análise de tarefas, possibilidades e limitações da plataforma em que o sistema será executado e princípios gerais de design de IHC. Nesse processo, as metas de usabilidade costumam ser representadas em “guias de estilos” para auxiliar sua verificação durante as demais atividades do processo (veja Seção 8.4).

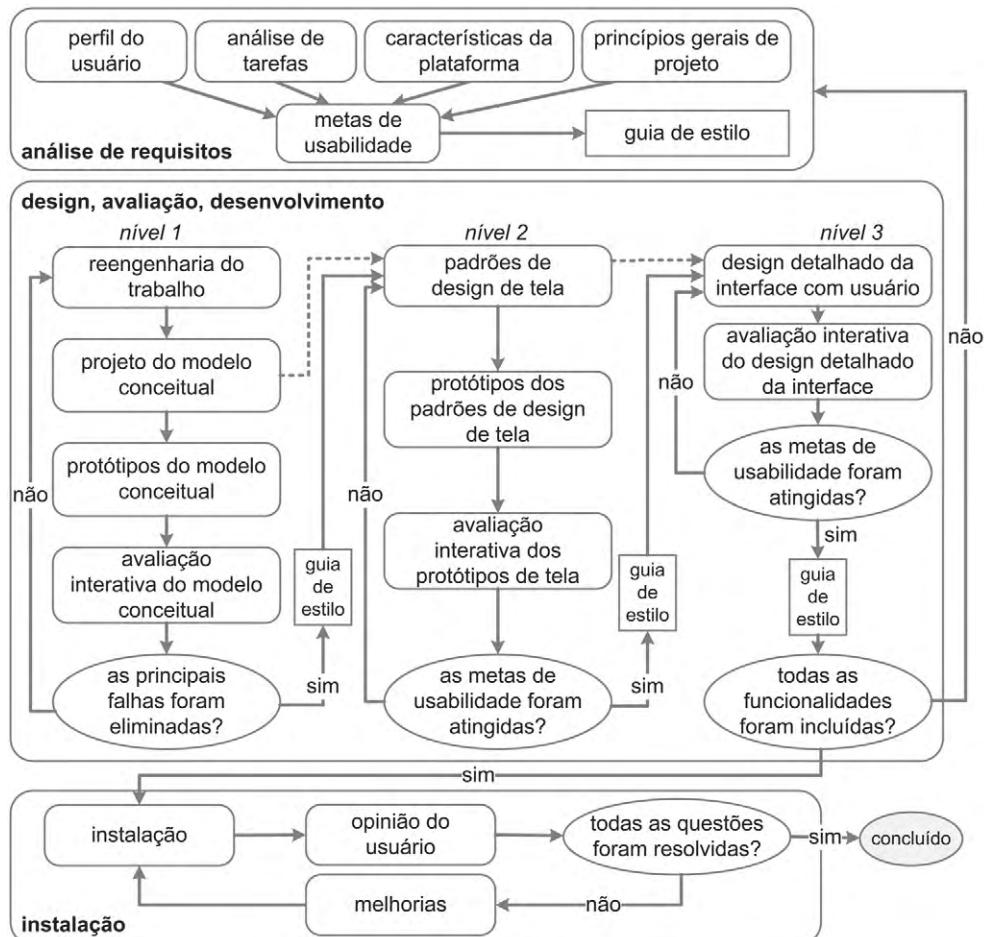


Figura 4.7 Ciclo de vida para a engenharia de usabilidade (adaptado de Mayhew, 1999).

A fase de **design, avaliação e desenvolvimento** tem por objetivo conceber uma solução de IHC que atenda às metas de usabilidade estabelecidas na fase anterior. Esse processo propõe projetar a solução de IHC em três níveis de detalhes. No primeiro nível, o designer precisa realizar a reengenharia do trabalho, repensando a execução das tarefas para alcançar os objetivos dos usuários, elaborar alternativas de solução do modelo conceitual, elaborar protótipos de baixa fidelidade e avaliar tais protótipos. No segundo nível, o designer deve estabelecer padrões de design de IHC para a solução sendo concebida, construir protótipos de média fidelidade de acordo com esses padrões e avaliá-los. No terceiro nível, o designer realiza o projeto detalhado da interface, com alta fidelidade, para ser implementado. Durante o desenvolvimento do sistema, a interface deve ser avaliada com a participação dos usuários.

Na fase de **instalação**, o designer deve coletar opiniões dos usuários depois de algum tempo de uso. Essas opiniões serão úteis para melhorar o sistema em versões futuras ou até mesmo para apontar a necessidade de desenvolver novos sistemas interativos ainda não previstos.

4.3.4 Design Contextual

O design contextual (*contextual design*) é um processo de design de IHC que orienta o designer a compreender profundamente as necessidades dos usuários¹ através de uma investigação minuciosa do contexto de uso (Beyer e Holtzblatt, 1998; Holtzblatt *et al.*, 2001). Essa apreciação cuidadosa do que ocorre em contexto é fundamental para o designer elaborar uma solução de IHC adequada. As atividades do design contextual são: investigação contextual, modelagem do trabalho, consolidação, reprojeto do trabalho, projeto do ambiente do usuário, prototipação e teste com usuários.

Na **investigação contextual** (*contextual inquiry*), o designer busca conhecer quem são os usuários, suas necessidades, seus objetivos e a forma como ele trabalha no seu dia a dia. Essa investigação ocorre diretamente no ambiente de trabalho do usuário para que o designer possa ter acesso a informações do contexto. O objetivo da investigação contextual é revelar detalhes e motivações implícitas no trabalho dos usuários a fim de informar o designer sobre suas necessidades reais. Essas informações contextuais são importantes para apoiar as decisões de design. O método de investigação contextual é descrito em detalhes na Seção 5.5.7.

O conhecimento adquirido na investigação contextual permite ao designer **modelar o trabalho** de cada usuário investigado separadamente. Existem cinco tipos de modelos de trabalho utilizados no design contextual: modelo de fluxo, modelo de sequência, modelo de artefato, modelo de cultura e modelo físico. Cada um representa um aspecto do trabalho sob a perspectiva de um usuário investigado, podendo conter conceitos complexos e muitos detalhes. Eles são utilizados para registrar e compartilhar com a equipe de projeto os conhecimentos adquiridos na investigação contextual.

A **consolidação** dos modelos de trabalho possibilita organizar e atribuir significado ao trabalho desempenhado por cada papel, perfil ou classe de usuário investigado. Um diagrama de afinidade deve ser elaborado para estruturar coletivamente

1 Beyer e Holtzblatt empregam o termo *customers* (clientes) para referenciar aqueles que usam o sistema. Neste livro, utilizamos o termo *usuário* para nos referirmos às pessoas que realizam as atividades que já são ou serão apoiadas pelos sistemas computacionais interativos investigados ou sendo concebidos. Denominamos *stakeholders* as demais partes interessadas no sistema.

a forma como os usuários trabalham, sem perder as particularidades de cada caso. O diagrama de afinidade sintetiza grande quantidade de dados qualitativos em um grande mapa. Além disso, os modelos de trabalho de todos os usuários investigados devem ser combinados em cinco modelos de trabalho consolidados, um para cada tipo de modelo. O resultado dessa consolidação é um conjunto de dados corporativos que vai guiar o projeto de IHC e podem ser reutilizados em projetos futuros. Os diagramas de afinidade são descritos na Seção 5.5.4, no contexto de sessões de *brainstorming*.

A consolidação dos modelos de trabalho fornece insumos para o designer **re-projetar** a forma como os usuários trabalham. O designer utiliza *storyboards* para explorar ideias sobre como melhorar a prática de trabalho com o suporte oferecido pela tecnologia.

Uma vez concebida uma nova maneira de trabalhar, o designer segue **projetando uma solução de interação e de interface** que apoie essa nova forma de trabalhar. Por fim, o designer deve construir **protótipos** do sistema e **avaliá-los** junto aos usuários. Isso permite revisar e refinar o projeto iterativamente até chegar a uma solução satisfatória.

4.3.5 Design Baseado em Cenários

O **design baseado em cenários** é um processo que utiliza diferentes tipos de cenários como representação básica e fundamental durante todas as atividades envolvidas na concepção de uma solução de IHC (Rosson e Carroll, 2002; Carroll, 1995). Um *cenário* é “simplesmente uma história sobre pessoas executando uma atividade” (Rosson e Carroll, 2002, p. 2). Como os cenários são geralmente escritos em linguagem natural, o seu uso motiva todos os interessados no sistema a participarem e contribuírem com as decisões de design, direta ou indiretamente.

Ao escrever, ler e revisar cenários, a equipe de design (incluindo representantes dos usuários) tem a oportunidade de discutir e analisar como as atividades dos usuários são afetadas pela tecnologia existente e como elas poderiam ser afetadas pelo sistema sendo desenvolvido. As histórias dos cenários estimulam a imaginação da equipe de design e encorajam a análise de caminhos alternativos. Por exemplo, perguntas do tipo “E se...” permitem à equipe de design imaginar outros caminhos para a história descrita no cenário, possivelmente contendo alguns elementos diferentes. Desse modo, cenários são uma ferramenta útil e barata para gerar e avaliar diversas ideias durante as atividades de design.

A Figura 4.8 ilustra as atividades propostas pelo design baseado em cenários. Basicamente, as atividades são: análise do problema, projeto de uma solução de IHC,

prototipação e avaliação da solução proposta. O diferencial desse processo está na forma como essas atividades são executadas. Esse processo inicia com a elaboração de cenários de problema, e continua com sucessivas análises e transformações de cenários de acordo com a atividade sendo executada. Apesar de as atividades serem apresentadas sequencialmente, o processo é iterativo, ou seja, sempre que necessário, a equipe de design pode revisar o que foi feito anteriormente.

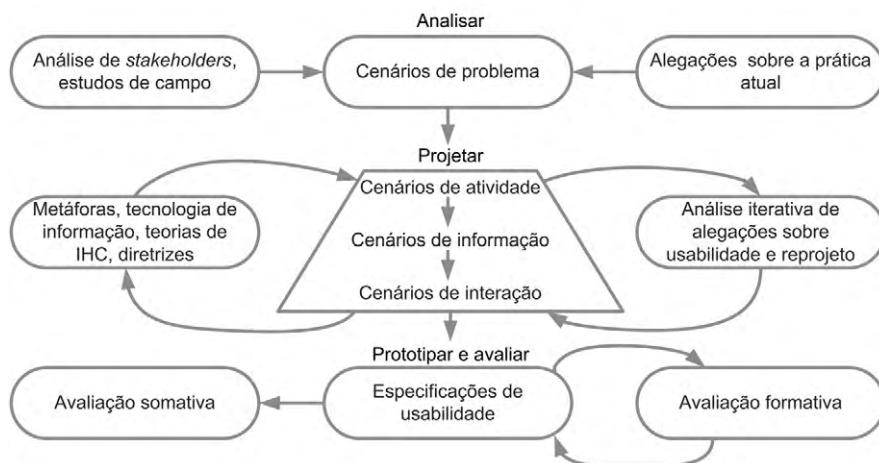


Figura 4.8 Atividades do design baseado em cenários.

Na análise do problema, a equipe de design estuda a situação atual junto aos interessados no sistema (*stakeholders*: clientes, usuários etc.). Com o conhecimento adquirido sobre a situação atual, a equipe de design deve formular **cenários de problemas** que cobrem características dos usuários, suas atividades típicas e críticas, os artefatos que eles utilizam e o contexto de uso (Rosson e Carroll, 2002). Uma vez que a compreensão da equipe de design sobre a situação atual tenha sido consolidada em cenários de problema, o próximo passo é elaborar uma solução de IHC adequada que resolva os problemas descritos.

Na atividade de projeto, a equipe de design deve explorar ideias para a solução de IHC elaborando três tipos de cenários: cenários de atividade, de informação e de interação. Um **cenário de atividade** é uma narrativa sobre as tarefas típicas e críticas que os usuários vão executar com ajuda do sistema. Eles concentram-se em relatar as funcionalidades do sistema, sem, no entanto, especificar ainda como os usuários vão utilizá-lo ou como deve ser a aparência do sistema. Um **cenário de informação** é uma elaboração de um cenário de atividade que descreve as informações fornecidas pelo sistema ao usuário durante a interação. Um **cenário de interação** especifica em

detalhes as ações do usuário e as respectivas respostas (*feedback*) do sistema necessárias para executar as tarefas apoiadas pelo sistema.

As ideias para a solução de IHC devem ser avaliadas continuamente durante o processo de design. Isso normalmente é realizado através de um protótipo que implementa ou demonstra partes da solução de IHC descritas em cenários. Os cenários também são responsáveis por guiar a avaliação, seja durante a concepção da solução de IHC ou depois que ela estiver pronta. Os cenários descrevem hipóteses sobre o uso da solução de IHC que permitem prever: os perfis dos usuários, seus objetivos, algumas tarefas que tentarão realizar para atingirem seus objetivos, algumas sequências de ações que tentarão realizar para concluir as tarefas escolhidas, as respectivas respostas do sistema e algumas possíveis reações dos usuários. Cenários são discutidos nas Seções 6.3 e 7.2, no contexto de análise e design de IHC, respectivamente.

4.3.6 Design Dirigido por Objetivos

O processo de design dirigido por objetivos orienta o designer a projetar uma solução de IHC criativa que apoie os usuários em atingirem seus objetivos (Cooper *et al.*, 2007). O diferencial desse processo é incentivar o designer a explorar as tecnologias disponíveis da melhor forma possível para oferecer aos usuários maneiras mais criativas, inovadoras e eficientes de alcançarem seus objetivos. Por exemplo, se a tecnologia permite sacar dinheiro de contas no banco em um caixa eletrônico que pode funcionar 24 horas durante toda a semana, por que limitar esse objetivo ao atendimento de um funcionário que trabalha seis horas em cinco dias na semana? Se a tecnologia permite enviar fotos em poucos minutos para familiares em locais bem distantes, como em outros países, por que esperar dias para as fotos chegarem ao destinatário através do correio tradicional? É claro que as tecnologias possuem várias limitações e não possuem as mesmas capacidades humanas. Contudo, o importante é explorar as possibilidades da tecnologia em favor dos usuários.

Segundo Cooper e seus colegas, alguns designers podem tentar projetar um sistema que permita ao usuário realizar as mesmas tarefas ou ações que ele realizava anteriormente. Dessa forma, a nova solução será mediana, pois sempre oferecerá o mesmo tipo de apoio ao usuário, limitado à forma como os usuários atingem seus objetivos atualmente. As novas soluções continuarão limitadas ao que as tecnologias anteriores já permitiam fazer, sem explorar novas possibilidades oferecidas pelas tecnologias atuais. Essa perspectiva de design abre pouco espaço para soluções criativas, capazes de explorar tecnologias antigas e novas de maneira inovadora para apoiar o usuário. Se por um lado é preciso estar ciente do que as pessoas sabem e estão acostumadas a fazer, por outro a nova solução não pode estar limitada a isso.

Como ser criativo e inovar sem estar limitado às tarefas executadas anteriormente pelos usuários? Para responder essa pergunta, primeiro é preciso diferenciar objetivos de tarefas ou ações do usuário. Cooper, Reimann e Cronin (2007, p. 15) definem um **objetivo** como sendo “uma expectativa de uma condição final, em que ações e tarefas são passos intermediários (em diferentes níveis de organização) que ajudam alguém a atingir um objetivo ou conjunto de objetivos”. Segundo essa definição, um objetivo seria o destino final, alcançado quando um usuário percorre certos caminhos definidos por tarefas e ações. Por exemplo, um objetivo do usuário pode ser comunicar algo a um colega. Ele pode fazer isso realizando diferentes tarefas, como, por exemplo, escrevendo uma carta convencional, enviando um e-mail, fazendo uma ligação telefônica, ou enviando uma mensagem de texto via telefone celular. Uma tarefa pode ser composta de outras tarefas mais simples. Por exemplo, a tarefa de escrever um e-mail pode ser composta pelas tarefas de digitar um texto e formatá-lo.

Os objetivos representam as motivações que levam o usuário a realizar suas tarefas. Conhecer esses objetivos permite compreender o significado das tarefas realizadas atualmente. Com isso, é possível repensar as tarefas com liberdade para imaginar novas possibilidades de atingir os objetivos do usuário, aproveitando ao máximo as tecnologias antigas e novas de forma criativa, inovadora e eficiente. Quando o design de IHC é dirigido pelos objetivos do usuário, é possível explorar a tecnologia para eliminar tarefas irrelevantes e aperfeiçoar as demais.

Por exemplo, pense no objetivo de comprar produtos em uma loja. Uma sequência de tarefas possível seria o caixa pegar cada produto, digitar seu valor na caixa registradora (ou obter a informação através da leitura de um código de barras), o cliente pagar e levar a mercadoria. Outra sequência possível de tarefas seria o próprio sistema identificar no carrinho do cliente o tipo e a quantidade dos produtos escolhidos (por exemplo, através de RFID — identificação por rádio frequência) e o cliente pagar e levar os produtos. Hoje em dia já é comum o próprio cliente informar a um sistema de comércio eletrônico quais produtos deseja comprar, como vai pagar, qual meio de transporte deve ser utilizado e onde os produtos devem ser entregues pela empresa.

Essas são apenas três formas diferentes de atingir o objetivo de comprar produtos, cada qual com uma sequência de tarefas diferente e com vantagens e desvantagens. Ao enunciar o objetivo do usuário, é possível explorar e comparar diferentes formas de atingi-lo. Assim, podemos empregar as tecnologias de maneira mais apropriada para apoiar e satisfazer os objetivos dos usuários.

O design dirigido por objetivos é um processo sistemático proposto para investigar e atender às necessidades e aos objetivos dos usuários, bem como atender aos

requisitos técnicos, do negócio e da organização. Esse processo é dividido em seis fases (Figura 4.9): pesquisar, modelar, definir requisitos, projetar, refiniar e manter.



Figura 4.9 Processo de design dirigido por objetivos (adaptado de Cooper *et al.*, 2007).

Na fase de **pesquisa**, o designer está interessado em conhecer o usuário, o domínio do sistema e o contexto de uso. Ele investiga comportamentos dos usuários que sugerem seus objetivos e motivações ao realizar suas atividades enquanto manipulam certos artefatos. O comportamento dos usuários pode estar associado a um papel ou função exercida, ou ainda corresponder a suas preferências pessoais.

A fase de **modelagem** tem por objetivo organizar e registrar o conhecimento adquirido na fase de pesquisa através da elaboração de modelos do usuário, domínio e contexto de uso. Os modelos são úteis para representar conceitos e relações entre eles, facilitando à equipe de design registrar, compreender, visualizar e discutir sobre o conhecimento adquirido na fase anterior. Sem a organização proporcionada por modelos, é possível que dados coletados na fase de pesquisa permaneçam inexplorados durante o processo de design por falta de atenção ou de um trabalho sistemático.

Na fase de **definição de requisitos**, o designer interpreta as informações coletadas e estruturadas nos modelos para definir os requisitos do usuário, do negócio e técnicos. Algumas vezes esses requisitos são conflitantes, e é preciso fazer concessões. Por exemplo, em um sistema de comércio eletrônico, alguns usuários podem estar interessados em comprar os produtos mais baratos, enquanto a empresa também deseja vender produtos mais caros. Como equilibrar isso? Uma possível solução seria expor tanto os produtos mais caros quanto os mais baratos em diferentes “vitrines” do sistema, e também permitir ao usuário consultar os produtos mais baratos de alguma maneira.

Na fase de **projeto conceitual** (*framework definition*), o designer concebe uma solução de interação e um esboço de interface pouco detalhado. Sua preocupação principal está na concepção da estrutura e do comportamento da interface. Na fase de **refinamento**, o foco é detalhar a solução de interface, definindo todas as características dos elementos de interface, tais como tamanho, cores e ícones. Nessa fase o designer verifica a coerência das tarefas percorrendo a interface. A solução detalhada pode ser avaliada junto aos usuários, e, caso seja necessário, ela é revisada. O resultado da fase de refinamento é uma documentação detalhada da solução de interação e de interface projetada.

Durante a implementação da solução projetada, podem surgir limitações técnicas que impeçam ou atrapalhem sua construção. Por isso, na última fase do processo, o designer tem a responsabilidade de **manter** a coerência da solução proposta enquanto acomoda as limitações técnicas imprevistas. A presença do designer nessa fase ainda é de extrema importância. Se o designer não estiver mais disponível durante a construção da solução, a equipe de desenvolvimento pode ter de tomar decisões de IHC sem o conhecimento necessário sobre usuários, objetivos e contexto de uso.

Cada fase do design dirigido por objetivos é iterativa. Podemos observar que não existe atividade exclusiva para avaliação nesse processo. A avaliação deve ser realizada durante cada atividade, principalmente nas fases de projeto e de refinamento da solução.

4.3.7 Design Centrado na Comunicação

O design centrado na comunicação (Barbosa *et al.*, 2004) tem como base teórica a engenharia semiótica (de Souza, 2005a), apresentada na Seção 3.8. Essa teoria compreende a interação humano-computador como um processo de comunicação entre o usuário e o designer do sistema, através da sua interface. Em outras palavras, a interface revela, durante o uso do sistema, a metacomunicação do designer (ou seja, as intenções de design e os princípios interativos). Quando o usuário tem acesso a essa metacomunicação, ele possui melhores condições de aprender e usar o sistema de forma produtiva, eficiente e criativa. A motivação principal do design centrado na comunicação é elaborar uma solução de IHC que transmita a metacomunicação do designer de forma eficiente e eficaz, ou seja, produzir um sistema interativo com alta comunicabilidade (conceito apresentado na Seção 2.2.3). Para isso, esse processo orienta o designer a se posicionar como um dos interlocutores das conversas que ocorrem durante a interação.

Para aumentar as chances de projetarmos uma interface que transmita adequadamente a metacomunicação do designer, a equipe de projeto deve definir o conteúdo dessa mensagem e compartilhá-la efetivamente entre seus membros. Em outras palavras, se a equipe de design não possui uma visão de design compartilhada consistente, ela dificilmente terá sucesso em comunicá-la aos usuários através da interface. O design centrado na comunicação parte do pressuposto que, se os designers conseguirem registrar a metacomunicação em um conjunto de artefatos e comunicá-la aos membros da equipe, eles terão melhores condições de comunicá-la aos usuários através da interface (Barbosa *et al.*, 2004).

Desse modo, o design centrado na comunicação busca representar a metacomunicação e promover uma compreensão compartilhada dela por todos os membros da

equipe de projeto. Essa compreensão deve ser registrada desde o início do processo de design, ainda na fase de análise do problema, e chegar até as atividades de implementação e testes do sistema. Isso ajuda a evitar e corrigir interpretações incompletas ou equivocadas, pois o compartilhamento da metacomunicação representa uma oportunidade para todos os membros da equipe contribuírem e revisarem o que foi produzido até o momento. Cada membro contribui com a sua perspectiva particular sobre o assunto em questão.

Mas onde os designers buscam insumos para construir a metacomunicação? Além do levantamento e da análise tradicionais dos objetivos, necessidades e preferências dos usuários, o design centrado na comunicação faz uso dos resultados de pesquisa sobre a construção da ajuda on-line de Silveira e colegas (Silveira *et al.*, 2004; Silveira, 2002).

A ajuda on-line é uma forma privilegiada de transmitir a metacomunicação ao usuário, pois fornece meios de transmitir direta e explicitamente as intenções de design e os princípios de interação. Silveira explora dúvidas comuns dos usuários para construir um sistema de ajuda on-line (veja Seção 7.5). Ela propõe coletar informações durante todo o processo de desenvolvimento para responder as perguntas que os usuários costumam fazer quando encontram problemas durante o uso. Por exemplo, a ajuda on-line deve buscar responder perguntas como “O que é isto?”, “Para que serve isto?” e “Como faço isto?” quando se referem a conceitos representados na interface. Essas perguntas são utilizadas para auxiliar na elaboração do conteúdo da metacomunicação e na sua comunicação aos usuários através da engenharia dos sistemas de signos da interface e da ajuda on-line.

O design centrado na comunicação leva adiante a proposta de Silveira para apoiar todas as atividades envolvidas no projeto de uma solução de IHC com alta comunicação, e não apenas a construção da ajuda on-line. Desse modo, o design centrado na comunicação propõe a elaboração da metacomunicação orientada por um conjunto de perguntas derivadas das dúvidas comuns dos usuários. Essas perguntas serão respondidas durante a atividade de análise da situação atual e o projeto da solução de IHC. Por exemplo, durante a atividade de análise, o designer deve procurar responder perguntas do tipo: “O que o usuário deseja ou desejaria fazer com o sistema?”, “Quem pode fazer isso?” e “O que deve ser feito antes disso?”. Já durante o projeto da solução, o designer deve buscar responder perguntas do tipo: “Como o usuário costuma atingir esse objetivo atualmente?”, “Como ele gostaria de fazer isso no futuro?”, “Que problemas podem ocorrer enquanto o usuário busca atingir esse objetivo?”, “Como solucionar tais problemas?” e “Como desfazer os resultados de uma ação indesejada?”. Em particular, a preocupação com as possíveis dúvidas dos usuários motiva a equipe

de design a tentar prever possíveis problemas durante o uso, a projetar formas de prevenção e recuperação desses problemas, bem como a comunicar sob demanda suas intenções de design e princípios interativos através da interface.

O design centrado na comunicação propõe três atividades: a análise do usuário, domínio e contexto de uso, o projeto de interação e interface, e a avaliação do que foi projetado (Figura 4.10). O diferencial desse processo consiste em nortear os esforços de design desde o início do processo pelas dúvidas que os usuários costumam ter durante a interação. Dessa forma, a solução de IHC é projetada para evitar o surgimento das dúvidas dos usuários e comunicar adequadamente informações necessárias para sanar dúvidas que eventualmente surjam.

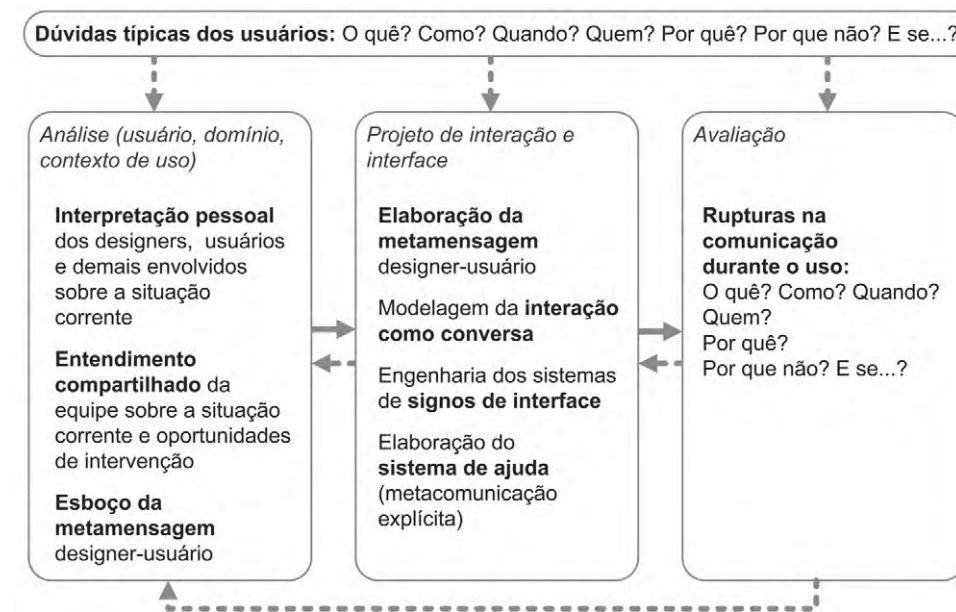


Figura 4.10 Design centrado na comunicação.

Embora um objetivo importante seja evitar que os usuários tenham dúvidas durante a interação, o design centrado na comunicação reconhece que é inevitável que diferentes usuários tenham dúvidas em alguns momentos, assim como é inevitável que mal-entendidos ocorram durante a conversação humana. Portanto, ele orienta o designer a projetar uma solução de IHC que comunique adequadamente não apenas as situações esperadas (que “dão certo”), mas também que ajude o usuário a evitar e se recuperar das rupturas de comunicação durante a interação (veja Seção 7.3.3).

O design centrado na comunicação diferencia a interface e a interação a ponto de propor o projeto de cada uma separadamente, mantendo a consistência e coerência entre elas. Recomenda que o projeto da interação seja realizado utilizando a MoLIC,

Linguagem para a Modelagem da Interação como Conversa (Barbosa e Paula, 2003; Silva e Barbosa, 2007). Para o projeto de interfaces gráficas, esse processo recomenda o uso de esboços de tela, mas não faz recomendações específicas para outros tipos de interface (via voz, gestos, realidade virtual etc.). O Capítulo 7 descreve as atividades de projeto de interação e interface desse processo de design, bem como os modelos e representações utilizados.

Os objetivos do usuário serão atingidos através de conversas entre usuário e o preposto do designer durante a sua interação com o sistema. Segundo essa interpretação, a interface representa uma linguagem e um meio de comunicação entre usuário e o preposto do designer, linguagem essa que define a forma e o conteúdo do que os interlocutores podem falar, em que ordem as falas podem ocorrer e quem pode falar em cada momento da conversa. Sendo assim, o designer deve projetar a conversa (a interação) e a forma de representá-la na interface. Em linha com Hoover, Rinderle e Fischer (1991), o design centrado na comunicação envolve modelos com diferentes focos e níveis de abstração e detalhes. Ele recomenda primeiro projetarmos a conversa usuário–designer em um modelo de interação, concentrando-nos nas trocas de turno de fala e nos tópicos e foco de cada turno de fala, e abstraindo-nos dos detalhes da interface. Em seguida, devemos projetar estrutura, *layout* e formatação da interface propriamente dita (isto é, projetar a expressão da conversa usuário–designer na interface).

Como sempre, a proposta de solução de IHC (interação e interface) deve ser avaliada para verificarmos se ela atende às necessidades dos usuários. No design centrado na comunicação, avaliamos se a metacomunicação foi enviada e recebida adequadamente. Se algum problema for encontrado, devemos rever o projeto de interação, o de interface, ou ambos. Durante as atividades de projeto e de avaliação, também pode ser necessário ampliar, refinar ou revisar a análise realizada anteriormente. Por isso, o design centrado na comunicação é bastante iterativo, com ciclos envolvendo as três atividades básicas de design: análise, síntese e avaliação, até que uma solução seja avaliada como satisfatória. O Capítulo 10 apresenta os métodos de inspeção semiótica e de avaliação de comunicabilidade, utilizados no design centrado na comunicação.

Os objetivos, as necessidades, as preferências e os pontos de vista dos usuários ganham importância em todas as atividades deste processo de design, pois desde o início a atenção da equipe está centrada na comunicação com os usuários. A definição compartilhada da metacomunicação que guia o projeto de IHC facilita construir um sistema interativo consistente, coerente e com alta comunicabilidade.

Vale observar que, embora os usuários e demais *stakeholders* participem intensamente das atividades de análise e avaliação ao longo de todo o projeto, cabe aos

designers elaborarem a metacomunicação e a solução de IHC com base no que observaram e ouviram durante essas atividades. Embora a participação dos usuários seja imprescindível para a qualidade e o sucesso da solução, eles não são *diretamente* codesigners da solução projetada. Em outras palavras, no design centrado na comunicação, a solução de IHC é projetada envolvendo os usuários e *para* eles, mas não *por* eles.

4.4 Integração das Atividades de IHC com Engenharia de Software

As áreas de IHC e de Engenharia de Software (ES) possuem diferentes perspectivas sobre o que é importante em um sistema interativo, sobre o que significa utilizá-lo e sobre como desenvolvê-lo. Cada área vem evoluindo historicamente por um caminho próprio e independente. Embora a preocupação com a qualidade de uso apareça desde os primeiros trabalhos sobre qualidade de software no final de década de 1970 (McCall *et al.*, 1977; Cavano e McCall, 1978; Boehm, 1981), a ES tem direcionado seus esforços para fatores de qualidade mais relacionados com engenharia — construção, instalação e manutenção — deixando para segundo plano a forma como os sistemas interativos serão utilizados.

Na **perspectiva de design centrada no sistema**, comum na área de Engenharia de Software, um sistema interativo é um artefato circunscrito e encapsulado por uma interface que recebe dados de entrada, processa esses dados com algum programa (codificado em software ou hardware) e retorna dados de saída (Figura 4.11). O que mais importa nessa perspectiva é aquilo que ocorre *dentro* do sistema. Tudo o que ocorre na fronteira ou fora dele, inclusive a própria interface, acaba recebendo pouca ou nenhuma atenção. O objetivo principal seria construir um sistema fidedigno (*dependable*) que seja capaz de processar adequadamente os dados de entrada e saída transmitidos através de uma interface bem definida. Os fatores de qualidade mais valorizados por essa perspectiva estão relacionados com a **construção** de um sistema interativo, como, por exemplo, disponibilidade, integridade, robustez, manutenibilidade e recuperabilidade (Avizienis *et al.*, 2004). Para atender a esses fatores de qualidade é preciso concentrar em conceitos fundamentais para construção de sistemas, tais como: arquitetura do sistema, estrutura de dados, mecanismos de persistência de dados, formas de comunicação entre sistemas, algoritmos, sistema operacional, linguagem de programação, ambiente de desenvolvimento, e assim por diante.

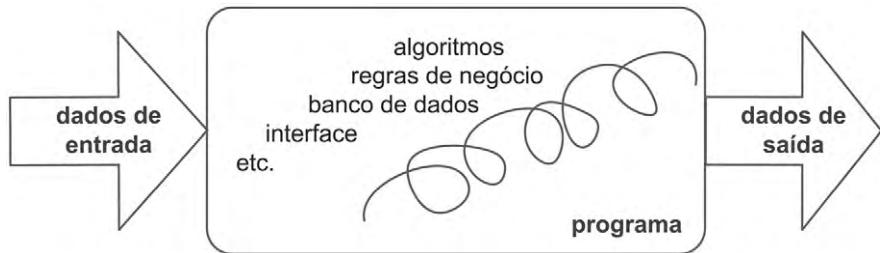


Figura 4.11 Perspectiva de design centrado no sistema.

A definição de uma interface permite ao engenheiro de software especificar a forma como um sistema irá interagir com o mundo externo. Tudo o que é possível solicitar ao sistema e receber dele será definido pela interface. Dessa maneira, o engenheiro de software geralmente abstrai o mundo externo ao construir o sistema, pois ele espera que o mundo se comunique “corretamente” com o sistema, conforme estabelecido pela interface. Abstrair o mundo externo pode funcionar bem quando estamos lidando com a comunicação entre sistemas, porque eles podem ser construídos para se comunicarem obedecendo a interfaces bem definidas. Contudo, a abstração do mundo externo costuma trazer problemas quando igualamos a interface com pessoas à interface com outros sistemas computacionais. A comunicação entre sistemas computacionais possui características bem diferentes da comunicação usuário-sistema. Quando o usuário entra em cena, as características humanas devem ser consideradas e endereçadas adequadamente durante a interação. Além disso, o contexto e o ambiente em que o usuário e o sistema estão inseridos também influenciam o uso do sistema e devem ser considerados.

Compreender e endereçar os problemas que ocorrem durante a interação usuário-sistema exigem conhecimentos além daqueles de base Matemática e Lógica que fundamentam a Engenharia de Software e a Computação como um todo. Exigem conhecimentos relacionados com características particulares das pessoas e das culturas em que estão inseridas. Um sistema interativo é construído para sempre executar um conjunto bem definido de instruções, que processam dados de entrada e saída. Diferente das máquinas, uma pessoa sente, pensa, interpreta, reinterpreta, é criativa, compartilha de uma cultura, vive em sociedade, muda de opinião, gostos, costumes, enfim, uma pessoa está sempre evoluindo. Não é possível prever ou controlar com exatidão a interpretação e o comportamento de alguém durante o uso de um sistema. De fato, tratar adequadamente o uso de sistemas interativos exige conhecimentos e esforços multidisciplinares muito além da Computação, incluindo áreas como Linguística, Psicologia, Sociologia, Ergonomia, Artes, dentre outras (Sharp *et al.*, 2007; Seção 1.4).

Como os conhecimentos matemáticos e lógicos que fundamentam a ES não dão conta das questões relacionadas ao uso que uma pessoa faz de sistemas interativos, a área de IHC vem se estabelecendo como área multidisciplinar que propõe uma perspectiva diferente para o desenvolvimento de sistemas interativos. Na perspectiva do design centrado no uso, comum a profissionais de IHC, conforme discutido no Capítulo 1, um sistema interativo é um artefato com o qual o usuário interage durante a realização de suas atividades em determinado contexto. O foco dessa perspectiva deixou de ser o que ocorre dentro do sistema e passou para aquilo que ocorre *fora* do sistema e *através* da sua interface. O mais importante nessa perspectiva é a forma como o usuário se apropria daquilo que o sistema pode oferecer em apoio aos seus objetivos em um determinado contexto. Assim, o objetivo do design centrado no uso é conceber (mais no sentido de projetar e avaliar do que implementar) um sistema interativo que sirva de apoio ao usuário na realização de suas atividades e no alcance dos seus objetivos. Para isso, é preciso trabalhar com conceitos relacionados ao uso do sistema, como, por exemplo, o contexto de uso, os objetivos do usuário, as formas como ele costuma alcançá-los, as características do usuário (formação, habilidades, prática na atividade sendo apoiada pelo sistema, cultura, gostos, preferências etc.) e possíveis formas de interação através da sua interface com usuário.

Desde sua origem, IHC tem sido proposta em contraste com o desenvolvimento centrado no sistema tipicamente praticado pela ES (Norman e Draper, 1986; Shneiderman, 1998; Sharp *et al.*, 2007). Para IHC, o uso que as pessoas vão fazer do sistema é o que deve guiar seu desenvolvimento. O usuário não deveria ser obrigado a adequar ao sistema sua forma de pensar, de realizar suas atividades, de trabalhar, de interagir com outras pessoas ou com instituições, e assim por diante. Na verdade, o sistema é que deveria ser construído de forma adequada ao usuário e suas necessidades e desejos.

As diferentes perspectivas de IHC e ES sobre o desenvolvimento de sistemas interativos deram origem a métodos, técnicas e processos próprios de cada área. Recentemente, alguns pesquisadores têm investigado a integração de métodos e técnicas de IHC em processos de desenvolvimento de software propostos em ES (Seffah *et al.*, 2005). As principais abordagens de integração de processos de IHC e ES são:

- definição de características de um processo de desenvolvimento que se preocupa com a qualidade de uso;
- definição de processos de IHC paralelos que devem ser incorporados aos processos propostos pela ES;
- indicação de pontos em processos propostos pela ES em que atividades e métodos de IHC podem ser inseridos.

Uma maneira de integrar as áreas de IHC e de ES é definir as características que um processo de desenvolvimento deve ter para tratar adequadamente a qualidade de uso. Gulliksen e seus colegas (2005) identificaram 12 princípios-chave que um processo de desenvolvimento deve ter para cuidar adequadamente da qualidade de uso. São eles:

- *foco no usuário*: os objetivos e as necessidades do usuário devem guiar o processo de desenvolvimento desde o início, para evitar que o desenvolvimento seja guiado pela tecnologia;
- *participação ativa do usuário*: representantes dos usuários devem participarativamente durante todo o processo de desenvolvimento;
- *desenvolvimento iterativo e incremental*: o desenvolvimento do sistema deve ser iterativo e incremental para permitir a avaliação e revisão das propostas de solução, bem como liberar logo para o usuário partes do sistema que já tenham sido desenvolvidas;
- *representações de design simples*: o resultado do design deve ser representado de forma que possa ser facilmente compreendido pelos usuários e demais envolvidos no processo de desenvolvimento;
- *prototipação*: protótipos em diferentes níveis de detalhes devem ser utilizados para visualizar e avaliar propostas de solução junto aos usuários;
- *avaliar o uso em contexto*: avaliar as propostas de solução considerando os critérios de qualidade de uso definidos como prioridade para o sistema em questão, sempre atento às reações dos usuários no contexto de uso;
- *atividade de design explícita e consciente*: o processo de desenvolvimento deve conter atividades dedicadas ao design da solução de interação e de interface com usuário;
- *atitude profissional*: o processo de desenvolvimento deve ser executado por uma equipe multidisciplinar;
- *defensor da qualidade de uso*: um profissional de IHC deve participar continuamente do processo de desenvolvimento com a responsabilidade de tomar as decisões necessárias para favorecer a qualidade de uso;
- *design holístico*: todos os aspectos que influenciam o uso devem ser considerados em conjunto durante o processo de desenvolvimento;
- *customização do processo*: o processo de desenvolvimento deve ser adaptado a cada organização;
- *atitude centrada no usuário*: todos os envolvidos no processo de desenvolvimento devem estar cientes de e concordar com a importância da qualidade de uso e da participação ativa do usuário durante o processo.

Esses princípios-chave são comuns em processos de design de IHC, porém não costumam ser considerados em processos de desenvolvimento propostos na ES. Certamente um profissional de IHC está mais bem preparado para colocar em prática esses princípios dentro de processos de desenvolvimento, pois são necessários muitos conhecimentos de IHC que engenheiros de software geralmente não possuem na profundidade necessária. Além disso, é importante manter a perspectiva do usuário e do sistema durante o desenvolvimento para que as qualidades de uso e de construção recebam a atenção e o cuidado desejados.

Outra forma de integrar IHC e ES é através da execução de processos de IHC paralelos a processos de ES (Seffah *et al.*, 2005). O ciclo de vida em estrela, o design dirigido por objetivos e o design centrado na comunicação, por exemplo, poderiam ser executados em paralelo a processos propostos pela ES. Nesse caso, é necessário manter a consistência entre os resultados das atividades de cada processo. Se considerássemos a interface com usuário como algo completamente separado e isolado do restante do sistema, essa integração seria fácil de ser resolvida. Entretanto, as decisões de design de interação e de interface também afetam as funcionalidades do sistema e o que está por trás da interface com usuário. Por exemplo, John e seus colegas (2004) mostram como questões de IHC estão relacionadas com a arquitetura do software e não somente a uma camada ou “casca” independente.

O terceiro tipo de abordagem para a integração de IHC e ES aponta as atividades nos processos da ES em que métodos e práticas de IHC podem ser aplicados. Ferre e seus colegas (Ferre, 2003; Ferre, *et al.*, 2004; 2005) realizaram uma revisão bibliográfica sobre os métodos e as práticas de IHC e definiram onde podem ser utilizados em um processo genérico de desenvolvimento de software. A Figura 4.12 apresenta o mapeamento das atividades de IHC em atividades de um processo genérico de desenvolvimento de software da ES que eles propuseram. Identificar onde os conhecimentos de IHC podem ser empregados num processo de desenvolvimento representa um passo importante para a ampla utilização desses conhecimentos na prática.

Na elicitação de requisitos em um processo de ES, devem ser coletadas informações sobre os usuários, seus objetivos, como costumam atingi-los (suas tarefas) e sobre o contexto de uso. A intenção dessa atividade é coletar dados sobre os elementos envolvidos durante o uso, que vão além da interface com usuário.

A interpretação desses dados é realizada durante a análise de requisitos em um processo de ES. Durante essa atividade, o engenheiro de requisitos costuma representar informações e tomar decisões que influenciam ou dizem respeito à interação e à interface com usuário. Por exemplo, quando ele define em um cenário ou em um caso de uso a sequência de ações que deve ser executada no sistema para o usuário

atingir um objetivo, ele toma decisões relacionadas com a interação e com a interface com usuário. De forma mais explícita, ele pode utilizar protótipos (que incluem a interface) para representar suas ideias sobre o que o sistema deve fazer e como ele vai apoiar o usuário durante a análise de requisitos. Essas decisões sobre a interação e a interface com usuário deveriam ser resultado de um projeto cuidadoso sob a responsabilidade de um profissional com conhecimentos de IHC, e não um subproduto das representações utilizadas durante a análise de requisitos. Os requisitos de IHC e as metas de design devem ser definidos com os requisitos do sistema durante a atividade de especificação de requisitos em um processo de ES. Desse modo, idealmente o design de IHC deveria ser iniciado durante a especificação de requisitos, mas sob a responsabilidade de um profissional de IHC. E mesmo após a definição da interação e da interface, e durante o projeto e a implementação do sistema, pode ser necessário também rever o projeto de IHC quando surgir alguma dificuldade para construir a solução projetada ou quando algo não tiver sido completamente especificado.

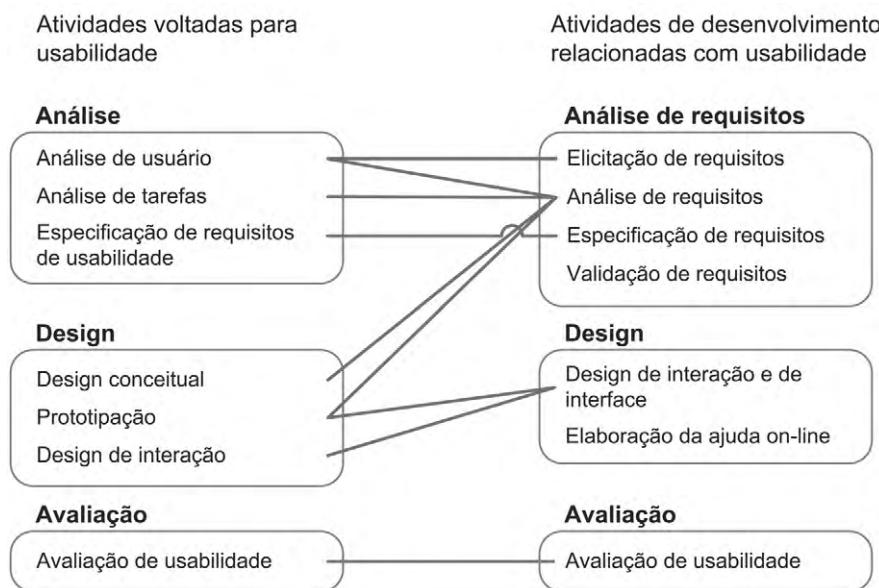


Figura 4.12 Mapeamento entre atividades de IHC e de ES adaptado de Ferre, 2003).

A avaliação do sistema deve ir além da validação de requisitos do sistema e testes do sistema, e englobar também uma avaliação de IHC da solução proposta, considerando os critérios de qualidade de uso definidos ao longo do processo de design e desenvolvimento. Sempre que possível, representantes dos usuários devem participar da avaliação de IHC do sistema para que a equipe de desenvolvimento possa perce-

ber os problemas que ocorrem durante o uso e coletar a opinião dos usuários sobre o sistema.

Como em qualquer equipe multidisciplinar, quando engenheiros de software e profissionais de IHC trabalham em conjunto, podem surgir problemas de comunicação, colaboração e coordenação. Parte do trabalho desses profissionais é solucionar o mesmo problema: definir a interação e a interface com usuário do sistema interativo sendo desenvolvido. Entretanto, eles possuem formações distintas, diferentes vocabulários, objetivos, perspectivas e focos na resolução desse mesmo problema. Portanto, é importante que os engenheiros de software e os profissionais de IHC reconheçam e valorizem o conhecimento, as preocupações e o trabalho uns dos outros. As atividades desses profissionais inevitavelmente influenciam umas as outras e, desse modo, devem estar bem coordenadas para produzirem um resultado consistente e coerente. A decisão de como será a interação e a interface deve ser tomada em conjunto pelos diferentes profissionais que compõem a equipe de design, sempre ponderando os diferentes pontos de vista e as questões levantadas por todos os membros da equipe de design.

O desenvolvimento de um sistema interativo exige muitos conhecimentos para cuidar adequadamente da sua construção e do seu uso, e envolve várias atividades realizadas em perspectivas diferentes do mesmo problema. Sendo assim, dificilmente um profissional será capaz de cuidar sozinho desses dois interesses, seja porque ele não possui os recursos necessários para adquirir conhecimento das duas áreas na profundidade e amplitude exigidas, seja porque ele não consegue articular ao mesmo tempo dois interesses tão distintos.

4.5 Métodos Ágeis e IHC

Os métodos ágeis de desenvolvimento de software, como o eXtreme Programming (Beck, 2000) e Scrum (Schwaber e Beedle, 2002), podem ser interessantes para IHC porque buscam colaborar com o cliente (*customer*) através de pequenos ciclos de desenvolvimento de forma iterativa e incremental, para obter retorno (*feedback*) do cliente e corrigir o rumo do processo de desenvolvimento (Armitage, 2004; Blomkvist, 2005). Contudo, ainda carecem de cuidado adequado em relação à qualidade de uso. Segundo Armitage (2004, p. 18), “a comunidade de métodos ágeis raramente menciona os usuários ou a interface com usuário como um todo, o que significa que ou eles negligenciam a experiência de uso, ou estão focando projetos com menor necessidade de uma experiência de uso sofisticada”.

Quando se trata de métodos ágeis, nem sempre existe uma distinção entre clientes e usuários do sistema sendo desenvolvido. Em IHC, é fundamental fazer essa dis-

tinção. Os clientes (que não os usuários) possuem apenas uma visão limitada e frequentemente equivocada das atividades dos usuários propriamente ditos.

Quando clientes ou usuários são envolvidos nos processos de desenvolvimento ágeis, existe um grande risco de eles serem os responsáveis por especificar os requisitos do sistema e por projetar a interação e a interface (Jokela e Abrahamsson, 2004). Apesar de conhecerem bem o domínio, eles podem não ter os conhecimentos necessários sobre a tecnologia e sobre design para realizarem essas atividades. Além disso, podem ter uma visão restrita do domínio, limitada pela sua atuação em apenas parte do processo que precisa ser apoiado pelo sistema.

O desenvolvimento incremental proposto pelos métodos ágeis exige modificações na interface a cada incremento construído. Isso dificulta a manutenção da consistência e coerência da interação e da interface, afetando direta e significativamente a qualidade de uso. Quando se trata de qualidades relacionadas com a construção do sistema, os testes automáticos do sistema são uma ferramenta adequada e eficiente para manter a qualidade do código quando um novo incremento for desenvolvido. Entretanto, esses tipos de testes não são adequados para avaliar a qualidade de uso (Blomkvist, 2005). Mesmo em um processo de desenvolvimento ágil, ainda se faz necessário avaliar a interação e a interface de acordo com os critérios de qualidade de uso, com a participação de usuários reais.

Blomkvist (2005) faz algumas sugestões concretas para integrar métodos e práticas de IHC em processos de desenvolvimento ágil:

- o objetivo principal é entregar ao cliente software que *funcione* e que seja *usável*. Atividades de IHC são importantes, mas não podem consumir muito tempo. Devemos equilibrar o tempo necessário para entregar um sistema que funcione com a qualidade de uso oferecida;
- é comum existir a necessidade de *priorizar as funcionalidades* que o sistema deve possuir para permanecer dentro do prazo disponível. Os usuários indicam de quais funcionalidades precisam, mas o designer de IHC deve auxiliá-los nessa decisão, para melhor atender aos seus objetivos e promover uma alta qualidade de uso;
- *envolverativamente os usuários* e não somente os clientes em todas as fases do desenvolvimento. O designer deve buscar informações sobre o contexto de uso, e não apenas consultar os usuários e clientes no ambiente de desenvolvimento;
- *o designer de IHC deve ser responsável* pelas decisões relacionadas com a qualidade de uso;

- é necessário *realizar avaliações de IHC* durante diferentes estágios do ciclo de desenvolvimento. Essas avaliações não podem ser executadas com a mesma frequência que testes automáticos de funcionalidades. Entretanto, sempre que possível, devemos executar avaliações de IHC mais simples e rápidas, idealmente com a participação dos usuários;
- devemos realizar uma *análise da situação atual* mais abrangente e rica em contexto de uso do que as histórias de uso (*user stories*) e os casos de uso (*use cases*) amplamente utilizados em métodos ágeis.

Vimos neste capítulo que o design de IHC é um processo iterativo que revisa e refina interpretações sobre uma situação para realizar uma intervenção na forma de um sistema computacional interativo. De modo geral, as atividades de um processo de design de IHC envolvem a análise da situação atual, a identificação do que deve ou pode ser melhorado, a elaboração e a avaliação de uma intervenção nessa situação. Os processos de design de IHC organizam essas atividades de diferentes maneiras, sugerem ou prescrevem os artefatos consumidos e produzidos em cada uma delas, e orientam como cada atividade deve ser executada.

O restante deste livro apresenta algumas técnicas e representações utilizadas ao longo do processo de design. O Capítulo 5 descreve técnicas de análise, o Capítulo 6 descreve representações que organizam o espaço de problema, o Capítulo 7 descreve a atividade de design de IHC, o Capítulo 8 apresenta diretrizes e padrões de IHC, e os Capítulos 9 e 10 descrevem a atividade e os métodos de avaliação de IHC.

Atividades

1. *O que é design.* Escolha uma situação cotidiana em que é preciso realizar uma atividade de design explorando a criatividade. Por exemplo, comprar uma roupa ou calçado, preparar uma refeição ou planejar as férias. Analise a situação escolhida, identificando o que geralmente é feito na:
 - análise da situação atual;
 - definição das necessidades e oportunidades de intervenção (i.e., do que é possível melhorar na situação analisada);
 - proposta de uma intervenção;
 - avaliação da intervenção.Faça em seguida a mesma análise, visando construir um sistema que permite aos usuários atingir os mesmos objetivos.
2. *Processo de design de IHC.* Investigue um processo de design de IHC de um sistema interativo. Escolha um sistema a cuja documentação ou equipe de design você tenha acesso direto, ou ainda um sistema de software livre que disponibilize

na Internet o material gerado durante seu processo de design de IHC. Por exemplo, investigue o processo de design do OpenOffice, do Mozilla Thunderbird, do KDE ou do GNOME. Identifique, descreva e ilustre:

- as atividades executadas;
 - os artefatos consumidos e produzidos em cada atividade;
 - algumas decisões de design tomadas em cada atividade.
3. *Processos de design de IHC.* Discuta as principais diferenças entre os processos de design de IHC descritos neste capítulo. Identifique características dos projetos que ajudam a tomar decisões sobre quais processos podem ou devem ser adotados.
 4. *Conhecimento de IHC envolvido nos processos de design.* Identifique quais conhecimentos de IHC precisam ser adquiridos por equipes de desenvolvimento para desempenhar atividades de IHC voltadas à qualidade de uso do sistema sendo projetado.
 5. *Integração das atividades de IHC com engenharia de software.* Investigue alguma experiência de integrar atividades de IHC com atividades de ES, seja em um processo ágil ou não. Você pode conversar com colegas que tenham participado de iniciativas desse tipo ou ler relatos de iniciativas do gênero. Analise e discuta os pontos positivos e negativos da experiência de integração investigada.
 6. *Inserção de atividades de IHC em processos de desenvolvimento de software.* Considerando um processo de desenvolvimento de software de que você tenha participado, identifique os pontos desse processo em que uma ou mais atividades de IHC voltadas à qualidade de uso poderiam ser realizadas.

5

Identificação de Necessidades dos Usuários e Requisitos de IHC

Objetivos do Capítulo

- Caracterizar o espaço de análise no processo de design de IHC.
 - Descrever o planejamento da coleta de dados de análise em IHC.
 - Discutir os aspectos éticos de pesquisas envolvendo pessoas.
 - Apresentar técnicas de investigação e análise comumente utilizadas: entrevistas, questionários, grupos de foco, *brainstorming*, classificação de cartões, estudos de campo e investigação contextual.
-

Este capítulo descreve os tipos de dados coletados durante a análise da situação atual, as fontes de informação que fornecem esses dados e os cuidados éticos envolvidos na captura dos dados e em pesquisas que envolvem pessoas. Apresenta ainda algumas técnicas de investigação e análise visando entender as necessidades dos usuários e definir os requisitos de IHC de um sistema interativo (Hackos e Redish, 1998; Courage e Baxter, 2005; Beyer e Holtzblatt, 1998).

5.1 Introdução

Conforme visto no Capítulo 4, a atividade de análise envolve uma pesquisa inicial da situação atual para identificar necessidades dos usuários e oportunidades de melhoria, a fim de determinar as características do produto de design como proposta de intervenção. Nessa atividade, devemos coletar requisitos de uma variedade de fontes (e.g., usuários finais, gerentes da empresa, clientes, instrutores, técnicos de suporte ou atendimento ao usuário) e utilizar essa informação para determinar que funcionalidades devem ser incluídas no produto, que tecnologias devem ser utilizadas, que fatores devem ser privilegiados, que tarefas devem ser apoiadas e por quê.

O principal objetivo da atividade de análise é identificar os requisitos dos usuários e as metas de design de IHC. Os **requisitos do usuário** se referem tanto aos objetivos dos usuários que o produto deve apoiar, como características e atributos que um produto deve ter ou de que maneira deve se comportar, do ponto de vista do usuário (Courage e Baxter, 2005). Tais requisitos incluem desde as funcionalidades de que os usuários precisam até critérios de qualidade de IHC que devem ser satisfeitos para que o produto de design seja considerado bem-sucedido.

O principal erro cometido por uma equipe de design é prescindir do estudo ou pesquisa inicial para a coleta de dados e prosseguir diretamente para realizar a análise com dados incompletos, inválidos, corrompidos ou pouco confiáveis (Courage e Baxter, 2005). Não podemos simplesmente pressupor que os usuários interagem com um produto de uma certa maneira, e não devemos confiar em dados que não tenham sido obtidos por pesquisas cuidadosamente conduzidas e documentadas. Mesmo que alguém tenha conversado com os usuários, essa pessoa pode não ter conhecimento e experiência suficiente sobre levantamento de dados para fazer um relato confiável e sem muitos erros de interpretação.

Um outro problema se refere ao termo “requisitos”. Muitas vezes é utilizado sem fazer uma distinção entre diferentes tipos de informação, tais como funcionalidades, atributos, restrições e expectativas. E nem sempre discrimina o grau de importância de cada informação. Por exemplo, é importante fazer uma distinção entre informações obrigatórias oriundas de regras de negócio, definições de processos e normas ou

restrições tecnológicas, e informações desejáveis e, portanto, passíveis de negociação, adaptações ou até mesmo descarte.

Sharp e coautoras destacam quatro pontos principais envolvidos na coleta de dados (Sharp *et al.*, 2007): definição dos objetivos da coleta de dados, relacionamento com participantes, triangulação e estudos-piloto.

A **definição dos objetivos** envolve identificar as razões para coletarmos dados. Podemos querer identificar como a tecnologia se encaixa no quotidiano de um grupo de pessoas; quais dificuldades elas encontram no seu dia a dia que podem ser reduzidas com a introdução de novas tecnologias; qual dentre duas ou mais alternativas de design melhor satisfaz os desejos de uma classe de usuários, entre outras questões. Os **objetivos** da coleta de dados determinam quais dados devem ser coletados e quais técnicas de coleta de dados podem ser utilizadas. Portanto, o primeiro passo para a coleta de dados é definir clara e concisamente os seus objetivos.

Tendo definido os objetivos da coleta de dados, os participantes que fornecerão os dados devem ser informados sobre esses objetivos e consentir com a sua coleta, com as condições de privacidade e anonimato previstas, com a forma como os dados serão utilizados, por quem e para quê. Esse esclarecimento ajuda a formar um **relacionamento profissional** entre as partes, bem como assegurar aos participantes o uso adequado das informações que eles forneçam. Em geral, a autorização dos participantes é obtida através da assinatura de um formulário de consentimento. Vale observar que, quando já existe um contrato envolvendo os participantes e as pessoas que coletam os dados (por exemplo, quando os participantes são empregados de uma empresa que contrata consultores para fazer o levantamento de requisitos), o formulário de consentimento pode não ser necessário. A Seção 5.4 discute os aspectos éticos da pesquisa com seres humanos e apresenta um exemplo de termo de consentimento.

A **triangulação** é uma estratégia de utilizar mais do que uma técnica de coleta ou análise de dados para obter diferentes perspectivas e confirmar as descobertas, permitindo obter resultados mais rigorosos e válidos. Por exemplo, uma estratégia de triangulação envolve utilizar observação para entender o contexto de realização das tarefas, realizar entrevistas para endereçar grupos de usuários específicos e distribuir questionários para alcançar uma população mais ampla, além de realizar grupos de foco para obter uma visão de consenso.

Um **estudo-piloto** é uma pequena prévia do estudo principal, com o objetivo de assegurar que o estudo é viável e permitirá coletar os dados desejados e realizar as análises planejadas. O estudo-piloto permite avaliar o material elaborado, como, por exemplo, avaliar se as perguntas de uma entrevista ou de um questionário estão

confusas. Caso o acesso à população-alvo seja limitado, podemos pedir para pessoas de perfil semelhante ou mesmo colegas participarem do estudo-piloto. É importante observar que qualquer pessoa envolvida num estudo-piloto não deve estar envolvida no estudo principal, pois essas pessoas saberão mais sobre o estudo e poderão distorcer os resultados.

5.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas nos usuários “melhores” ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quanto bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e discontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;

- *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;
- *atitudes e valores*: preferências de produto, medo de tecnologia etc. O usuário costuma assumir riscos e explorar novas formas de fazer o mesmo trabalho? Ou evita novas experiências, preferindo caminhos já percorridos e testados? Ou prefere que alguém lhes mostre cada passo de uma nova tarefa sendo aprendida?;
- *conhecimento do domínio*: o que e quanto o usuário conhece sobre o assunto em questão? É especialista? É esperado que se torne um especialista?;
- *objetivos*: quais são os principais objetivos dos usuário? Como eles são alcançados atualmente?;
- *tarefas*: quais são as tarefas do usuário que precisam ser apoiadas? Quais dessas são consideradas primárias, e quais são secundárias? Há quanto tempo realiza essas tarefas? São tarefas frequentes ou infrequentes? São tarefas inovadoras? Que experiência ele possui em tarefas semelhantes?;
- *gravidade dos erros*: em geral, as possíveis consequências dos erros de um usuário;
- *motivação para o trabalho*: o usuário se limita a cumprir a carga horária ou trabalha além do expediente, por prazer? Gosta da interação social no local de trabalho? Tem ambição de ser promovido?;
- *idiomas e jargões*: que idiomas o usuário conhece e utiliza fluentemente? Ele possui um jargão profissional particular, um vocabulário próprio da empresa, da sua atividade ou de algum grupo social relevante para o seu projeto?

À medida que conduzimos atividades de levantamento de requisitos, coletamos informações para (re)alimentar diversos artefatos utilizados na análise de IHC, tais como: perfis de usuários, personas, cenários e modelos de tarefa. Esses artefatos são apresentados no Capítulo 6. A próxima seção enumera diversas fontes de informação que podem fornecer os dados necessários ao projeto.

5.3 De Quem Coletar Dados?

Um aspecto importante da coleta de dados é definir quem fornecerá qual tipo de informação. Ao coletar dados sobre os usuários do sistema, é essencial encontrar fontes confiáveis, relevantes e representativas dos usuários e do seu trabalho. Caso contrário, não apenas os dados coletados serão de pouca utilidade, como também podem prejudicar seu produto, sua credibilidade e a credibilidade das atividades de IHC em geral.

O termo “usuário” geralmente diz respeito aos usuários finais, aqueles que são ou serão usuários diretos do seu produto, sejam **primários**, que utilizam o produto regularmente, ou **secundários**, que o utilizam ocasionalmente, por exemplo, em atividades de configuração eventuais. Além deles, pode ser importante traçar o perfil de outras partes interessadas (**stakeholders**), que não utilizam o produto diretamente mas são afetados pelo seu uso, como, por exemplo, pessoas que devem receber informações ou artefatos resultantes do uso do produto. Note que o termo *stakeholders* costuma se aplicar a todas as partes interessadas, incluindo os próprios usuários. Além disso, um único *stakeholder* pode exercer diversos papéis num sistema ou organização, os quais podem ter necessidades contraditórias.

Para identificar as partes interessadas que podem fornecer informações relevantes ao projeto de um sistema, devemos descobrir: quem utilizará o sistema? Quem será afetado por ele? Quem é responsável por decidir quais objetivos o sistema deve apoiar e quais funcionalidades ele deve ter? Quem definiu os processos a serem apoiados pelo sistema?

Caso o projeto em questão seja de melhoria ou expansão de um sistema existente, é importante conhecer também: quem utiliza o sistema atualmente? Além desses, quem passará a utilizá-lo? Quem são os usuários satisfeitos com o sistema? E quem são os insatisfeitos? Quem concebeu o sistema? Quem preparou a documentação do sistema? Quem dá treinamento aos usuários? Quem dá suporte aos usuários? Quem faz a manutenção do sistema? Quem projetou o sistema?

Para escolher uma técnica de coleta de dados, é necessário identificar o tipo de acesso a cada fonte de informação. A disponibilidade e localização das pessoas restringem o tipo de técnica de coleta de dados que pode ser utilizada. Algumas pessoas podem ter saído da empresa onde o sistema é utilizado e se tornado inacessíveis; outras podem ter sido promovidas ou deslocadas para outros setores; e ainda outras podem estar envolvidas em outras atividades e não ter tempo hábil para participar dessa etapa do projeto. Pessoas dispersas geograficamente também restringem o tipo de técnica de coleta de dados a ser utilizada. Por exemplo, é muito difícil realizar um grupo de foco à distância, mesmo com as tecnologias de videoconferência disponíveis hoje em dia.

Antes de começar a trabalhar com um usuário sequer, precisamos entender o domínio em que estamos trabalhando. Quando o produto já é conhecido, Beyer e Holtzblatt (1998) sugerem identificar necessidades que ainda não foram reconhecidas. Quando se trata de uma melhoria no produto (*upgrade*), os desafios são entender as razões das solicitações de melhoria e projetar uma solução que satisfaça a necessidade mantendo a prática de trabalho coerente e preservando a integridade do design do



sistema. É preciso examinar toda a prática de trabalho para entender de que maneira a mudança afeta o trabalho como um todo, e olhar em detalhes o uso de ferramentas para ver quais mecanismos de interação funcionam, quais atrapalham os usuários e quais podem ser reaproveitados para outras situações ou solicitações de mudança.

Já ao endereçar um novo domínio de trabalho, é importante definir o trabalho que o novo produto substituirá, e estudar esse trabalho para aprender o que é importante e como ele é estruturado, de modo a facilitar a transição para o novo produto. Para isso é importante coletar informações sobre as intenções dos usuários e como são realizadas utilizando as ferramentas atuais. O resultado de projetar um sistema novo é definir novas formas de trabalhar e os sistemas que as apoiam. Além disso, quando uma nova tecnologia está em jogo, é importante buscar analogias com as tecnologias existentes e como elas são utilizadas.

Podemos buscar dados que nos ajudem a aprender sobre o produto através de diferentes fontes, tais como: *feedback* dos usuários, arquivos de log, análise competitiva e pesquisa em geral (Courage e Baxter, 2005).

Se estamos trabalhando com um produto que já possui uma versão em produção e a empresa possui um grupo de suporte aos usuários, podemos aprender bastante sobre o produto conversando com esse grupo. Caso tenhamos um registro do **feedback dos usuários**, podemos precisar conduzir entrevistas ou estudos de campo junto aos usuários para entender melhor as questões levantadas.

Embora os **arquivos de log** indiquem caminhos que os usuários percorreram durante a interação com a aplicação, eles possuem diversas limitações quanto ao que pode ser capturado. Na Web, nem sempre há uma identificação única de cada usuário. Além disso, a funcionalidade de *cache* no navegador e o uso do botão de voltar podem deixar lacunas no registro. Examinando apenas o log, não é possível inferir as razões pelas quais o usuário demorou o tempo registrado numa página Web ou num módulo de um sistema tradicional. Por exemplo, se um usuário desviar sua atenção do sistema para atender um telefonema, o tempo de utilização registrado para o módulo correspondente não corresponderá ao uso real do sistema. E quase sempre é difícil ou até mesmo impossível inferir se um usuário atingiu seu objetivo. Sendo assim, a análise de arquivos de log deve ser complementada por outras técnicas. Além disso, ao analisarmos o tempo despendido numa parte do sistema, é melhor utilizar o valor mediano em vez do valor médio, pois este é mais suscetível a situações extremas incomuns.

A **análise competitiva** pode ser uma forma eficiente de obter vantagem sobre seus competidores. Além de examinarmos os competidores diretos, devemos também

analisar os produtos que os substituem ou complementam.¹ Esses produtos podem ou não competir diretamente com o seu, mas podem ter características semelhantes a ele e devem ser estudados para aprender seus pontos fortes e fracos, conhecer o perfil de seus usuários e clientes, a disponibilidade do produto, suas características e funcionalidades únicas, sua reputação e seus requisitos de hardware e software. Uma análise competitiva voltada para IHC vai além da abrangência das funcionalidades do sistema, e se concentra em aspectos da experiência do usuário, como estilo e características da interface com usuário, estrutura das tarefas, terminologia, satisfação dos usuários e qualidade de uso em geral. O produto de uma análise competitiva geralmente é uma tabela comparativa do seu produto com os dos seus competidores, que pode ser consultada e atualizada ao longo do processo de desenvolvimento.

A documentação de **processos e normas** também é um insumo importante para a análise, pois define restrições sobre o que o usuário poderá ou não fazer através do sistema, e às vezes até como ele poderá utilizá-lo.

Para conhecermos um produto, também devemos utilizá-lo. Em geral, a equipe de design está cercada por pessoas que conhecem o domínio e o produto. Devemos encontrar as pessoas que apoiam o produto atual e ler seus relatórios de problemas e acompanhamento do uso, bem como as pessoas que escrevem o conteúdo técnico da empresa e que elaboram os manuais, a ajuda on-line e o material de treinamento. Tudo o que for difícil de documentar pode ser resultado de um produto complicado demais para explicar.

É importante definir uma visão inicial do que está em jogo: quem são os usuários, clientes e demais partes interessadas; quais seus objetivos e quais tarefas realizam para atingi-los. Essa visão permitirá escolher as técnicas de análise utilizadas ao longo do projeto.

5.4 Aspectos Éticos de Pesquisas Envolvendo Pessoas

Muitas profissões possuem códigos de ética que regulam seu exercício. Em geral, os códigos de ética recebem maior destaque em profissões que atuam diretamente sobre as pessoas, como na área de saúde, por exemplo. A Computação também possui uma forte preocupação com a ética em suas pesquisas e intervenções (Johnson, 2001). Algumas associações de profissionais da Computação, como a ACM e a IEEE, possuem códigos de ética que orientam o trabalho dos seus associados. No código de ética da ACM (1992), podemos destacar os seguintes cuidados éticos (ou deveres morais):

1 É importante observar que algumas empresas de software proíbem, na sua licença, a condução de análise competitiva com o seu produto. Examine a licença de qualquer produto para se certificar de que ela não será violada por esse tipo de análise.

evitar causar danos ou consequências negativas aos outros, tais como perda de informação, perda de bens, danos a propriedades, ou impactos ambientais indesejados; respeitar a privacidade dos outros; e honrar a confidencialidade de informações a que tivermos acesso. No código de ética da IEEE (2006), podemos destacar o seguinte cuidado ético: evitar prejudicar ou causar dano a outras pessoas, seus bens, reputação ou emprego. No Brasil, apesar de a Sociedade Brasileira de Computação ainda não ter um código de ética, os currículos de referência da área abordam o tema.

É de responsabilidade da equipe de design proteger o bem-estar físico e psicológico dos participantes de qualquer estudo, pesquisa ou análise realizada (Johnson, 2001). No Brasil, a Resolução nº 196/96 do Conselho Nacional de Saúde (1996) regulamenta as pesquisas *científicas* envolvendo pessoas, em qualquer área do conhecimento. Apesar de essa resolução não se aplicar à execução de métodos de avaliação com objetivos *técnicos*, suas recomendações são muito úteis para orientar os avaliadores no cuidado ético durante seu trabalho. Segundo essa resolução, esses cuidados envolvem considerar os seguintes princípios (p. 2):

- **princípio da autonomia**, que envolve o *consentimento livre e esclarecido dos indivíduos* e a proteção a grupos vulneráveis e aos legalmente incapazes, tais como: menores de idade, alunos ou subordinados. Nesse sentido, a pesquisa envolvendo seres humanos deverá sempre tratá-los com dignidade, respeitá-los em sua autonomia e defendê-los em sua vulnerabilidade;
- **princípio da beneficência**, que envolve a *ponderação entre riscos e benefícios*, tanto atuais como potenciais, individuais ou coletivos, comprometendo-se com o máximo de benefícios e o mínimo de danos e riscos. Os danos podem ocorrer na dimensão física, psíquica, moral, intelectual, social, cultural ou espiritual do ser humano, em qualquer fase da pesquisa ou depois dela;
- **princípio da não maleficência**, que envolve a *garantia de evitar danos prevíveis* relacionados à pesquisa, tanto os imediatos quanto os tardios;
- **princípio da justiça e equidade**, relacionado à *relevância social da pesquisa*, com vantagens significativas para os participantes da pesquisa e minimização do ônus para os participantes vulneráveis, o que garante a igual consideração dos interesses envolvidos, não perdendo o sentido de sua destinação sócio-humanitária.

Com base nesses princípios éticos da Resolução nº 196/96 e na literatura (Johnson, 2001; Courage e Baxter, 2005; Sharp *et al.*, 2007), podemos sugerir diversas diretrizes para as pesquisas e avaliações de IHC, descritas a seguir.

O pesquisador deve **explicar os objetivos** da pesquisa aos participantes e dizer exatamente como deverá ser a participação deles. Ele deve deixar claro o que vai

ocorrer durante a coleta de dados, o tempo aproximado da coleta, os tipos de dados que serão coletados, e ainda como eles serão analisados. Qualquer dúvida do participante sobre a pesquisa deve ser esclarecida prontamente pelo avaliador.

O pesquisador deve garantir aos participantes a **confidencialidade e a privacidade** dos dados brutos coletados. Com o consentimento dos participantes, os dados brutos são compartilhados apenas com os pesquisadores. Ninguém mais deve ter acesso a esses dados brutos.

Ao divulgar os resultados da avaliação, o avaliador deve garantir o **anonimato** dos participantes, a preservação das suas imagens e a utilização cuidadosa das informações coletadas. Isso significa não divulgar seus nomes ou qualquer outra informação que possa identificá-los. O objetivo principal é não prejudicar os participantes, direta ou indiretamente, seja em termos de autoestima, de prestígio, na profissão, ou de qualquer outra forma. Esse cuidado ético deve ser tomado em todas as mídias em que as informações serão publicadas, seja em veículos impressos ou digitais, em textos, imagens, áudios ou vídeos. É comum utilizar nomes fictícios ou números em todo o material coletado para identificá-los apenas assim no relato dos resultados de uma pesquisa. Por exemplo, um participante pode citar nomes de pessoas com quem ele se relaciona, como no domínio de correio eletrônico. Nem o nome do participante, nem os nomes dos seus conhecidos podem ser divulgados, devendo ser substituídos por nomes fictícios ou por alguma máscara, como “[*nome*]”. Além disso, se o participante utilizar algum jargão ou expressão particular, devemos evitar divulgá-la nos resultados da avaliação porque seus conhecidos podem identificá-lo. A idade, a profissão e o sexo dos participantes também podem identificá-los, caso o participante seja diferente dos demais, como uma mulher dentre vários homens, ou uma pessoa de mais idade dentre vários mais jovens. O anonimato, a princípio, é voltado para terceiros. Entretanto, quando o pesquisador critica os dados obtidos, é desejável que nem o participante que os forneceu se reconheça nos relatos, pois eventuais críticas podem afetar sua autoestima. Nesses casos, o pesquisador deve relatar o que ocorreu com um grupo de participantes e os motivos para isso, em vez de criticar ou apresentar falhas de um participante específico. Sempre que possível, os participantes devem ter acesso aos resultados da pesquisa antes que eles sejam publicados.

É necessário obter **permissão para gravar** a voz ou imagem de qualquer pessoa, antes de começar a gravação. Devemos informar aos participantes logo no recrutamento sobre os tipos de gravações que serão realizadas, para evitar mal-entendidos ou desistências no momento da atividade.

A participação na pesquisa deve ocorrer apenas com o **consentimento livre e esclarecido** dos participantes. Todo participante de qualquer estudo tem o direito de



saber o objetivo do estudo, a duração estimada, os procedimentos de coleta de dados, o uso que será feito da informação coletada, os seus direitos enquanto participante do estudo e quaisquer riscos, desconfortos ou efeitos adversos relacionados à sua participação no estudo. Essas informações devem ser comunicadas ao participante durante o processo de recrutamento e depois reiteradas no início da atividade através de um **termo de consentimento**. Ao assinar esse termo, o participante atesta que entende as garantias e os riscos do estudo e concorda com sua participação naquelas condições. Caso o participante tenha menos de 18 anos, o termo de consentimento deve ser assinado pelo seu responsável legal. O termo de consentimento também deve ser assinado pelo responsável pelo estudo, atestando sua responsabilidade e comprometimento com as garantias ali asseguradas. Uma das vias assinadas do termo de consentimento permanece com o pesquisador, enquanto a outra é entregue ao participante. O Exemplo 5.1 apresenta um termo de consentimento para a realização de uma entrevista.

O **conforto dos participantes** deve ser cuidadosamente considerado. Os participantes de um estudo nunca devem se sentir desconfortáveis, seja física ou psicologicamente. Isso inclui coisas simples, como oferecer pausas para beberem água ou irem ao banheiro, além de instalações confortáveis. Eles devem ser tratados com respeito o tempo todo. Devemos evitar utilizar termos pejorativos ao nos referirmos ao participante, como *cobaia*, por exemplo. Além disso, no caso de observação de uso de um produto, devemos enfatizar que é o *produto* que será avaliado, e não o participante.

O participante tem o **direito e a liberdade de se recusar a participar** ou retirar seu consentimento e abandonar o estudo em qualquer fase da pesquisa, sem ser penalizado por isso. Sempre que o pesquisador perceber que o participante está passando por algum tipo de constrangimento ou incômodo físico, emocional ou psíquico, deve interromper a pesquisa antes que o participante tenha mais sofrimentos desnecessários. Uma pesquisa de IHC não deve deixar os participantes excessivamente exaustos, nervosos, ou levá-los ao pranto. Ela deve ser interrompida bem antes disso.

Exemplo 5.1 – Termo de consentimento

Somos uma equipe de consultoria da <<empresa>>, que está participando do projeto do sistema <<nome e breve descrição do sistema>>. Nessa etapa do projeto, queremos conhecer o que algumas das pessoas que irão <<usar o/ser afetadas pelo>> sistema pensam a respeito do <<sistema atual/processo atual>> e como imaginam que o novo sistema deveria apoiar o seu trabalho.

Estamos realizando uma série de pesquisas, e solicitamos seu consentimento para a realização e gravação de uma entrevista. Para decidir sobre o seu consentimento, é importante que você conheça as seguintes informações sobre a pesquisa:

- Os dados coletados durante a entrevista destinam-se estritamente a atividades de análise e desenvolvimento do sistema <<nome do sistema>>.

- Nossa equipe tem o compromisso de divulgar os resultados de nossas pesquisas para o cliente. A divulgação desses resultados pauta-se no respeito à sua privacidade, e o anonimato dos participantes será preservado em quaisquer documentos que elaborarmos.
- O consentimento para a entrevista é uma escolha livre, feita mediante a prestação de todos os esclarecimentos necessários sobre a pesquisa.
- A entrevista pode ser interrompida a qualquer momento, segundo a sua disponibilidade e vontade.
- Nossa equipe encontra-se disponível para contato através do e-mail <>e-mail<>.

De posse dessas informações, gostaríamos que você se pronunciasse acerca da entrevista:

Dou meu consentimento para a sua realização.

Não consinto com a sua realização.

<>local<>, <>data<>

<>assinatura do entrevistador<> <>assinatura do entrevistado<>

<>nome do entrevistador<> <>nome do entrevistado<>

Os participantes devem preferencialmente ter **autonomia** plena para serem capazes de decidir participar ou não do estudo ou coleta de dados. Devemos evitar a participação de sujeitos vulneráveis, tais como: menores de idade, alunos ou subordinados, a menos que este seja explicitamente o perfil dos participantes. Nesses casos, o pesquisador deve tomar um cuidado ainda maior para não causar constrangimentos ou danos aos participantes.

Antes de começar a pesquisa, o pesquisador deve combinar com o participante formas de **incentivo** à participação da avaliação, como, por exemplo, livros, brindes, vale-presentes, ou pagamento em dinheiro. É importante notar que no Brasil não é permitido pagar para as pessoas participarem de uma pesquisa científica, como ocorre em outros países. Aqui só é permitido ressarcir as despesas decorrentes da participação da pesquisa, basicamente transporte e alimentação nos dias de participação. Essa restrição não se aplica a pesquisas de cunho técnico ou prático, como, por exemplo, uma pesquisa visando ao reprojeto de um sistema comercial.

Embora os cuidados ao lidar com os participantes sejam sempre prioritários, é importante também considerar os aspectos éticos relacionados aos **dados coletados** (Lazar *et al.*, 2010; Cooper, 1999), em particular no que diz respeito à validade e confiabilidade dos dados, e à retenção de dados e documentação. Devemos proteger os dados coletados, para que não sejam mal interpretados ou corrompidos. Se isso ocorrer, todo o tempo e outros recursos despendidos na coleta dos dados serão desperdiçados, pois os dados terão de ser descartados. Os dados devem ser válidos e confiáveis, senão o risco de causar mais prejuízo do que benefícios é grande, pois dados distorcidos, corrompidos ou inválidos podem resultar em decisões de projeto



inadequadas. Além disso, os dados devem ser mantidos apenas enquanto forem relevantes. Quando não precisarem mais ser utilizados, devem ser descartados.

As perguntas feitas aos participantes e as reações dos designers aos seus comentários podem afetar os dados coletados. Devemos assegurar que os dados coletados sejam livres de tendenciosidades ou inclinações indevidas (*bias*), sejam corretos, válidos e confiáveis. Caso haja suspeita que um dado seja inválido ou não confiável, ele deve ser descartado. Além disso, as partes interessadas nos resultados do estudo devem ser informadas sobre as limitações dos dados coletados.

Além dos cuidados com os dados do participante, pode ser necessário assegurar a confidencialidade dos dados ou sistemas apresentados aos participantes, principalmente quando se trata de produtos comerciais. Nesses casos, o participante deve assinar um **acordo de confidencialidade**, no qual se compromete a manter todas as informações relacionadas ao produto e ao estudo confidenciais por um determinado período de tempo. Em geral, acordos de confidencialidade são elaborados pelo departamento jurídico da empresa proprietária do produto.

5.5 Como Coletar Dados dos Usuários?

Dentre as técnicas utilizadas frequentemente para coletar dados e levantar os requisitos dos usuários, destacamos:

- entrevistas;
- grupos de foco;
- questionários;
- *brainstorming* de necessidades e desejos dos usuários;
- classificação de cartões (*card sorting*);
- estudos de campo;
- investigação contextual.

Essas técnicas podem ser caracterizadas quanto ao seu objetivo, suas vantagens e o nível de esforço necessário para sua aplicação, conforme a Tabela 5.1.

Tabela 5.1 Quadro comparativo de técnicas de levantamento de requisitos (adaptado de Courage e Baxter, 2005)

técnica	objetivo	vantagens	esforço
entrevistas	<ul style="list-style-type: none"> ■ coletar informações detalhadas e profundas de usuários individuais 	<ul style="list-style-type: none"> ■ permite coletar muitas informações dos usuários individualmente ■ flexível: permite fazer perguntas de <i>follow-up</i> e se aprofundar mais do que questionários ou grupos de foco 	<ul style="list-style-type: none"> ■ é necessário treinar os entrevistadores ■ leva tempo para entrevistar muitos usuários
questionários	<ul style="list-style-type: none"> ■ coletar rapidamente dados (principalmente quantitativos) de muitos usuários 	<ul style="list-style-type: none"> ■ permite coletar informações de muitos usuários ■ pode ser rápido e fácil analisar os dados ■ relativamente baratos 	<ul style="list-style-type: none"> ■ avaliador deve ser experiente para evitar perguntas que induzem certas respostas ■ na Web, requer pouco esforço de distribuição
grupos de foco	<ul style="list-style-type: none"> ■ avaliar atitudes, opiniões e impressões dos usuários 	<ul style="list-style-type: none"> ■ permite coletar informações de muitos usuários simultaneamente (em grupo) ■ discussão em grupo com frequência dispara novas ideias 	<ul style="list-style-type: none"> ■ recrutar usuários suficientes pode requerer muitos recursos
brainstorming	<ul style="list-style-type: none"> ■ coletar uma lista priorizada de necessidades e desejos percebidos dos usuários 	<ul style="list-style-type: none"> ■ pode-se preparar, conduzir e analisar dados da atividade em pouco tempo e com poucos recursos 	<ul style="list-style-type: none"> ■ moderação em grupo requer esforço razoável ■ recrutar usuários suficientes pode requerer muitos recursos ■ pouco esforço para conduzir e analisar dados
classificação de cartões	<ul style="list-style-type: none"> ■ identificar como usuários agrupam informações ou objetos (para arquitetura da informação) 	<ul style="list-style-type: none"> ■ técnica simples de conduzir ■ se feita em grupo, permite coletar dados de vários usuários de uma vez ■ motiva a própria equipe a detalhar o produto em componentes 	<ul style="list-style-type: none"> ■ esforço de detalhar informações e definições ■ baixo esforço de condução ■ esforço para análise depende de ferramenta, número de cartões e de participantes
estudos de campo e investigação contextual	<ul style="list-style-type: none"> ■ entender usuários, seu ambiente e suas tarefas em contexto 	<ul style="list-style-type: none"> ■ permite descobrir o que se faz de fato (vs. o que se diz que se faz) ■ permite coletar muitos dados ricos ■ validade ecológica 	<ul style="list-style-type: none"> ■ nível de esforço mais alto para preparar as visitas, conduzir e analisar os dados



5.5.1 Entrevistas

A **entrevista** é uma das técnicas mais utilizadas de coleta de dados e levantamento de requisitos. Trata-se de uma conversa guiada por um roteiro de perguntas ou tópicos, na qual um entrevistador busca obter informação de um entrevistado (Seidman, 1998). Quando há mais de um entrevistado, costumamos chamar essa atividade de grupo de foco, conforme visto na próxima subseção. Há diversos tipos de entrevista, dependendo das suas limitações e necessidades.

Numa entrevista, as perguntas podem ser abertas ou fechadas (Lazar *et al.*, 2010; Sharp *et al.*, 2007). As **perguntas abertas** têm natureza exploratória. Nelas, não há qualquer restrição sobre o tipo ou tamanho de resposta que o entrevistado poderá fornecer. Uma pergunta aberta é bem útil quando temos pouco ou nenhum entendimento sobre a situação e quando queremos obter a opinião e as reações das pessoas sobre uma nova ideia de design. Mesmo quando a situação é conhecida, as perguntas abertas permitem revelar opiniões ou fatos desconhecidos e inesperados. Um exemplo de pergunta aberta é “O que você acha do mecanismo de busca do Web site CompreMais?”

As **perguntas fechadas** apresentam um conjunto predefinido de respostas dentre as quais o entrevistado deve selecionar. Perguntas fechadas requerem que o entrevistado conheça as respostas prováveis. Esse tipo de pergunta costuma ser mais utilizado em questionários. Uma pergunta fechada pode ser utilizada para coletar *feedback* rápido sobre uma opção de design específica. Por exemplo, a pergunta “Num Web site de comércio eletrônico, você prefere navegar pelas seções dos produtos ou fazer diretamente uma busca pelo produto desejado?” restringe o espaço de resposta do usuário às duas opções oferecidas.

Do ponto de vista de análise dos dados coletados, as perguntas fechadas são analisadas mais rapidamente do que perguntas abertas, e, em geral, se destinam à coleta de dados quantitativos ou quantificáveis, ao passo que as perguntas abertas se destinam principalmente à coleta de dados qualitativos e estudos em profundidade.

As entrevistas podem ser classificadas em estruturadas, não estruturadas e semiestruturadas. Em uma **entrevista estruturada**, o entrevistador se mantém fiel a um roteiro, fazendo as perguntas previamente definidas na ordem especificada. O entrevistador não possui muita liberdade para explorar tópicos novos que surjam durante a entrevista. Em geral, essa entrevista é composta na sua maioria de respostas fechadas. Em contrapartida, em uma **entrevista não estruturada**, o entrevistador realiza perguntas de modo bastante flexível, usando perguntas abertas e se aprofundando mais em alguns tópicos. Nesse tipo de entrevista, o único comprometimento do entrevistador é com o tópico abordado.

Em geral, buscamos um meio termo, e conduzimos **entrevistas semiestruturadas**. Nessas entrevistas, o roteiro é composto dos tópicos ou perguntas (geralmente abertas) que devem ser endereçados na entrevista, em uma ordem lógica. O entrevistador tem liberdade para explorar em maior profundidade as respostas fornecidas pelo entrevistado e até mesmo modificar a ordem dos tópicos abordados, mas deve manter o foco nos objetivos da entrevista.

O **roteiro** de entrevistas pode conter perguntas completas ou apenas os tópicos que devem ser endereçados durante a entrevista. Para manter o tom natural da conversa, principalmente em entrevistas semiestruturadas, algumas pessoas evitam redigir no roteiro as perguntas literais que devem ser feitas. Por exemplo, em vez de incluir no roteiro a pergunta: “O que você acha do mecanismo de busca do site CompreMais?”, o roteiro pode conter algo como “mecanismo de busca – opinião geral”.

Embora perguntas completas assegurem que serão feitas exatamente da mesma forma para todos os entrevistados, é possível que a conversa se torne menos fluida, com o entrevistador lendo a pergunta de modo a quebrar o ritmo da conversa. No caso de uma lista tópicos, a conversa se torna mais “natural”, pois o entrevistador apenas consulta o roteiro e tem liberdade de formular a pergunta relacionada a cada tópico de forma mais adequada ao perfil do entrevistado, buscando manter o tom da conversa até o momento. No entanto, dependemos mais da experiência do entrevistador em formular as perguntas de modo a não alterar o conteúdo das perguntas ou se desviar dos seus objetivos. Podemos ainda fazer um roteiro híbrido, com tópicos atuando como lembretes para o entrevistador coletar as informações necessárias, e com perguntas de exemplo para auxiliar entrevistadores menos experientes (Exemplo 5.2).

Já no caso de entrevistas estruturadas, costumamos formular as questões tal como serão perguntadas, para assegurar a validade de uma análise comparativa das respostas por toda a amostra de entrevistados. Seja qual for o tipo de entrevista, o roteiro deve ser revisado perante os objetivos da coleta de dados, para evitar incluir perguntas espúrias ou deixar de incluir perguntas que visem coletar dados necessários ao atingimento dos objetivos da entrevista.

Uma sessão de entrevista costuma seguir a seguinte **estrutura**: uma *apresentação*, na qual o entrevistador se apresenta e explica o objetivo da entrevista; um período de *aquecimento*, no qual são feitas perguntas de fácil resposta, como dados demográficos; a parte *principal* da entrevista, na qual o roteiro é explorado; um período de *desaquecimento*, para desfazer alguma tensão que tenha surgido; e a *conclusão*, na qual o entrevistador agradece ao entrevistado pelo seu tempo, desliga o gravador e guarda suas anotações, indicando que a entrevista terminou (Sharp *et al.*, 2007).

Exemplo 5.2 – Roteiro (parcial) de entrevista para um professor universitário.

- Experiência como professor de curso (tempo – área – nível):
Há quantos anos? Que área(s)?
Que nível (graduação/pós-graduação/extensão)?
 - Função (atividades – frequência – satisfação)
Quais as principais atividades? Quais as mais frequentes? E as menos frequentes?
De quais gosta mais de realizar? E de quais gosta menos? Por quê?
 - Divisão de responsabilidades (divisão – responsável – satisfação – desejos)
[professor, coordenação, suporte, universidade]
Quem faz o quê (definição do programa, critério de avaliação)?
Satisfação com a divisão atual? Delegaria o quê? Centralizaria o quê?
 - Utilização de tecnologias computacionais para apoiar o seu trabalho (tecnologia/atividade – frequência – satisfação – desejos)
Usa?
SIM: *Quais? Para quê? Com que frequência?*
 O que mais gosta? O que menos gosta? O que faria diferente?
NÃO: *Já usou? Por que não usa (mais)? O que precisaria ter para você usar?*
 - Sistema ideal
 - Comentários adicionais
-

O entrevistador deve evitar influenciar as respostas dos entrevistados com a formulação das perguntas, expressões faciais, gestos ou entonação de voz. Ao perguntar “Por que você gosta do mecanismo de busca do site CompreMais?”, por exemplo, o entrevistador já está pressupondo que o entrevistado gosta desse mecanismo, o que pode não corresponder à realidade. Esse tipo de pergunta pode desmotivar o entrevistado a dar sua opinião sincera e fazer com que ele responda o que acredita que o entrevistador queira ouvir.

Perguntas do tipo “sim ou não” costumam ser utilizadas para filtrar algumas perguntas subsequentes e definir o rumo da entrevista. Por exemplo, o entrevistador pode perguntar “Você já utilizou o mecanismo de busca do site CompreMais?”, para decidir se deve ou não se aprofundar nesse ponto. Para outros propósitos, perguntas desse tipo devem ser evitadas, pois podem induzir o entrevistado a dar a resposta que ele acredita que o entrevistador gostaria que ele desse. Por exemplo, ao perguntar “Você gosta do mecanismo de busca do site CompreMais?”, o entrevistado pode querer agradar o entrevistador e responder rapidamente “sim”. Não apenas essa resposta pode não refletir corretamente a opinião do entrevistado, como também traz pouca informação à entrevista. Se, no entanto, o entrevistador pergunta “O que você acha do mecanismo de busca do site CompreMais?”, o entrevistado é motivado a fornecer uma resposta mais elaborada.

Há casos em que o entrevistado fornece uma resposta muito sucinta para uma pergunta aberta, como “Bom”, “Ruim”, “Gosto”, “Não gosto”. Nessas situações, cabe ao

entrevistador explorar um pouco mais a resposta fazendo perguntas adicionais, como por exemplo: “O que você *mais* gosta nesse mecanismo de busca?”; “E o que você *menos* gosta nesse mecanismo de busca?”; “Houve algum momento em que o mecanismo de busca não trouxe o resultado esperado?”; e, em caso de resposta positiva: “O que você estava buscando nesse momento?”

Dependendo do objetivo da entrevista, perguntas fechadas podem prejudicar a coleta dos dados, por restringirem demais a expressão do entrevistado, que precisa encaixar a sua resposta naquelas previstas no roteiro. Considere, numa fase preliminar de levantamento de dados, uma pergunta do tipo “A ou B”, como, por exemplo, “Você prefere procurar um apartamento fornecendo o nome da rua ou pelo mapa?”. Embora haja outras formas de buscar um apartamento (e.g., pelo número de cômodos e vagas na garagem, faixa de preço etc.), a pergunta fechada pode fazer com que o entrevistado deixe de responder o que realmente pensa para fornecer uma resposta que se encaixe nas opções fornecidas, falseando, assim, o dado coletado.

Como a entrevista se assemelha a uma conversa, devemos evitar perguntas muito longas ou complexas, que sobrecarreguem a memória do entrevistado e prejudiquem sua resposta. Perguntas compostas devem ser desdobradas em perguntas menores. Por exemplo, em vez de perguntarmos “O que você acha da estrutura dos menus e submenus e da terminologia utilizada, em comparação com outros sites semelhantes?”, podemos perguntar primeiro “O que você acha da estrutura de menus e submenus do site?”, depois “O que você acha dos rótulos dos menus e submenus?” e, finalmente, “Você conhece algum site semelhante?” e, em caso positivo: “Como você compara esses menus com os desse site?” Caso contrário, aumentamos o risco de o entrevistado responder apenas parte da pergunta.

A terminologia utilizada numa entrevista também deve ser cuidadosa. Devemos evitar utilizar termos técnicos com os quais os entrevistados não tenham familiaridade. Quando um jargão desconhecido é utilizado numa entrevista, é possível que o entrevistado não entenda direito a pergunta e, portanto, forneça uma resposta pouco informativa ou até mesmo incorreta.

As entrevistas costumam ser gravadas em áudio. Além dessa gravação, os entrevistadores podem fazer algumas anotações durante a entrevista. Nesse caso, o roteiro pode ser projetado com espaço para anotar as respostas. Além disso, para perguntas fechadas, o roteiro impresso pode incluir as respostas predefinidas de modo que o entrevistador marque rapidamente a que tiver sido selecionada pelo entrevistado, como, por exemplo:

Você já fez compras em sites de comércio eletrônico alguma vez?

- sim não não se lembra

Além de entrevistas presenciais ou face a face, é possível conduzir entrevistas pelo telefone ou on-line, em geral através de um sistema de comunicação síncrona (como chat ou videoconferência) ou assíncrona (como e-mail).

Os entrevistadores devem ser treinados para realizar a entrevista. É importante que eles conheçam a fundo o roteiro, tenham segurança sobre os seus objetivos e prestem atenção ao que os entrevistados dizem para que possam formular perguntas que lhes permitam se aprofundarem nos tópicos investigados. Essa é uma característica importante e vantajosa da entrevista, principalmente em comparação com questionários, e que pode ser prejudicada pela inexperiência ou falta de treinamento do entrevistador. Por exemplo, um entrevistador que não conheça bem o roteiro pode ficar tão preocupado com a próxima pergunta a ser feita que deixa de prestar atenção ao que o entrevistado está dizendo, confiando exclusivamente na gravação de áudio e perdendo oportunidades preciosas de fazer perguntas adicionais para se aprofundar nas respostas.

As entrevistas são muito flexíveis e podem ser utilizadas de forma independente ou em conjunto com alguma outra atividade de coleta de dados e levantamento de requisitos. Como ocorre em toda triangulação de dados, algumas informações fornecidas em uma entrevista podem ser contestadas por dados coletados utilizando outras técnicas. Por exemplo, após perguntar quais sites o entrevistado mais visita ou quanto tempo passa navegando na Web, é possível comparar suas respostas com algum registro de software que monitore sua navegação real, permitindo triangular a percepção do entrevistado (dado subjetivo) com os fatos, dados objetivos.

Vale observar que nem sempre o que uma pessoa diz que faz é o que ela realmente faz. Algumas pessoas se esquecem o que exatamente aconteceu numa situação ou quanto tempo levaram para realizar uma determinada tarefa, enquanto outras podem querer projetar uma boa imagem de si mesmas e do seu trabalho. Para evitar esse tipo de problema, algumas entrevistas podem envolver a observação do entrevistado enquanto ele realiza uma ou mais atividades. A técnica de investigação contextual é uma forma bem conhecida desse tipo de entrevista com observação, e é apresentada na Seção 5.5.7.

O resultado de um conjunto de entrevistas é uma integração de perspectivas de múltiplos usuários, com base nos comentários recorrentes dos entrevistados. A análise das entrevistas pode ser feita interparticipante e intraparticipante (Nicolaci-da-Costa, 1994 ; Nicolaci-da-Costa *et al.*, 2004). Na **análise interparticipante**, para cada

pergunta (ou item do roteiro) individual, todas as respostas de todos os entrevistados são analisadas sistemática e rigorosamente. Essa análise revela as tendências centrais das respostas. Já na **análise intraparticipante**, para cada entrevistado individual, todas as suas respostas (para todas as perguntas) são analisadas, buscando identificar possíveis conflitos de opiniões, inconsistências entre respostas, sentimentos contraditórios etc. Essas duas formas de análise podem ser feitas alternadamente, visando aprofundar o resultado da análise e permitir detectar ausências notáveis, ou seja, o que os entrevistados deixaram de dizer em certas respostas mas disseram ou sugeriram em outras, bem como detalhes sobre seus sentimentos e atitudes, incluindo eventuais conflitos internos.

Ao conduzirmos entrevistas com diversos perfis de usuários sobre o mesmo processo, sistema ou organização, podemos obter uma visão profunda e abrangente dos tópicos investigados. Como não costuma ser viável realizar entrevistas com muitas pessoas, podemos utilizar o resultado de uma entrevista para elaborar questionários que permitam coletar informações de um número maior de pessoas, e assim obter resultados estatisticamente significativos ou pelo menos mais representativos da população de interesse.

5.5.2 Questionários

O uso de questionários também é uma das técnicas de coleta de dados mais frequentemente utilizadas. Um **questionário** é um formulário impresso ou on-line com perguntas que os usuários e demais participantes devem responder, a fim de fornecer os dados necessários em uma pesquisa, análise ou avaliação. Diferentemente de entrevistas, questionários permitem coletar dados de um grande número de pessoas, até mesmo geograficamente dispersas, compondo amostras muito maiores do que com entrevistas ou grupos de foco. Assim como entrevistas, questionários podem conter perguntas abertas e fechadas, mas costumam privilegiar as perguntas fechadas, de preenchimento rápido e de fácil análise.

As pessoas podem responder questionários no seu próprio tempo e no conforto do seu lar ou local de trabalho. No entanto, como o respondente não terá como tirar dúvidas sobre as perguntas no momento de responder ao questionário, a formulação da pergunta (e das respostas) deve ser ainda mais cuidadosa do que no caso de entrevistas, evitando ambiguidades e mal-entendidos (Lazar *et al.*, 2010; Sharp *et al.*, 2007). Além disso, um questionário deve conter instruções claras sobre como responder cada pergunta, indicando explicitamente se uma pergunta admite uma única resposta ou múltiplas respostas e utilizando símbolos informativos de forma consistente.



A taxa de respostas de um questionário é muito variável. Segundo Courage e Baxter (2005), varia de 1% em questionários de filantropia a 95% em questionários de censo. Em alguns casos, para aumentar o número de questionários respondidos, adotamos uma estratégia de sortear brindes dentre os respondentes. No entanto, essa estratégia pode falsear os dados coletados, fornecidos por pessoas interessadas mais nos brindes do que em contribuir para a pesquisa.

Diferentemente de entrevistas, em questionários não devemos fazer muitas perguntas abertas, porque isso reduz a taxa de respostas. Utilizamos questionários em geral quando temos uma boa noção das respostas mais prováveis e queremos conhecer a proporção de respostas numa amostra mais ampla da população de usuários. Embora restrinja os dados coletados, essa estratégia torna a análise dos dados mais rápida e fácil. Muitas vezes os questionários são utilizados em conjunto com entrevistas. Após entrevistas exploratórias, questionários podem ser utilizados para corroborar os resultados das entrevistas. Além disso, caso as estatísticas coletadas através de questionários sejam inesperadas, novas entrevistas podem ser formuladas para descobrir os motivos por trás das surpresas encontradas.

Como não há oportunidade de discutir sobre o questionário ou tirar dúvidas no momento de respondê-lo, as perguntas fechadas geralmente incluem respostas neutras ou alternativas, como “não sei”, “não quero responder” ou “outros”. Muitos pesquisadores costumam omitir perguntas negativas nos questionários, para não confundir os respondentes (Sharp *et al.*, 2007). Outros misturam esses dois tipos de perguntas justamente para ajudar a verificar a consistência das respostas dos usuários.

Segundo Sharp e coautoras (2007), um questionário típico inicia com a seguinte **estrutura**: informações demográficas básicas (e.g., sexo, idade) e detalhes relevantes sobre sua experiência (e.g., há quanto tempo utiliza computadores e nível de experiência com o domínio em questão). Essas perguntas auxiliam os pesquisadores a agruparem respostas de diferentes usuários. Vale observar, no entanto, que essas informações contextuais devem se limitar àquelas que forem relevantes aos objetivos do estudo. Perguntas gerais costumam preceder perguntas específicas. Aliás, a ordem das perguntas deve ser cuidadosamente projetada, pois a resposta a uma pergunta pode ser influenciada por uma das perguntas anteriores. Quando um questionário é longo, suas perguntas podem ser agrupadas em tópicos relacionados, formando uma estrutura lógica e de preenchimento mais fácil. Quando queremos obter informações de grupos distintos de usuários (e.g., professores e alunos, gerentes e técnicos etc.), pode ser necessário elaborar um questionário diferente para cada grupo.

Existem diversos tipos de perguntas e respostas utilizados em questionários, dentre os quais destacamos: múltipla escolha, faixas de valores, escalas e perguntas abertas.

Existem perguntas cujas respostas são previsíveis, como, por exemplo, sexo (feminino ou masculino). Nesses casos, podemos oferecer um conjunto de respostas de **múltipla escolha**, como no exemplo a seguir:

Sexo: masculino feminino prefiro não informar

Em alguns casos, o usuário pode escolher mais do que uma resposta. Esses casos devem ser bem marcados, para diferenciar das perguntas de uma única resposta. Por exemplo, quando queremos descobrir quais as atividades mais frequentes dentro de um conjunto de atividades predefinidas, podemos solicitar ao usuário que marque todas as opções relevantes, ou um número máximo de opções, como a seguir:

Quais atividades você realiza mais frequentemente on-line? (marque **até duas** opções)

- | | |
|--|---|
| <input type="checkbox"/> e-mail | <input type="checkbox"/> pesquisas gerais |
| <input type="checkbox"/> leitura de notícias | <input type="checkbox"/> compra de produtos |
| <input type="checkbox"/> transações bancárias | <input type="checkbox"/> contrato de serviços |
| <input type="checkbox"/> participação em redes sociais | <input type="checkbox"/> outros |

Algumas perguntas se referem a valores específicos, como idade ou renda mensal. Como alguns respondentes não se sentem à vontade em fornecer esses valores exatos, e como a análise desses dados costuma ser feita de forma agregada, é comum oferecermos **faixas de valores** como opções de resposta. Nesse tipo de pergunta, é importante evitar a sobreposição de valores (e.g., 20–30 e 30–40), para que a ambiguidade não prejudique a acurácia dos dados coletados. Por exemplo:

Idade: abaixo de 21 21–30 31–40 41–50 acima de 50

Para facilitar a comparação das respostas dos usuários, com frequência são utilizadas **escalas**, dentre as quais as mais conhecidas são as escalas de Likert (1932) e as escalas de diferenciais semânticos. A **escala de Likert** é comumente utilizada para medir opiniões, atitudes, crenças e, no caso de IHC, satisfação dos usuários com um produto ou ideia de design, como no exemplo a seguir:

É fácil encontrar o produto desejado navegando pelas seções do site:

- concordo plenamente
- concordo parcialmente
- não concordo nem discordo
- discordo parcialmente
- discordo totalmente

A escala de **diferenciais semânticos**, por sua vez, explora atitudes bipolares sobre um item particular.

Para cada par de adjetivos a seguir, marque o valor correspondente à sua opinião sobre a página de um produto do site:

- | | | | | | | |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------|
| atraente | <input type="radio"/> | feia |
| clara | <input type="radio"/> | confusa |
| útil | <input type="radio"/> | inútil |

O número de valores de uma escala é variável. Em geral, utilizamos um número ímpar de valores, a menos que queiramos evitar que os usuários fiquem “em cima do muro” (Sharp *et al.*, 2007). Em escalas Likert, costumamos utilizar 5 pontos, e em escalas de diferencial semântico utilizamos 5, 7 ou mesmo 9 pontos, este último quando queremos que os usuários façam julgamentos sutis sobre as características indicadas.

Perguntas abertas são utilizadas para obter informações livres e possivelmente mais detalhadas sobre alguns pontos. É importante fornecer espaço suficiente para o usuário se expressar. Em geral, apenas duas ou três linhas não são suficientes nem motivam uma resposta extensa. Compare o formato das perguntas a seguir. Qual formato incentiva o respondente a fornecer uma resposta mais completa?

(a) O que você acha do mecanismo de busca do site?

(b) O que você acha do mecanismo de busca do site?

Devemos tomar cuidado para não incluirmos muitas perguntas abertas em um questionário, pois isso pode desmotivar os respondentes a completá-los.

Para coletar dados dos usuários através de questionários, atualmente é possível utilizar formulários on-line. Existem diversos serviços gratuitos ou pagos,² que permitem não apenas criar formulários sofisticados para coletarmos os dados, mas também oferecem diversas ferramentas de análise dos dados coletados.

Como dito na seção anterior, muitas vezes os questionários são utilizados após entrevistas exploratórias. Isso ocorre principalmente por dois motivos: (1) nessas entrevistas são coletadas informações que permitem elaborar melhores perguntas no questionário, e (2) os questionários permitem descobrir o quanto as informações fornecidas pelos entrevistados são representativas da população-alvo. Além disso, caso os questionários forneçam resultados surpreendentes, é possível conduzir uma nova rodada de entrevistas para auxiliar na interpretação desses resultados.

5.5.3 Grupos de Foco

Em um grupo de foco, diversas pessoas (geralmente entre três e dez) são reunidas por uma ou duas horas numa espécie de discussão ou entrevista coletiva, guiada por um moderador experiente. Quando são bem conduzidos, os grupos de foco podem fornecer uma ampla gama de informações num curto período de tempo.

Grupos de foco permitem coletar informações sobre um público-alvo sobre quem tenhamos pouca informação. Podem ser realizados para gerar ideias; obter opiniões de pessoas sobre tópicos, conceitos ou demonstrações; obter respostas a uma série de questões; identificar conflitos relacionados a terminologias; identificar expectativas de diferentes grupos de pessoas; e descobrir problemas, desafios, frustrações, atitudes, preferências e aversões que surgem apenas num contexto social e por isso podem ser ignoradas por outras técnicas (Lazar *et al.*, 2010; Sharp *et al.*, 2007; Courage e Baxter, 2005). Os grupos de foco têm como vantagem permitir obter, em pouco tempo, múltiplos pontos de vista de um grupo de pessoas.

O papel do **moderador** de um grupo de foco é muito importante para assegurar que pessoas mais quietas ou tímidas participem e evitar que as extrovertidas e agressivas dominem a discussão.

Courage e Baxter (2005) sugerem evitar pedir para os participantes fazerem previsões sobre algo que eles ainda não experimentaram, como, por exemplo, pedir para avaliar a utilidade de algo que ainda não utilizaram. Além disso, devemos evitar endereçar tópicos polêmicos, relacionados, por exemplo, com política e valores morais. Algumas questões típicas exploradas em grupo de foco são (Courage e Baxter, 2005):

- um “dia típico” de um usuário ou o dia de trabalho mais recente;

² LimeSurvey (www.limesurvey.org) e o SurveyMonkey (www.surveymonkey.com) são exemplos desse tipo de serviço.

- as tarefas que os usuários realizam e como eles as realizam;
- o domínio em geral (e.g., terminologia, procedimentos normatizados);
- preferências e aversões dos usuários;
- resultados desejados ou objetivos dos usuários;
- reações, opiniões ou atitudes dos usuários sobre um determinado produto ou conceito;
- resultados desejados para novos produtos ou funcionalidades.

Além de perguntas, é comum fornecer aos participantes materiais concretos e protótipos do produto para que eles tenham um foco bem definido sobre o que falar. No caso de protótipos, podemos também pedir para eles realizarem algumas tarefas e relatarem suas experiências.

5.5.4 Brainstorming de Necessidades e Desejos dos Usuários

Essa técnica fornece informações sobre os tipos de conteúdo e características que os usuários querem e desejam em um produto (Courage e Baxter, 2005). Essa atividade de *brainstorming* funciona para qualquer produto ou serviço, e resulta numa lista priorizada de necessidades e desejos dos usuários. Em geral, essa técnica é utilizada para levantar requisitos e aprender sobre novas características que os usuários apreciam em um produto, e fornece mais benefícios quando utilizada durante o estágio conceitual do desenvolvimento do produto. Diferentemente de um grupo de foco, que busca endereçar perguntas específicas, uma sessão de *brainstorming* busca levantar de forma bastante livre um conjunto grande e abrangente de opiniões dos participantes em torno de um tema. Os resultados dessa atividade podem alimentar diretamente a especificação funcional e documentação de design.

Uma sessão de *brainstorming* pode ser conduzida em aproximadamente uma hora, e leva menos tempo ainda para analisar os dados de uma sessão, o que torna essa técnica leve em termos de recursos, mas poderosa em termos de resultados. Em geral, uma sessão de *brainstorming* envolve entre 8 e 12 usuários finais, de preferência com perfil semelhante. Caso haja mais do que um perfil, devemos conduzir mais de uma sessão.

Uma sessão eficiente de *brainstorming* começa com uma pergunta que sumariza o objetivo de entender o que os usuários querem e precisam no produto. Em vez de pedir para falarem sobre qualquer coisa que queiram, é mais eficiente fazer uma pergunta visando identificar *conteúdo*, *tarefas* ou *características* do produto. Sendo assim, a pergunta inicial pode ser feita de três diferentes formas: (1) para identificar as informações que os usuários querem ou precisam que o sistema forneça; (2) para

identificar os tipos de atividades ou ações que os usuários esperam realizar com o sistema; e (3) para identificar características como, por exemplo, confiabilidade, rapidez, segurança (Courage e Baxter, 2005). A pergunta deve se referir ao “sistema ideal”, para que os participantes não se limitem ao que eles acreditam que a tecnologia possa fazer. Alguns exemplos de pergunta são: “Que informações o sistema ideal deve fornecer?”; “Que tarefas você precisaria ou gostaria de realizar com o sistema ideal?”; “Que características o sistema ideal deve apresentar?”. Uma sessão de levantamento de necessidades e desejos dos usuários também pode ser dividida em duas etapas, uma para o levantamento das informações e outra para o levantamento das tarefas.

Cada sessão deve ter um **moderador**, que é responsável por fazer perguntas para esclarecer o que for dito; manter o foco no objetivo da sessão; manter a atividade em andamento, mas sem oferecer suas próprias opiniões ou influenciar indevidamente as respostas dos participantes; manter os participantes motivados; não criticar o que eles disserem; certificar-se de que todos participem, mas que ninguém domine a sessão. Além do moderador, uma sessão pode envolver um secretário, um cinegrafista e, caso as instalações permitam, outras partes interessadas.

No início da sessão, os participantes devem ser informados sobre o objetivo e procedimento da atividade. Courage e Baxter (2005) sugerem uma introdução como a seguir (p. 382):

Estamos projetando *<Descrição do produto>* e precisamos entender quais *<informações, tarefas ou características>* vocês querem e precisam nesse produto. Isso ajudará a nos certificarmos de que o produto seja projetado para satisfazer seus desejos e necessidades. Esta sessão terá duas partes. Na primeira, faremos um *brainstorming* de *<informações, tarefas ou características>* de um sistema ideal; e na segunda parte da atividade pediremos que vocês priorizem individualmente os itens que foram levantados.

Eles sugerem definir as seguintes regras para a sessão:

- Este é um sistema *ideal*, então todas as ideias são corretas. Os participantes não devem censurar a si próprios ou aos outros, mas sim exercitar sua criatividade.
- Não se trata de uma sessão de design, então os participantes não devem tentar *projetar ou construir* o sistema. Caso os participantes começem a sugerir soluções de design, eles serão interrompidos e perguntas serão feitas para identificar a motivação por trás dessas sugestões. A questão inicial tem um papel importante nesse ponto. Em vez de perguntar “Qual é o sistema de agenda ideal para o seu dispositivo móvel?”, por exemplo, podemos perguntar “Qual é o sistema de agenda ideal para quando você está fora do

escritório?”, a fim de evitar que os participantes se refiram especificamente aos seus próprios dispositivos.

- O moderador pode fazer perguntas sobre sugestões duplicadas. Quando um participante faz uma sugestão que parece semelhante a uma sugestão anterior, é possível que ele esteja querendo dizer algo um pouco diferente. Nesse caso, o moderador deve pedir mais detalhes para descobrir de que maneira as sugestões diferem.
- O secretário escreve apenas o que o moderador parafrasear. Isso significa que o moderador precisa entender o que os participantes realmente querem com cada sugestão, para então transmitir ao secretário o que deve ser registrado. O secretário então deve registrar a sugestão, numerando-a para facilitar referências futuras.

Durante a sessão, caso os participantes desviam dos objetivos, a pergunta inicial e até mesmo as regras podem ser repetidas. Caso os usuários comecem a sugerir informações quando o foco são as tarefas, ou vice-versa, o moderador deve fazer perguntas para que o participante reformule sua sugestão e assim permita eliciar as tarefas relevantes.

Devemos fornecer papel e lápis a fim de que os participantes possam registrar suas ideias para que, em momentos de grande atividade, as ideias individuais não se percam enquanto o moderador estiver tentando entender uma outra ideia fornecida anteriormente.

Quando os participantes de uma sessão de *brainstorming* começam a se calar, pode ser um sinal de que todas as ideias que tinham já tenham sido registradas e de que está na hora de passar para a próxima atividade. Segundo Courage e Baxter (2005), isso costuma acontecer por volta de 40 minutos do início da atividade.

Na atividade de priorização dos itens registrados, geralmente solicitamos que cada participante registre, num formulário, os cinco itens que considera essenciais para o produto, indicando, para cada item, seu número, sua descrição e por que esse item é importante para ele. Devemos informar aos participantes que todos os cinco itens têm o mesmo peso, e que, caso um participante tente “votar” num mesmo item mais de uma vez, as duplicatas serão descartadas.

Uma alternativa à sessão de *brainstorming* consiste em utilizar diagramas de afinidade (Beyer e Holtzblatt, 1998). Um **diagrama de afinidade** é construído da seguinte maneira: cada pessoa escreve cada item por ela levantado numa pequena folha de papel autocolante e a afixa num painel ou parede. Em seguida, os participantes examinam as folhas afixadas por todos, agrupando os itens semelhantes ou fortemente relacionados. Dependendo do objetivo do diagrama, podemos acrescentar rótulos

aos grupos identificados. Para priorizar os itens, podemos pedir que cada participante marque com três asteriscos o item que julgar mais importante, dois asteriscos o segundo mais importante e um único asterisco o terceiro item mais importante para ele.

A principal vantagem da utilização de diagramas de afinidade é assegurar a oportunidade de cada participante em expressar suas ideias, o que pode não ocorrer numa sessão de *brainstorming* em que alguns participantes sejam bem mais extrovertidos ou agressivos do que outros.

Na preparação para a análise, itens semelhantes devem ser agrupados. Itens repetidos de um mesmo participante devem ser contabilizados apenas uma vez. Para cada item, devemos determinar a porcentagem de participantes que o escolheu. Os dados de múltiplas sessões devem ser analisados em dois momentos: examinando cada sessão individualmente, e depois combinando os dados de todas as sessões.

Devemos comparar os itens que foram priorizados pelos participantes com os itens que constem na especificação funcional do produto, caso haja. Em geral, os itens priorizados pelos participantes devem ser priorizados pela equipe de design do produto. Caso surjam sugestões inesperadas, podemos voltar ao registro de áudio da sessão e rever a discussão sobre esses itens, para identificar por que os participantes os julgaram importantes.

É importante observar que algumas sugestões podem parecer tão óbvias aos participantes que eles não as mencionam. Por exemplo, num sistema de comércio eletrônico, é possível que ninguém mencione “preço do produto” como um item de informação desejado, assumindo que isso seja óbvio. Como em qualquer outra técnica, é importante considerar também o conhecimento que a equipe de design já adquiriu sobre o domínio, os usuários e o resultado de outras atividades de levantamento e análise ao interpretar os dados coletados.

O resultado da análise também pode ajudar a reduzir a lista de funcionalidades especificadas para o produto. Em vez de empregar tempo e recursos para projetar e desenvolver uma funcionalidade que nunca foi mencionada pelos participantes, devemos questionar a necessidade dessa funcionalidade ou até mesmo planejar um novo estudo para esclarecer a discrepância entre as expectativas da equipe de projeto e as dos usuários finais.

Em geral, os resultados de uma análise de desejos e necessidades dos usuários são sumarizados em uma tabela com: item ou categoria; exemplos do item ou categoria, obtidos dos exemplos que os próprios participantes forneceram; porcentagem de participantes que selecionaram o item como um dos cinco itens prioritários. A tabela deve ser ordenada por prioridade, da maior para a menor. Além disso, a lista completa



de ideias deve ser incluída num relatório, para consulta futura sobre funcionalidades adicionais ou mesmo inspiração.

Finalmente, como toda atividade que envolve pessoas, é importante realizar uma sessão-piloto, para avaliar a questão oferecida, o procedimento, o treinamento e as habilidades do moderador e do escriba, o tempo estimado para as atividades e todo o material preparado.

5.5.5 Classificação de Cartões

A técnica de **classificação de cartões** (*card sorting*) é utilizada principalmente para informar ou guiar o projeto da arquitetura de informação de um produto. Por exemplo, pode ajudar a determinar a estrutura de menus e submenus numa aplicação, de índices de navegação em um Web site e de um sistema de ajuda on-line. Pode ajudar também a criar um esquema de classificação para sistemas de gerenciamento de documentos e identificar categorias potenciais para uma base de conhecimento. Finalmente, pode ajudar a identificar os passos e subpassos de um processo. Também pode fornecer informações para decidir como organizar controles numa interface. Spencer (2009) afirma ainda que, mais do que um método colaborativo para criar estruturas de navegação, a classificação de cartões é uma ferramenta que nos auxilia a entender as pessoas para as quais estamos projetando um produto.

Essa técnica pode ser utilizada para levantar diferentes modelos de classificação; para explorar como as pessoas pensam sobre certos tópicos; para descobrir que categorias parecem semelhantes ou complementares; para descobrir sobre o que pode ser agrupado e o que não pode; e para coletar listas de palavras ou expressões que as pessoas utilizam para descrever grupos de informação (Spencer, 2009).

O método em si é relativamente simples. Um conjunto de cartões ou fichas são preparados com amostras ou descrições de conteúdo e fornecidos a um grupo de pessoas que devem organizá-los em grupos, de acordo com a similaridade entre os cartões. Note que o critério de similaridade é definido pelos próprios participantes. No caso da **classificação de cartões aberta**, os participantes então descrevem os grupos de cartões que eles criaram. Já no caso da **classificação de cartões fechada**, são fornecidos também cartões que representam categorias, e nesse caso os participantes devem classificar os cartões de conteúdo nessas categorias predefinidas. Em ambos os casos, os resultados são registrados, analisados e aplicados no projeto de um produto. A classificação de cartões nos permite aprender sobre como as pessoas pensam em categorias e conceitos, como os descrevem e quais informações pertencem a quais categorias. A Figura 5.1 ilustra os cartões utilizados em uma sessão de *card sorting*.



Figura 5.1 Cartões utilizados em uma sessão de *card sorting*.

Spencer (2009) enumera os seguintes passos para a condução de uma classificação de cartões:

1. decidir o que queremos descobrir;
2. selecionar o método (aberto ou fechado; individual ou em grupo; presencial ou remoto; manual ou por software);
3. selecionar o conteúdo;
4. selecionar e convidar os participantes;
5. conduzir a sessão de classificação de cartões e registrar os dados;
6. analisar os resultados;
7. utilizar os resultados no seu projeto.

Definição dos Objetivos

O primeiro passo é definir o objetivo do estudo. Spencer (2009) enumera os seguintes objetivos típicos:

- aprender sobre como as pessoas pensam sobre o conteúdo e seus principais agrupamentos e utilizar essa informação para criar categorias de alto nível e subcategorias;
- aprender se há conceitos de alto nível nesse conteúdo e utilizar essa informação para entender melhor os relacionamentos entre os itens de conteúdo;

- envolver autores de um Web site como forma de mostrar-lhes que as pessoas pensam de diferentes maneiras, e em particular como elas pensam sobre o seu próprio conteúdo;
- explorar se há um esquema principal de classificação para o conteúdo ou se há mais de um esquema. Utilizar essa informação para definir se as informações devem ser oferecidas de mais de uma maneira;
- descobrir por que uma pequena seção do Web site não está funcionando ao explorar diferentes métodos de categorização. Utilizar essa informação para decidir se e como o conteúdo deve ser reorganizado;
- coletar termos e expressões a serem utilizados para rotular grupos e categorias de conteúdo.

Seleção do Tipo de Classificação

Para selecionar o tipo de classificação de cartões — aberta ou fechada —, devemos ter em mente que a classificação aberta permite aprender mais sobre os esquemas de classificação dos usuários. Dependendo do nosso objetivo, podemos pedir aos participantes que considerem alguns critérios particulares, como, por exemplo: principais grupos de usuários-alvo; principais tarefas que os usuários devem realizar; passos ou estágios de um processo. Existem casos, no entanto, em que uma classificação fechada é preferencial: quando temos um conjunto de categorias que não pode ser modificado; quando estamos acrescentando pouco conteúdo a uma estrutura existente; quando estamos confiantes de que os grupos atuais são adequados e desejamos explorar detalhes mais específicos de posicionamento dos itens de conteúdo (Spencer, 2009).

A classificação de cartões pode ser conduzida com indivíduos ou com um grupo de usuários trabalhando individualmente (Courage e Baxter, 2005; Spencer, 2009). O principal risco de realizarmos uma classificação de cartões em grupo é haver um membro dominante que force sua opinião sobre as opiniões dos outros. Quando isso não acontece, e os participantes conversam e discutem sobre os diferentes tipos de conteúdo e as diferentes formas de classificá-lo e utilizá-lo, eles fornecem insumos preciosos para o estudo, às vezes mais úteis do que o resultado final da classificação em si. Spencer recomenda que, sempre que possível, essa atividade seja conduzida em grupo.

Seleção de Conteúdo

Uma das principais causas para o insucesso de uma sessão de classificação de cartões é a seleção de conteúdo inadequado. O conteúdo selecionado deve permitir atingir os objetivos do estudo. No caso de Web sites, o conteúdo pode ser: tópicos ou assuntos; páginas de conteúdo do próprio Web site; produtos de um catálogo; navegação ou

páginas de índice; pequenas seções do Web site que tenhamos confiança de que deveriam estar juntas. Já no caso de aplicações, devemos buscar tarefas e funções: itens de menu da aplicação; funcionalidades-chave da aplicação; passos em um processo; tarefas-chave. Quando o objetivo principal é explorar uma ideia ou tópico, o conteúdo selecionado pode ser resultante de *brainstorming* ou de conteúdos disponíveis no domínio ou em produtos semelhantes (Spencer, 2009).

Ao selecionarmos o conteúdo, devemos assegurar que os itens selecionados sejam representativos e relevantes para os participantes, e que possam ser agrupados, ou seja, que a amostra não tenha itens completamente diferentes (e.g., informações ou funcionalidades, mas não ambos) e não relacionados. Outro cuidado a ser tomado é não incluir itens de conteúdo que já pareçam categorias (e.g., “Utilidades Domésticas”), não abusar de termos que induzam os participantes a certas classificações (e.g., diversos cartões intitulados “Manual de...” possivelmente serão agrupados, mesmo que seu conteúdo seja bem diferente) e procurar manter todos os itens num nível semelhante de abstração, para não influenciar indevidamente a sua atividade. Para evitar classificações indevidas, no caso de uma classificação fechada, os cartões que representem categorias devem ser bem diferenciados dos cartões que representam conteúdos. Para isso, costumamos utilizar cartões de cores diferentes.

Condução da Sessão

Como em toda observação de alguma atividade do usuário, é importante conduzir um estudo piloto antes da sessão com os participantes reais. Um estudo-piloto visa avaliar não apenas as instruções fornecidas aos participantes, mas todo o material gerado. Devemos avaliar se os textos escritos nos cartões refletem adequadamente o conteúdo e se estão claros e corretos ortográfica e gramaticalmente. No piloto é possível identificar também se é possível formar grupos com a amostra de cartões selecionada.

No início da sessão, devemos informar aos participantes sobre o objetivo do estudo (e.g., um Web site, um software) e o conteúdo que eles vão encontrar nos cartões. Em seguida, devemos instruí-los a agruparem os cartões que receberão, sempre que fizer sentido que os cartões fiquem juntos no sistema indicado. No caso de uma classificação fechada, eles devem classificar os cartões nas categorias predefinidas. Perguntas frequentes devem ser antecipadas: cartões que os participantes não entendam devem ser separados do restante, e um cartão pode ser classificado em mais de uma categoria. Nesse último caso, o participante deve copiar o conteúdo do cartão para um outro e colocar cada um nos grupos desejados.



Após fornecer as instruções, os cartões são espalhados em uma mesa para que os participantes iniciem a atividade. Caso seja possível, devemos gravar os comentários e discussões que surgem, para registrar as negociações de significado e a forma de pensar dos participantes. Além disso, no caso de uma sessão em grupo, devemos cuidar para que um participante não domine a atividade. Ao término da atividade, caso se trate de uma classificação aberta, devemos solicitar aos participantes que deem nomes aos grupos. Finalmente, podemos pedir a opinião dos participantes sobre os grupos formados, para avaliar o quanto estão satisfeitos e confiantes com o resultado. Podemos pedir também que descrevam os critérios utilizados na classificação e que indiquem os melhores exemplos de cada grupo.

Análise dos Resultados

A análise dos cartões consiste em verificar quais grupos foram formados, qual esquema de classificação as pessoas utilizaram, quais itens de conteúdo foram classificados em cada grupo e quais termos e expressões foram utilizados para descrever os grupos (no caso de classificação aberta). Essa análise pode ser feita utilizando uma planilha ou software específico para classificação de cartões. Em seguida, costumamos fazer uma análise estatística dos dados, utilizando os algoritmos de agrupamento ou aglomeração (*clustering*) de k-médias (*k-means cluster analysis*) ou hierárquico (*hierarchical cluster analysis*), ou ainda de escalonamento multidimensional (*multidimensional scaling, MDS*).

Informações podem ser organizadas utilizando diferentes esquemas, como, por exemplo: tópico; cronologia; geografia; ordem alfabética; ordem numérica; tarefa; público-alvo (Spencer, 2009). Ao fornecer indícios sobre como as pessoas organizam um certo conjunto de informações, a técnica de classificação de cartões pode levar a equipe de design a questionar suas suposições e preconceitos. Pode fornecer uma nova perspectiva sobre as informações e seus esquemas organizacionais. Segundo Courage e Baxter (2005), essa técnica nos diz como as características de um produto podem ser estruturadas para se encaixarem nas expectativas dos usuários sobre de que maneira essas características estão relacionadas.

Spencer (2009) alerta que o resultado da classificação de cartões não deve definir cegamente a arquitetura de informação de um produto. Em outras palavras, ele não fornece uma resposta direta sobre como organizar as informações, mas sim ideias para a elaboração da arquitetura de informação. Segundo ela, o real valor da classificação de cartões é aprender sobre o seu usuário coisas que não saberíamos de outro jeito. Por exemplo, se planejamos organizar um Web site de receitas por método de cozimento e os usuários pensam em termos de ingredientes, precisamos saber disso.

Cabe ao projetista, no entanto, combinar o que aprendeu sobre a pesquisa com usuários, os objetivos do produto e a análise de conteúdo, sintetizando os dados coletados numa arquitetura de informação adequada.

Além disso, os resultados de uma classificação de cartões não indicam se os usuários serão capazes de encontrar informações com o esquema de categorias resultante, pois a atividade envolve partir de um conteúdo e associá-lo a uma categoria, e não o contrário (i.e., encontrar um conteúdo a partir de um conjunto de categorias).

5.5.6 Estudos de Campo

A expressão “estudo de campo” inclui uma categoria ampla de atividades relacionadas com usabilidade que podem incluir investigação contextual, entrevistas no ambiente do usuário e observações simples. Durante um estudo de campo, um pesquisador visita usuários finais no seu próprio ambiente (e.g., lar ou local de trabalho) e os observa enquanto desempenham uma atividade. Estudos de campo podem durar desde algumas poucas horas até diversos dias, dependendo dos objetivos do estudo e dos recursos disponíveis.

O principal objetivo de um **estudo de campo** é entender o comportamento natural do usuário final no contexto do seu próprio ambiente de atuação (Courage e Baxter, 2005). Ao observarmos os usuários em seu próprio ambiente, podemos capturar informações que afetam o uso de um produto — incluindo interrupções, distrações e outras demandas de tarefa — e contexto adicional que não podem ser capturados ou replicados num ambiente de laboratório. Coletamos dados ricos e detalhados, que nos permitem obter uma visão holística do processo ou do domínio. Trata-se de uma investigação da realidade dos usuários, e não de suposições. O resultado é tornar explícitos os aspectos e processos implícitos do ambiente do usuário.

Estudos de campo podem ser utilizados em qualquer momento durante o ciclo de vida de desenvolvimento de um produto, mas geralmente são mais proveitosos durante as atividades iniciais de levantamento. Os estudos de campo permitem alcançar diferentes objetivos (Courage e Baxter, 2005):

- identificar novas funcionalidades e produtos;
- desafiar ou verificar suposições que as partes interessadas tenham sobre os usuários, suas tarefas e seu ambiente;
- identificar uma falta de correspondência entre a forma como o usuário trabalha e pensa e a forma como as ferramentas e os procedimentos lhes obrigam a trabalhar;
- entender os objetivos dos usuários;
- identificar os materiais de treinamento necessários;

- criar designs iniciais;
- desenvolver um inventário das tarefas;
- definir uma hierarquia de tarefas;
- coletar artefatos (i.e., objetos ou itens que os usuários utilizam para complementar suas tarefas, ou que resultam de suas tarefas);
- verificar se os usuários correspondem aos perfis de usuários traçados inicialmente;
- elaborar personas a partir de observações de usuários reais;
- coletar informações necessárias para outras atividades voltadas à qualidade de uso (e.g., elaborar um questionário, identificar tarefas para um teste de usabilidade).

Courage e Baxter (2005) alertam para a possibilidade de um investigador sem familiaridade com o domínio fazer anotações demasiadamente simplificadas. Para evitar isso, sugerem que as anotações sejam revisadas por um especialista no domínio. Os próprios participantes do estudo podem tentar “traduzir” o seu trabalho de forma simplificada demais, com objetivo de ajudar o investigador. Para evitar isso, o investigador pode solicitar aos participantes que o considerem um aprendiz e lhe ensinem sobre o trabalho tal como deve ser realizado, sem omitir etapas.

Mesmo no ambiente de atuação do participante, é possível que ele se comporte de forma diferente apenas porque sabe que está sendo observado (Draper, 2009). Sendo assim, pode ser necessário fazer o estudo durante um tempo prolongado, para que as pessoas se acostumem com a presença do observador e voltem a se comportar normalmente.

Existem diversas formas de estudo de campo. A forma mais simples é a observação pura, sem interação do observador com os participantes. Uma variação é a observação guiada por um conjunto de tópicos de interesse. Outras formas envolvem entrevistas estruturadas ou semiestruturadas, possivelmente com o registro de fotos, áudio e vídeo do ambiente de atuação dos participantes, e com a coleta ou cópia dos artefatos utilizados por eles. Existe ainda a possibilidade de registrar as informações sem a presença do observador. Esse é o caso de diários, mantidos pelos próprios participantes, e de registros de vídeo por câmeras instaladas no ambiente dos participantes (Courage e Baxter, 2005).

Uma das formas mais comuns de estudo de campo é a investigação contextual, apresentada na próxima seção. Trata-se de um estudo de campo com o envolvimento intenso do investigador como um participante aprendiz, incluindo entrevistas e observação.

5.5.7 Investigação Contextual

Como visto na Seção 4.3.4, Beyer e Holtzblatt (1998) elaboraram o ciclo de vida de design contextual, uma abordagem centrada no cliente na qual a atividade de investigação contextual exerce um papel central.

O objetivo da **investigação contextual** é revelar todos os aspectos da prática do trabalho. A investigação contextual parte da hipótese de que, quando boa parte do trabalho não pode ser articulada adequadamente por aqueles que o praticam, é necessário que *vejamos* o trabalho. Para isso, a investigação contextual advoga ir aonde o usuário trabalha, observar o usuário enquanto ele trabalha e conversar com ele sobre o seu trabalho. Mais especificamente, os principais objetivos da investigação contextual são:

- obter dados sobre a estrutura do trabalho na prática, em vez de uma caracterização de marketing abstrata ou dissociada da prática real;
- tornar explícito o conhecimento tácito e não articulado sobre o trabalho, para que os designers, que não o realizam, possam entendê-lo;
- conhecer os detalhes do trabalho que se tornaram habituais e invisíveis.

Reconhecendo que muitas vezes é difícil para uma pessoa articular o seu trabalho de forma a comunicar suas práticas para uma equipe de design, a investigação contextual adota um modelo de **mestre-aprendiz**. Nesse modelo, o entrevistador, membro da equipe de design, exerce o papel de *aprendiz* do trabalho do usuário. O usuário, no papel de *mestre*, ensina o seu trabalho exercendo-o e falando sobre ele com o aprendiz, enquanto o trabalho é realizado. Isso torna o compartilhamento de conhecimento uma tarefa mais simples e natural. Os usuários nem sempre têm consciência de tudo o que fazem ou por que o fazem; eles se tornam conscientes disso ao fazê-lo.

O modelo mestre-aprendiz é eficiente para coletar, do próprio usuário, os dados que a equipe de design precisa conhecer sobre o trabalho desse usuário. Esse modelo facilita:

- tornar os usuários cientes do que fazem, ao fazê-lo;
- interromper o trabalho para pensar sobre ele;
- revelar todos os detalhes de uma prática de trabalho;
- apontar e explicar as diferenças entre o essencial e o irrelevante;
- revelar padrões ou princípios atuantes, que influenciam ou determinam a forma como trabalham.

A investigação contextual se baseia nos seguintes princípios: contexto, parceria, interpretação e foco.



O investigador deve ir ao local de trabalho dos seus usuários e observá-los realizando seu trabalho, **em contexto**. Estar presente enquanto o trabalho ocorre torna detalhes visíveis e os dados coletados concretos, permitindo conhecer a experiência em andamento, em vez de um sumário sobre uma experiência passada ou idealizada, envolvendo dados abstratos.

A **parceria** com os usuários é firmada através de conversas sobre o trabalho deles, com o objetivo de nos tornarmos seu colaborador no entendimento da prática de trabalho. Durante o trabalho, o usuário está engajado na sua atividade e o entrevistador observa os detalhes que se desdobram, buscando identificar *padrões* e *estruturas*, e pensando sobre as razões subjacentes às ações do usuário. Quando uma estrutura é identificada ou quando algo parece não se encaixar, o entrevistador interrompe o trabalho para conversar com o usuário sobre o que observou, ou sobre uma ideia de design que possa ter tido. É importante observar que, para que um processo seja verdadeiramente centrado no cliente, os usuários devem poder modificar o entendimento inicial do trabalho que os designers possam ter formado. Para isso, devemos evitar os modelos de entrevistador–entrevistado, o do designer como especialista e o do designer como visitante, e adotar o modelo de mestre–aprendiz.

A **interpretação** consiste em desenvolver um entendimento compartilhado com o usuário sobre os aspectos relevantes do trabalho. A partir de um fato, de um evento observável, o designer levanta uma hipótese, uma interpretação inicial do que o fato significa ou do que está por trás dele, atribuindo significado a suas observações. Essa hipótese tem implicações para o design, que podem ser concretizadas numa ideia de design para o sistema. É necessário compartilhar essa interpretação com o usuário para nos certificarmos de que ela está correta, ou seja, que o trabalho foi entendido corretamente. Quando um evento contradiz nossas suposições, devemos investigá-lo para corrigirmos o entendimento falho por um melhor. E mesmo quando o usuário já sugere diretamente uma ideia de design, é importante seguirmos a cadeia inversa para entendermos o contexto do trabalho que gerou aquele desejo.

O princípio de **foco** afirma que a investigação deve ser guiada por um entendimento claro do seu objetivo, para atribuir significado ao trabalho. Devemos ter em mente perguntas sobre aspectos do trabalho (e.g., o que é o trabalho que deve ser apoiado? Como esse trabalho se encaixa nas atividades do usuário? Quais são as principais tarefas?), sobre as pessoas com quem devemos conversar (e.g., quem está envolvido nessa atividade? Quem são os ajudantes informais? Quem fornece a informação necessária para realizar o trabalho? Quem utiliza o resultado do trabalho?) e sobre o contexto do trabalho (e.g., onde [fisicamente] ocorre o trabalho? Qual é o contexto cultural e social em que ele ocorre?). Devemos buscar metáforas para o

trabalho — tipos de trabalho que tenham estrutura semelhante àquele que queremos apoiar. Podemos utilizar metáforas para estruturar nosso pensamento e conduzir entrevistas explorando-as, se isso ajudar no nosso entendimento.

Uma investigação contextual geralmente envolve, para cada papel, 6 a 10 entrevistas de três horas com pessoas que trabalham de forma bem diferente (Beyer e Holtzblatt, 1998), e segue a seguinte estrutura: uma entrevista convencional curta (de aproximadamente 15 minutos), uma transição para a entrevista contextual propriamente dita, e uma conclusão.

Durante a **entrevista convencional**, um membro da equipe de design, assumindo o papel de entrevistador, se encontra com o usuário no local de trabalho dele, se apresenta e descreve o foco da entrevista. Explica que o usuário e seu trabalho são primordiais e que ele depende do usuário para aprender sobre o trabalho e corrigir seus mal-entendidos. O entrevistador pede opinião sobre as ferramentas que o usuário utiliza e obtém uma visão geral do trabalho como um todo e do que precisa ser feito naquele dia. Esses dados são sumarizados, e não contextuais, então nenhuma questão deve ser aprofundada neste momento.

Logo após a entrevista inicial, é feita uma rápida **transição**, na qual o entrevistador informa que a partir daquele momento o usuário deve realizar o seu trabalho enquanto o entrevistador o observa, que o entrevistador vai interrompê-lo quando vir algo interessante e que o usuário pode avisar quando não for um bom momento para ser interrompido.

Durante a **entrevista contextual** propriamente dita, o usuário começa a fazer o seu trabalho, enquanto o entrevistador observa e interpreta. O entrevistador deve analisar os artefatos utilizados e eliciar relatos em retrospectiva. Ele deve tentar manter o usuário falando concretamente sobre seu trabalho (*vs. "em geral..."*) e fazer anotações o tempo todo, para não depender exclusivamente da gravação, que pode falhar ou deixar de capturar informações importantes, como o momento de um gesto ou o uso de um artefato que não foi comentado verbalmente, como, por exemplo, uma rápida consulta visual a uma anotação afixada no monitor do usuário. A conversa entre os entrevistadores e os usuários deve focar principalmente o trabalho, e não aspectos de design do sistema. Segundo Beyer e Holtzblatt (1998), o entrevistador deve ser curioso e pode ser até um pouco intrometido.

Durante a observação do trabalho, é comum ocorrerem situações que lembrem o usuário de **eventos passados** relevantes para o entendimento do trabalho. Também durante o trabalho, o usuário recorre a artefatos (formulários, anotações, manuais etc.) que disparam conversas sobre como são utilizados, como foram criados e como sua estrutura apoiou seu uso numa situação particular. O entrevistador observa o



usuário realizando o trabalho em que a equipe está interessada e, sempre que julgar necessário, interrompe e discute com o usuário sobre algum aspecto relevante observado. Caso o usuário pegue um papel, formulário ou anotação, eles analisam juntos o artefato em detalhes. Utilizando esses artefatos para apoiar a conversa, o entrevistador descobre eventos que ocorreram há mais tempo.

Na **conclusão** da entrevista contextual, o entrevistador repassa rapidamente suas anotações e sumariza o que aprendeu, tentando não repetir literalmente o que aconteceu, mas dizendo o que ele identificou que é importante sobre o trabalho, para o usuário e para a organização. Isso permite que o usuário corrija e refine o entendimento do entrevistador, e deve ser encorajado a fazê-lo. Após a entrevista, toda a equipe de design trabalha com o entrevistador para interpretar os resultados da entrevista perante o problema de design.

Entrevistadores que observem múltiplos eventos e múltiplos usuários aprendem a ver as **estratégias** comuns que os usuários adotam ao trabalhar. Uma vez que as estratégias básicas são compreendidas, os designers podem começar a imaginar um sistema que apoie essas estratégias.

Caso o processo a ser observado seja extremamente longo, podemos tentar entrevistar um conjunto mais amplo de usuários, que assumam diferentes papéis em diversos pontos do processo. Podemos solicitar também que eles forneçam um relato aprofundado, em retrospectiva, de situações exemplares ou excepcionais.

Atividades

1. *Definição dos dados a serem coletados.* Imagine que você foi contratado para elaborar um sistema acadêmico de apoio a professores e alunos na Web. Para o professor, o sistema deve apoiar objetivos relacionados ao planejamento de aulas, divulgação de material didático e agendamento de trabalhos, provas e outras atividades, bem como o cálculo e a divulgação de notas. Já para o aluno, o sistema deve facilitar a organização do material e das atividades que precisa realizar em cada disciplina, a comunicação com o professor e com os colegas. Enumere os dados que deseja coletar, indicando por que cada dado é relevante para o projeto do sistema.
2. *Definição de quem fornecerá as informações.* Para o sistema acadêmico da atividade anterior, defina o perfil dos fornecedores de informação e outras possíveis fontes de informação, tais como manuais e normas. Relacione os dados que quer coletar (definidos na atividade anterior) com cada fonte. Para variar a amostra de pessoas a serem consultadas, pense em diferentes dimensões, tais como:

- tempo de experiência naquele papel (e.g., professor antigo/recente; aluno calouro/veterano);
 - área de atuação (e.g., tecnológica/humanas);
 - atitude com relação a tecnologias computacionais (e.g., “antenado”/resistente a tecnologias);
 - experiência com tecnologias computacionais (usa tecnologia simples/avanhada; usa há vários anos/não usa ainda; uso frequente/ocasional).
- 3.** *Elaboração de roteiro de entrevista.* Considerando o sistema acadêmico descrito anteriormente, elabore um roteiro de entrevista para o professor e outro para o aluno. Para isso, responda às seguintes perguntas:
- Quais os objetivos da entrevista?
 - Quem serão os entrevistados?
 - Que tópicos serão explorados na entrevista, e em que profundidade?
 - Quais tópicos seriam mais adequados para um questionário?
 - O quanto as perguntas elaboradas permitem obter os dados que você quer coletar?
 - Há perguntas compostas ou complexas, que precisam ser simplificadas ou segmentadas em múltiplas perguntas?
 - Há perguntas que dificultam o aprofundamento das respostas (por exemplo, perguntas que pedem respostas do tipo sim/não)?
 - Que cuidados foram tomados na elaboração das perguntas para que elas não induzam certas respostas?
 - Como aumentar a chance para que entrevistados reticentes falem mais sobre cada tópico?
- 4.** *Elaboração de questionário.* Considerando o sistema acadêmico descrito anteriormente, elabore um questionário para o professor e outro para o aluno. Além das perguntas enumeradas no item anterior, responda também:
- Quais tópicos seriam mais adequados para uma entrevista?
 - Para cada pergunta fechada, de onde extraio as possíveis respostas? Todo respondente conseguirá encaixar sua resposta numa das opções fornecidas?
 - Há perguntas excessivamente fechadas?
 - Há um número muito grande de perguntas abertas?
 - Há ambiguidades na enunciação das perguntas?



- O questionário apresenta claramente o seu objetivo e as instruções para preenchimento (e devolução, no caso de questionário impresso)?
 - Quanto tempo você estima que o respondente leve para completar o questionário? Esse tempo é adequado?
5. *Brainstorming.* Conduza duas pesquisas utilizando a técnica de *brainstorming*: uma para levantar os objetivos dos usuários e outra para levantar os itens de informação relacionados a esses objetivos.
 6. *Classificação de cartões.* Considerando o resultado de uma pesquisa de levantamento de objetivos dos usuários e dos itens de informações relacionados a esses objetivos (como sugerido na atividade anterior), crie um conjunto de cartões que permitam avaliar como os usuários organizariam esses itens de informação relacionados aos objetivos que querem atingir com o sistema.

6

Organização do Espaço de Problema

Objetivos do Capítulo

- Apresentar representações utilizadas para organizar o espaço de problema: personas e seus objetivos, cenários de problema e modelos de tarefas.
 - Discutir como essas representações permitem registrar as informações elicitadas durante o levantamento e a análise de objetivos e necessidades dos usuários.
-

Uma parte importante de qualquer método de análise ou design são os modelos e as representações utilizados para registrar o que foi aprendido ou definido. Eles definem um recorte no mundo de interesse, sob uma determinada perspectiva, com um determinado foco e em um determinado nível de detalhes (Hoover *et al.*, 2001). Como produto da etapa de análise, este capítulo apresenta diversas representações e modelos utilizados para registrar, organizar, refinar e analisar os dados coletados: perfil de usuário (Courage e Baxter, 2005; Hackos e Redish, 1998), personas e seus objetivos (Cooper *et al.*, 2007; Pruitt e Adlin, 2006; Cooper, 1999), cenários de análise ou de problema (Carroll, 1995; Carroll, 2000; Rosson e Carroll, 2002) e modelos de tarefas, que visam organizar e estruturar como os objetivos dos usuários podem ser alcançados (Diaper e Stanton, 2003; Card *et al.*, 1983; Paternò, 2000).

Na engenharia semiótica, a atividade de análise pode ser vista como meio de completar a primeira parte da metamensagem do designer para o usuário (de Souza, 2005a, p.25; Seção 3.8):

Este é o meu entendimento, como designer, de quem você, usuário, é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão.

Vemos a seguir diversas representações que podem auxiliar a registrar esse conhecimento adquirido dos usuários.

6.1 Perfil de Usuário

O primeiro passo para registrarmos nosso entendimento sobre os usuários é traçarmos um perfil deles. Quem são? Quais são seus objetivos? Além de nos ajudar a entender para quem estamos construindo o produto, o perfil de usuários também auxilia no recrutamento de participantes para futuras atividades de análise e avaliação (Courage e Baxter, 2005; Hackos e Redish, 1998).

Perfil de usuário é uma descrição detalhada das características dos usuários cujos objetivos devem ser apoiados pelo sistema sendo projetado. Como visto na Seção 5.2, devemos identificar as características de interesse (e.g., cargo, função, experiência, nível de instrução, atividades principais, faixa etária etc.) e conduzir um estudo (e.g., através de entrevistas e questionários) para coletar os dados dos usuários. A partir dos dados coletados, podemos agregar os valores em grupos e faixas na qual os usuários se encaixam (e.g., idade 18-25), e assim traçar os perfis de usuários com características semelhantes e calcular a proporção de usuários que se encaixam em cada perfil. A elaboração de um perfil de usuário é um processo iterativo. Em geral,

um designer começa seu trabalho com uma ideia inicial de quem são seus usuários, mas essa ideia não costuma ser suficientemente detalhada e pode até ser apenas uma impressão equivocada.

As características de um perfil de usuário podem ser priorizadas conforme o produto e projeto em questão. Nesse caso, a maior parte dos recursos para capturar informações deve ser destinada a essas características-chave do seu produto. Em geral, um perfil de usuário é caracterizado por dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto (Hackos e Redish, 1998; Courage e Baxter, 2005). Esses tipos de dados foram enumerados na Seção 5.2.

Uma vez que a faixa de respostas para cada uma das características e a porcentagem de usuários nessa faixa tiverem sido determinadas, podemos categorizar seus usuários em grupos, com base em suas semelhanças. Alguns grupos comuns são definidos por: *idade* (criança, jovem, adulto, terceira idade etc.); *experiência* (leigo/novato, especialista); *atitudes* (tecnófilos, tecnófobos); e *tarefas primárias* (compra, venda). O Exemplo 6.1 apresenta um quadro representando dois perfis de professores universitários.

Exemplo 6.1 – Quadro para análise de perfil de coordenadores de curso

papel	coordenador	
	A	B
Percentual de professores no perfil	47%	43%
Número de professores no perfil (total: 15)	7	8
Faixa etária	[30,40)	[40,50)
Quanto tempo como professor (anos)	[5,10)	[10,15)
Frequência de uso de tecnologia <i>constante:</i> 5 [várias vezes ao dia] <i>alta:</i> 4 [todo dia] <i>média:</i> 3 [4-6 vezes/semana] <i>ocasional:</i> 2 [1-3 vezes/semana] <i>baixa:</i> 1 [menos de 1 vez/semana]	5	5
Experiência com tecnologia <i>alta:</i> 5 - faz tudo sem ajuda <i>baixa:</i> 1 - precisa de muita ajuda	5	4
Atitude perante tecnologia <i>adora:</i> 5 <i>odeia:</i> 1 (só usa porque é obrigado)	5	4
Estilo de aprendizado	aprende fazendo; busca na Web	lê manual; pergunta ao colega

papel	coordenador	
perfil	A	B
Aplicações mais utilizadas	[1. e-mail, 2. leitor RSS, 3. ed. texto, 4. ed. slides, 5. ferramenta de busca]	[1. e-mail, 2. ed. texto. 3. ed. slides, 4. ferramenta de busca]
Opinião sobre sistema atual		
adora: 5 odeia: 1	3	4
útil: 5 inútil: 1	5	5
funcionalidades necessárias: tem: 5 não tem: 1	3	3
eficiente: 5 ineficiente: 1	4	5

Nesse exemplo, observamos que nem todos os professores se encaixam nesses dois perfis. Alguns (10% do total) não se encaixam em nenhum grupo que possa ser considerado homogêneo.

Os perfis de usuários facilitam a criação de personas, descritas na próxima seção.

6.2 Personas

Uma **persona** é um personagem fictício, arquétipo hipotético de um grupo de usuários reais, criada para descrever um usuário típico (Cooper *et al.*, 2007; Pruitt e Adlin, 2006; Cooper, 1999). É utilizada principalmente para representar um grupo de usuários finais durante discussões de design, mantendo todos focados no mesmo alvo. As personas são definidas principalmente por seus *objetivos*, que são determinados num processo de refinamentos sucessivos durante a investigação inicial do domínio de atividade do usuário. Em geral, começamos com uma aproximação razoável e convergimos numa população plausível de personas.

Cooper afirma que, em vez de ampliarmos a funcionalidade do produto para acomodar a maior parte das pessoas, devemos tentar projetar especificamente para uma *única* persona. Ele argumenta que, a cada vez que estendemos a funcionalidade para incluir mais um grupo de usuários, colocamos mais um obstáculo no caminho de todos os outros usuários. Em outras palavras, os recursos que agradam alguns usuários interferem na satisfação e no desempenho de outros. Segundo ele, tentar agradar muitos pontos de vista diferentes pode arruinar um bom produto.

Segundo ele, o próprio termo “usuário” é problemático, pois sua imprecisão o torna pouco útil. Ele acredita que esse termo genérico denota um “usuário elástico”, que precisa se contorcer para se adaptar às necessidades do momento. Por exemplo, alguns sistemas tratam o usuário ora como um usuário iniciante, forçando-o a passar por diversos assistentes para realizar sua tarefa, ora como um especialista, que preci-

sa entender e configurar múltiplos parâmetros obscuros e interdependentes, muitas vezes conforme a conveniência dos programadores. No entanto, o objetivo principal é projetar sistemas que se adaptem às necessidades do usuário. E usuários reais não são elásticos. Sendo assim, os designers nunca devem ser vagos e dizerem que o seu programa é projetado para “o usuário” ou que será “amigável”. Em vez disso, Cooper advoga falar de um usuário bem específico: uma *persona*.

Para definir uma persona, Courage e Baxter (2005) enumeraram os seguintes elementos característicos:

- *identidade*: dê a uma persona nome e sobrenome. Forneça uma idade e outros dados demográficos que seriam representativos do perfil do usuário. Inclua também uma foto, para tornar a persona ainda mais realista e memorável;
- *status*: defina se esta persona é primária, secundária, outro *stakeholder* ou representa um antusuário do seu sistema. Um antusuário é alguém que não vai utilizar o produto e, portanto, não deve influenciar as decisões de projeto;
- *objetivos*: quais são os objetivos desta persona? Não se limite a objetivos relacionados ao seu produto específico;
- *habilidades*: qual é a especialidade da sua persona? Isso inclui educação, treinamento e competências específicas. Novamente, não se limite a detalhes relacionados ao seu produto específico;
- *tarefas*: em linhas gerais, quais as tarefas básicas ou críticas que a persona realiza? Qual é a frequência, importância e duração dessas tarefas? Deixe as informações mais detalhadas sobre como as tarefas são realizadas para os cenários (veja a Seção 6.3);
- *relacionamentos*: entender com quem a persona se relaciona é importante, pois ajuda a identificar outros *stakeholders*;
- *requisitos*: de que a persona precisa? Inclua citações que ajudam a dar mais vida a essas necessidades;
- *expectativas*: como a persona acredita que o produto funciona? Como ela organiza as informações no seu domínio ou trabalho?

Embora personas sejam fictícias (i.e., não correspondem a uma pessoa real), elas são definidas com rigor e detalhes para representar usuários “típicos”. Elas são derivadas de um processo de investigação que levanta as características dos usuários e descreve seus perfis. Apenas seus nomes e detalhes pessoais são inventados. Quanto mais específicas forem as personas, mais eficientes elas serão como *ferramentas de design e comunicação*.

O Exemplo 6.2 ilustra duas personas de um sistema de apoio acadêmico.

Exemplo 6.2 – Personas de um sistema de apoio acadêmico

Paulo Correa, técnico de suporte – “comandos para máxima eficiência”

Paulo Correa, de 43 anos, trabalhou durante muitos anos consertando e configurando computadores. Atualmente, trabalha na universidade AprendaMais, configurando PCs e as contas dos alunos de cada turma. Ele fez um curso de administração de rede, mas prefere aprender fazendo do que assistindo a cursos ou lendo manuais. Quando tem alguma dúvida, ele faz uma busca na Internet por informações que lhe ajudem a resolver os seus problemas. Usuário “das antigas”, Paulo prefere utilizar linguagem de comando do que assistentes em interface gráfica, pois acredita que assim seja mais eficiente. Sempre que uma tarefa se repete com frequência, ele tenta elaborar um script ou fazer alguma configuração que acelere o seu trabalho.



Todo início de período, Paulo precisa configurar dezenas de contas para cada turma, com diferentes perfis, fornecendo acesso diferenciado para alunos regulares, monitores, instrutores e coordenadores de cada disciplina. Precisa atender aos pedidos dos professores sobre o que deve estar disponível na intranet de cada disciplina (e.g., publicação de material didático; fórum de discussão; recebimento de trabalhos dos alunos; cadastramento de notas; pedidos de revisão). Seu maior objetivo é atender aos professores com a maior eficiência possível. Para isso, é importante ele poder acessar o sistema onde quer que esteja, no horário que for, para realizar qualquer tarefa remotamente.

Lúcio Marques, professor – “é mais prático usar apenas o básico”

Lúcio Marques é professor da universidade AprendaMais há cinco anos e já lecionou diversas disciplinas diferentes. Ele tenta sempre aproveitar ao máximo o que já tiver utilizado em outros períodos, mas sempre busca atualizar seu material com conceitos extras e novos exemplos reais, que ele lê em blogs de profissionais da área.



Lúcio gostaria de poder, ele próprio, configurar o sistema, mas como sua preocupação principal é lecionar uma boa aula, não tem tempo para decifrar as dezenas de funcionalidades ocultas nos diversos menus do sistema. Ele costuma sempre solicitar ao suporte a configuração básica inicial com módulos apenas para divulgação de material e fórum de discussão. Dependendo do perfil da turma, ele pede para o suporte acrescentar mais funcionalidades ao longo do curso, mas isso raramente ocorre.

Segundo Cooper (1999), um problema com os “representantes” de usuários reais é que um usuário real pode ter idiossincrasias que interferem com o processo de design e reduzem sua capacidade de representar o grupo de usuários almejado. Além disso, os usuários não podem estar presentes em todas as etapas do projeto. Por outro lado, falar genericamente de “o usuário”, que não é definido com precisão, dificulta o estabelecimento de uma visão de design compartilhada. Quando cada designer possui sua própria visão de quem é o usuário, “o usuário” se torna um alvo móvel — pode mudar de especialista para novato e dali para a tia do desenvolvedor, numa única discussão.

Uma persona assume uma solidez tangível que coloca os pressupostos de design em perspectiva. À medida que uma persona perde sua elasticidade, podemos identificar suas habilidades, suas motivações e o que ela quer alcançar. Munidos desse conhecimento, podemos então examiná-la à luz do domínio do software para ver se ela realmente é um arquétipo de usuário. Dar um nome à persona é uma parte importante da sua elaboração, para que ela se torne um indivíduo concreto na mente dos designers.

Cooper (1999) considera as personas como a ferramenta de design mais poderosa utilizada por ele e sua equipe. São a base para todo o design orientado a objetivos apresentado na Seção 4.3.6. Elas tornam claros os objetivos dos usuários, para que possamos ver o que o produto deve fazer — ou pode deixar de fazer. As personas ajudam a equipe de design a justificar suas decisões de design para os desenvolvedores e gerentes. Projetar para um conjunto pequeno de personas aumenta as chances de a equipe de design acertar o alvo. Uma persona pode ser utilizada em reuniões como uma ferramenta de discussão (e.g., “Lucia nunca utilizaria essa funcionalidade”), em avaliações por inspeção (Seção 9.6), *storyboarding*, encenações (*role-playing*) e outras atividades voltadas à qualidade de uso do sistema. Finalmente, personas também podem ajudar novos membros da equipe a aprender rapidamente sobre quem são os usuários.

É essencial que todos na equipe de design não apenas conheçam o elenco de personas, mas que cada persona se torne como uma pessoa real, como membro da equipe. Cada designer deve insistir em expressar todas as questões de design em termos de personas, através de seus nomes. Todos devem evitar falar em “o usuário”. À medida que começamos a desenvolver ideias para soluções de design, constantemente as avaliamos perante as personas. Quando conseguimos nos colocar no lugar de uma persona e examinar um produto ou uma tarefa, conseguimos apreciar melhor se o design foi ou não bem-sucedido em fazer essa persona feliz (Cooper, 1999).

Segundo Courage e Baxter (2005), devemos criar pelo menos uma persona por papel de usuário (e.g., ao menos uma para o professor e outra para o aluno, num ambiente acadêmico). Cada projeto possui seu próprio **elenco de personas**, que consiste de três a 12 personas distintas. Não concebemos o design para todas elas, mas todas são úteis em articular parte da população de usuários. Algumas são definidas apenas para tornar claro que *não* estamos projetando para elas; são as antiperonas.

Cada elenco de personas possui ao menos uma **persona primária**. Trata-se do indivíduo que é o foco principal do design. Para ser primária uma persona é alguém que *tem de* ser satisfeita, mas que não pode ser satisfeita por uma interface projetada

para uma outra persona qualquer. Em geral, cada persona primária requer uma interface distinta e única (Cooper, 1999).

Courage e Baxter (2005) apontam um cuidado na escolha do número de personas elaboradas. É importante que as personas sejam memoráveis e, para isso, o elenco de personas deve ser reduzido. Se houver muitas personas para representar os grupos de usuários, elas vão se misturar na mente dos designers e desenvolvedores, e com isso reduzimos os benefícios dessa técnica. No entanto, o elenco deve cobrir os principais grupos de usuários, para ajudar a desenvolver um produto que funciona para todos. Ao nos limitarmos a uma única persona, podemos deixar de fora dados valiosos de usuários finais que não correspondam a um mesmo grupo. Uma recomendação comum é que o elenco de personas inclua três personas primárias.

Objetivos das Personas

Um “bom design de interação” só tem sentido no contexto de uma pessoa utilizando o sistema com algum objetivo. E não há objetivos sem pessoas. É por isso que os elementos-chave do processo de design dirigido por objetivos são: objetivos e pessoas, representadas pelas personas.

Objetivos não são a mesma coisa que tarefas. Como visto na Seção 4.3.6, um objetivo é uma condição final, ao passo que uma tarefa é um processo intermediário necessário para atingir o objetivo. Existe uma maneira simples de fazer a distinção entre tarefas e objetivos. As tarefas mudam com a tecnologia, mas os objetivos são bem mais estáveis. Para descobrir os objetivos, Cooper (1999) sugere realizar sessões de *brainstorming* considerando um “computador mágico” (Seção 5.5.4), que não possui qualquer restrição. Ele acredita que esse exercício ajuda a fazer a distinção entre objetivos e tarefas.

Muitos desenvolvedores se aproximam do design perguntando “Quais são as tarefas?”. Embora isso possa ajudar a resolver um problema, em geral não ajuda a produzir uma solução brilhante e pode até mesmo não satisfazer o usuário. Projetar a partir de tarefas em vez de objetivos é uma das principais causas de interação inefficiente e frustrante. Perguntar “Quais são os objetivos de Marta (uma persona)?” nos permite desfazer a confusão e criar um design mais adequado e satisfatório. Quando designers analisam objetivos para resolverem problemas, eles geralmente encontram soluções bem diferentes, e muitas vezes melhores e mais criativas (Cooper, 1999).

Em 1999, Cooper descreveu três tipos de objetivos: objetivos pessoais, corporativos e práticos. **Objetivos pessoais** são simples, universais e pessoais: manter sua dignidade e não se sentir estúpido, não cometer erros, conseguir realizar uma quantidade de trabalho razoável, se divertir ou ao menos não ficar completamente ente-

diado. Muitas pessoas não admitem o objetivo de “não se sentir estúpido”, pois são orgulhosas, inteligentes e costumam gostar de enfrentar desafios e dominar situações complexas. Mas esse é um dos principais objetivos pessoais.

Objetivos corporativos típicos são: aumentar lucro, aumentar dominação de mercado, derrotar a competição, contratar mais pessoas, oferecer mais produtos e serviços, abrir a empresa para o mercado de ações. O designer utiliza esses objetivos para manter o foco nas questões mais amplas e evitar se distrair com tarefas ou outros objetivos falsos.

Os **objetivos práticos** fazem a ponte entre os objetivos corporativos e os do usuário individual. A empresa quer que todos trabalhem bastante para maximizar o retorno. Um objetivo prático de processar as requisições do cliente conecta os objetivos corporativos de maior lucro com o objetivo pessoal do usuário de ser produtivo. Alguns objetivos práticos típicos são: evitar reuniões, processar as requisições do cliente, registrar um pedido de um cliente.

Segundo Cooper, os objetivos mais importantes são os objetivos pessoais. Trata-se de uma pessoa real interagindo com o seu produto, e não uma corporação abstrata. Seus usuários vão se esforçar para atingir os objetivos corporativos, mas somente após seus próprios objetivos pessoais terem sido atingidos. Embora uma persona não precise atingir todos seus objetivos práticos de uma vez, ela nunca deve ter um de seus objetivos pessoais violados.

Norman (2003) propôs três níveis de processamento cognitivo: visceral, comportamental e reflexivo. Com base nesses níveis de processamento, Cooper, Reimann e Cronin (2007) definiram três tipos de objetivos do usuário: objetivos de experiência, finais e de vida, respectivamente.

Os **objetivos de experiência** (*experience goals*) dizem respeito a *como* o usuário deseja se *sentir*, por exemplo: sentir-se no controle, se divertir, relaxar, permanecer alerta, manter o foco ou não se sentir estúpido. Eles estão relacionados com o *processamento cognitivo visceral*, responsável por perceber e reagir rapidamente aos estímulos do ambiente através dos cinco sentidos — visão, audição, tato, paladar e olfato — e o sistema motor humano. Esse processamento nos auxilia a decidir rapidamente sobre o que é bom, ruim, agradável, desagradável, seguro, perigoso etc. Apoiar os objetivos de experiência exige maior atenção às características físicas do sistema interativo para afetar as primeiras reações e sentimentos do usuário de forma esperada.

Os **objetivos finais** (*end goals*) representam o *que* o usuário deseja *fazer*, como, por exemplo: manter contato com amigos e familiares, concluir sua lista de tarefas no final de um dia de trabalho, encontrar músicas que ele gosta de ouvir ou ser notificado de qualquer problema antes que se torne crítico. Eles têm forte relação com o *proces-*

samento cognitivo comportamental, que permite gerenciar comportamentos simples e cotidianos. Atender aos objetivos finais significa projetar um sistema interativo capaz de estar alinhado com e complementar os comportamentos do usuário.

Os **objetivos de vida** (*life goals*) estão relacionados com *quem* o usuário deseja ser, como, por exemplo: viver uma boa vida, ter sucesso em suas ambições, se tornar um especialista em determinado assunto ou atividade ou ser popular e respeitado pelos colegas. Esses objetivos estão relacionados com o *processamento cognitivo reflexivo*, envolvendo considerações e reflexões conscientes sobre experiências anteriores e conhecimentos adquiridos. Para dar atenção aos objetivos de vida, o projeto de um sistema interativo deve integrar os desejos e as expectativas de longo prazo dos usuários com a experiência proporcionada durante o uso desse sistema. Durante o uso, essa integração ocorre através de um processo reflexivo do usuário que atribui significado e valor de longo prazo à experiência proporcionada pelo uso.

Cooper (1999) alerta para o perigo de se definir como objetivos o que ele chama de **falsos objetivos**. Tratam-se de *meios* para se atingir um fim, e não objetivos finais. Os objetivos verdadeiros são sempre um fim. Cooper considera como um exemplo de falso objetivo “rodar num navegador”, quando o que o usuário quer mesmo é ter acesso ao sistema em qualquer lugar, o que pode ser realizado de outras formas. Sempre que suspeitarmos que um objetivo seja falso, devemos investigá-lo, perguntando por que a persona gostaria de atingir aquele objetivo, até chegarmos ao objetivo verdadeiro.

O Exemplo 6.3 descreve uma terceira persona do sistema de apoio acadêmico, desta vez com os seus objetivos destacados no final da descrição.

Exemplo 6.3 – Objetivos de uma persona

Marta Batista, professora – “cada turma é uma turma”

Marta Batista é professora da universidade AprendaMais há dois anos. Embora lecione apenas duas disciplinas diferentes, ela gosta de configurar o sistema sob medida para cada turma, pois sente que isso contribui para a qualidade do curso.



Elá não se importa em ler instruções sobre como proceder para atingir um objetivo, mas gosta-ria que essas instruções estivessem no ponto em que são necessárias, em vez de ter de buscar num manual separado. Marta gostaria de agilizar o seu trabalho, com acesso mais rápido às funcionalida-des que utiliza com frequência, como divulgar material, ver se há novidades no fórum de discussão, descobrir quem já entregou cada trabalho e quem está devendo, além de divulgar as correções dos trabalhos dos alunos.

Objetivos pessoais:

- não perder tempo e
- trabalhar da melhor maneira possível

Objetivos práticos:

- utilizar um sistema adequado a cada disciplina e a cada turma;
 - divulgar material didático;
 - acompanhar e participar das discussões no fórum da disciplina;
 - acompanhar a entrega dos trabalhos dos alunos; e
 - divulgar as correções dos trabalhos dos alunos.
-

6.3 Cenários

Como visto na Seção 4.3.5, um cenário é basicamente uma história sobre pessoas realizando uma atividade (Rosson e Carroll, 2002). 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. Os cenários podem ser utilizados em diversas etapas do processo, com diferentes objetivos: para descrever uma história num domínio de atividade, visando capturar requisitos e auxiliar no entendimento da atividade, levantar questões sobre a introdução de tecnologia, explorar diferentes soluções de design e avaliar se um produto satisfaz a necessidade dos seus usuários (Rosson e Carroll, 2002; Cooper, 1999). Além de poderosos, os cenários requerem menos custo e tempo quando comparados com modelos e protótipos complexos, o que os torna uma ferramenta importante em todo o processo de design de IHC.

Os cenários descrevem o comportamento e as experiências dos atores. Cada ator possui objetivos que dirigem as tarefas que ele realiza. Um cenário possui um enredo, que inclui sequências de ações e eventos: o que os usuários fazem, o que acontece com eles, que mudanças ocorrem no ambiente, e assim por diante. Essas ações e eventos podem ajudar, atrapalhar ou ser irrelevantes para o atingimento do objetivo.

Em geral, cada cenário apresenta um ator principal e um objetivo principal. Tal objetivo pode ser desdobrado em subobjetivos, numa atividade de planejamento que se passa na cabeça dos atores. Quando essa atividade mental for importante para uma situação, o cenário pode incluir informações sobre planejamento e avaliação das ações realizadas. Cada cenário costuma ter um título que descreve brevemente a situação, sem muitos detalhes; os atores que participam do cenário; uma breve descrição da situação inicial em que os atores se encontram; e referências a outros cenários que permitam aos atores atingir os mesmos objetivos de diferentes maneiras. Segundo Rosson e Carroll (2002), os elementos característicos de um cenário são:

- *ambiente ou contexto:* detalhes da situação que motivam ou explicam os objetivos, ações e reações dos atores do cenário;
- *atores:* pessoas interagindo com o computador ou outros elementos do ambiente; características pessoais relevantes ao cenário;

- *objetivos*: efeitos na situação que motivam as ações realizadas pelos atores;
- *planejamento*: atividade mental dirigida para transformar um objetivo em um comportamento ou conjunto de ações;
- *ações*: comportamento observável;
- *eventos*: ações externas ou reações produzidas pelo computador ou outras características do ambiente; algumas delas podem ser ocultas ao ator mas importantes para o cenário;
- *avaliação*: atividade mental dirigida para interpretar a situação.

A descrição de um ator no cenário deve incluir as suas características pessoais que forem relevantes ao cenário. Caso os cenários sejam utilizados em conjunto com pessoas, os atores dos cenários são as pessoas elaboradas previamente.

Na atividade de análise, em particular, utilizamos **cenários de análise** (por vezes denominados cenários de problema), histórias sobre o domínio de atividade do usuário, tal como ele existe antes da introdução de tecnologia (Rosson e Carroll, 2002). O Exemplo 6.4 ilustra dois cenários de análise relacionados para um sistema de administração acadêmica.

Exemplo 6.4 – Cenários de análise

Cadastro de projetos finais com coorientador externo não cadastrado

Atores: Joana Marinho (secretária), Fernando Couto (aluno)

Na primeira semana de aula, Joana Marinho, secretária do curso de Engenharia Ambiental, precisa cadastrar entre vinte e trinta projetos finais dos alunos no período atual. Um projeto final é um trabalho individual de um aluno sob a orientação de um ou dois professores. Cada aluno preenche um formulário impresso e o entrega na secretaria. Em vez de cadastrar os projetos finais à medida que são entregues, Joana prefere juntar vários para cadastrá-los de uma vez, pois acha que assim perde menos tempo. Joana confere o formulário, verificando se o aluno definiu seu(s) orientador(es) e o título e formato de entrega do seu trabalho (e.g., relatório, software), para então cadastrar os dados no sistema. No caso do aluno Fernando Couto, após informar o título do trabalho e o orientador principal, Joana descobre que o seu coorientador, que não é professor regular do curso, não está cadastrado no sistema. Ela interrompe o cadastramento, pega o e-mail de Fernando da sua ficha cadastral (impressa) e lhe envia uma mensagem solicitando os dados do seu coorientador externo: nome completo, CPF e e-mail para contato. No dia seguinte, Joana recebe a mensagem de resposta de Fernando com os dados solicitados. Ela então reinicia o cadastro do projeto final de Fernando, sem poder aproveitar o que havia feito na véspera. Ao terminar o cadastro, Joana entra no seu sistema de correio eletrônico e envia uma mensagem para todos os envolvidos (aluno e coorientadores), para que eles confirmem os dados cadastrados e confirmem sua participação no projeto.

Confirmação do cadastro de projetos finais

Atores: Marcos Correa (professor)

Marcos Correa, orientador de Fernando, é um professor requisitado que orienta diversos alunos ao mesmo tempo. Todo início de período ele recebe diversas mensagens informando-lhe sobre os

projetos finais cadastrados sob sua supervisão. Infelizmente, as mensagens não apresentam os dados cadastrados, então ele precisa entrar no sistema para conferir os dados. Além disso, mesmo já estando no sistema e verificando um projeto, ele não consegue ver os dados dos outros projetos pendentes. Sendo assim, tem de ficar alternando entre o seu cliente de e-mail e o sistema acadêmico, e às vezes ele acaba visitando os dados de um mesmo projeto mais de uma vez.

Análise dos cenários

No primeiro cenário, observamos alguns pontos que podem ser considerados problemáticos e devem ser considerados em um reprojeto de IHC:

- a necessidade de transcrição para o sistema de dados preenchidos pelo aluno em um formulário impresso;
- a falta de integração do sistema com as informações dos alunos (o e-mail de Fernando deve ser buscado em uma ficha impressa);
- a incapacidade de enviar uma mensagem através do sistema para as pessoas envolvidas no cadastro de um projeto final (Joana precisa acessar seu sistema de correio eletrônico para enviar uma mensagem para o orientador, coorientador e aluno);
- a impossibilidade de o professor acessar outros projetos com pendências, uma vez que já esteja no sistema.

Os cenários destacam objetivos sugeridos pela aparência e o comportamento de um sistema; o que as pessoas tentam fazer com ele; que procedimentos são adotados, não são adotados e realizados com sucesso ou falha; e quais interpretações as pessoas fazem do que acontece com elas (Rosson e Carroll, 2002). Por serem ricos em contextualização, os cenários permitem explorar com detalhes os impactos do produto nas atividades e nos processos de trabalho dos usuários.

Cenários podem incluir exceções. Que eventos importantes acontecem raramente? Ao compreendermos situações extremas ou infrequentes que os usuários enfrentam, podemos identificar situações em que o produto pode se tornar problemático. Também podemos identificar características-chave que podem beneficiar seus usuários finais (Cooper, 1999).

Uma diferença importante entre cenários e casos de uso utilizados em engenharia de software é que os cenários podem enfatizar mudanças de objetivos, planos e entendimentos (Rosson e Carroll, 2002). Trata-se da descrição de um ator específico realizando ações específicas. As ações potenciais e os fluxos alternativos que são descritos num caso de uso só devem ser incluídos num cenário caso façam parte do processo de planejamento ou avaliação dos atores daquele cenário específico. Em outras palavras, cada cenário descreve apenas um dos caminhos descritos em um caso de uso. Carroll (2000) acredita que diagramas e especificações abstraem a riqueza do uso do sistema em situações reais, podendo se tornar um obstáculo tanto para uma exploração mais ampla do espaço de design quanto para o envolvimento e a participação dos clientes e futuros usuários do sistema no processo de design.

As técnicas baseadas em cenários vêm ganhando grande aceitação por parte dos projetistas e seus clientes. Isso se deve principalmente à natureza da atividade de design de software, que não é uma forma fácil ou rotineira de resolução de problemas. Problemas de design costumam ser especificados de forma incompleta, as soluções não são conhecidas *a priori*, envolvem equilibrar tensões 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, através da concretização de situações de uso, explorar a complexidade da resolução de problemas de design (Carroll, 2000).

Algumas críticas ao uso de cenários se referem à frequência com que ficam incompletos ou ambíguos. Com o objetivo de elaborar cenários mais completos, descobrindo informações que tenham sido omitidas, Carroll *et al.* (1994) propõem a técnica de **questionamento sistemático**. Trata-se de segmentar o cenário em proposições e investigar mais profundamente cada proposição a partir de um conjunto geral de perguntas, cujas respostas, por sua vez, geram novas proposições, repetindo o ciclo até que o conjunto de proposições seja considerado suficientemente completo. Os tipos gerais de perguntas que eles sugerem são: *Por quê? Como? O que é? X pode ser feito da forma Y? X faz parte de Y?* Associado a cada questão eles indicam o conteúdo esperado das respostas à questão, conforme ilustrado na Tabela 6.1.

Tabela 6.1 Questões para refinamento de cenários (Carroll *et al.*, 1994)

questão	conteúdo das respostas
questões exploratórias	
Por que ...?	<ul style="list-style-type: none"> ▪ condições para a realização da atividade ▪ consequências da atividade ▪ estados e eventos anteriores ou posteriores à atividade
Como ...?	<ul style="list-style-type: none"> ▪ detalhes sobre a sequência de ações que compõem uma atividade ▪ pode revelar também objetos que não constavam do cenário original
O que é ...?	<ul style="list-style-type: none"> ▪ objetos e seus atributos, organizados em uma hierarquia ou outro modelo de dados
questões de verificação	
<X> pode ser feito da maneira <Y>?	<ul style="list-style-type: none"> ▪ respostas sim ou não ▪ servem para avaliar se uma ação ou atributo está bem definido e localizado no nível certo da hierarquia
<X> faz parte de <Y>?	

Para enriquecer as informações dos cenários, elaborar narrativas mais detalhadas e assim permitir uma análise mais profunda, o designer pode responder também perguntas mais específicas (adaptadas de Silveira *et al.*, 2004; Aureliano, 2007), relacionadas com os elementos de um cenário (Tabela 6.2).

Tabela 6.2 Perguntas utilizadas para refinar cada elemento de um cenário ou auxiliar a análise

elemento	perguntas
objetivo	Por que os atores querem ou precisam alcançar esse objetivo? Quais as precondições para esse objetivo? De que informações ou conhecimento os atores precisam para realizar esse objetivo? Quais informações são (ou deveriam ser) criadas, consumidas, manipuladas ou destruídas pelo alcance do objetivo? Que outros objetivos (de quais atores) estão relacionados a esse?
ambiente	Em que situações o cenário ocorre (quando, onde e por quê)? Que dispositivos e outros recursos (inclusive tempo) estão disponíveis para o alcance do objetivo? Quais pressões existem para o alcance do objetivo? Quais são as tecnologias utilizadas no ambiente de trabalho? Como os usuários as utilizam?
ator(es)	Quem pode alcançar o objetivo descrito no cenário? Quais características dos atores lhes auxiliam ou atrapalham em alcançar o objetivo? De quem depende o alcance do objetivo? Quem fornece as informações necessárias ao alcance do objetivo? Quem depende do resultado do objetivo? Quem consome quais informações geradas pelo alcance do objetivo? Quem precisa ser notificado da conclusão (bem-sucedida ou malsucedida) do objetivo?
planejamento	Como os atores alcançam o objetivo atualmente? Como gostariam de fazê-lo? Quais são as estratégias alternativas para realizar o objetivo? Quando e por que cada estratégia é (ou deveria ser) seguida? Os atores conhecem todas as estratégias disponíveis? Que decisões os atores precisam tomar a cada momento? De que maneira o ambiente e o sistema auxiliam ou impedem que os atores tomem decisões adequadas? Quais as consequências de uma decisão errada? Que ações realizam? Como essas ações estão relacionadas? Em que ordem os atores precisam realizar as ações? Gostariam de realizá-las em outra ordem?

elemento	perguntas
ação	Quais as precondições para essa ação? Como os atores as realizam? Há diferentes formas de realizá-la? Qual deve ser adotada em que situações? Os atores gostariam de fazer isso de outra maneira? Como o fariam? De que informações ou conhecimento os atores precisam para realizar essa ação? Que recursos estão disponíveis para realizá-la? Quais problemas ou dificuldades podem surgir ao realizá-la? Como podem ser resolvidos ou contornados? Quais erros podem ser cometidos ao realizá-la? Como podem ser desfeitos? Quais suas consequências? Quais informações são (ou deveriam ser) criadas, consumidas, manipuladas ou destruídas pela realização da ação? Quais eventos são (ou deveriam ser) disparados pela realização da ação?
evento	Quais eventos disparam a necessidade de alcançar o objetivo? Quais eventos são (ou deveriam ser) disparados pela conclusão desse objetivo?
avaliação	[de uma ação] Como os atores conseguem saber se uma ação foi concluída e realizada com sucesso? [do objetivo] Como os atores conseguem saber se o objetivo foi concluído e alcançado com sucesso? [do objetivo] Qual é o resultado do alcance do objetivo?

Com cenários bem elaborados, os designers têm melhores condições de investigar quais atividades dos usuários poderiam ser executadas de forma mais eficiente, o que se pode modificar nos processos e sistemas atuais e como um sistema computacional interativo novo ou reprojeto pode melhor apoiar essas atividades, de forma a se encaixarem adequadamente no ambiente de trabalho.

Para assegurar que os cenários sejam representativos do produto, Cooper (1999) sugere que o conjunto de cenários enderece os cinco tópicos a seguir:

- *ciclo de vida do processo*: um processo em ampla escala deve ser decomposto em diversos passos, e cada passo pode ser representado por um cenário diferente;
- *segmentos de público*: seus cenários devem examinar os diferentes tipos de usuário e suas experiências, objetivos habilidades, padrões de uso etc.;
- *funções do produto*: um produto pode ter diferentes funcionalidades, que apoiam tarefas diferentes e não relacionadas. Seu conjunto de cenários deve cobrir a gama de funcionalidades que seu produto apoie;

- *variantes de uma classe de situações de tarefa:* uma simples tarefa (ou objetivo) pode ser realizada de diferentes formas. Idealmente, o conjunto de cenários deve examinar essas variações para cada tarefa;
- *métodos para realizar uma tarefa:* uma única tarefa é selecionada e diferentes funcionalidades e métodos para realizá-la são examinados.

Cooper (1999) comenta, no entanto, que a elaboração de cenários pode consumir bastante tempo. Segundo ele, não é necessário criar cenários para todas as tarefas e situações que os usuários possam enfrentar. Em vez disso, devemos inicialmente elaborar cenários para as tarefas principais, e para as tarefas secundárias à medida que o tempo permitir. Ele destaca os **cenários de uso diário**, que envolvem as principais ações que os usuários vão realizar, e com a maior frequência. Em geral, a maioria dos usuários possui poucos cenários de uso diário. Esses cenários são os que precisam de um apoio mais robusto do sistema sendo projetado. Os novos usuários precisam dominar esses cenários rapidamente, e para isso as instruções devem ser claras e didáticas, embutidas no próprio sistema de forma contextualizada. Em contrapartida, como são utilizados com frequência, à medida que os usuários se tornarem experientes vão requerer atalhos e possibilidades de customização para que a interação se torne adequada às suas preferências e estilo individual. Esses cenários são em grande parte responsáveis pelo sucesso do produto.

Algumas tarefas são realizadas periodicamente mas com pouca frequência como, por exemplo, gerar um relatório anual. Elas também requerem instruções claras e didáticas embutidas no produto, mas geralmente podem prescindir de atalhos, pois raramente seriam aprendidos pelos usuários.

Personas e cenários podem ser utilizados no âmbito da engenharia semiótica para ajudar a definir a primeira parte da metamensagem designer–usuário: “*Este é o meu (designer) entendimento de quem você (usuário) é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê.*” (de Souza, 2005a, p. 25). Para melhor coletar as informações necessárias à elaboração da metamensagem, Paula (2003) propõe complementar os cenários com perguntas que revelem a intenção do designer ao elaborar os cenários e identifiquem os pontos que o designer almeja descobrir, explorar ou ratificar junto aos usuários. O designer pode gerar uma lista única de perguntas para um conjunto de cenários, numeradas para que estes possam se referir a cada pergunta individualmente, incluindo o número da pergunta entre colchetes, logo após o trecho correspondente no cenário, conforme ilustrado pelo Exemplo 6.5.

Exemplo 6.5 – Perguntas exploradas nos cenários

Conjunto de perguntas e parte do cenário do Exemplo 6.4 anotado com referências às perguntas.

1. Quem pode/deve cadastrar os dados dos projetos finais no sistema?
2. Quando são cadastrados os projetos finais?
3. Quem fornece os dados dos projetos finais?
4. Quais dados de projeto final devem ser cadastrados?
5. Quantos projetos são cadastrados a cada período?
6. Quem pode orientar um trabalho final?
7. Que dados são necessários para cadastrar um coorientador externo?
8. Como são obtidos os dados de um coorientador externo?
9. De quem depende a conclusão do cadastramento de projeto final?
10. De que informações os responsáveis pelo projeto precisam para confirmarem o cadastro?
11. Como um envolvido efetua a confirmação do cadastro?
12. Em que pontos a interação pode ser mais eficiente?
13. Como entrar em contato com um aluno?
14. Quem precisa ser notificado da conclusão do cadastro?

Cadastro de projetos finais com coorientador externo não cadastrado

Atores: Joana Marinho (secretária), Fernando Couto (aluno)

Na primeira semana de aula [2], Joana Marinho, secretária do curso de Engenharia Ambiental, precisa cadastrar entre 20 e 30 projetos finais dos alunos no período atual [5]. Um projeto final é um trabalho individual de um aluno sob a orientação de um ou dois professores [6]. Cada aluno preenche um formulário impresso e o entrega na secretaria [3]. Em vez de cadastrar os projetos finais à medida que são entregues, Joana prefere juntar vários para cadastrá-los de uma vez, pois acha que assim perde menos tempo [2]. Joana confere o formulário, verificando se o aluno definiu seu(s) orientador(es) e o título e formato de entrega do seu trabalho (e.g., relatório, software [4]), para então cadastrar os dados no sistema [1]. No caso do aluno Fernando Couto, após informar o título do trabalho e o orientador principal, Joana descobre que o seu coorientador, que não é professor regular do curso [6], não está cadastrado no sistema. Ela interrompe o cadastramento, pega o e-mail de Fernando da sua ficha cadastral (impressa) [13] e lhe envia uma mensagem [8] solicitando os dados do seu coorientador externo: nome completo, CPF e e-mail para contato [7]. No dia seguinte, Joana recebe a mensagem de resposta de Fernando com os dados solicitados. Ela então reinicia o cadastro do projeto final de Fernando, sem poder aproveitar o que havia feito na véspera [12]. Ao terminar o cadastro, Joana entra no seu sistema de correio eletrônico e envia uma mensagem para todos os envolvidos (aluno e orientadores) [14], para que eles confirmem os dados cadastrados e confirmem sua participação no projeto [9].

Naturalmente, nem todas as perguntas são exploradas em todos os cenários. No entanto, assim como no questionamento sistemático proposto por Carroll, pensar na forma de perguntas pode levar o designer a descobrir lacunas no cenário. Por exemplo, a pergunta 10 — *Quais são as informações fornecidas para os responsáveis pelo projeto confirmarem o cadastro?* — revela que o cenário, embora envolva o envio da mensagem de confirmação, não define claramente quais são as informações contidas na mensagem. É possível que a omissão seja proposital naquele momento, mas é

importante tornar explícitas as perguntas que permitem elaborar a metamensagem designer-usuário, para que elas fiquem registradas e sejam respondidas em algum momento do processo de design. Além disso, a leitura de um cenário pode, por sua vez, levantar novas questões a serem incorporadas ao conjunto de questões e respondidas naquele ou em outros cenários.

6.4 Análise de Tarefas

Uma análise de tarefas é utilizada para se ter um entendimento sobre qual é o trabalho dos usuários, como eles o realizam e por quê. Nesse tipo de análise, o trabalho é definido em termos dos objetivos que os usuários querem ou precisam atingir. Segundo Diaper (2003), a análise de tarefas é “a expressão utilizada no campo da ergonomia, que inclui IHC, para representar todos os métodos de coletar, classificar e interpretar dados sobre o desempenho de um sistema que possua ao menos uma pessoa como componente” (p. 14). A questão crucial passa a ser como definir um “desempenho satisfatório” para um sistema e seus componentes. Trata-se não apenas de listar ações, mas entender como um sistema de trabalho (composto ao menos de um sistema computacional e uma pessoa) afeta o domínio de aplicação e como, em contrapartida, o domínio de aplicação afeta o sistema de trabalho.

Em IHC, a análise de tarefas pode ser utilizada nas três atividades habituais: para análise da situação atual (apoizada ou não por um sistema computacional), para o (re)design de um sistema computacional ou para a avaliação do resultado de uma intervenção que inclua a introdução de um (novo) sistema computacional. Quando visa a avaliar um sistema computacional existente, a análise de tarefas pode ser bem concreta, descrevendo o comportamento de forma detalhada. Já numa situação de design, a análise de tarefas geralmente será realizada num nível maior de abstração, pois diversos pontos ainda não terão sido definidos no início da atividade de design.

Um dos primeiros passos numa análise de tarefas é coletarmos um conjunto de objetivos, definidos em termos psicológicos, ou seja, objetivos das pessoas. É claro que organizações também podem possuir objetivos, como, por exemplo, “aumentar o lucro”, mas estes não costumam ser endereçados em análises de tarefa. Para cada objetivo, elaboramos uma lista das ações realizadas (no mundo físico e através do sistema computacional) por um agente para alcançar esse objetivo. Quando há múltiplos agentes, Diaper recomenda representar as ações de cada agente numa coluna diferente.

Diaper (2003) ressalta que, independentemente da forma como os dados para uma análise de tarefas forem coletados, só teremos uma *simulação* das verdadeiras tarefas de interesse. Primeiro, há um número potencialmente infinito de tarefas reali-

zadas por diferentes pessoas, mas apenas algumas podem ser selecionadas para a análise. Segundo, apenas uma pequena porção do trabalho pode ser observada, e, portanto, os dados das tarefas são sempre incompletos. Finalmente, o próprio ato de coletar dados costuma alterar o que está sendo estudado. Embora um insumo importante da análise de tarefas seja a observação do *desempenho*, Diaper afirma que ela envolve também dados obtidos através de entrevistas, questionários, documentação, programas de treinamento e sistemas existentes. A análise de tarefas deve buscar identificar dados conflitantes e disparidades entre o relato oficial e a prática do trabalho.

Dentre os métodos de análise de tarefas mais comuns, podemos destacar a Análise Hierárquica de Tarefas (HTA – *Hierarchical Task Analysis*, Annett, 2003; Annett e Duncan, 1967), o GOMS (*Goals, Operators, Methods, and Selection Rules*, Kieras, 2003) e o ConcurTaskTrees (CTT; Paternò, 2000), descritos a seguir.

6.4.1 Análise Hierárquica de Tarefas

A Análise Hierárquica de Tarefas (HTA – *Hierarchical Task Analysis*) foi desenvolvida na década de 1960 para entender as competências e habilidades exibidas em tarefas complexas e não repetitivas, bem como para auxiliar na identificação de problemas de desempenho (Annett, 2003; Annett e Duncan, 1967). Ela ajuda a relacionar *o que* as pessoas fazem (ou se recomenda que façam), *por que* o fazem, e quais as *consequências* caso não o façam corretamente. Ela se baseia em psicologia funcional, e não comportamental, como eram as abordagens da época em que foi criada.

Uma **tarefa** é qualquer parte do trabalho que precisa ser realizada. Toda tarefa pode ser definida em termo de seu(s) objetivo(s). Tarefas complexas são definidas em termos de objetivos e subobjetivos, num desdobramento hierárquico. Esse desdobramento é chamado de decomposição de tarefas ou *redescrição* (Diaper, 2003). Observe que essa definição é mais ampla e difere da definição adotada pelo design baseado em objetivos e considerada nas seções anteriores, segundo a qual uma tarefa é um meio para se atingir um objetivo (Cooper *et al.*, 2007). Em HTA, tarefa se aproxima do conceito de atividade.

Uma análise funcional de tarefas começa pela definição dos objetivos das pessoas, antes de considerarmos as ações através das quais a tarefa pode ser realizada e o objetivo ser atingido. Um **objetivo** é um estado específico de coisas, um estado final. Esse estado pode ser definido por um ou mais eventos ou por valores fisicamente observáveis de uma ou mais variáveis, que atuam como critério de alcance do objetivo e, em última instância, do desempenho do sistema. Em vez de identificar uma lista de ações, a HTA inicia com uma definição dos objetivos das pessoas.

A HTA examina primeiramente os objetivos de alto nível (por exemplo, marcar uma reunião), decompondo-os em subobjetivos (por exemplo, decidir a data, decidir o local, convidar os participantes etc.), buscando identificar quais subobjetivos são mais difíceis de atingir (ou que geram mais erros) e que, portanto, limitam ou mesmo impedem o atingimento do objetivo maior.

Os subobjetivos de um objetivo e as relações entre eles é denominada de plano. Um **plano** define os subobjetivos necessários para alcançar um outro objetivo maior, e a ordem em que esses subobjetivos devem ser alcançados. Os planos podem definir diversas relações entre os subobjetivos: sequência fixa (um objetivo deve ser atingido antes do próximo); regra de seleção ou decisão (quais objetivos que deverão ser atingidos dependem das circunstâncias); ou em paralelo (mais de um objetivo deve ser atingido ao mesmo tempo).

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA. A Figura 6.1 apresenta os elementos de um diagrama HTA.

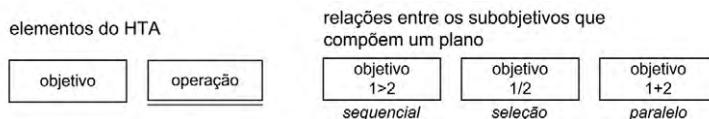


Figura 6.1 Elementos de um diagrama HTA.

Uma **operação** é especificada pelas circunstâncias nas quais o objetivo é ativado (*input* ou *entrada*), pelas atividades ou ações (*actions*) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (*feedback*). Uma **ação** pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o **input** como estados e o **feedback** como testes ou avaliação do estado final. A ação pode ser vista formalmente como uma regra de transformação entre estados. Por exemplo, em um sistema de agenda, é possível ter a seguinte tupla de *<inputs, ações, feedback>*: *<[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>*. Em outras palavras, as principais características de uma operação são as diversas ações que devem ser desempenhadas para atingir um objetivo e as condições que o satisfazem. Dessa maneira, a análise visa identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

A Figura 6.2 ilustra um diagrama HTA para o cadastro de um projeto final em um sistema acadêmico, e a Tabela 6.3 apresenta a tabela equivalente ao diagrama.

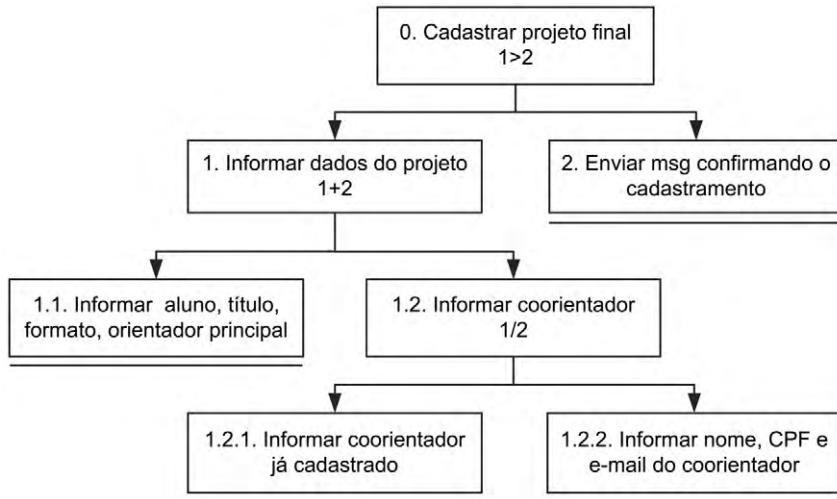


Figura 6.2 Diagrama HTA, para o objetivo de cadastrar um projeto final em um sistema acadêmico.

Tabela 6.3 Exemplo de representação das tarefas da HTA em tabela

objetivos / operações	problemas e recomendações
0. Cadastrar projeto final 1>2	<p><i>input:</i> formulário de cadastro de projeto final, com título, orientador(es) e formato do trabalho</p> <p><i>feedback:</i> novo projeto aparece para a secretaria na lista de projetos cadastrados como pendente enquanto os envolvidos não confirmarem</p> <p><i>plano:</i> informar dados do projeto e depois enviar mensagem de confirmação do cadastramento</p> <p><i>recomendação:</i> permitir que o aluno efetue o cadastro on-line</p>
1. Informar dados do projeto 1+2	<p><i>plano:</i> informar aluno, título, formato, orientador principal e informar coorientador</p>
1.1. Informar aluno, título, formato, orientador principal	
1.2 Informar coorientador 1/2	<p><i>plano:</i> informar coorientador já cadastrado ou informar nome, CPF e e-mail do orientador</p>
1.2.1. Informar coorientador já cadastrado	

objetivos / operações	problemas e recomendações
1.2.2. Informar nome, CPF e e-mail do coorientador	<p><i>problema:</i> ao cadastrar novo orientador, perde os dados já cadastrados do projeto, caso haja</p> <p><i>recomendação:</i> incluir o CPF de orientadores externos no formulário preenchido pelo aluno</p>
2. Enviar mensagem de confirmação do cadastramento	<p><i>ação:</i> cadastro deve ser confirmado em até sete dias</p> <p><i>recomendação 1:</i> tornar a confirmação mais eficiente</p> <p><i>recomendação 2:</i> alertar sobre o prazo de confirmação</p>

Um mesmo objetivo pode ser atingido de diferentes maneiras, dependendo de circunstâncias peculiares a cada situação. E uma ação pode ser utilizada como parte do alcance de diferentes objetivos. Por exemplo, consultar os compromissos em uma agenda pode fazer parte de um objetivo de marcar uma reunião, de um objetivo de saber o que tem de ser feito em um determinado dia, ou diversos outros. Sendo assim, apenas listar as ações sem entender para que servem pode causar interpretações equivocadas (Diaper, 2003).

Uma questão recorrente em decomposição de tarefas diz respeito à decisão sobre quando parar de decompor. Em geral, a decomposição termina quando já se tem todas as informações necessárias para atingir os objetivos da análise. Um critério de parada é o critério $p \times c$ (Annett e Duncan, 1967), que significa parar quando o produto da probabilidade de falha (p) e o custo da falha (c) for julgado aceitável. Uma outra razão para a parada ocorre quando a origem de um erro ou problema já tiver sido identificada e, portanto, o analista já pode propor uma correção, seja em termos de design de sistema, procedimentos operacionais ou treinamento.

Segundo Diaper (2003), uma Análise Hierárquica de Tarefas consiste nos seguintes passos:

1. *Decidir os objetivos da análise.* Em geral, envolvem projetar um sistema novo, modificar um sistema existente e desenvolver treinamento para os operadores.
2. *Obter consenso entre as partes interessadas na definição dos objetivos da tarefa e medidas de sucesso.* Devem ser definidos os resultados de desempenho desejados, tal como frequência de erros, e as formas como esses resultados serão aferidos. Para cada objetivo, as questões-chave são: “qual evidência objetiva indicará que esse objetivo foi atingido?” e “quais as consequências da falha em atingir esse objetivo?”.
3. *Identificar as fontes de informações das tarefas e selecionar as formas de aquisição de dados.* Embora a observação direta seja amplamente utilizada como forma de aquisição de dados, geralmente contribui pouco sobre eventos incomuns que possam ser essenciais. Entrevistas com especialistas, registros

do desempenho real e relatos de incidentes costumam trazer informações preciosas para a HTA.

4. *Coletar dados e esboçar uma tabela ou diagrama de decomposição.* Numa decomposição, os subobjetivos devem ser mutuamente exclusivos e exaustivos, ou seja, devem definir completamente o objetivo a que estão subordinados, sem que haja sobreposições entre os subobjetivos. A decomposição é esboçada em um diagrama hierárquico (Figura 6.2) e em uma tabela, que contém mais detalhes sobre as circunstâncias (*inputs*) que disparam o objetivo, ações e *feedbacks* de cada objetivo (Tabela 6.3).
5. *Verificar a validade da decomposição* junto às partes interessadas, visando assegurar a confiabilidade da análise.
6. *Identificar operações significativas*, tendo em vista o objetivo da análise, geralmente utilizando o critério $p \times c$.
7. *Gerar e, se possível, testar hipóteses relacionadas aos fatores que afetam o aprendizado e o desempenho.* Nessa etapa, pode ser útil utilizar as classificações de erro humano propostas por Reason (1990): desempenho baseado em *habilidades*, em *regras*, ou em *conhecimento*.

O desempenho baseado em *habilidades* envolve padrões de instruções pré-programadas. Erros nesse nível estão relacionados a variações de força e de coordenação espaço-tempo das pessoas para operar e atuar sobre os dispositivos. O desempenho baseado em *regras* envolve resolver problemas familiares cujas soluções são governadas por regras do tipo “e-se”. Erros nesse nível estão relacionados com a classificação equivocada de situações, levando à aplicação de regras ou procedimentos errados. Finalmente, o desempenho baseado em *conhecimento* ocorre em situações novas para as quais as ações ainda precisam ser planejadas, utilizando processos analíticos conscientes. Erros nesse nível se devem a limitações de recursos e conhecimento incompleto ou incorreto (Reason, 1990).

6.4.2 GOMS (*Goals, Operators, Methods, and Selection Rules*)

Card *et al.* (1983) propuseram um conjunto de modelos chamado de família GOMS (*Goals, Operators, Methods, and Selection Rules* — Objetivos, Operadores, Métodos e Regras de Seleção) para analisar o desempenho de usuários competentes de sistemas computacionais, realizando tarefas dentro da sua competência e sem cometer erros. Muitos sistemas são projetados considerando que as pessoas se tornam habilidosas no seu uso e, portanto, vão querer formas eficientes de realizar tarefas rotineiras. Os modelos GOMS têm se mostrado úteis para prever o desempenho, ou seja, predizer o impacto de decisões de design no desempenho competente (John, 2003).

O GOMS é um método para descrever uma tarefa e o conhecimento do usuário sobre como realizá-la em termos de objetivos (*goals*), operadores (*operators*), métodos (*methods*) e regras de seleção (*selection rules*). Os **objetivos** representam o que o usuário quer realizar utilizando o software (e.g., editar um texto). Os **operadores** são primitivas internas (cognitivas) ou externas (as ações concretas que o software permite que os usuários façam, tal como um comando e seus parâmetros digitados num teclado; a seleção de menus; o clique de um botão). Os **métodos** são sequências bem conhecidas de subobjetivos e operadores que permitem atingir um objetivo maior. Quando há mais do que um método para atingir um mesmo objetivo, são necessárias **regras de seleção**, que representam tomadas de decisão dos usuários sobre qual método utilizar numa determinada situação. Em suma, o GOMS caracteriza o conhecimento procedural de uma pessoa ao realizar tarefas num determinado dispositivo (Kieras, 2003).

A análise GOMS se aplica principalmente a situações em que os usuários realizam tarefas que já dominam. Esses usuários não estão resolvendo problemas ou tentando identificar o que precisam fazer em seguida. Essa análise pressupõe que eles sabem o que fazer, e só precisam fazê-lo. Em geral, o GOMS é utilizado após uma análise básica de tarefas, com o objetivo de fornecer uma representação formalizada que pode ser utilizada para prever o desempenho da tarefa, de modo que possa substituir parte dos testes empíricos com usuários. Isso é feito atribuindo um tempo estimado de execução para cada operador primitivo. Um operador cognitivo toma aproximadamente 50 ms, ao passo que o tempo associado a operadores externos é baseado em dados de observação. Para efeitos de engenharia, o modelo psicológico oferecido pelo GOMS fornece resultados bastante robustos e informativos.

O GOMS pode ser utilizado tanto quantitativamente, de modo a fornecer previsões sobre o tempo necessário para realizar tarefas, como qualitativamente, no sentido de auxiliar na elaboração de programas de treinamento, sistemas de ajuda e sistemas tutores inteligentes, pois um modelo GOMS contém uma descrição detalhada do conhecimento necessário para realizar cada tarefa. Também pode ser utilizado para reprojetar um sistema: pode revelar um objetivo frequente apoiado por um método muito ineficiente; pode mostrar que alguns objetivos não são apoiados por nenhum método; e pode revelar onde objetivos semelhantes são apoiados por métodos inconsistentes, uma situação em que os usuários podem ter problemas para lembrar o que fazer (John, 2003).

A análise com GOMS requer que o designer comece com uma lista de objetivos de usuário (ou tarefas de alto nível), que pode ser obtida de entrevistas de usuários potenciais, observações de usuários de sistemas existentes ou semelhantes. A princí-

pio, uma análise GOMS não revela objetivos que o analista tenha deixado de identificar, nem corrige uma formulação equivocada dos objetivos dos usuários. O GOMS também pode ser utilizado como ferramenta de design numa avaliação formativa, realizada ao longo de todo processo de design para avaliar as soluções alternativas elaboradas a cada iteração.

Dentre os modelos da família GOMS, destacamos: KLM (Card *et al.*, 1983), CMN-GOMS (Card *et al.*, 1983) e CPM-GOMS (John e Gray, 1995), descritos a seguir.

KLM

O KLM é a técnica mais simples de GOMS, limitada a um conjunto predefinido de operadores primitivos: K para pressionar uma tecla ou botão; P para apontar com o mouse um alvo num dispositivo visual; H para mover as mãos para o teclado ou outro dispositivo; D para desenhar um segmento de reta em um *grid*; M para se preparar mentalmente para realizar uma ação ou uma série de ações primitivas fortemente relacionadas entre si; e R para o tempo de resposta do sistema, durante o qual o usuário precisa esperar (Tabela 6.4).

Tabela 6.4 Algumas operações do KLM-GOMS e suas durações médias (Kieras, 1993)

operação	duração média
K: pressionar e soltar uma tecla do teclado	
exímio digitador (135 ppm)	0,08 s
bom digitador (90 ppm)	0,12 s
digitador mediano (55 ppm)	0,20 s
digitador inexperiente (40 ppm)	0,28 s
digitação de letras aleatórias	0,50 s
digitação de códigos complexos	0,75 s
digitador não familiarizado com o teclado	1,20 s
P: apontar o cursor do mouse num objeto da tela	1,10 s
B: pressionar ou soltar o botão do mouse	0,10 s
H: levar a mão do teclado ao mouse ou vice-versa	0,40 s
M: preparação mental	1,20 s
T (n): digitação de cadeia de caracteres	n x K s
W(t): espera pela resposta do sistema	depende do sistema

O KLM também inclui um conjunto de heurísticas sobre como posicionar operadores mentais durante a preparação. A estimativa de tempos de execução pode ser utilizada para comparar ideias em tarefas de *benchmark*, fazer uma avaliação paramétrica para

explorar o espaço definido por importantes variáveis (e.g., o tamanho de nomes de arquivos em uma linguagem de comando) e fazer análises sobre as suposições feitas (e.g., velocidade de digitação do usuário) (John, 2003).

O Exemplo 6.6 ilustra o tipo de análise que pode ser feita com o KLM.

Exemplo 6.6 – Análise do desempenho com o KLM

Análise da Tarefa: Salvar arquivo

método	operador	descrição	tempo (em s)
menu Arquivo > Salvar	M	preparação	1,20
	H	levar a mão do teclado ao mouse	0,40
	P	levar cursor até menu Arquivo	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
	P	levar cursor até menu Salvar	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
TOTAL			4,60
botão Salvar na barra de ferramentas	M	preparação	1,20
	H	levar a mão do teclado ao mouse	0,40
	P	levar cursor até botão Salvar	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
	TOTAL		3,10
teclas de atalho (Ctrl+S), considerando um digitador mediano	M	preparação	1,20
	K	teclar Ctrl	0,20
	K	teclar S	0,20
	TOTAL		1,60

Essa análise evidencia que usar as teclas de atalho é quase duas vezes mais eficiente do que usar o botão correspondente na barra de ferramentas, e quase três vezes mais eficiente do que usar o item de menu correspondente.

CMN-GOMS

O CMN-GOMS se refere à proposta original de GOMS, elaborada por Card, Moran e Newell. No CMN-GOMS, há uma hierarquia estrita de objetivos, os operadores são executados estritamente em ordem sequencial, e os métodos são representados numa notação semelhante a um pseudocódigo, que inclui submétodos e condicionais. O

Exemplo 6.7 apresenta um modelo GOMS parcial representando as tarefas envolvidas em descobrir a direção de tráfego de uma rua utilizando o Google® Maps.¹ Os objetivos e métodos foram numerados para facilitar sua identificação. Algarismos indicam sequência, e letras indicam alternativas.

Exemplo 6.7 – Modelo GOMS sem detalhes

GOAL 0: descobrir direção de tráfego de uma rua

GOAL 1: encontrar a rua

METHOD 1.A: zoom até o nível de ruas

(SEL. RULE: a região em que se situa a rua está visível no mapa e o usuário conhece o local)

METHOD 1.B: fazer busca pelo nome da rua

(SEL.RULE: o usuário não conhece o local ou o mapa visível está longe de lá)

GOAL 2: identificar a direção do tráfego na rua

Ao elaborar um modelo GOMS, devemos definir cuidadosamente o que representar e o que não representar. Tarefas mentais podem ser complexas, mas apenas aquelas que estejam relacionadas ao design do sistema devem ser incluídas no modelo (Kieras, 2003). Além disso, o nível de detalhes utilizado deve atender aos objetivos da análise. Em etapas iniciais, costumamos representar as estratégias alternativas que o usuário poderá seguir para atingir seus objetivos (Exemplo 6.7). Já para uma análise mais precisa do desempenho, os passos são mais detalhados (Exemplo 6.8).

Exemplo 6.8 – Modelo GOMS detalhado

GOAL 0: descobrir direção de tráfego de uma rua

GOAL 1: encontrar a rua

METHOD 1.A: zoom até o nível de ruas

(SEL. RULE: o local está visível no mapa e o usuário sabe onde fica a rua)

METHOD 1.A.A: zoom utilizando roda do mouse

(SEL. RULE: rua não centralizada no mapa, cursor distante da escala e preferência do usuário)

OP. 1.A.A.1: deslocar o cursor do mouse para a rua desejada

OP. 1.A.A.2: girar a roda do mouse para a frente

OP. 1.A.A.3: verificar enquadramento da rua no mapa

METHOD 1.A.B: zoom utilizando o menu pop-up

(SEL. RULE: rua centralizada no mapa, cursor distante da escala e pref. do usuário)

OP. 1.A.B.1: clicar com o botão direito do mouse

OP. 1.A.B.2: deslocar o mouse para a opção “zoom in”

OP. 1.A.B.3: clicar com o botão esquerdo do mouse

OP. 1.A.B.4: verificar enquadramento da rua no mapa

METHOD 1.A.C: zoom utilizando régua de escala

(SEL. RULE: cursor próximo da escala e preferência do usuário)

1 <http://maps.google.com>.

OP. 1.A.C.1: deslocar o cursor do mouse para a régua de escala na posição de zoom desejada

OP. 1.A.C.2: clicar com o botão esquerdo do mouse

OP. 1.A.C.3: verificar enquadramento da rua no mapa

METHOD 1.A.D: zoom utilizando botão de *zoom in*

(SEL. RULE: cursor próximo da escala e preferência do usuário)

OP. 1.A.D.1: deslocar o cursor do mouse para o botão de *zoom in*

OP. 1.A.D.2: clicar com o botão esquerdo do mouse

OP. 1.A.D.3: verificar enquadramento da rua no mapa

METHOD 1.B: fazer busca pelo nome da rua

(SEL.RULE: o usuário não conhece o local ou o mapa visível está longe de lá)

OP. 1.B.1: deslocar o cursor do mouse para o campo de busca

OP. 1.B.2: digitar o nome da rua desejada

OP. 1.B.3: ativar a busca

OP. 1.B.4: verificar resultados de busca

GOAL 1.B.5: localizar a rua

METHOD 1.B.5.A: selecionar a rua da lista de ruas encontradas

(SEL. RULE: mais de uma rua encontrada; rua não está visível no mapa; nível de zoom inadequado)

OP. 1.B.5.A.1: deslocar o cursor do mouse para a lista

OP. 1.B.5.A.2: clicar sobre a rua desejada

OP. 1.B.5.A.3: verificar enquadramento da rua no mapa

METHOD 1.B.5.B: localizar visualmente a rua no mapa

(SEL. RULE: rua está visível no mapa)

OP. 1.B.5.B.1: examinar marcador que identifica a rua

GOAL 2: identificar a direção do tráfego na rua

OP. 2.1: examinar setas desenhadas ao longo da rua desejada

Quantitativamente, os modelos CMN-GOMS permitem prever a sequência de operadores e o tempo de execução. Qualitativamente, eles focam métodos para alcançar objetivos: métodos semelhantes são facilmente identificados, métodos atípicamente curtos ou longos se destacam e podem disparar ideias de design, como, por exemplo, a inclusão de teclas de atalho para comandos frequentes e pontos de *feedback* para o usuário.

Uma diferença importante entre os modelos KLM e CMN-GOMS é que o CMN-GOMS é representado na forma de programa, e, portanto, a análise é geral e executável. Qualquer instância de classe de tarefas descrita pode ser realizada ou simulada seguindo os passos do modelo que podem passar por caminhos diferentes dependendo da situação específica da tarefa (John, 2003).

CPM-GOMS

O CPM-GOMS foi assim designado por dois motivos: por representar operadores *cognitivos, perceptivos e motores*, e por seguir a abordagem de *Critical Path Method*

(técnica de análise do caminho crítico). CPM-GOMS é uma versão do GOMS baseada diretamente no processador humano de informações (MHP, veja Seção 3.3.1) e, portanto, no modelo de estágios paralelos de processamento do processamento humano de informações. Isso significa que o CPM-GOMS não supõe que os operadores são executados sequencialmente. Em outras palavras, operadores cognitivos, perceptivos e motores podem ser tornar paralelos conforme a tarefa. O CPM-GOMS utiliza um diagrama tipo PERT para representar os operadores e as dependências entre eles. Nessa análise, o caminho crítico fornece uma previsão simples do tempo total da tarefa (Figura 6.3).

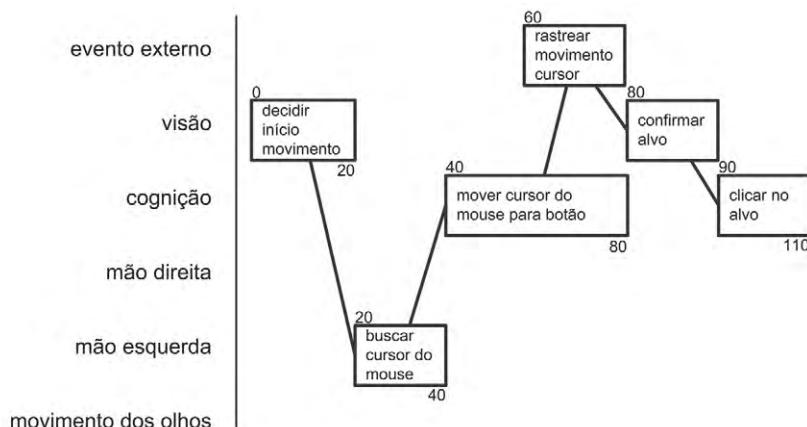


Figura 6.3 Exemplo de modelo CPM-GOMS.

A construção de um modelo CPM-GOMS inicia com a construção do modelo CMN-GOMS, cujos operadores são em seguida classificados em operadores cognitivos, perceptivos e motores do MHP. Atribuímos então uma duração estimada a cada operador e calculamos o tempo de execução previsto para a tarefa. É possível ainda efetuar uma análise qualitativa da relação entre aspectos do design e o tempo de execução, bem como fazer simulações de designs alternativos e ajudar a identificar por que um terá um desempenho melhor do que o outro.

Em geral, o CPM-GOMS assume que o usuário é extremamente experiente e executa as tarefas tão rápido quanto a arquitetura MHP permite. Isso significa que o usuário sabe exatamente onde procurar visualmente um determinado item de informação, e que não há atividades cognitivas substanciais associadas à seleção de métodos ou decisões complexas. Portanto, os modelos CPM-GOMS devem ser utilizados apenas para tarefas nas quais a seleção do método se baseia em dicas óbvias do ambiente ou envolve decisões triviais (John, 2003).

6.4.3 Árvores de Tarefas Concorrentes (*ConcurTaskTrees* – CTT)

O modelo de árvores de tarefas concorrentes (*ConcurTaskTrees* – CTT) foi criado para auxiliar a avaliação e o design e avaliação de IHC (Paternò, 2000). Nesse modelo, existem quatro tipos de tarefas (Figura 6.4a):

- *tarefas do usuário*, realizadas fora do sistema;
- *tarefas do sistema*, em que o sistema realiza um processamento sem interagir com o usuário;
- *tarefas interativas*, em que ocorrem os diálogos usuário–sistema; e
- *tarefas abstratas*, que não são tarefas em si, mas sim uma representação de uma composição de tarefas que auxilie a decomposição.

Assim como na análise hierárquica de tarefas, os diferentes níveis hierárquicos devem ser lidos como “para considerar T1 como tendo sido realizada, as tarefas T2 e T3 devem ter sido realizadas” (Figura 6.4b).

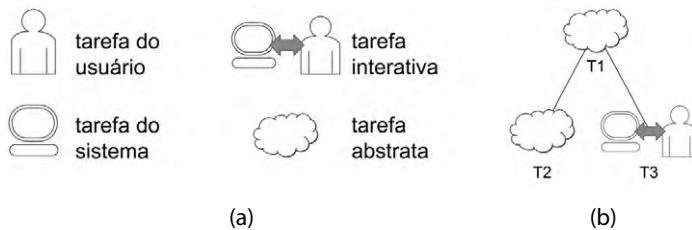


Figura 6.4 Representações (a) dos tipos de tarefas e (b) da sua hierarquia no CTT.

Além da hierarquia, o CTT permite representar diversas relações entre as tarefas, que aumentam a expressividade da notação (Figura 6.5). Os significados dessas relações são os seguintes:

- *ativação*: $T1 >> T2$ significa que a segunda tarefa (T2) só pode iniciar após a primeira tarefa (T1) terminar;
- *ativação com passagem de informação*: $T1 [] >> T2$ especifica que, além de T2 só poder ser iniciada após T1, a informação produzida por T1 é passada para T2;
- *escolha* (tarefas alternativas): $T1 \sqcap T2$ especifica duas tarefas que estejam habilitadas num momento, mas que, uma vez que uma delas é iniciada, a outra é desabilitada;
- *tarefas concorrentes*: $T1 \parallel T2$ especifica que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo;

- *tarefas concorrentes e comunicantes*: $T1 \mid [] \mid T2$ especifica que, além de as tarefas poderem ser realizadas em qualquer ordem ou ao mesmo tempo, elas podem trocar informações;
- *tarefas independentes*: $T1 \mid= T2$ especifica que as tarefas podem ser realizadas em qualquer ordem, mas quando uma delas é iniciada, precisa terminar para que a outra possa ser iniciada;
- *desativação*: $T1 > T2$ especifica que $T1$ é completamente interrompida por $T2$;
- *suspensão/retomada*: $T1 > T2$ especifica que $T1$ pode ser interrompida por $T2$ e é retomada do ponto em que parou assim que $T2$ terminar.

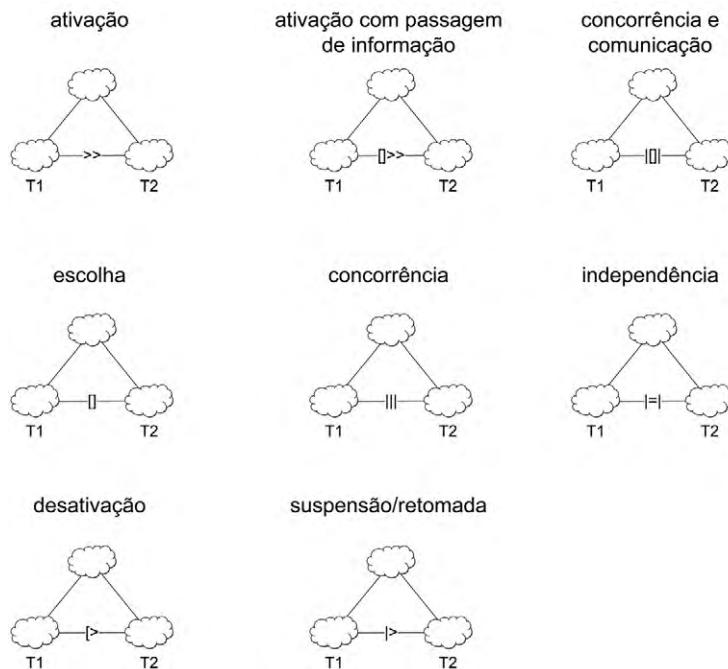


Figura 6.5 Relações entre tarefas no CTT.

A Figura 6.6 apresenta um exemplo de modelo de tarefas representado em CTT para um objetivo de marcar um compromisso em uma agenda.

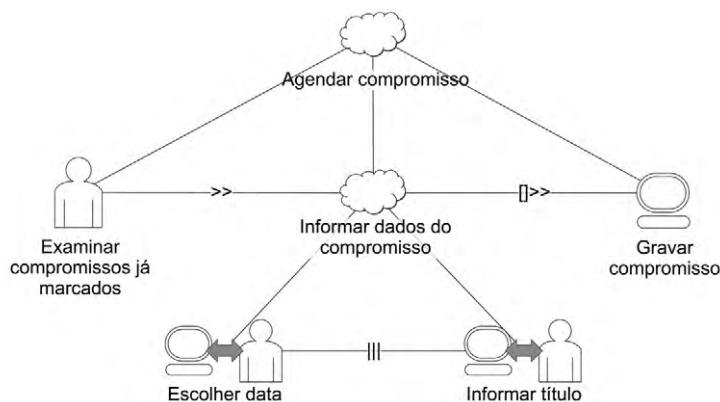


Figura 6.6 Exemplo de modelo de tarefas representado em CTT.

Dentre as vantagens do CTT com relação a outros modelos de tarefas, destacamos a possibilidade do registro explícito das relações entre as tarefas. Observamos que, uma vez que há tarefas interativas, do sistema e do usuário, o CTT vai além da análise de tarefas tradicional para representar uma solução de design da interação. Uma desvantagem com relação a modelos especificamente projetados para a interação é a ausência de elementos destinados à representação de mecanismos de prevenção e tratamento de erros na interação usuário-sistema (Seção 7.3.3).

Atividades

1. *Elaboração de perfil de usuário.* Trace os perfis de alunos e professores que deverão utilizar um sistema de apoio ao planejamento de aulas, divulgação de material didático e agendamento de trabalhos, provas e outras atividades. Identifique quais perguntas de uma entrevista ou de um questionário fornecem as informações necessárias para traçar esses perfis.
2. *Elaboração de personas.* Com base nos perfis de alunos e professores traçados anteriormente, crie o elenco de personas que representam os usuários do seu sistema. Identifique as personas primárias e secundárias, bem como as que representam demais *stakeholders* e antusuários.
3. *Elaboração de cenários.* Elabore cenários de problema para as personas atingirem seus objetivos. Considere os objetivos mais frequentes e os mais infrequentes de cada persona. Indique quais perguntas são respondidas ou endereçadas pelo cenário.
4. *Análise Hierárquica de Tarefas.* Elabore os diagramas hierárquicos de tarefas e suas respectivas tabelas, correspondentes aos cenários de problema criados na

atividade anterior. Identifique, a partir desses modelos, pontos em que a atividade pode ser mais eficiente.

5. *Elaboração de modelo GOMS.* Elabore um modelo GOMS detalhado para dois dos objetivos cenarizados anteriormente. Identifique a necessidade ou oportunidade de diferentes estratégias para alcançar um mesmo objetivo.

7

Design de IHC

Objetivos do Capítulo

- Apresentar representações e modelos utilizados no design da interação e da interface com usuário.
 - Discutir como as representações utilizadas favorecem certos tipos de reflexão sobre o design de IHC.
 - Apresentar diferentes estilos de interação que podem ser adotados no design de IHC.
 - Descrever representações da interface com usuário em diferentes níveis de abstração e enfocando diversos aspectos da solução.
-

Este capítulo apresenta representações e modelos utilizados ao longo de diferentes processos de design de IHC, focando as atividades de concepção da solução interativa. Descreve cenários de interação, que são utilizados em diversos processos de design apresentados no Capítulo 4, em particular nos processos de design baseado em cenários (Seção 4.3.5), orientado por objetivos (Seção 4.3.6) e centrado na comunicação (Seção 4.3.7). O capítulo se aprofunda nos modelos utilizados pelo design centrado na comunicação, em particular: mapa de objetivos, esquema conceitual de signos, modelo de tarefas e modelo de interação (Paula, 2003; Silva, 2005). Esses modelos são utilizados para motivar a reflexão de designers, durante o design de IHC, sobre estratégias alternativas de uso do sistema e sobre mecanismos de prevenção e recuperação de rupturas comunicativas que podem causar falhas na interação. Em seguida, são analisadas algumas decisões tomadas ao partir do design da interação para o design da interface. Finalmente, o capítulo descreve brevemente questões relacionadas ao sistema de ajuda (Silveira *et al.*, 2004) e questões de adaptação e adaptatividade (Schneider-Hufschmidt *et al.*, 1993; de Souza e Barbosa, 2006) que apresentam desafios adicionais à atividade de design.

7.1 Introdução

Os modelos e as representações utilizados no Capítulo 6 permitem descrever quem usa ou utilizará o sistema (através de perfis de usuário e personas); quais são seus objetivos, motivações, e em que contexto ele será utilizado (cenários de problema); e como os usuários alcançam esses objetivos atualmente (cenários e modelos de tarefa). Essas informações e artefatos são utilizados também para o design da interação. Naquele capítulo, o foco era a análise da *situação atual*. Neste capítulo, nos concentramos no projeto da *intervenção* que será feita, através do design do sistema computacional interativo visando apoiar melhor os usuários no alcance dos seus objetivos.

O design de IHC visa elaborar um modelo conceitual de entidades e atributos do domínio e do sistema, estruturar as tarefas e projetar a interação e a interface de um sistema interativo que apoie os objetivos do usuário. Para isso, podem ser utilizadas diversas representações, cada qual endereçando uma ou mais questões de IHC, como pode ser visto na Figura 7.1.

Personas, cenários de problema e alguns modelos de tarefas foram descritos no Capítulo 6. As demais representações são descritas nas próximas seções.

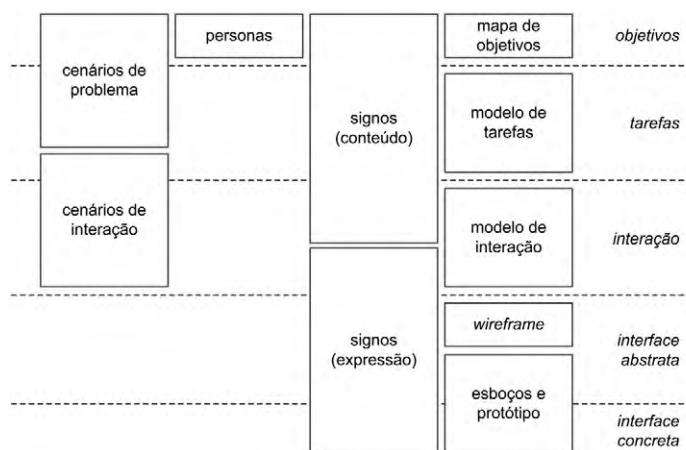


Figura 7.1 Representações utilizadas para explorar diferentes aspectos de IHC.

7.2 Cenários de Interação

Como visto nas Seções 4.3.5 e 6.3, cenários são narrativas sobre pessoas realizando uma atividade para alcançar um ou mais objetivos (Rosson e Carroll, 2002). A Seção 4.3.5 descreve diversos tipos de cenários: de problema, de atividade, de informação e de interação, e a Seção 6.3 explora cenários de problemas em detalhes. Os cenários de problemas investigados na atividade de análise descrevem a situação atual, evidenciando problemas e oportunidades de melhoria. Já os cenários de interação representam intervenções para endereçar esses problemas e oportunidades.

Um **cenário de interação** especifica em mais detalhes as ações do usuário e as respectivas respostas (*feedback*) do sistema necessárias para alcançar os objetivos apoiados pelo sistema. Apesar de ricos em detalhes e contextualizados, os cenários de interação elaborados em fases iniciais de design não devem conter detalhes da *interface* propriamente dita, como textos e rótulos, tipos de elementos de interface (*widgets*) utilizados etc. Pretendemos com isso 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. Por exemplo, não devemos dizer que “Joana seleciona relatório técnico da lista expansível (*dropdown list*) denominada Formato de entrega”. Tais decisões sobre a escolha dos elementos de interface devem ser postergadas para o momento adequado no processo de design.

O Exemplo 7.1 define dois cenários de interação que buscam resolver os problemas apontados por um dos cenários do Exemplo 6.4, reproduzido a seguir para facilitar a análise das soluções propostas.

Exemplo 7.1 – Cenários de interação

Cenário de problema: Cadastro de projetos finais com coorientador externo não cadastrado

Atores: Joana Marinho (secretária), Fernando Couto (aluno)

Na primeira semana de aula [2], Joana Marinho, secretária do curso de Engenharia Ambiental, precisa cadastrar entre 20 e 30 projetos finais dos alunos no período atual [5]. Um projeto final é um trabalho individual de um aluno sob a orientação de um ou dois professores [6]. Cada aluno preenche um formulário impresso e o entrega na secretaria [3]. Em vez de cadastrar os projetos finais à medida que são entregues, Joana prefere juntar vários para cadastrá-los de uma vez, pois acha que assim perde menos tempo [2]. Joana confere o formulário, verificando se o aluno definiu seu(s) orientador(es) e o título e formato de entrega do seu trabalho (e.g., relatório, software [4]), para então cadastrar os dados no sistema [1]. No caso do aluno Fernando Couto, após informar o título do trabalho e o orientador principal, Joana descobre que o seu coorientador, que não é professor regular do curso [6], não está cadastrado no sistema. Ela interrompe o cadastramento, pega o e-mail de Fernando da sua ficha cadastral (impressa) [13] e lhe envia uma mensagem [8] solicitando os dados do seu coorientador externo: nome completo, CPF e e-mail para contato [7]. No dia seguinte, Joana recebe a mensagem de resposta de Fernando com os dados solicitados. Ela então reinicia o cadastro do projeto final de Fernando, sem poder aproveitar o que havia feito na véspera [12]. Ao terminar o cadastro, Joana entra no seu sistema de correio eletrônico e envia uma mensagem para todos os envolvidos (aluno e orientadores) [14], para que eles confirmem os dados cadastrados e confirmem sua participação no projeto [9].

Cenário de interação A: Cadastro de projetos finais pelos professores

Atores: Joana (secretária), Fernando Couto (aluno), Marcos Correa (professor, orientador principal do projeto final), Pedro Melo (coorientador externo)

Na primeira semana de aula [2], Joana, secretária do curso de Engenharia Ambiental, precisa se certificar de que os projetos finais dos alunos iniciados no período atual estão cadastrados. Como costumam ser entre 20 e 30 projetos [5], e seu cadastramento deve ser efetuado numa época em que o pessoal da secretaria está sobrecarregado de trabalho, cada professor deve cadastrar os projetos dos seus alunos [1]. Para isso, Joana envia uma mensagem a todos os professores solicitando que cadastrem os projetos sob sua orientação e informando que eles têm apenas uma semana para fazê-lo, sob risco de os alunos terem suas matrículas em Projeto Final I canceladas.

Ao receber a mensagem de Joana, Marcos Correa entra no sistema para cadastrar o projeto final do seu aluno Fernando Couto [1,3]. Ele informa o nome e a matrícula do aluno, além do título e do formato de entrega do seu trabalho (e.g., relatório ou software) [4]. Ao informar os dados do coorientador externo (nome completo, e-mail e CPF) [7], percebe que não possui o CPF do seu colega, Pedro Melo. Marcos então pede que o próprio sistema envie uma mensagem a Pedro solicitando essa informação [8] e confirma o cadastramento [9]. Ao concluir o cadastramento, Marcos é informado de que o sistema enviará uma mensagem de solicitação de informações adicionais para seu colega Pedro e uma mensagem de *feedback* para o aluno Fernando Couto.

Cenário de interação B: Cadastro de projetos finais pelos próprios alunos

Atores: Joana (secretária), Fernando Couto (aluno), Marcos Correa (professor, orientador principal do projeto final), Pedro Melo (coorientador externo)

Na primeira semana de aula [2], Joana, secretária do curso de Engenharia Ambiental, precisa se certificar de que os projetos finais dos alunos iniciados no período atual estão cadastrados. Como costumam ser entre 20 e 30 projetos [5], e seu cadastramento deve ser efetuado numa época em que o pessoal da secretaria está sobrecarregado de trabalho, cada aluno deve cadastrar o seu próprio projeto [1]. Para isso, Joana extrai do sistema uma lista de alunos matriculados em Projeto Final I, contendo número de matrícula, nome e e-mail de cada aluno, e envia uma mensagem para esses alunos com instruções sobre como devem cadastrar o seu projeto e informando que eles têm apenas uma semana para fazê-lo, sob risco de terem suas matrículas em Projeto Final I canceladas.

Ao receber a mensagem de Joana, Fernando Couto entra no sistema para cadastrar seu projeto final [1,3]. Ele informa seu orientador principal, verifica que seu coorientador, que não pertence ao quadro de professores da universidade [6], não está cadastrado e inicia o cadastramento dos seus dados: nome completo, e-mail e CPF [7]. Percebe que não possui o CPF do seu coorientador, Pedro Melo. Fernando então pede que o próprio sistema envie uma mensagem a Pedro solicitando essa informação [8]. Fernando então prossegue informando os demais dados do projeto final: título e formato de entrega do seu trabalho (e.g., relatório ou software) [4]. Ao concluir o cadastramento, Fernando é informado de que o sistema enviará uma mensagem de confirmação para o seu orientador e uma mensagem de solicitação de informações adicionais para seu coorientador, e ficará marcado como pendência até que a confirmação ocorra [9].

Conjunto de perguntas exploradas nos cenários:

1. Quem pode/deve cadastrar os dados dos projetos finais no sistema?
2. Quando são cadastrados os projetos finais?
3. Quem fornece os dados dos projetos finais?
4. Quais dados de projeto final devem ser cadastrados?
5. Quantos projetos são cadastrados a cada período?
6. Quem pode orientar um trabalho final?
7. Que dados são necessários para cadastrar um coorientador externo?
8. Como são obtidos os dados de um coorientador externo?
9. De quem depende a conclusão do cadastramento de projeto final?
10. De que informações os responsáveis pelo projeto precisam para confirmarem o cadastro?
11. Como um envolvido efetua a confirmação do cadastro?
12. Em que pontos a interação pode ser mais eficiente?
13. Como entrar em contato com um aluno?
14. Quem precisa ser notificado da conclusão do cadastro?

Também nos cenários de interação são feitas referências às perguntas exploradas na sua narrativa. Na tentativa de solucionar o problema de cadastramento pela secretária Joana, foram propostos dois cenários de interação: cenário A, no qual o cadastramento é realizado pelo professor orientador; e cenário B, no qual o cadastramento é efetuado pelo próprio aluno. Durante as discussões sobre qual das soluções é a mais adequada, é possível que nenhuma das duas seja escolhida, e que uma nova solu-

ção seja elaborada, diferente de A e de B, ou uma combinação das duas. Além disso, num processo de design reflexivo (em linha com a perspectiva de reflexão-em-ação discutida na Seção 4.2), é possível ainda que o próprio problema seja revisto e que o novo enquadramento do problema requeira a elaboração de novos cenários, tanto de problema como de interação.

7.3 Design Centrado na Comunicação

Na engenharia semiótica, o objetivo do design da interação é completar a segunda parte da metamensagem do designer para o usuário (de Souza, 2005a, p. 25; Seção 3.8):

Este é o meu entendimento, como designer, de quem você, usuário, é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. *Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão.*

Em outras palavras, é responsabilidade do designer comunicar aos usuários sua visão de design e dar-lhes melhores condições de entender e aprender sobre o sistema projetado e como podem utilizá-lo.

Como visto na Seção 4.3.7, o design centrado na comunicação visa elaborar essa metacomunicação de modo a evitar rupturas comunicativas durante a interação do usuário com o preposto do designer. Como essa interação é vista como uma conversa, projetar a interação significa definir as conversas que o usuário poderá travar com o preposto do designer para alcançar seus objetivos. As representações e o embasamento teórico do design centrado na comunicação têm por objetivo motivar o designer a refletir sobre o seu papel como interlocutor dessas conversas. Essa reflexão resulta no projeto da linguagem de interface, através da qual o usuário e o preposto do designer, seu representante durante a interação, serão capazes de travar essas conversas em direção aos objetivos do usuário, segundo a visão do designer.

O design da interação pode então ser considerado como a especificação de todas as conversas que os usuários poderão travar com o preposto do designer, incluindo conversas alternativas representando diferentes estratégias para alcançar um objetivo e conversas para a recuperação de rupturas comunicativas. Precisamos estabelecer *sobre o que* o usuário e o preposto do designer estão conversando a cada momento e *de que forma* essa conversa ocorre durante o uso do sistema interativo.

Toda **conversa** tem um **tópico**, que é o assunto geral por ela endereçado. Essa conversa pode se desdobrar em **diálogos**, que endereçam **subtópicos** relacionados ao tópico da conversa. A cada momento, a conversa tem um **foco**, que pode ser um dos

elementos do modelo de comunicação de Jakobson (1960): contexto, emissor, receptor, mensagem, código e canal (veja Seção 3.8.3). Para se referir a qualquer um desses elementos, um interlocutor faz uso de **signos**, compondo as **falas** que ele emite.

Os tópicos e o foco da conversa são mantidos ou alterados por trocas de turno, e em cada turno é emitida uma ou mais falas de um dos interlocutores da conversa (usuário ou preposto do designer). A Tabela 7.1 apresenta um exemplo de interação como uma conversa, cujo objetivo é cadastrar o enunciado de um trabalho. Nela, os signos que constituem o foco da conversa a cada momento aparecem em negrito.

Tabela 7.1 Exemplo de representação da interação como uma conversa entre usuário (U) e preposto do designer (D)

tópico > subtópico (diálogo)	falas e signos
cadastrar trabalho	U: Preciso cadastrar um trabalho para os meus alunos de IHC.
> informar dados do trabalho	D: Qual é o título e a descrição do trabalho? Até quando deve ser entregue? Pode ser feito em grupo ? Quantos pontos vale o trabalho?
> consultar datas importantes	U: Antes, quero consultar os prazos da universidade e feriados desse semestre. D: Ei-los.
> informar dados do trabalho	U: Preciso de uma semana para corrigir os trabalhos, e preciso entregar as notas até dia 2 de junho. Então vou pedir para os alunos entregarem os trabalhos até o dia 26 de maio (data de entrega). Eles devem receber um lembrete do prazo de entrega . D: OK, o trabalho deverá ser entregue até o dia 26 de maio e os alunos serão lembrados no dia 23 de maio (três dias antes).
> informar dados do trabalho	D: E qual é o título e a descrição do trabalho? Pode ser feito em grupo ? Quantos pontos vale o trabalho? U: O trabalho pode ser feito em dupla, e vale 20% da nota. O título é (...) e a descrição é (...). D: OK, o trabalho já foi cadastrado.
conferir cadastro do trabalho > examinar dados do trabalho	U: Deixa eu conferir os dados do trabalho... Estão OK.
notificar alunos	U: Agora quero avisar aos alunos de que o enunciado do trabalho já está disponível. D: OK, posso enviar a mensagem padrão ?
> informar conteúdo da mensagem	U: Sim.
conferir mensagem > conteúdo e destinatários da mensagem	D: A mensagem (...) foi enviada para os alunos (...).

As próximas subseções descrevem as representações utilizadas para o projeto da interação como uma conversa.

7.3.1 Mapa de Objetivos dos Usuários

Como primeiro passo do design da interação, devemos organizar os objetivos dos usuários que tenham sido identificados na etapa de análise. Em um primeiro momento, devemos representar apenas **o que** o usuário deseja realizar, sem considerar **como** ele o fará. Esses objetivos correspondem aos objetivos práticos descritos em personas e cenários.

Os objetivos podem ser classificados em finais e instrumentais. Os **objetivos finais** são aqueles que levam o usuário a utilizar o sistema. Já os **objetivos instrumentais** são utilizados como facilitadores para os objetivos finais, um meio para o fim desejado. Por exemplo, considere o objetivo final de cadastrar um projeto. Um projeto final possui um tipo, cujo valor deverá ser indicado dentre um conjunto de tipos válidos. Caso o usuário possa definir novos tipos, cadastrar um novo tipo de projeto pode ser considerado como um objetivo instrumental do primeiro.

Os objetivos instrumentais podem ainda ser subclassificados em instrumentais diretos ou indiretos. Um objetivo **instrumental direto** pode ser considerado como facilitador imediato do objetivo que instrumenta, durante a interação. Um objetivo **instrumental indireto**, por sua vez, prepara o terreno para que o objetivo por ele instrumentado seja atingido em algum momento futuro.

Retomando o exemplo anterior, caso o usuário possa definir novos tipos de projeto enquanto cadastra um projeto, o objetivo cadastrar um novo tipo de projeto é considerado instrumental direto. Já se o usuário precisará planejar quais tipos de projeto devem ficar disponíveis e cadastrá-los *a priori*, o objetivo cadastrar um novo tipo de projeto é considerado instrumental indireto. Podemos observar que a escolha entre tornar um objetivo instrumental direto ou indireto já envolve tomar decisões sobre a interação usuário-sistema. Considerando as estratégias de ação planejada e situada, descritas na Seção 3.5, podemos dizer que os objetivos instrumentais indiretos favorecem uma interação mais planejada, ao passo que os objetivos instrumentais diretos favorecem uma interação mais situada e oportunista.

A Tabela 7.2 fornece modelos comumente utilizados na formulação dos diversos tipos de objetivos, do ponto de vista do designer.

Tabela 7.2 Formulações comumente utilizadas para descrever diferentes tipos de objetivos

tipo de objetivo	formulação
final	Você (usuário no papel <Papel>) quer utilizar o sistema para <atingir objetivoFinal>
instrumental	Quer <atingir objetivoInstrumental> para <atingir objetivoFinal> [de forma mais eficiente/fácil/flexível...]
instrumental direto	Quer <atingir objetivoInstrumental> para <atingir objetivoFinal> [de forma mais eficiente/fácil/flexível...] agora
instrumental indireto	Quer <atingir objetivoInstrumental> para <atingir objetivoFinal> [de forma mais eficiente/fácil/flexível...] no futuro

Há casos ainda em que um objetivo final pode ser considerado também um facilitador de um outro objetivo, como se fosse, ao mesmo tempo, final e instrumental. Por exemplo, considere os objetivos finais consultar um projeto e alterar o cadastro de um projeto. O objetivo consultar um projeto pode ser considerado, além de objetivo final, objetivo instrumental direto para alterar o cadastro de um projeto. Dessa maneira, assim que o usuário tiver atingido o objetivo de consultar um projeto, ele poderá imediatamente passar para o objetivo de alterá-lo. A representação desse tipo de relação envolve decisões de design sobre a estruturação da atividade do usuário, que será representada em modelos de tarefa e de interação, descritos nas próximas subseções.

Independentemente da representação utilizada para organizar os objetivos dos usuários, o designer pode relacionar e anotar esses objetivos de acordo com algumas dimensões de interesse. Essas dimensões variam com o tipo de projeto. Alguns elementos que podem ser utilizados para organizar os objetivos são:

- papéis de usuários (i.e., papéis que podem ou devem atingir cada objetivo, tais como aluno/professor, autor/editor);
- tipo de objetivo (objetivo final ou instrumental direto ou indireto);
- “entidade” (conceito do domínio ou do sistema) principal relacionada ao objetivo (e.g., professor, disciplina, atividade, apresentação, categoria etc.)

O Exemplo 7.2 apresenta um mapa de objetivos de um sistema de apoio a aulas presenciais em uma universidade.

Exemplo 7.2 – Mapa de objetivos de usuário

Considere um sistema de apoio à divulgação de material, atividades e notas de uma disciplina em uma universidade. Professores e alunos utilizam o sistema com diferentes objetivos. Professores podem gerenciar e consultar suas disciplinas, alunos, avisos, materiais didáticos e trabalhos a serem realizados pelos alunos. Já os alunos utilizam o sistema principalmente para consultas, mas também para “entregar” os trabalhos realizados. Todos podem se comunicar através de um fórum de discussão.

No mapa de objetivos da Figura 7.2, os objetivos que manipulam um mesmo tipo de entidade estão agrupados: *aviso*, *material didático*, *trabalho*, *fórum de discussão* e *sistema*. Esses agrupamentos podem auxiliar a tomar decisões sobre a interface, como, por exemplo, sobre a organização e a estrutura de menus e submenus.

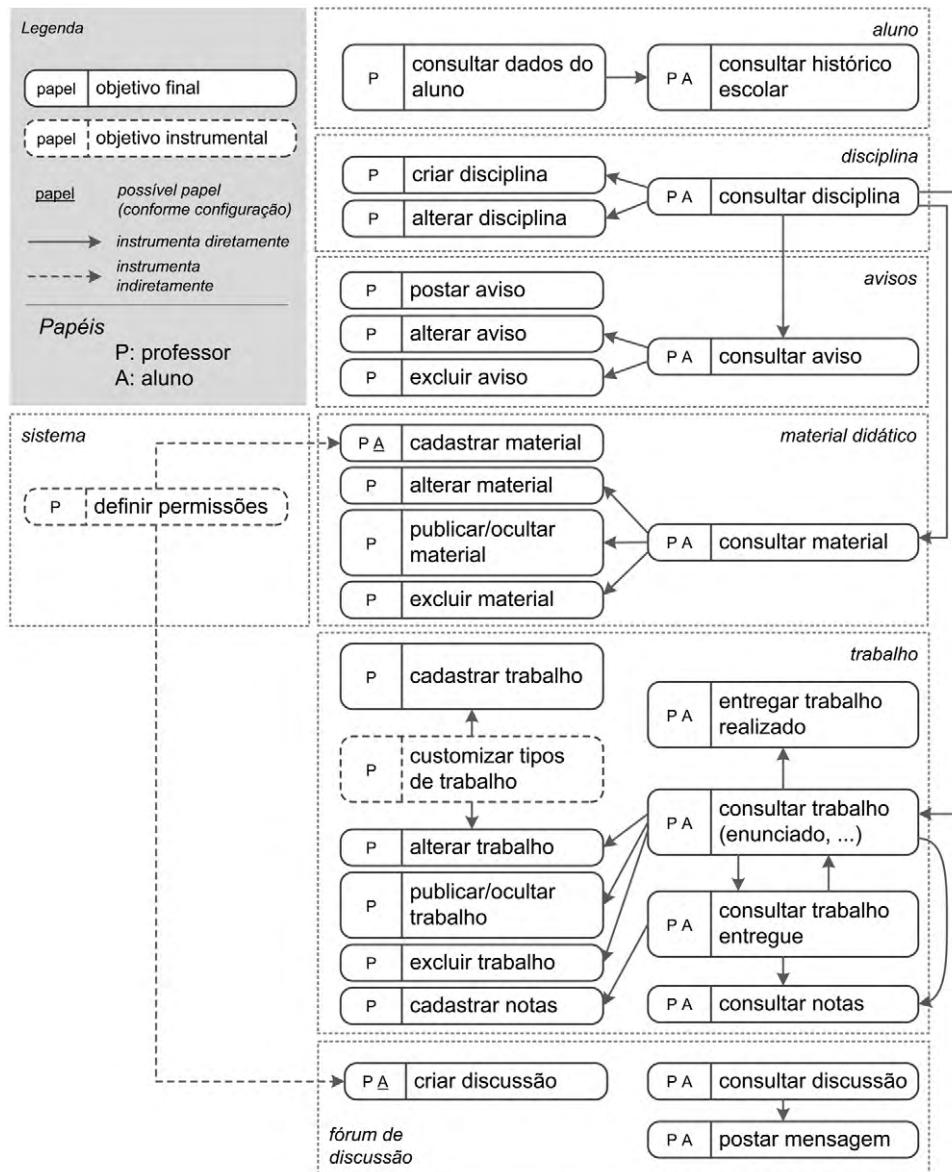


Figura 7.2 Mapa de objetivos dos usuários.

Nesse exemplo, observamos que o objetivo final *alterar trabalho* é instrumentado diretamente pelo objetivo (também final) *consultar trabalho*. A maioria dos objetivos representados na figura são objetivos finais. Os únicos objetivos instrumentais representados são os objetivos: *customizar tipo de*

trabalho, que permite alcançar os objetivos *cadastrar trabalho* e *alterar trabalho* de modo mais preciso, qualificando as atividades sendo cadastradas; e *definir permissões*, que permite configurar quem pode fazer o que no sistema.

Os papéis que são afetados por algum tipo de configuração de permissões são sublinhados. No exemplo, observamos que a definição de permissões afeta apenas os objetivos *cadastrar material* e *criar discussão*. Isso significa que o professor (papel que pode alcançar o objetivo *definir permissões*) pode permitir ou não que os alunos (representados pela letra A sublinhada) alcancem esses objetivos através do sistema.

A representação das relações de instrumentação entre objetivos finais ajuda a tomar decisões sobre consistência no design da interação e da interface. Considerando os objetivos Consultar material e Excluir material do exemplo, se a relação entre eles não estiver representada, pode acontecer de um cenário para Excluir material definir uma forma de localizar o material didático diferente da forma projetada no cenário Consultar material. Isso pode causar inconsistências e rupturas na interação usuário-sistema.

Uma situação extrema aconteceria se o usuário consultasse um material e, percebendo que deveria removê-lo, ele não pudesse fazê-lo naquele momento. Em vez disso, teria de abandonar a “atividade de consulta”, iniciar a conversa (caminho de interação) para remover o material, localizá-lo novamente e só então removê-lo. Esse exemplo trivial ilustra uma possível inconsistência ou ineficiência que pode passar despercebida quando utilizamos representações que tratam objetivos separadamente, como algumas formas de cenários e modelos de tarefas.

O Exemplo 7.3 apresenta uma situação semelhante, encontrada em um sistema de administração acadêmica.

Exemplo 7.3 – Decisões de design da interação relacionadas com o mapa de objetivos

Considere um sistema de administração acadêmica, que permite a professores buscarem um aluno matriculado na universidade e consultarem seu histórico escolar. Suponha que o professor Jorge Nunes queira consultar o histórico escolar do aluno Frederico Barbosa, candidato a uma bolsa de projeto. Esses objetivos podem ser representados de duas maneiras, cada qual relacionada com uma conversa usuário-sistema (Figura 7.3). Os elementos destacados indicam a diferença entre as duas alternativas de design.



Figura 7.3 Duas formas alternativas de organizar os objetivos consultar dados de aluno e consultar histórico escolar de aluno.

A conversa na direção do objetivo buscar um aluno é comum às duas soluções:

U: Quero procurar o aluno Frederico Barbosa.

S: Existem três alunos com esse nome: Frederico Arruda Barbosa, aluno de Direito, matrícula 0720253; Frederico Barbosa Torres, aluno de Economia, matrícula 0819247; e Frederico Martins Barbosa, aluno de Comunicação Social, matrícula 0723298.

U: Quero ver detalhes sobre o aluno Frederico Martins Barbosa.

S: Seu número de matrícula é 0723298. Ele é aluno de graduação que cursa o Programa de Comunicação, habilitação Jornalismo. Segue o currículo 2006.2.

O objetivo buscar um aluno foi concluído. As conversas em direção ao objetivo consultar histórico escolar são diferentes para cada solução.

Solução a: início de nova conversa em direção do objetivo — novo tópico e **novo foco**

U: Quero consultar um histórico escolar.

S: Qual é o número de matrícula do aluno?

U: 0723298.

S: As disciplinas cursadas até hoje são (...)

Solução b: mudança na direção da conversa — novo tópico, **mesmo foco**

U: Quero consultar o histórico escolar dele.

S: As disciplinas cursadas até hoje são (...)

O objetivo consultar histórico escolar foi concluído.

Observamos que, quando a relação entre os objetivos não é representada, aumenta o risco de o foco da conversa se perder quando o tópico da conversa muda, ou seja, quando o usuário alcança um objetivo e inicia a conversa em direção a um outro.

7.3.2 Esquema Conceitual de Signos: Conteúdo (Parte I)

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 fala (ação) do usuário, sistema ou interlocutor externo que afete a interação usuário-sistema. A partir das atividades e representações de análise (veja Capítulos 5 e 6), podemos identificar os signos que farão parte da conversa entre o usuário e o preposto do designer. O esquema conceitual de signos é definido ao longo do design da interação. Inicialmente, definimos o **conteúdo** dos signos, especificando restrições sobre valores associados ao signo, seus valores default, e os mecanismos de prevenção e tratamento de rupturas associadas aos signos (Seção 7.3.3). À medida que o design da interface avança, definimos também a **expressão** dos signos, especificando como eles se manifestam na interface e como os usuários podem “falar sobre” eles. Esta seção descreve como o conteúdo dos signos deve ser definido. A expressão dos signos é descrita na Seção 7.4.4.

Alguns signos estão relacionados a conceitos ou **entidades** do domínio ou do próprio sistema (denominados *signo-entidade*, ou simplesmente entidade); outros

correspondem a **atributos** desses signos-entidade (*signo-atributo*, ou simplesmente atributo), ou ainda como **valores** de um signo-atributo.

A Tabela 7.3 mostra alguns signos extraídos e extrapolados da conversa representada na Tabela 7.1.

Tabela 7.3 Definição parcial dos signos-entidade do sistema de apoio acadêmico do exemplo

Enunciado de trabalho (E) – enunciado de trabalho de disciplina de graduação		
signo	origem	observações
+ título	domínio	
descrição	domínio	
data de entrega	domínio	
formato de entrega	domínio	(e.g., relatório, protótipo)
número máximo de alunos	domínio	indica se o trabalho deve ser realizado individualmente ou em grupo
peso	domínio	peso do trabalho na pontuação (porcentagem)
lembrete do prazo de entrega	aplicação	indica se o sistema deve ou não enviar aos alunos um lembrete alguns dias (<u>prazo para lembrete</u>) antes da data final para entrega do trabalho
<i>prazo para lembrete</i>	aplicação	para cada turma, o professor define a data de lembrete pelo número de dias antes da data de entrega
Trabalho entregue (T) – trabalho realizado por um ou mais alunos		
signo	origem	observações
+ Enunciado (E)	domínio	T é definido por E
+ Alunos (A) A.[matrícula, nome]	domínio	A realiza T; cardinalidade depende de <i>E.número máximo de alunos</i>
relatório	domínio	
data de entrega	domínio	
nota	domínio	
Aluno (A) – aluno de graduação		
signo	origem	observações
+ matrícula	domínio	
nome	domínio	
período	domínio	calculado a partir da data de ingresso do aluno

O signo-entidade Enunciado de trabalho é composto dos signos título, descrição, data de entrega, número máximo de alunos (individual/em grupo), peso e lembrete do prazo de entrega. O signo-entidade Trabalho entregue é composto dos signos Aluno(s), relatório, data de entrega e nota. Uma letra inicial maiúscula indica um signo-entidade, ao passo que uma letra inicial minúscula indica um signo-atributo. Notamos que o

signo Aluno é, ele próprio, uma entidade, relacionado com o signo-entidade Trabalho entregue através da relação indicada na coluna de observações. As letras maiúsculas identificam as entidades para indicar a direção do relacionamento (e.g., A realiza T). Além disso, podemos indicar quais são os signos-atributo de interesse, utilizando a representação Entidade.atributo ou Entidade.[atributo1, atributo2, ...]. Por exemplo, o aluno matriculado na disciplina é identificado pelo seu número de matrícula e nome (A.[matrícula, nome]), ao passo que o enunciado do trabalho é identificado apenas pelo seu título (T.título).

Essas composições e relacionamentos podem auxiliar a definição de um modelo de entidades e relacionamentos (ER) ou podem ser extraídos, com adaptações, de um ER existente. No entanto, as entidades, os atributos e os relacionamentos no modelo de dados nem sempre coincidem com os signos-entidade e signos-atributo correspondentes. Por exemplo, podemos ter um signo-atributo Aluno.período (período em que o aluno se encontra) relacionado a um trecho da interação, ao passo que no modelo de dados constaria apenas o período em que ele foi admitido na universidade, a partir do qual o período em que ele se encontra seria calculado. Nesse momento do design de IHC, o foco está na identificação de quais signos estão relacionados aos objetivos dos usuários, e não como esses signos estão representados no modelo de dados. Em outras palavras, do ponto de vista do usuário, a forma como os dados estão representados no modelo de dados não é relevante, mas sim a sua conceitualização e expressão na interface.

Nem todo atributo possui o mesmo *status*. Alguns são utilizados para *identificar univocamente* uma entidade, e não apenas para uma *caracterização* (parcial) da entidade. Eles são indicados por um sinal de mais (+). Essa distinção é importante para o designer saber, no momento do projeto da interação e da interface propriamente dita, quais informações são necessárias e suficientes para o usuário distinguir duas instâncias de uma mesma entidade. Esses signos nem sempre correspondem a chaves primárias de tabelas em bases de dados. Por exemplo, um professor pode ser identificado em uma base de dados pelo seu número de matrícula, mas os usuários do sistema identificam-no pelo seu nome. É claro que essa identificação *informal* precisará ser mapeada para uma identificação *formal* que o sistema seja capaz de “interpretar”. Caso haja uma ambiguidade (e.g., dois professores com o mesmo nome), a representação informal deve ser modificada para removê-la (e.g., utilizando um “nome de guerra” ou apelido, em vez do nome real ou em complemento a ele).

Origem de um Signo

Quanto à sua origem, os signos podem ser classificados como signos de domínio, convencionais, transformados ou de aplicação. Signos encontrados no mundo do usuário, independentemente do sistema, são chamados de signos do **domínio** (por exemplo, nome e endereço de uma pessoa). Signos originados no domínio, mas que sofrem alguma transformação ao serem incorporados ao sistema, como uma transformação resultante de analogias ou metáforas, são representados como signos **transformados** (por exemplo, diretório de arquivos digitais para substituir pastas de arquivos físicas; login para identidade e senha para assinatura). Já signos que só fazem sentido dentro do sistema, e que não têm significado prévio para os usuários, são chamados de signos da **aplicação** (por exemplo, a porcentagem de download de um arquivo). Signos transformados ou de aplicação que já tenham sido estabelecidos como convenções na cultura dos usuários são chamados de signos **convencionais** (por exemplo, zoom em editores gráficos ou interfaces baseadas em mapas).

Por que classificar os signos dessa forma? Porque tipos de signos diferentes requerem diversas tomadas de decisão por parte do designer, para aumentar a chance de o usuário interpretar adequadamente o signo, entender os valores a ele associados e saber como “falar sobre” ele (i.e., manipulá-lo) utilizando a linguagem de interface.

Signos do *domínio* e signos *convencionados* costumam ser facilmente entendidos pelos usuários. Entretanto, se o sistema impõe limitações na forma como o usuário pode falar sobre o signo, explicações se fazem necessárias. Pode haver uma restrição sobre a expressão do signo (e.g., formato ou unidade esperada do dado de entrada) ou sobre o seu 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 cinco anos; somente dias úteis). Já para os signos *transformados*, o designer deve fornecer aos usuários informações sobre os limites da analogia ou metáfora realizada de modo a transportá-los para o sistema. Por exemplo, uma explicação sobre as pastas em um ambiente de trabalho deveria indicar que o volume de documentos que uma pasta pode conter não é determinado pela pasta em si, mas sim pelo espaço disponível no disco rígido. Finalmente, os signos de *aplicação*, que podem ser totalmente desconhecidos pelos usuários, requerem uma explicação completa sobre o que significam e como são utilizados. Por exemplo, um signo enquadramento em um editor gráfico pode ser considerado como signo de aplicação.

Com o passar do tempo e aumento da familiaridade dos usuários com certos sistemas de significação, há uma tendência de os signos de aplicação e transformados se tornarem convencionados, ou seja, de serem incorporados à cultura dos usuários. Além disso, alguns signos podem gerar dúvidas no momento de classificação em um

dos tipos. Por exemplo, um signo senha pode ser classificado como convencional ou transformado, se considerarmos que ele é uma analogia à assinatura, impressão digital ou algo que identifica uma pessoa. Nesse tipo de situação, cabe ao designer definir o tipo do signo, com base no conhecimento e na experiência prévia dos usuários, bem como a quantidade de informação ou instruções a serem fornecidas sobre o signo. Em todo caso, a classificação auxilia o designer a refletir sobre a explicação a ser associada a cada signo.

Examinando a descrição dos signos, podemos identificar oportunidades de customização. No caso do signo lembrete do prazo de entrega, achamos que é interessante permitir que o usuário altere o conteúdo (valor) associado ao signo. Para isso, criamos um novo signo prazo para lembrete, que aparece sublinhado no campo de observações correspondente ao signo lembrete do prazo de entrega. Os **signos de customização** são definidos separadamente, indicando a qual elemento estão associados: ao usuário, ao dispositivo, a uma entidade, a um período de tempo ou algum outro elemento de contexto ao qual o sistema tenha acesso (e.g., a rede na qual o dispositivo está conectado).

Restrições sobre a Manipulação e o Conteúdo do Signo

À medida que o design avança, é possível definir mais informações acerca dos signos. A Tabela 7.4 apresenta informações sobre os tipos de conteúdo e restrições sobre o conteúdo de alguns signos.

Tabela 7.4 Definição do conteúdo dos signos que compõem o signo enunciado de trabalho

Enunciado de trabalho (E) – enunciado de trabalho de disciplina de graduação			
signo	tipo de conteúdo	restrição sobre o conteúdo	valor default
+ título	texto	não pode ser nulo	—
descrição	texto		—
data de entrega	data	data futura	—
formato de entrega	seleção simples	conjunto flexível: inicialmente = {relatório, protótipo}	relatório
núm. máx. de alunos	seleção simples	[1,6]	1 (individual)
peso	número real	[0,1]	1 (100%)
lembrete do prazo de entrega	seleção simples	sim/não	sim
<i>prazo para lembrete</i>	número	[1,7]	3

Observamos que formato de entrega, número máximo de alunos e lembrete do prazo de entrega são do tipo seleção simples. No entanto, há diferenças entre eles: o formato

de entrega possui um conjunto flexível de valores. Isso significa, provavelmente, que deverá haver um objetivo instrumental para a manutenção desse conjunto.

7.3.3 Prevenção e Recuperação de Rupturas Comunicativas

Como visto na Seção 3.8, a engenharia semiótica ressalta a importância de tentarmos prever, durante o design de uma solução de IHC, **rupturas (breakdowns)** na comunicação entre o preposto do designer e o usuário que podem ocorrer durante a interação. Para cada ruptura identificada, o designer deve representar os tipos de apoio à prevenção e à recuperação da ruptura que pretende oferecer aos usuários. Esses tipos de apoio podem ser classificados nas seguintes categorias (Barbosa e Paula, 2003):

- **prevenção passiva (PP)**: o preposto do designer *tenta evitar* que haja uma ruptura, fornecendo explicações sobre a linguagem de interface. Por exemplo, apresenta uma dica de formato como “(dd/mm/aaaa)” ao lado de um campo de data; ou uma instrução explícita como “asterisco (*) indica campo obrigatório”;
- **prevenção ativa (PA)**: o preposto do designer *impede* que o usuário emita falas inválidas que causem uma ruptura. Por exemplo, habilita ou desabilita um botão de acordo com o estado atual do sistema; impede que o usuário digite letras ou símbolos em campos numéricos; apresenta um conjunto fechado de opções em uma lista ou um controle de calendário que impede que o usuário indique uma data inválida;
- **prevenção apoiada (ou alerta, AL)**: o preposto do designer, ao identificar uma situação como causa potencial de uma ruptura, descreve a situação e solicita que o usuário tome uma decisão informada sobre os rumos da interação. Um exemplo de causa potencial de ruptura ocorre quando o usuário expressa a intenção de gravar um arquivo com um nome diferente (através de um item de menu Salvar como...), mas em seguida informa o nome de um arquivo existente. Geralmente esse mecanismo é concretizado na interface por diálogos de confirmação (por exemplo, “Arquivo já existe, deseja sobrescrevê-lo?”; “Foram feitas alterações no trabalho. Deseja armazená-las?”);
- **recuperação apoiada (RA)**: após uma ruptura ter ocorrido, o preposto do designer auxilia o usuário a *se recuperar* da ruptura. Ele descreve a ruptura e oferece ao usuário a oportunidade de retomar a conversa de forma produtiva. Por exemplo, quando o usuário preenche um campo incorretamente, o preposto apresenta uma mensagem descrevendo o erro no preenchimento e destaca o campo a ser corrigido, esperando que o usuário assim o faça;
- **captura de erro (CE)**: após uma ruptura ter ocorrido, o preposto do designer identifica que não será possível ao usuário se recuperar dela através da

linguagem de interface do próprio sistema. Nesse caso, o preposto descreve a ruptura e, se possível, indica ao usuário algo que ele possa fazer fora do sistema para retomar uma conversa produtiva com o sistema no futuro. Por exemplo, no caso de um arquivo corrompido, o preposto pode apresentar a mensagem “O arquivo está corrompido. Tente copiá-lo novamente da sua origem”.

Esses mecanismos de prevenção e tratamento de rupturas podem ser relacionados a uma tarefa (veja Seção 7.3.4), a um trecho de interação (veja Seção 7.3.5) ou a um signo. A Tabela 7.5 define os mecanismos de prevenção e recuperação de rupturas comunicativas associadas aos signos definidos na Tabela 7.3.

Tabela 7.5 Mecanismos de prevenção e recuperação dos signos-atributo do signo-entidade enunciado de trabalho

Enunciado de trabalho (E) – enunciado de trabalho de disciplina de graduação		
signo	prevenção	recuperação
+ título	PP: campo obrigatório	RA
descrição	—	—
data de entrega	PP+PA: apenas datas futuras podem ser informadas	—
formato de entrega	PA: ao menos uma opção está sempre selecionada	—
número máximo de alunos	PA: ao menos uma opção está sempre selecionada	—
peso	PP: campo numérico entre 0 e 1	RA
lembrete do prazo de entrega	PA: ao menos uma opção está sempre selecionada	—

Nesse exemplo, são utilizados três tipos de mecanismos de prevenção e recuperação de rupturas. A prevenção ativa (PA) associada aos signos data de entrega, formato de entrega, número máximo de alunos e lembrete do prazo de entrega impede o usuário de deixar campos obrigatórios em branco. Isso poderá ser mapeado, na interface, em uma lista ou grupo de botões de opção (*radio buttons*), nos quais um valor default vem selecionado, e não é possível ter uma seleção nula.

A prevenção passiva (PP) associada aos signos título e peso indica que a interface fornecerá instruções para evitar uma ruptura comunicativa durante a interação (e.g., uso de * para indicar os campos obrigatórios, juntamente com uma instrução textual “* indica os campos obrigatórios”). No entanto, como não há garantias de que o usuário vá entender ou respeitar essas instruções, sempre que um signo estiver associado a uma forma de prevenção passiva, também deve estar associado a alguma forma de recuperação apoiada (RA) equivalente (RA: campo obrigatório ou, como a prevenção passiva já indica a natureza ou causa da possível ruptura, apenas RA).

7.3.4 Modelagem de Tarefas

A partir das atividades e representações de análise (veja Capítulos 5 e 6), podemos estruturar cada objetivo do usuário de forma a explorar diferentes estratégias que o usuário poderá seguir para alcançá-lo. No design pautado pela engenharia semiótica, os modelos de tarefas representam não apenas a estrutura hierárquica das tarefas do usuário que compõem um objetivo, mas também estruturas de sequência e iteração, além de tarefas alternativas, independentes de ordem, opcionais e ubíquas. Além disso, para cada tarefa são representados os signos associados, os mecanismos de prevenção e tratamento de rupturas de comunicação e as precondições para a tarefa, se houver.

A representação de tarefas utilizada na engenharia semiótica (Paula, 2003; de Souza, 2005a; Prates e Barbosa, 2007) também segue uma decomposição hierárquica (Seção 6.4). O **objetivo** de mais alto nível é representado por um retângulo com bordas arredondadas. Ele é composto de **tarefas**, representadas por retângulos. A decomposição prossegue até chegarmos a **operadores**, que representam ações atômicas, ou seja, que podem ser mapeadas diretamente em ações sobre um elemento de interação na interface (*widget*). São exemplos de operadores: Escolher tipo de busca e Efetuar busca. Os operadores são representados como um retângulo sobre uma linha.

Ao elaborarmos um modelo de tarefa, devemos evitar nos concentrar em um ambiente ou uma plataforma tecnológica específica. A decomposição do objetivo ou de uma tarefa em tarefas menores e operadores deve parar antes que o modelo inclua detalhes de interface, tais como Digitar X, Pressionar botão Y etc. Essa consideração não apenas facilita o reúso 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. Observamos ainda que tudo o que o usuário vai realizar diretamente na interface está representado no último nível da estrutura hierárquica, em geral por operadores. Uma tarefa é realizada apenas indiretamente, com a realização dos operadores que a compõem. Para fazer a distinção entre tarefa e operador, podemos tentar formular a atividade do usuário da seguinte maneira: “*Para realizar/atingir A, é preciso fazer X, Y e Z*”. Caso X, Y e Z sejam necessariamente descritos em termos de elementos de interação e de interface, A pode ser considerado um operador.

Considere um sistema que permita ao professor buscar uma apresentação existente, para ajudá-lo a preparar o material didático da sua disciplina atual. Na Figura 7.4, Buscar apresentação é o objetivo, Definir busca é uma tarefa, e Efetuar busca é um operador.

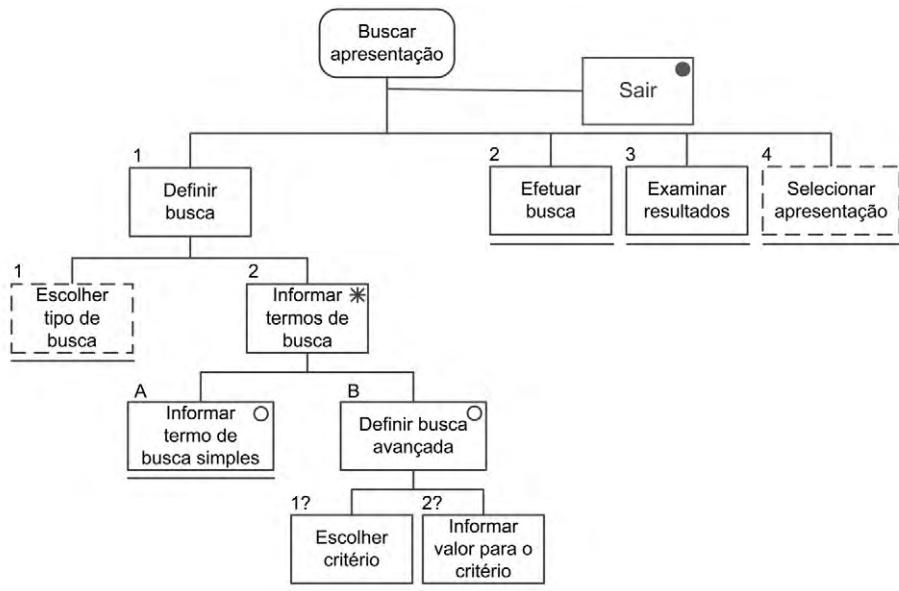


Figura 7.4 Modelo hierárquico de tarefas adaptado.

As tarefas e os operadores podem ser organizados nos seguintes tipos de estruturas: sequenciais, independentes de ordem, alternativas, iterativas e ubíquas. Além disso, uma tarefa pode ser representada como opcional (Figura 7.5).

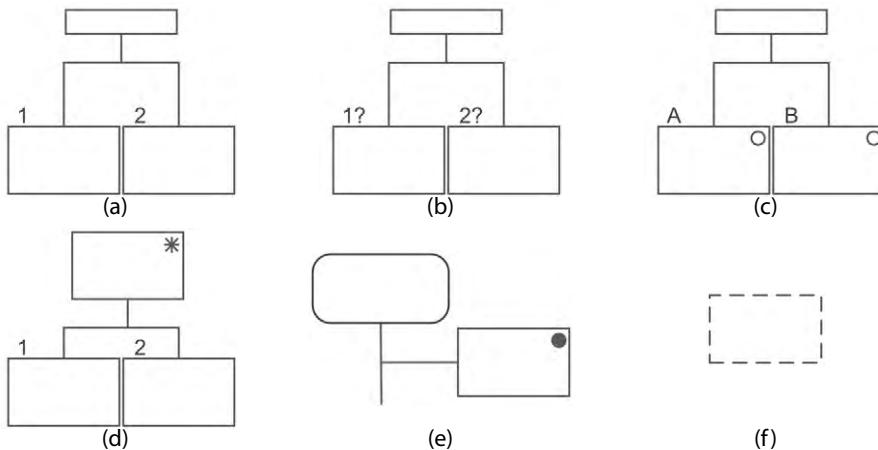


Figura 7.5 Representação de estrutura de tarefa (a) sequencial, (b) independente de ordem, (c) alternativa, (d) iterativa, (e) ubíqua e (f) opcional.

Em uma estrutura **sequencial**, existe uma ordem em que as tarefas devem necessariamente ser efetuadas pelo usuário. As tarefas, nessa estrutura, são representadas por retângulos contendo o nome da tarefa e um *número* indicando sua posição na sequ-

ência (Figura 7.5a). Vale observar que o nome da tarefa deve ser expresso do ponto de vista do usuário. Por exemplo, uma tarefa em que o usuário examina os resultados da busca apresentados pelo sistema deve se chamar *Examinar resultados* e não *Apresentar resultados*.

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. Geralmente, o projetista sugere uma ordem de execução, mas é o usuário quem determina, de fato, em que ordem as tarefas serão efetuadas. Nesse tipo de estrutura, as tarefas são representadas como as tarefas sequenciais, mas, como a ordem é apenas sugerida pelo designer, incluímos um *ponto de interrogação* após o número que indica a posição sugerida da tarefa na estrutura (Figura 7.5b). No exemplo da Figura 7.4, identificamos como independentes de ordem os operadores *Escolher critério* e *Informar valor para o critério*. Percebemos ainda que o designer *sugere* que seja essa a ordem de realização dessas tarefas, mas não a *impõe*.

Para o alcance de um objetivo, há momentos em que diversos cursos de ação são possíveis. Tais cursos de ação são representados por uma estrutura **alternativa**, em que o usuário deverá selecionar qual das tarefas da estrutura será efetuada, conforme a estratégia que queira adotar. Nessa estrutura, utilizamos *pequenos círculos* no canto superior direito do retângulo de cada tarefa alternativa e *letras* como identificadores em vez de números (Figura 7.5c). No exemplo da Figura 7.4, identificamos como alternativas as formas de busca: *Busca simples* ou *Busca avançada*.

Quando uma tarefa pode ser realizada diversas vezes, utilizamos uma estrutura **iterativa**. Um asterisco (*) no canto superior direito do retângulo é usado para indicar a iteração (Figura 7.5d). No exemplo da Figura 7.4, a tarefa *Definir critérios* é iterativa. Geralmente, uma tarefa iterativa representa tarefas que podem ser efetuadas uma ou mais vezes. Caso seja necessário definir um número mínimo ou máximo de repetições, podemos 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.

Quando o usuário pode realizar uma tarefa a partir de qualquer momento da interação para atingir o objetivo desejado, ela é dita **ubíqua**, e é representada por um retângulo com círculo cheio, ligado à aresta que liga o objetivo de mais alto nível aos seus descendentes (Figura 7.5e). Finalmente, quando o usuário pode optar por realizar ou não uma tarefa, ela é dita **opcional**, e é representada com uma *borda tracejada* (Figura 7.5f). No exemplo da Figura 7.4, o operador *Escolher tipo de busca*

é opcional, possivelmente indicando que deve haver um tipo default de busca (e.g., busca simples).

Podemos representar também, associados 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. Por exemplo, na tarefa iterativa Definir critérios da busca avançada, podemos representar o seguinte conjunto de signos e mecanismos de prevenção e tratamento de rupturas comunicativas:

definir critérios		
signo	prevenção	recuperação
critério	PP: pelo menos um critério deve ser informado	RA
valor	PP: pelo menos um valor deve ser informado	RA

Nesse caso, a interface fornecerá instruções sobre a obrigatoriedade do usuário informar pelo menos um critério e um valor (PP). No entanto, o usuário pode tentar realizar a busca sem definir um critério e um valor. Isso causará uma ruptura, que será tratada através de um mecanismo de recuperação apoiada (RA).

Uma solução alternativa consiste em impedir que o usuário consiga concluir a tarefa sem que as restrições sejam respeitadas. Nesse caso, existe uma forma de prevenção ativa (PA), associada ao signo que representa a operação propriamente dita, no caso, buscar. Isso poderia ser representado da seguinte maneira:

definir critérios		
signo	prevenção	recuperação
critério	PP: pelo menos um critério deve ser informado	—
valor	PP: pelo menos um valor deve ser informado	—
buscar	PP+PA: pelo menos um critério e um valor devem ser informados	—

Essa prevenção ativa pode ser concretizada, por exemplo, desabilitando o botão ou outro elemento de interface utilizado para acionar a busca. É importante destacar que, quando oferecemos um mecanismo de prevenção ativa, devemos deixar muito claro o motivo de uma determinada ação não estar disponível, e o que o usuário pode fazer para torná-la disponível. Em outras palavras, sempre que houver uma prevenção ativa associada a um signo de operação, deve haver também alguma forma de prevenção passiva associada a esse signo, indicado na tabela por PP+PA.

Em geral, não modelamos as tarefas relacionadas a todos os objetivos do sistema. A modelagem de tarefas se destina principalmente a explorar tarefas com estruturas mais complexas do que a do exemplo apresentado nesta seção. Para muitos sistemas,

os designers passam diretamente da definição do mapa de objetivos para a modelagem da interação, descrita a seguir.

7.3.5 Modelagem da Interação

Paula, Silva e Barbosa propuseram, no âmbito da engenharia semiótica, uma linguagem para a modelagem da interação humano-computador como uma conversa, denominada MoLIC (*Modeling Language for Interaction as Conversation* — Paula, 2003; Barbosa e Paula, 2003; Silva, 2005). A MoLIC foi projetada para apoiar os designers no planejamento da interação, motivando sua reflexão sobre as estratégias de realização de atividades e resolução de problemas dos usuários que deveriam ser apoiadas pelo sistema interativo.

A MoLIC permite representar a interação humano-computador como o conjunto de conversas que os usuários podem (ou devem) travar com o preposto do designer para atingir seus objetivos. Nessas conversas, o preposto do designer precisa comunicar adequadamente aos usuários: 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. A MoLIC foi projetada de modo a ser não apenas uma notação para especificar a interação, mas também como uma ferramenta epistêmica, ou seja, para aumentar a compreensão dos designers sobre o problema sendo resolvido e o artefato sendo projetado. É importante observar que a MoLIC foi proposta para uso humano e, portanto, não representa um modelo formal processável por computador.

A elaboração de um diagrama MoLIC parte geralmente da definição dos perfis de usuários ou personas, dos objetivos dos usuários, dos cenários de análise e/ou interação e dos signos mencionados nos cenários. O diagrama de interação representa **como** os objetivos poderão ser atingidos durante a interação. Assim como cenários e modelos de tarefas, o diagrama de interação MoLIC serve como ponte entre a definição dos objetivos dos usuários (veja Seção 7.3.1) e o projeto da interface propriamente dita (veja Seção 7.4.3). Diferentemente das outras representações, a MoLIC foi concebida para motivar os designers a refletir sobre a metacomunicação, incentivando-lhes a decidir como lidar com as rupturas de comunicação, a explorar conversas alternativas para o atingimento de um mesmo objetivo e a analisar o relacionamento e interferências entre objetivos (e, portanto, especificar as conversas relativas a objetivos instrumentais diretos). A representação diagramática promove uma visão global do sistema tal como o preposto vai apresentá-lo ao usuário. Além disso,

ajuda o designer a verificar a consistência na interação sendo projetada e a identificar oportunidades de simplificação da interação.

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 deseja realizar (e a forma 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 tornadas explícitas durante o design da interação.

A construção de diagramas MoLIC é realizada em duas etapas. Primeiro, os designers definem os tópicos de todas as possíveis conversas usuário-sistema e as trocas de turno entre o usuário e o preposto do designer que encadearão os tópicos dessas conversas. Esse nível de abstração encoraja reflexões, análises e discussões sobre a interação por uma equipe multidisciplinar cedo no processo de design (Paula *et al.*, 2005). Em seguida, os tópicos são detalhados, e os designers definem os diálogos e 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.

Definição da Estrutura da Conversa: Tópicos e Mudanças de Tópico

Na primeira etapa da construção de um diagrama MoLIC, os designers refletem sobre as seguintes questões:

- tópicos das conversas (troca de turnos entre usuário e preposto do designer) em direção a um objetivo;
- conversas alternativas em direção a um mesmo objetivo, possivelmente endereçando as necessidades e preferências de diferentes perfis de usuários;
- mudanças de tópico relativas a objetivos instrumentais diretos;
- conversas para a recuperação de rupturas, i.e., mecanismos para os usuários se recuperarem de problemas na comunicação com o preposto do usuário;
- a consistência entre caminhos de interação semelhantes ou análogos.

Podemos partir de uma visão simplificada, em que cada objetivo de usuário (Figura 7.2) é mapeado para um tópico de uma cena, conforme ilustrado pelos diagramas parciais de interação do professor e do aluno apresentados na Figura 7.6.

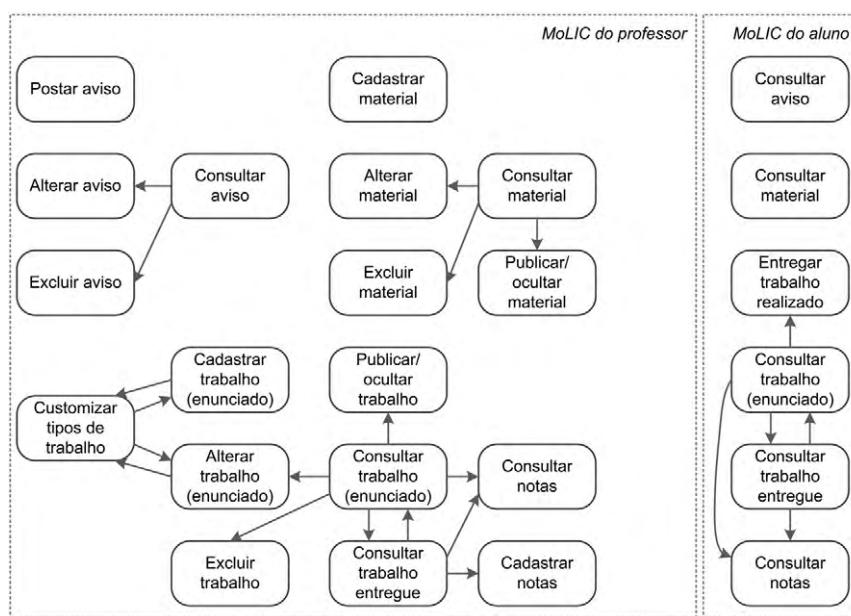


Figura 7.6 Diagramas parciais de interação do professor e do aluno, cujos tópicos foram extraídos do mapa de objetivos.

As cenas representam conversas sobre um determinado tópico, culminando na vez de o usuário dizer algo para concluir a conversa, suspender-a, desviar do tópico ou mesmo abandoná-la. Uma cena pode ser vista como uma cena real em uma peça teatral, em que ocorrem as trocas comunicativas entre usuário e preposto do designer. Uma cena é representada por um retângulo com bordas arredondadas.

O tópico de uma cena pode ser lido, do ponto de vista do preposto, como: “Neste momento, você (usuário) pode (ou deve) <tópico>”. No momento, não há uma diferenciação, na representação dos tópicos, entre “você pode X”, “você deve X” ou “eu recomendo que você X”.¹

Na MoLIC, as mudanças de tópico são representadas por falas de transição, sejam do usuário ou do preposto. Uma **fala de transição** 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. Desse modo, as setas da Figura 7.6 precisam ser rotuladas para indicar as mudanças de tópico. A Figura 7.7 apresenta apenas o diagrama do aluno, já com as falas de transição do usuário.

1 Estamos investigando formas de representar essa diferenciação, a fim de aumentar o caráter epistêmico da MoLIC e facilitar o mapeamento da MoLIC para a interface concreta.

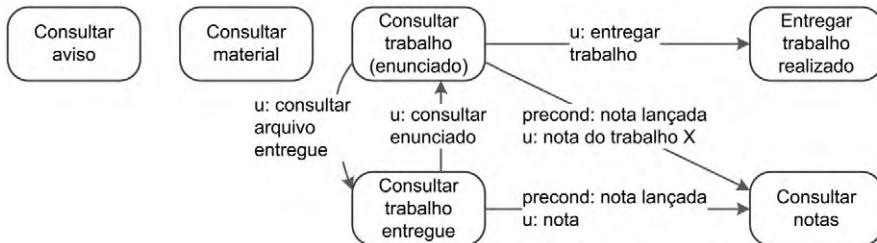


Figura 7.7 Diagrama (parcial) de interação do aluno com falas de transição de usuário.

Podemos observar que só faz sentido para um aluno consultar a nota de uma atividade caso ela já tenha sido lançada pelo professor. Sendo assim, as falas de transição para a cena Consultar notas possuem, além do rótulo que identifica a fala, uma expressão precedida da palavra-chave precond. Essa expressão define uma **precondição** para que o usuário possa emitir a fala correspondente. Isso significa que, quando um aluno estiver consultando uma atividade entregue, caso a nota correspondente ainda não tenha sido lançada pelo professor, ele não poderá emitir a fala u: nota. Uma precondição pode ser concretizada, na interface, por um link ou botão desabilitado, ou até mesmo pela ausência de um elemento com essa função. Isso evita que o aluno siga um link ou pressione um botão acreditando que sua nota já esteja disponível, apenas para receber uma mensagem de que a nota ainda não foi lançada.

A Figura 7.7 representa apenas mudanças de alguns tópicos para outros tópicos relacionados, através das falas de transição entre cenas. Além disso, o usuário deve ter a possibilidade de iniciar uma nova conversa em direção a um novo objetivo, seja ele representado por uma cena com tópico relacionado ao tópico da cena atual, ou não. Para isso, utilizamos **acessos ubíquos**, que representam o início de uma conversa em direção a um objetivo, e cujas falas de transição podem ser emitidas em qualquer momento durante a interação.

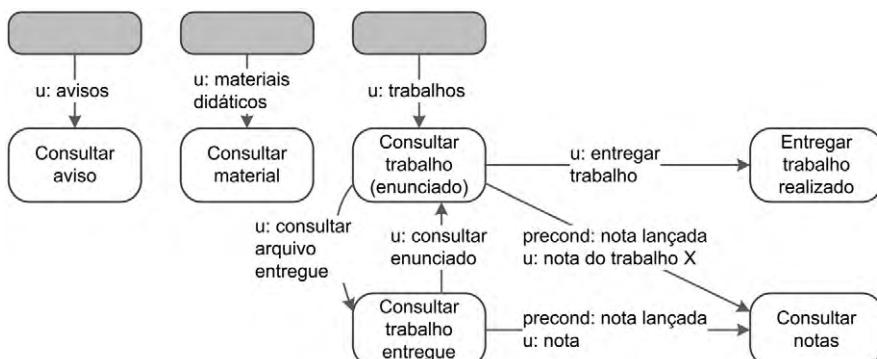


Figura 7.8 Diagrama (parcial) de interação do aluno, incluindo os acessos ubíquos.

Os acessos ubíquos são representados por uma cena anônima de fundo cinza mais a fala de transição do usuário para a cena de destino, conforme ilustrado pela Figura 7.8.

Até agora, não definimos quais são os **pontos de abertura** da conversa usuário–preposto, ou seja, não definimos em quais cenas pode começar a interação quando o usuário entra no sistema. Em sistemas multiusuários, geralmente a conversa se inicia com uma cena de login, que, por sua vez, costuma levar o usuário identificado a uma cena semelhante a uma homepage, com um quadro resumido com informações relevantes e recentes, indicando também os objetivos que o usuário pode atingir com o sistema. Analogamente, não definimos o **ponto de encerramento**, ou seja, o que ocorre quando o usuário sai do sistema. Pontos de abertura são representados por círculos preenchidos na cor preta, e pontos de encerramento por um círculo na cor preta circunscrito a um círculo branco, conforme ilustrado pela Figura 7.9.

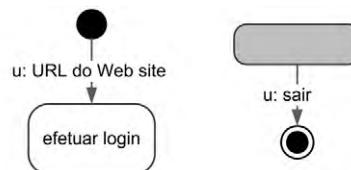


Figura 7.9 Diagrama (parcial) de interação do aluno, incluindo os pontos de abertura e encerramento da conversa usuário–preposto.

Na maioria dos ambientes, a conversa se inicia no momento em que o sistema é ativado através do sistema operacional. Em um navegador, no momento em que uma URL válida é digitada ou um link é seguido para a aplicação Web. Em sistemas baseados em documentos, por outro lado, geralmente há dois pontos de abertura: um acessado ativando-se o sistema, e outro acessado quando um documento produzido por aquele sistema é ativado. Em cada caso, a conversa pode iniciar de forma diferente: abrindo um documento em branco ou o documento acessado. A Figura 7.10 ilustra dois pontos de abertura em um editor de apresentações (*slides*).

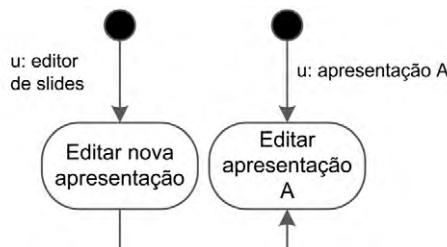


Figura 7.10 Pontos de abertura da conversa global usuário–preposto em um editor de apresentações (diagrama parcial).

Nessa figura, vemos que o usuário pode iniciar a conversa global do sistema solicitando ao sistema operacional que abra o editor (através da fala de usuário “u: editor de slides”) ou que abra uma apresentação específica (através da fala de usuário “u: apresentação A”).

Nem sempre uma fala de transição do usuário o leva diretamente para a cena de destino. Em alguns momentos da interação há uma troca de turno, em que o preposto do designer deve *interpretar* (i.e., processar) o que o usuário disse e responder adequadamente. Dois elementos da MoLIC são utilizados para indicar a troca de turno e a resposta do preposto: um processo de sistema e uma fala de transição (ou fala de troca de turnos).

Um **processo de sistema** é representado para indicar a vez de o sistema interpretar o que o usuário disse e decidir como a conversa irá prosseguir, ou seja, qual será o próximo tópico da conversa. Enquanto o sistema está “pensando”, o usuário não sabe o que está ocorrendo, a menos que o preposto lhe mantenha informado, durante e/ou após o processamento. A representação desse elemento visa motivar os designers a pensarem a respeito de como melhor comunicar aos usuários sobre o progresso e os resultados do processamento do sistema, sobre para onde a conversa vai dali e por quê. Um processo de sistema é representado num diagrama MoLIC por uma “caixa preta” (quadrado 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 durante o processamento), como e por quê. A Figura 7.11 apresenta um processo do sistema quando a fala do usuário u: iniciar busca passa o turno para o preposto do designer processar sua solicitação.

Existem duas formas de comunicação do preposto para o usuário sobre um processamento: consecutiva, logo após o processamento, e síncrona, durante o processamento. Uma **comunicação consecutiva** é representada como uma fala de transição do preposto para uma cena, e é rotulada como d: *resposta*, conforme ilustrado na Figura 7.11 para o resultado d: n itens encontrados. Essa comunicação poderá ser concretizada na interface de diferentes maneiras, por exemplo: como uma tela de mensagem independente, ou como um ou mais signos na tela correspondente à cena de destino (veja Seção 7.4.3). Já uma **comunicação síncrona** é projetada para comunicar ao usuário o progresso do processamento ou seus estados intermediários. É utilizada geralmente em processos longos, como, por exemplo, durante o download de um arquivo, um cálculo científico complexo ou a realização de uma busca em uma base de

dados muito grande. Nesses casos, o preposto do designer pode emitir diversas falas *durante* o processamento. Para representar esse tipo de comunicação, utilizamos uma cena acoplada à caixa preta por um canal de comunicação, que pode apresentar falas do designer sincronizadas com o processamento do sistema e sobre ele. Esse é o caso da cena Examinar progresso de busca, acoplada ao processo do sistema na figura.

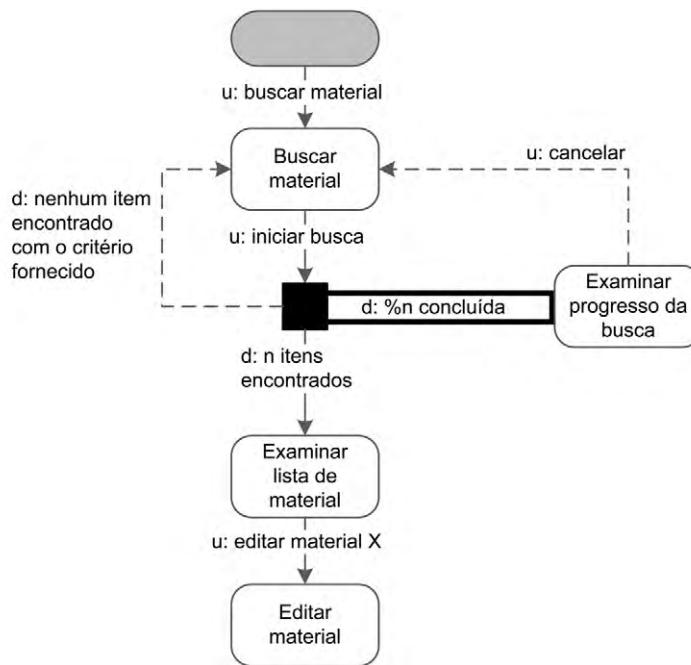


Figura 7.11 Processo do sistema e comunicação consecutiva e síncrona.

É importante assegurar que a comunicação sobre um processamento seja uma indicação correta do estado ou resultado 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 a ele algum controle sobre esse processamento, tal como suspendê-lo, cancelá-lo ou ajustá-lo. Portanto, pode haver falas de transição de usuário saindo da cena acoplada ao processamento, como ilustrado na figura.

Podemos observar, ainda na Figura 7.11, que a fala do usuário *u: cancelar* está traçejada. Trata-se de uma **fala de recuperação de ruptura** comunicativa ou *breakdown*. Esse tipo de fala representa uma oportunidade explicitamente projetada pelo designer para o usuário se recuperar de uma conversa accidental (não intencional) ou de uma conversa que não tomou o rumo esperado. Quando é emitida pelo usuário, a fala de recuperação de ruptura indica um momento em que o usuário pode mudar

de ideia e, consequentemente, o rumo da conversa. Na figura, u:cancelar indica que o usuário pode abandonar a busca durante seu processamento, por exemplo, ao perceber que está demorando mais do que ele esperava.

Quando a fala de recuperação de ruptura é emitida pelo preposto do designer, ela indica que o preposto não conseguiu interpretar uma ou mais falas do usuário adequadamente e é necessário que o usuário as retifique. Considerando ainda o exemplo da busca, a Figura 7.11 apresenta uma fala de recuperação de ruptura emitida pelo preposto: d: nenhum item encontrado com o critério fornecido. Essa fala representa uma conversa de recuperação para que o usuário possa retomar uma conversa produtiva em direção ao seu objetivo, tal como presumido pelo designer. Caso, na interpretação do designer, um dos possíveis objetivos da busca seja justamente se certificar de que um material didático não foi cadastrado, observe que essa fala de transição não seria considerada uma fala para recuperação de ruptura e seria representada por uma linha sólida.

Devemos representar um processo do sistema somente quando seu resultado puder causar uma mudança no rumo da conversa, ou quando for necessário que o preposto do designer notifique o usuário sobre o progresso ou resultado desse processamento. Sendo assim, é desnecessário representar um processo do sistema após uma fala de transição do usuário que sempre resulte na mudança de tópico esperada. Na Figura 7.11, a fala u: editar material X, a partir da cena Examinar lista de material, sempre muda o tópico da conversa para a cena Editar material e, portanto, não precisa ser intermediada por um processo do sistema.

Ao devolver o turno para o usuário, o preposto pode levá-lo a uma nova cena ou de volta à cena da qual foi emitida a fala de transição do usuário. Quando o designer deseja apenas comunicar o resultado do processamento do sistema, sem introduzir um novo tópico, ele pode levar o usuário para uma cena cujo tópico é representado apenas por “—”, indicando que a cena corresponde a uma resposta que conclui a conversa sobre o tópico da cena precedente. A Figura 7.12 ilustra dois designs de interação alternativos para o processo de inscrição em um Web site.

Na Figura 7.12a, após solicitar sua inscrição, o usuário é levado a uma cena com o tópico “—”, na qual o preposto lhe comunica que deve aguardar um e-mail do administrador do sistema confirmando sua inscrição. A partir dessa cena, o usuário pode ir para a cena de Efetuar login (através da fala u:login), se quiser. Já na Figura 7.12b, o preposto comunica ao usuário que ele deve aguardar e-mail do administrador e o leva diretamente para a cena Efetuar login.

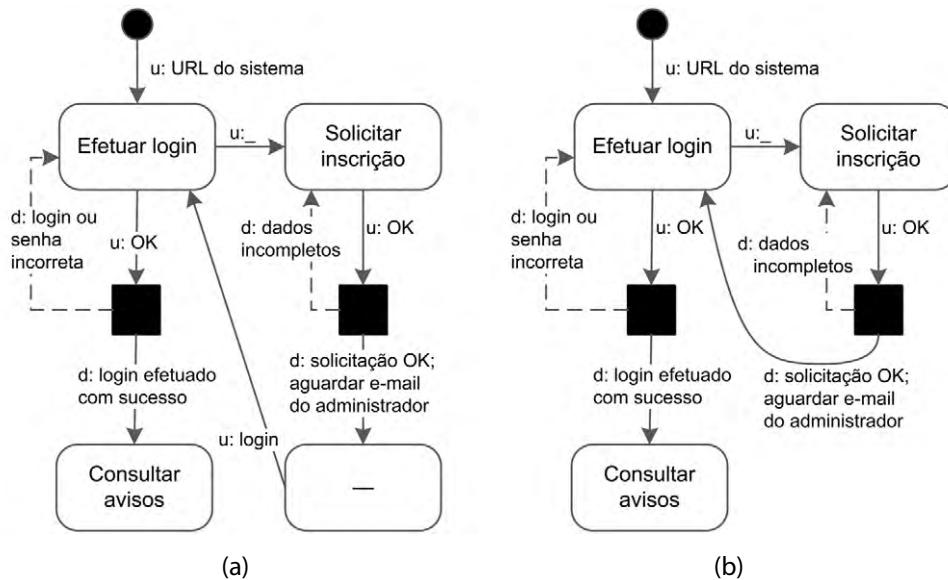


Figura 7.12 Resposta do preposto do designer sobre um processamento, a) sem mudança de tópico; b) com mudança de tópico.

Algumas falas de usuário são rotuladas como *u: _*. Essa notação indica que a fala corresponde ao tópico da cena de destino. Por exemplo, a fala que leva o usuário à cena *Solicitar inscrição* pode ser representada por *u: solicitar inscrição* ou por *u: _*. Não se trata apenas de uma conveniência sintática, mas também um lembrete de que deve haver correspondência entre as falas do usuário e os tópicos da cena. Por exemplo, em um mapeamento desse diagrama de interação para uma interface Web, um link *solicitar inscrição* (correspondente à fala *u: _*) levaria a uma página intitulada *Solicitar inscrição* ou *Solicitando inscrição* (correspondente à cena *Solicitar inscrição*). Essa página não deveria ser intitulada *Efetuar cadastro*, *Abrir conta* ou algo semelhante.

Além disso, algumas falas são marcadas como *u: OK*. Isso não significa que vá haver um botão com rótulo *OK* na interface. Trata-se apenas de uma convenção para indicar que o usuário deseja concluir seu objetivo, tal como indicado pelo tópico da cena de origem dessa fala. Analogamente, é possível utilizar *d: OK* para representar que a interpretação do designer conseguiu produzir o efeito desejado pelo usuário tal como expresso pela sua fala.

Como visto na Seção 7.3.3, existem situações que o designer identifica como sendo rupturas em potencial, mas cujo diagnóstico final deve ser feito pelo usuário. Cabe ao preposto descrever adequadamente a situação e solicitar que o usuário tome uma decisão informada sobre os rumos da interação, no mecanismo de prevenção denominado alerta (AL). Uma **cena de alerta** é representada com uma linha tracejada

da, conforme ilustrado na Figura 7.13. Uma **cena de captura de erro** também é representada com uma linha tracejada, mas não requer que o preposto do designer projete, a partir dela, uma fala de transição para recuperação da ruptura, pois geralmente o sistema não poderá ajudar o usuário a se recuperar dessa ruptura.

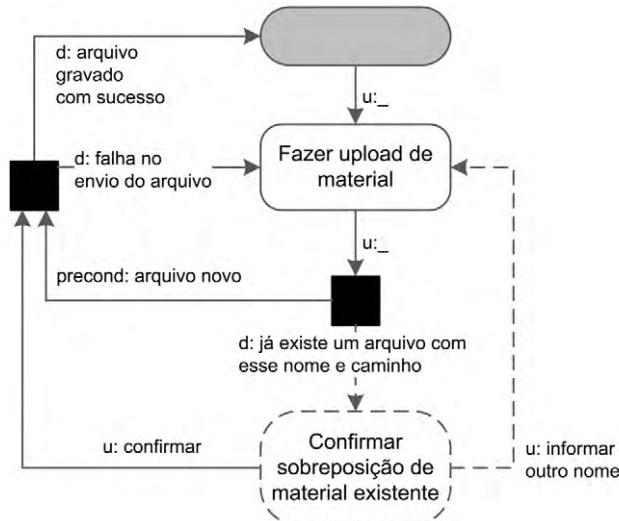


Figura 7.13 Alerta de sobreposição de arquivo existente.

Como é natural e frequente que mal-entendidos ocorram numa conversa, a engenharia semiótica destaca a importância de representar as rupturas comunicativas que podem ocorrer durante a interação. É necessário definir a forma como o preposto irá comunicar 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. No diagrama de interação, representamos o mecanismo de recuperação apoiada através de falas de transição para recuperação de ruptura, e os mecanismos de alerta e captura de erro como cenas tracejadas. Já os mecanismos de prevenção passiva e ativa são representados apenas no esquema conceitual de signos (veja Seção 7.3.2) e no detalhamento dos diálogos (visto na próxima seção).

À medida que o designer avança na modelagem da interação, podem surgir inconsistências na forma como o usuário poderá interagir com o sistema para alcançar objetivos semelhantes. Por exemplo, considere as conversas para postar avisos, cadastrar um material didático e cadastrar uma atividade projetadas conforme a Figura 7.14.

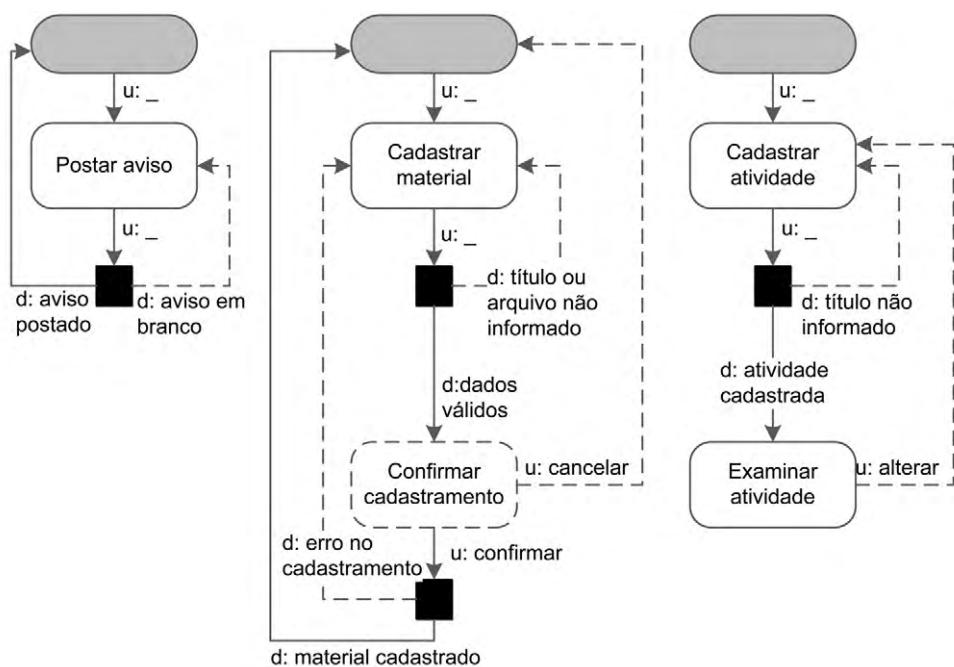


Figura 7.14 Diferentes tipos de conversa para o cadastramento de aviso, material didático e atividade.

Podemos observar três formas de diferentes de “cadastrar” algo no sistema:

- no caso de Postar aviso, o preposto grava os dados sem pedir confirmação e leva o usuário para a cena em que ele estava antes de postar o aviso. Isso dificulta a verificação do aviso postado;
- no caso de Cadastrar material, o preposto solicita a confirmação do usuário antes de efetuar o cadastramento propriamente dito. Isso facilita a verificação do material cadastrado e evita que a base de dados contenha momentaneamente informações incorretas. No entanto, é possível que o usuário, dependendo da sua experiência com sistemas semelhantes, acredite que o cadastro já tenha sido efetuado e acabe não efetuando a confirmação;
- no caso de Cadastrar atividade, o preposto não pede confirmação, grava os dados e leva o usuário para a cena em que ele pode examinar o que acabou de cadastrar e, se necessário, a partir da qual ele pode iniciar uma conversa para alterá-los. Diferentes conversas para alcançar objetivos semelhantes em um mesmo sistema podem confundir o usuário, que terá dificuldade em se lembrar qual tipo de conversa foi projetado para o cadastro de qual tipo de informação. Existem casos em que a diferença é intencional, por exemplo, o designer quer solicitar a confirmação prévia apenas para dados críticos. No

entanto, essa decisão deve ser tomada de forma deliberada, e não incidental. Os diagramas MoLIC ajudam a identificar esse tipo de inconsistência e promovem uma reflexão e tomada de decisão de design mais informada.

Para facilitar o entendimento e aumentar a consistência da conversa, podemos padronizar os verbos utilizados nos tópicos das cenas, como, por exemplo: buscar, ver (lista de), examinar (detalhes de), analisar, comparar, cadastrar (ou criar ou postar), alterar, remover, confirmar e controlar (González-Calleros *et al.*, 2009). Dependendo do domínio do sistema, um conjunto distinto de verbos pode ser utilizado. Cabe ao designer manter esse vocabulário controlado, considerando, naturalmente, o vocabulário utilizado pelos usuários, incluindo sinônimos e destacando eventuais diferenças sutis, sempre que necessário. Por exemplo, do ponto de vista do sistema, cadastrar e postar podem ser considerados sinônimos, pois ambos envolvem criar uma nova instância de uma entidade. Do ponto de vista do usuário, no entanto, ele não “cadastra” um aviso, mas sim “posta” um aviso. Cabe ao designer decidir o quanto essas semelhanças e diferenças na terminologia utilizada pelos usuários devem se refletir em consistência ou quebra de consistência na interação e, posteriormente, na interface com o usuário.

Detalhamento dos Diálogos e Falas

Na segunda etapa da construção de um diagrama MoLIC, os designers detalham a conversa sobre cada tópico, especificando os diálogos, as falas e os signos de cada cena.

Como visto no início desta seção, a conversa sobre um tópico pode ser composta por diversos diálogos, que, por sua vez, são compostos por falas sobre signos. Considerando a cena Entregar trabalho, no diagrama de interação do aluno, podemos identificar os seguintes diálogos: ver turma (dados de contexto), ver enunciado e informar dados do trabalho. Além disso, caso o trabalho deva ser feito em grupo, também temos o diálogo informar integrantes do grupo. Os diálogos são representados num segundo compartimento da cena (Figura 7.15). Em geral, são enunciados na forma *verbo + objeto*, em que o objeto costuma ser um signo-entidade ou signo-atributo do domínio.

Podemos observar, na Figura 7.15, o uso da palavra-chave precond, indicando a precondição que deve ser satisfeita para que preposto e usuário possam travar o diálogo correspondente.

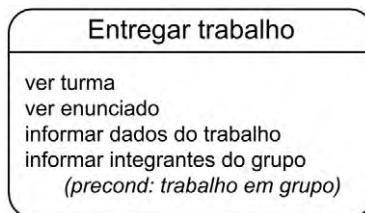


Figura 7.15 Cena com diálogos.

Assim como ocorre nos tópicos das cenas, para facilitar o entendimento e aumentar a consistência dos diálogos, podemos padronizar os verbos utilizados nos diálogos do sistema, agora com granularidade mais fina. Podemos utilizar, por exemplo, o seguinte conjunto: *ver, comparar, informar, selecionar, filtrar, ordenar, acrescentar, remover, ativar, desativar, iniciar, pausar, retomar, parar e abortar* (González-Calleros *et al.*, 2009). Cabe ao designer estabelecer o conjunto de verbos relevantes ao domínio, e decidir como eles serão concretizados na interface. Por exemplo, em interfaces que não sejam gráficas, não faz sentido “ver” algo. Nesse caso, o diálogo *ver turma* poderia ser substituído por algo como conferir dados da turma.

Existem casos em que os diálogos podem ser compostos de outros diálogos, segundo uma determinada estrutura. Caso haja dependência entre diálogos, e eles precisem ser travados em uma determinada ordem, podemos utilizar a palavra-chave SEQ, como a seguir:

```

SEQ {
    selecionar estado
    selecionar cidade
}
  
```

Já um diálogo para informar os dados de contato de um usuário poderia ser estruturado da seguinte forma:

```

informar dados para contato {
    d+u: forma de contato (e-mail, telefone ou correspondência)
    XOR {
        informar e-mail (se forma de contato = e-mail)
        informar telefone (se forma de contato = telefone)
        informar endereço (se forma de contato = correspondência)
    }
  
```

Nesse caso, a palavra-chave XOR indica que apenas um dos diálogos da estrutura pode ser travado pelo usuário e preposto. Caso a palavra-chave utilizada seja OR, um ou mais diálogos podem ser travados. E, caso a palavra-chave utilizada seja AND, todos

os diálogos devem ser travados. Como a MoLIC é uma linguagem para processamento humano, outros casos podem ser indicados por comentários, como, por exemplo, a restrição de que ao menos uma forma de contato precisa ser informada:

```

informar dados para contato {
    d+u: forma de contato preferencial (e-mail, telefone ou correspondência)
    OR (pelo menos uma forma de contato precisa ser informada) {
        informar e-mail
        informar telefone
        informar endereço
    }
}

```

A partir da definição dos diálogos, podemos detalhar os signos envolvidos em cada diálogo (Figura 7.16). Caso um signo seja emitido apenas pelo preposto do designer, ele é precedido por d:. Caso seja emitido por ambos, preposto do designer e usuário (ou seja, caso o designer dê oportunidade para o usuário falar sobre o signo), o signo é precedido por d+u:.

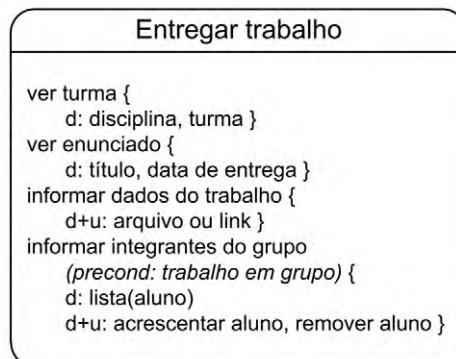


Figura 7.16 Cena com diálogos e signos.

Podemos observar que nem tudo o que está representado como “diálogo” se desdobra, durante a interação, em uma conversa entre os dois interlocutores (usuário e preposto do designer) sobre determinado subtópico. No diálogo ver turma, por exemplo, apenas o designer emite uma ou mais falas sobre os signos disciplina e turma. O mesmo ocorre com o diálogo ver enunciado. Já no diálogo informar dados do trabalho, ambos interlocutores falam sobre o subtópico: o preposto solicita ao usuário falar sobre um dos signos (ou ambos), arquivo ou link, e o usuário assim o faz. Finalmente, no diálogo informar integrantes do grupo, há uma parte em que ocorre um monólogo, na qual o preposto do designer emite falas sobre os alunos já cadastrados como integrantes do grupo (assumindo que o grupo já tenha sido definido anteriormente), tal como iden-

tificado pela expressão `d: lista(aluno)`. E há outra parte do diálogo em que o usuário pode alterar essa lista, acrescentando ou removendo alunos.

Em alguns casos, as falas de transição se referem a algum signo manipulado na cena de origem. Por exemplo, quando um aluno vê uma lista dos enunciados de trabalho da sua turma, pode “selecionar” um deles para entregar o trabalho realizado. Isso é representado por uma fala de transição `u: entregar trabalho T`, em que `T` representa um trabalho selecionado ou indicado pelo usuário. A forma como o usuário poderá selecionar ou indicar o trabalho desejado se refere a decisões de design da interface propriamente dita, em uma etapa posterior de design, quando boa parte das decisões sobre o design da *interação* já tiverem sido tomadas, como será visto na próxima seção.

Em certos momentos da interação, podem ocorrer alguns diálogos que não são parte da aplicação sendo projetada, mas sim do sistema operacional. Isso acontece, por exemplo, quando um usuário manipula uma página Web utilizando uma barra de rolamento. Nesse caso, ele não está conversando com a aplicação projetada, mas sim com o navegador. Em geral, o designer tem pouco ou nenhum controle sobre essa interação. Por isso, esses diálogos não costumam ser representados no diagrama de interação. Somente nos casos em que o designer queira que seu preposto “fale” sobre esses interlocutores “externos” é que ele deve representar os diálogos e signos correspondentes.

7.4 Design da Interface

À medida que o design da interação avança, o designer passa a definir a interface propriamente dita, a parte física do sistema com a qual o usuário entrará em contato. A definição da interface inicia com a escolha dos estilos de interação do sistema, para então passar para a representação da interface, em diferentes níveis de abstração.

As próximas subseções apresentam estilos de interação comumente adotados no design de sistemas interativos, notações utilizadas para representar a interface com usuário e algumas estratégias utilizadas para elaborar a interface com usuário a partir do design da interação representado através de diagramas MoLIC.

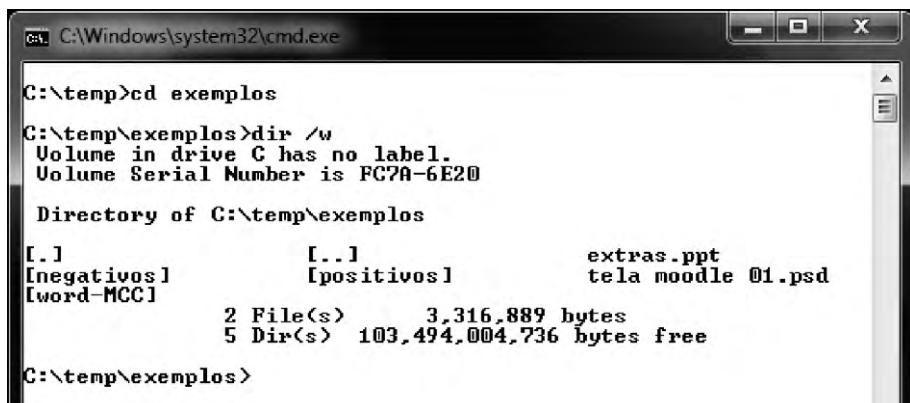
7.4.1 Estilos de Intereração

Dentre os estilos de interação mais comumente utilizados, encontramos as linguagens de comando, a linguagem natural e a interação por menus, por formulários, por manipulação direta e WIMP (*Windows, Icons, Menus, and Pointers*).

Em uma interação por **linguagem de comando**, o usuário deve digitar os comandos que realizam as ações na aplicação. Para isso, ele precisa memorizar os comandos

que precisa utilizar. Para facilitar esse aprendizado, os comandos devem ser construídos com base no vocabulário dos usuários, e a gramática da linguagem de comandos deve refletir a forma como eles conceitualizam as operações (Figura 7.17).

Segundo Shneiderman (1998), os objetivos básicos do design de uma linguagem de comando são: precisão, concisão, facilidade de escrita e leitura, completude, rapidez no aprendizado, simplicidade para reduzir erros e facilidade de retenção ao longo do tempo. Ele enumera ainda os seguintes objetivos de alto nível: correspondência entre a realidade e a notação, conveniência para realizar manipulações relevantes às tarefas dos usuários; compatibilidade com notações existentes; flexibilidade para acomodar usuários novatos e experientes; e expressividade para encorajar a criatividade.



```
C:\Windows\system32\cmd.exe
C:\temp>cd exemplos
C:\temp\exemplos>dir /w
Volume in drive C has no label.
Volume Serial Number is FC7A-6E20

Directory of C:\temp\exemplos

[.]           [...]          extras.ppt
[negativos]   [positivos]    tela moodle 01.psd
[word-MCC]      2 File(s)     3,316,889 bytes
                  5 Dir(s)    103,494,004,736 bytes free

C:\temp\exemplos>
```

Figura 7.17 Exemplo de interação através de linguagem de comando.

Em geral, uma linguagem de comando é organizada como um conjunto de comandos simples (e.g., DIR), comandos mais parâmetros (e.g., COPY *origem destino*) ou comandos seguidos de opções e argumentos (e.g., COPY /V *origem destino*). Para facilitar o aprendizado, os parâmetros devem ser ordenados de forma consistente, as opções devem ser representadas por palavras-chave em vez de símbolos arbitrários, e os comandos devem ser hierárquicos (Shneiderman, 1998).

A interação em **linguagem natural** visa permitir que o usuário se expresse como em uma conversa com uma outra pessoa, utilizando seu próprio idioma. O objetivo principal é facilitar o uso de um sistema por usuários novatos. Entretanto, existem grandes desafios para a implementação de uma interface capaz de negociar significados e resolver ambiguidades e imprecisões nas ilocuções dos usuários. Em geral, os usuários têm dificuldade para entenderem os limites do sistema, ou seja, qual é o subconjunto da linguagem natural que o sistema consegue interpretar. Além disso, esse estilo de interação se torna ineficiente para usuários experientes, quando comparado com a interação por linguagem de comando.

Na interação através de **menus**, o sistema oferece um conjunto de opções dentre as quais o usuário deve selecionar a que lhe interessa (Figura 7.18).

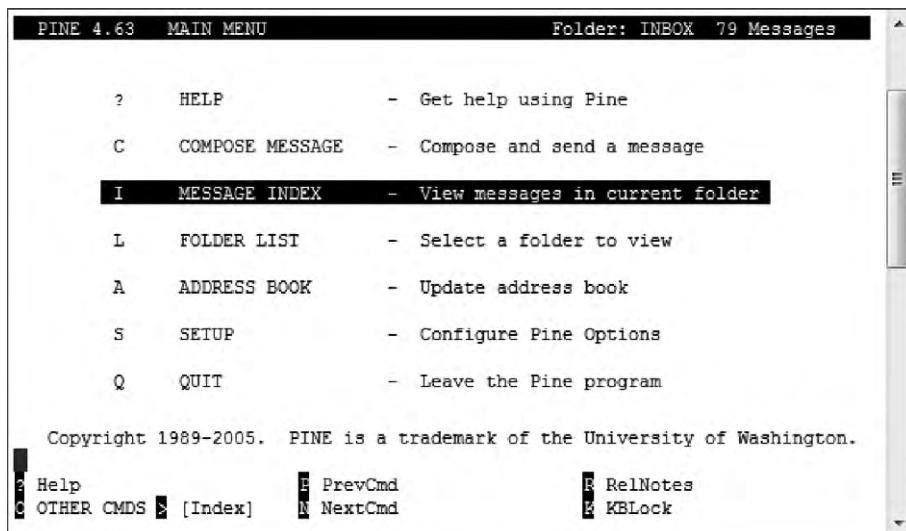


Figura 7.18 Exemplo de interação através de menu.

Além de barras de menu (*dropdown*), barras de navegação e menus contextuais (*pop-up*), Shneiderman também considera conjuntos de botões de seleção (*checkboxes*) e opção (*radio buttons*) como formas de interação por menu (Figura 7.19), embora estejam associados mais frequentemente com diálogos de fornecimento de informação e formulários na Web.

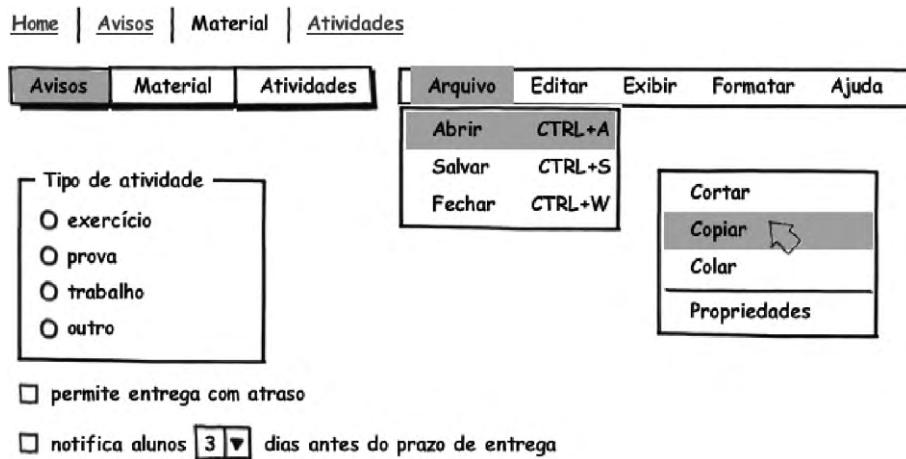


Figura 7.19 Exemplos de diferentes formas de menu.

Segundo Shneiderman (1998), o objetivo do design de menus é criar uma organização razoável, inteligível memorável e conveniente para as tarefas dos usuários. A estrutura das tarefas deve guiar a escolha da estrutura de menus: linear, hierárquica ou em rede; acíclica ou cíclica. Muitos menus são organizados em uma estrutura hierárquica. Para projetá-los, devemos considerar a profundidade e a largura da estrutura. Shneiderman recomenda optar por menus largos e rasos, em vez de estreitos e profundos.

Para qualquer tipo de menu, devemos decidir em que ordem as opções devem ser apresentadas, para refletir a estrutura das tarefas ou o modelo conceitual embutido no sistema. Alguns exemplos de ordenação são: alfabética, cronológica, numérica (ascendente ou descendente), itens mais frequentes, mais recentes ou mais importantes. Além disso, devemos criar grupos de itens logicamente semelhantes, nos certificar de que não há sobreposições entre os itens e utilizar terminologia familiar aos usuários. Sobre a redação dos itens de menu, Shneiderman sugere utilizar itens curtos, iniciados por uma palavra-chave. Como em qualquer estilo de interação, devemos utilizar gramática, *layout* e terminologia consistentes, fornecer atalhos para localizar e selecionar um item e apresentar instruções inteligíveis.

Na interação através de **formulário**, o sistema solicita dados do usuário através de campos que precisam ser preenchidos. Os formulários comumente encontrados em Web sites se encaixam nesse estilo de interação (Figura 7.20).

The screenshot shows a web form for entering shipping information. At the top, there is a placeholder text "Primeiro digite o CEP:" followed by a text input field and a note "(Ex. 99999-999)". Below it is a link "Não sabe o seu CEP? Consulte aqui". The next section is labeled "Tipo de Endereço:" with a dropdown menu. The "Endereço:" field contains a placeholder "clique aqui" and a note "n.º" followed by a numeric input field. To the right of this is a link "Dúvidas para o preenchimento do Endereço,". The "Complemento:" field has a placeholder "(Ex. ap. 1234)". Below these are fields for "Bairro:", "Cidade:", and "Estado:", each with a dropdown menu. There is also a field for "(Pedidos Internacionais)" with a dropdown menu. The "País:" field is set to "Brasil". Below these are fields for "Telefone 1:" and "Telefone 2:", both with placeholder text and notes about DDD+Telefone Fixo. The "Referência para entrega:" field has a placeholder "(Ex: travessa na altura do nº 4600 da Av. Celso Garcia.)". At the bottom right is a "Continuar >" button.

Figura 7.20 Exemplo de interação através de formulário.

Assim como em outros estilos de interação, Shneiderman (1998) recomenda criar grupos de itens relacionados e ordená-los de forma lógica; utilizar terminologia familiar aos usuários e consistente; fornecer atalhos para localizar e selecionar um item e apresentar instruções inteligíveis. Ele destaca ainda a importância de fornecer um movimento de cursor conveniente para navegação entre os campos do formulário, ou seja, que corresponda à direção natural de leitura dos usuários.

Formulários apresentam desafios interessantes para prevenção e recuperação de erros de preenchimento. Wroblewski (2008) explora diversos aspectos de formulários na Web através da comparação entre designs alternativos. Ele destaca também a importância de considerar o contexto mais amplo do formulário (público-alvo, aplicação, negócio), ou seja, informações sobre como o formulário será utilizado.

O estilo de interação por **manipulação direta** foi proposto com o objetivo de aproximar a interação da manipulação dos objetos no mundo real (Shneiderman, 1998). É necessário, para isso, que um objeto do mundo real tenha uma representação visual na interface e que cada manipulação sobre um objeto possa ser mapeada nas operações do mouse, como clique, duplo clique e clique-e-arrasto (Figura 7.21).

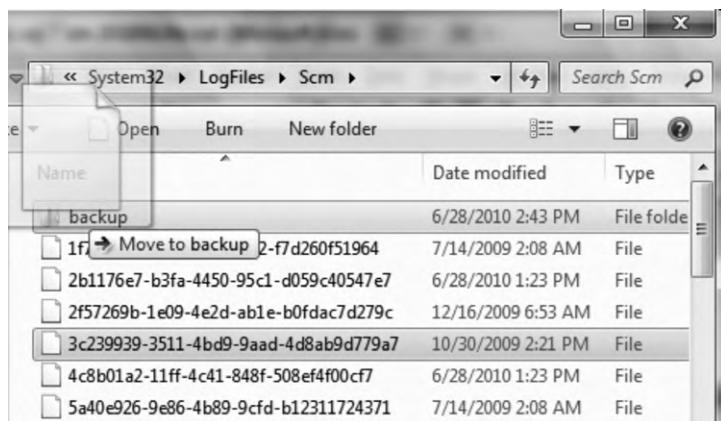


Figura 7.21 Exemplo de interação através de manipulação direta.

Em uma interação por manipulação direta, as ações devem ser rápidas, incrementais e reversíveis. Os resultados das ações devem ser imediatamente apresentados. Os benefícios desse estilo sobre a linguagem de comando são: redução das taxas de erro; aprendizado mais rápido; aumento na retenção (memorização) das operações; e engajamento e motivação para explorar o sistema.

Entretanto, uma interface por manipulação direta traz dificuldades para usuários com deficiência visual ou motora. Além disso, à medida que o número de objetos e ações aumenta, a interação se torna mais complexa e de difícil aprendizado e execu-

ção (e.g., pressionamento simultâneo de teclas do mouse e do teclado, enquanto um objeto visual é arrastado para seu destino).

Um mesmo sistema com frequência utiliza vários estilos em diferentes partes da interface, como é o caso do **WIMP** (*Windows, Icons, Menus, and Pointers* – Janelas, Ícones, Menus e Apontadores/Cursores), adotado nos ambientes baseados em janelas (Figura 7.22). Eles visam aproveitar os benefícios e contornar as limitações de cada estilo de interação individual.

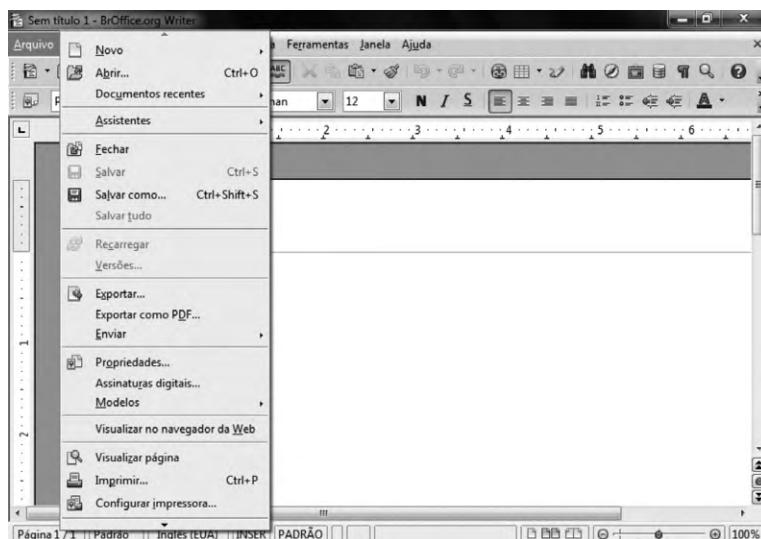


Figura 7.22 Exemplo de interação WIMP.

Recentemente vêm surgindo diversos outros estilos de interação, tais como: interação em 3D, com realidade virtual e aumentada; e interação com caneta, toque, multitoque e gestos. Consequentemente, estudos vêm sendo realizados para definir novos princípios e diretrizes específicos para esses estilos.

7.4.2 Representações da Interface com Usuário

Uma interface pode ser representada informalmente através de esboços, de forma estruturada através de modelos ou até mesmo através de protótipos funcionais. O design da interface pode ser realizado em diferentes níveis de abstração: da interface abstrata até a interface concreta. Na elaboração da interface abstrata, definimos os agrupamentos e as características dos elementos de interface, como, por exemplo, um grupo com um texto editável e com uma seleção simples dentre dez itens. Na elaboração de interface concreta, definimos o posicionamento e escolhemos os elementos de interface interativos (*widgets*). Nesse nível, por exemplo, decidimos entre

representar uma determinada entrada de dados como uma caixa de lista ou uma lista do tipo *dropdown*.

Alguns modelos utilizados para o registro da interface são representados através de linguagens de descrição de interfaces com usuário (UIDL — *user interface description languages*). Exemplos de UIDL são UIML², UsiXML³ e XAML.⁴ Outros modelos são gráficos, como os protótipos canônicos abstratos (Constantine e Lockwood, 1999). Em uma abordagem informal, o design da interface costuma ser representado em esboços (Buxton, 2007), *wireframes* e protótipos, que vão sendo refinados sucessivamente ao longo do processo.

Essas representações podem ser classificadas com relação ao seu grau de fidelidade. Uma representação é dita de baixa fidelidade quando se trata de um rascunho ou esboço da interface, sem muita preocupação com detalhes dos aspectos gráficos. Pode ser feita manualmente (Figura 7.23) ou utilizando ferramentas computacionais, como a Balsamiq Mockups⁵ (Figura 7.24). Já uma representação de alta fidelidade apresenta o desenho completo da interface, possivelmente feito em um editor de imagens, em que já estão incorporadas as decisões a respeito de tamanhos, posições, cores, fontes e outros detalhes visuais de cada elemento (Figura 7.25).



Figura 7.23 Representação de uma interface em baixa fidelidade.

2 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uiml.

3 <http://www.usixml.org/>.

4 http://msdn.microsoft.com/en-us/library/ms752059.aspx#xaml_files.

5 www.balsamiq.com. Uma licença desse software foi gentilmente cedida aos autores para a elaboração dos esboços de tela deste livro.

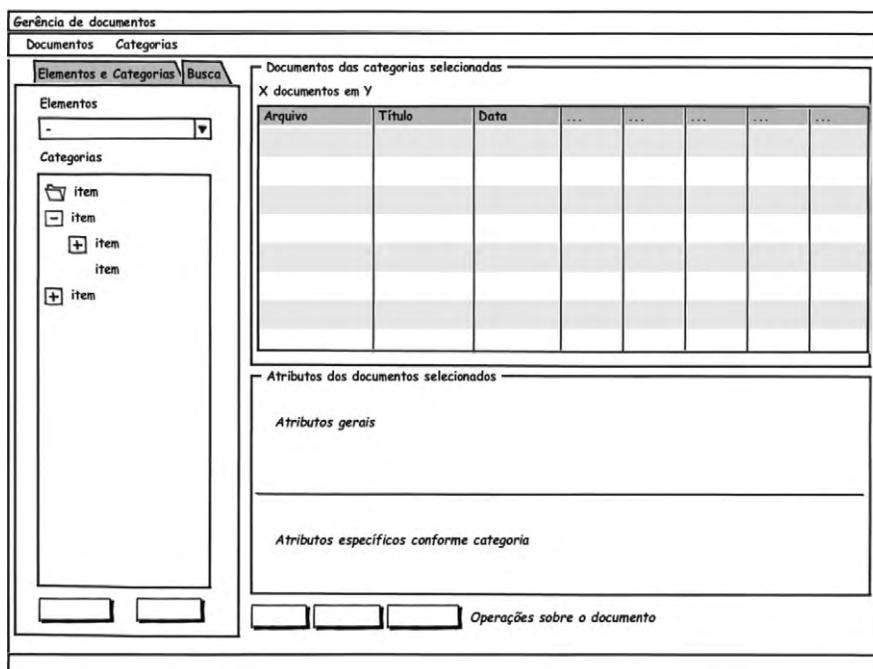


Figura 7.24 Representação de uma interface em baixa fidelidade elaborada com apoio de ferramenta computacional.

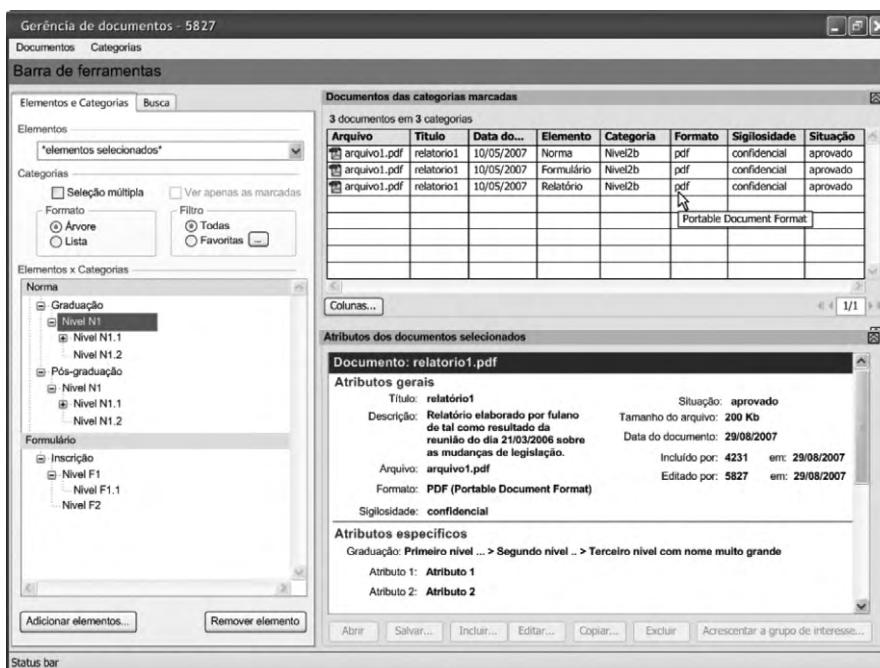


Figura 7.25 Representação de uma interface em alta fidelidade.

Com frequência, as representações de design de interface são denominadas **protótipos**. Outra classificação comumente utilizada diz respeito ao grau de funcionalidade embutida nesses protótipos: desde **wireframes** ou **maquetes**, que apresentam apenas os signos estáticos e metalingüísticos da interface, em diferentes níveis de detalhe, até **protótipos funcionais**, que incluem também os signos dinâmicos.

As atividades realizadas ao longo de todo o processo de design e os artefatos gerados fornecem insumos para o design da interface. A identificação dos principais objetivos e cenários de uso frequente permite decidir quais elementos de interface devem ser destacados (e.g., atalhos na barra de ferramentas, itens de menu pop-up e posição de destaque no *layout* da tela). Permite decidir também quais elementos podem ser deslocados para posições secundárias (e.g., link de volta no canto inferior esquerdo) ou até mesmo para telas secundárias (e.g., tela de gerenciamento de tipos de trabalho).

A atividade de análise, as pesquisas com usuários e outras fontes de informação auxiliam na definição do vocabulário utilizado na interface. Caso seja usada uma metáfora para comunicar o modelo conceitual, ela também contribui para a construção desse vocabulário. É importante manter um vocabulário consistente e familiar ao usuário a fim de tornar a comunicação com o usuário eficiente e clara.

Um desafio de design de interface enfrentado com frequência está relacionado à experiência e ao conhecimento do usuário. É comum projetar assistentes para usuários iniciantes e telas de configuração com múltiplas opções para especialistas, como se fossem duas interfaces distintas e muitas vezes isoladas. No entanto, segundo Cooper (1999), a maioria dos usuários deixa de ser iniciante rapidamente, mas permanece como usuário intermediário indefinidamente. Os designers devem avaliar a importância de fazer na interface uma ponte entre as diferentes estratégias de uso do sistema, para apoiar a curva de aprendizado do usuário até ele se tornar um especialista.

7.4.3 Da Interação como uma Conversa para o Design da Interface

Esta seção apresenta algumas decisões comumente tomadas ao projetar a interface com usuário a partir da modelagem da interação como uma conversa representada utilizando a MoLIC (Silva *et al.*, 2005).

Os acessos ubíquos são pontos de início de conversas dirigidas por objetivos e, em geral, devem estar disponíveis em qualquer momento da interação, desde que respeitadas suas precondições. Sendo assim, em geral, os acessos ubíquos costumam ser mapeados na interface em barras de navegação ou itens de menu (Figura 7.26).

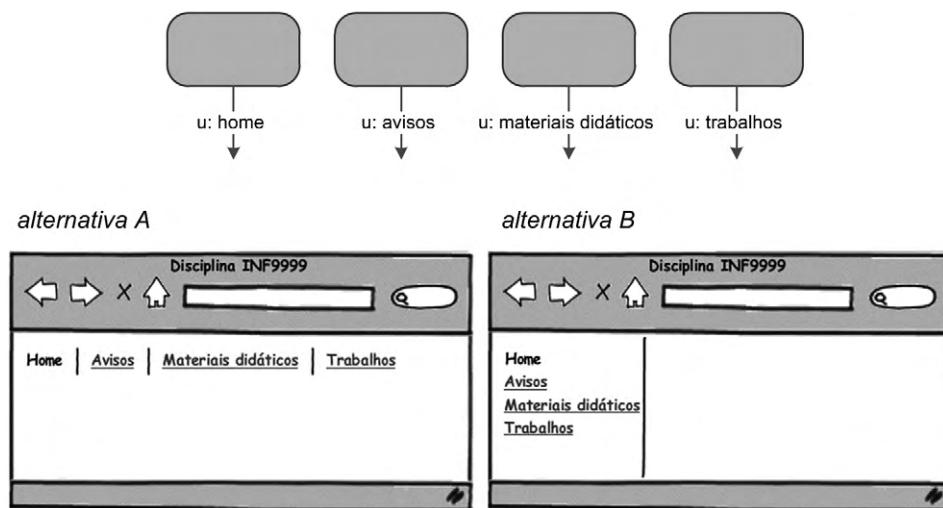


Figura 7.26 Acessos ubíquos mapeados para uma barra de navegação, considerando duas alternativas: (a) horizontal e (b) vertical.

A interface é composta de diferentes **unidades de apresentação**. Em uma interface gráfica, uma unidade de apresentação é uma tela ou janela. Já em interfaces Web, é uma página. Embora não haja necessariamente um mapeamento direto entre cena e unidade de apresentação, essa estratégia é bastante comum (Figura 7.27).

Data	Tipo	Título	Arquivo	Tamanho
10/03/2010	Slides	Introdução	00intro.pdf	247 KB
12/03/2010	Notas de aula	Apostila - Parte I	apostila01.pdf	1658 KB
17/03/2010	Slides	Vetores e Matrizes	01vetor.pdf	765 KB
24/03/2010	Slides	Listas	02lista.pdf	894 KB
31/03/2010	Slides	Pilhas	03pilha.pdf	425 KB
07/04/2010	Slides	Filas	04fila.pdf	364 KB

Figura 7.27 Mapeamento de uma cena para uma unidade de apresentação.

A Figura 7.28 apresenta uma alternativa de design de interface que decompõe a mesma cena em duas unidades de apresentação. Esse tipo de situação acontece com frequência no caso de diálogos que manipulam arquivos, como localizar um arquivo

para abri-lo ou localizar um diretório e definir um nome para salvar um arquivo. Também é comum em alguns dispositivos móveis com telas de tamanho reduzido.

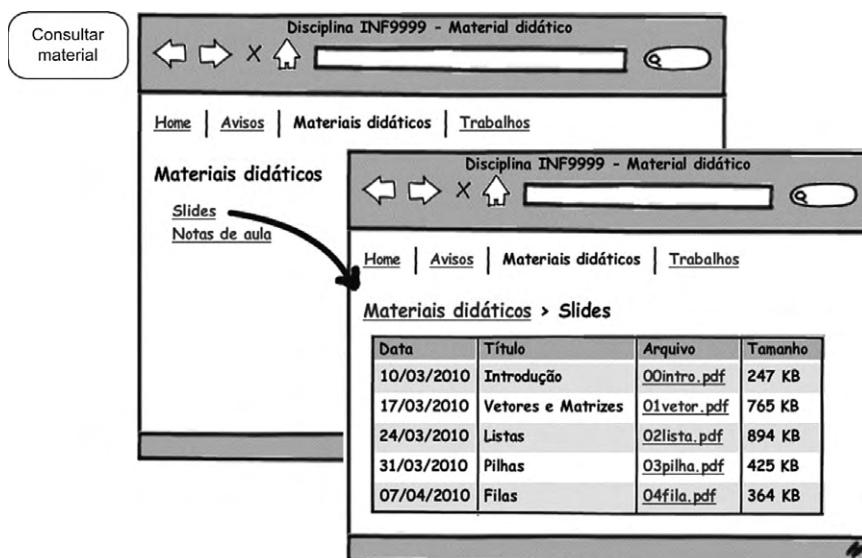


Figura 7.28 Mapeamento de uma cena para duas unidades de apresentação.

Grupos de diálogos costumam ser refletidos na interface com usuário em quadros ou contêineres de outros elementos de interface, marcados visivelmente ou definidos apenas por um *grid*. Cada signo, por sua vez, costuma ser representado como um elemento de interface (*widget*). E, caso haja alguma prevenção passiva prevista associada a um signo, ela geralmente é concretizada em uma instrução ou dica, textual ou pictórica, apresentada próxima ao elemento correspondente, como o asterisco para indicar campos obrigatórios.

Uma fala de transição de usuário geralmente se destina a uma mudança de tópico na conversa, de prosseguimento da conversa para o aprofundamento do tópico em direção ao seu objetivo, ou de conclusão da conversa para verificar o alcance do objetivo. Sendo assim, uma fala de transição do usuário geralmente é mapeada em um link, botão ou item de menu. Caso haja uma precondição associada à fala, pode haver uma mudança no estado (ativo ou inativo) ou na visibilidade (visível ou oculto) do elemento correspondente.

A Figura 7.29 ilustra um fragmento de diagrama de interação e diversos mapeamentos:

- cena Consultar material mapeada para unidade de apresentação Materiais didáticos (indicação nº 1 na figura)
- diálogo ver materiais mapeado para a tabela de materiais didáticos (nº 2)

- fala de usuário u: cadastrar novo material mapeada para link Cadastrar novo material didático (nº 3)
- fala de usuário u: editar material X mapeada para os links na tabela (nº 4)
- cena Editar material mapeada para duas unidades de apresentação semelhantes, conforme a fala de transição de usuário que leva até ela:
 - Cadastrando novo material didático, destino da fala u: cadastrar novo material didático (nº 5)
 - Editando material didático, destino da fala u: editar material X (nº 6)

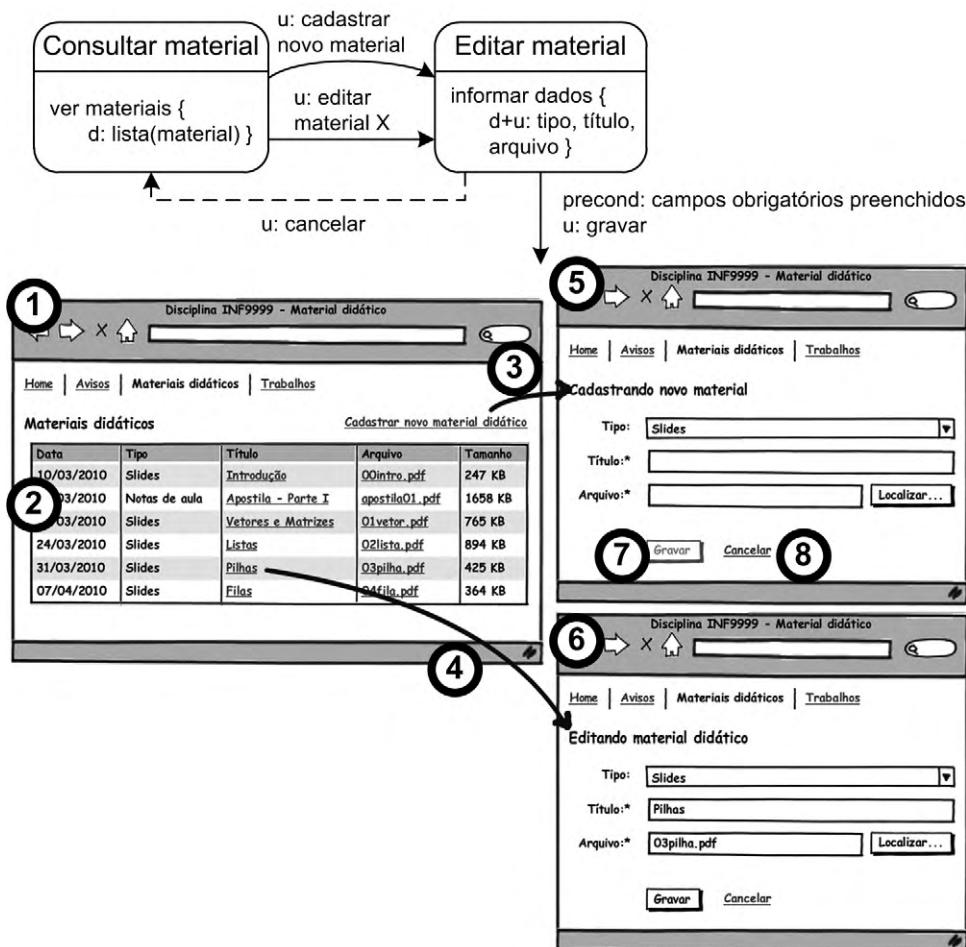


Figura 7.29 Diálogos e signos mapeados em elementos de interface.

Podemos observar ainda que o botão Gravar aparece desativado na unidade de apresentação Cadastrando novo material didático (nº 7), pois os campos obrigatórios ainda não foram preenchidos. Além disso, a fala de transição do usuário para recuperação

de ruptura u: cancelar foi mapeada para o link Cancelar (nº 8), que leva o usuário de volta para a página Materiais didáticos. No exemplo, as falas u: cadastrar novo material e u: editar material X levam o usuário para uma nova página. Mas poderiam ter aberto uma janela ou painel sobre a tela Materiais didáticos, por exemplo. Nesse caso, a fala u: cancelar teria como comportamento fechar essa janela.

As falas do preposto do designer se destinam a fornecer ao usuário *feedback* sobre o andamento ou a conclusão de um processamento do sistema, em resposta a uma solicitação do usuário. Elas comumente são concretizadas na interface como mensagens de erro (em particular no caso de falas de recuperação de uma ruptura comunicativa) e de *status*. Essas mensagens podem ser apresentadas de pelo menos duas formas distintas: embutidas na unidade de apresentação correspondente à cena de destino (nº 1) ou em unidades de apresentação independentes (nº 2), como pode ser visto na Figura 7.30, para as falas d: material gravado e d: problema na gravação, respectivamente.

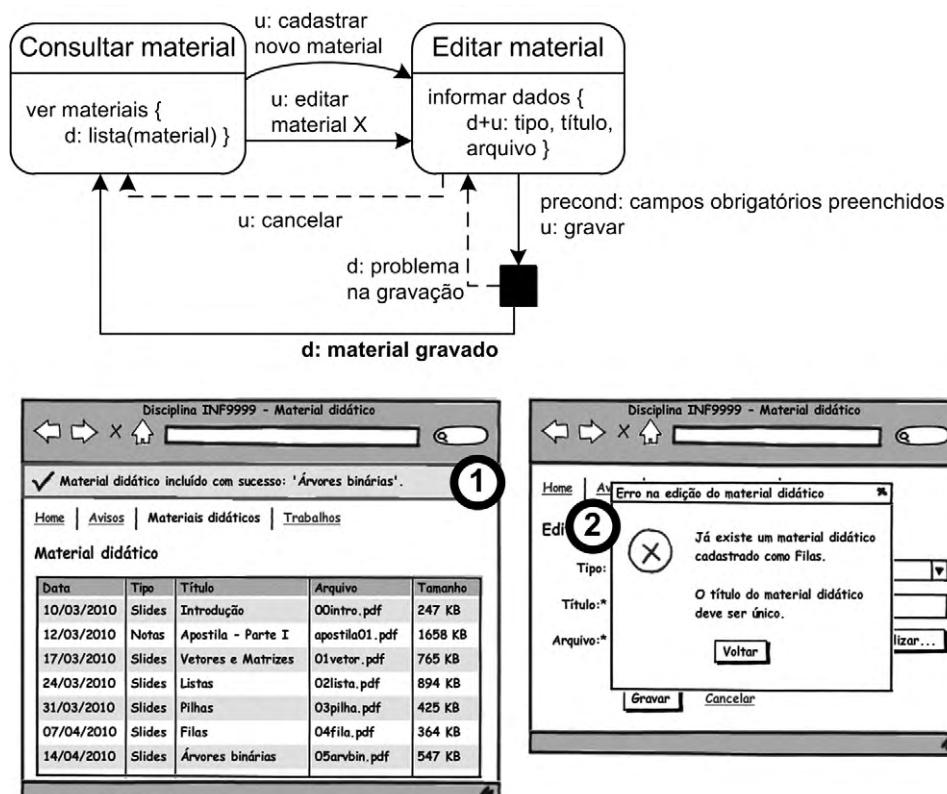


Figura 7.30 Falas de transição do preposto do designer mapeadas para elementos de interface.

7.4.4 Esquema Conceitual de Signos: Expressão (Parte II)

Em paralelo à elaboração da interface, podemos definir as expressões de cada signo, concluindo assim a definição do esquema de signos. A expressão de um signo define qual elemento de interface (*widget*) deverá ser utilizado para apresentar ao usuário ou permitir que ele manipule o conteúdo do signo.

Cada signo pode apresentar uma expressão diferente, conforme o seu interlocutor e o contexto de interação em que o signo ocorre (Tabela 7.6). Quando o interlocutor é o preposto do designer (emissor = d), o signo é apresentado ao usuário através de elementos de interface para saída de dados (*output*); ao passo que quando ambos, usuário e preposto, *falam* sobre o signo (emissor = d+u), suas expressões são elementos de interface para entrada de dados (*input*).

Destacamos na tabela a possibilidade de haver mais de uma expressão para um mesmo signo. Observamos que o signo data de entrega possui duas formas de expressão ao ser emitido pelo preposto do designer: por default, ele é apresentado pela data formatada como dd/mm/aaaa; mas na cena de avisos, ele é apresentado por uma representação em calendário (cena Consultar avisos: dd/mm/aaaa + calendário). A Figura 7.31 apresenta duas páginas que ilustram esses diferentes contextos de interação.

Tabela 7.6 Definição das expressões dos signos

Enunciado de trabalho (E) – enunciado de trabalho de disciplina de graduação			
signo	emissor	tipo de expressão	expressão default e em contexto
+ título	d+u	texto editável simples	caixa de texto
	d	texto simples	rótulo
descrição	d+u	texto formatado editável	caixa de texto com ferramentas de formatação
	d	texto simples (aprox. 150 palavras)	rótulo com múltiplas linhas
data de entrega	d+u	calendário	controle de calendário
	d	data	default: rótulo (dd/mm/aaaa); cena Consultar avisos: dd/mm/aaaa + calendário
formato de entrega	d+u	lista de seleção simples	default: combo
	d+u	texto editável simples	cena Cadastrar formato de entrega: caixa de texto
	d	texto simples	rótulo
número máximo de alunos	d+u	texto editável simples para números inteiros	caixa de texto com botões de incremento e decremento
	d	texto simples	rótulo

peso	d+u	texto editável simples	caixa de texto
	d	texto simples	rótulo
lembrete do prazo de entrega	d+u	grupo de opções	radio (sim,não)
	d	texto simples	rótulo(sim/não)

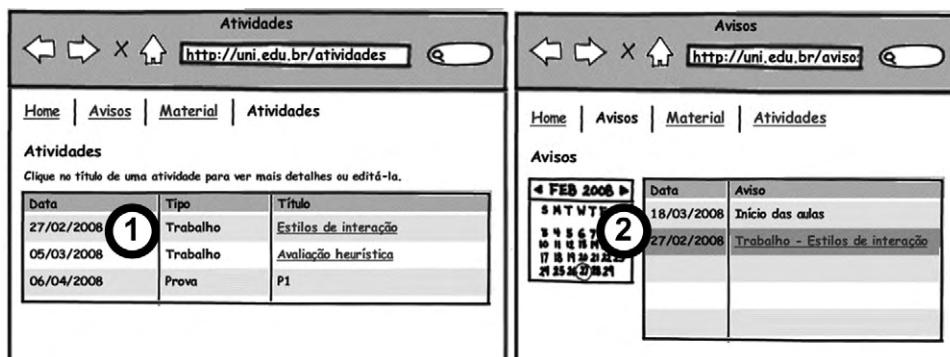


Figura 7.31 Esboços de tela de visualização de lista de projetos e dos detalhes de um projeto final.

A tabela de expressões dos signos é utilizada principalmente para assegurar a consistência de apresentação e manipulação dos signos. Pode haver mais do que uma expressão associada a um signo, mas essa decisão deve ser tomada de forma cuidadosa e consciente pelos designers do produto.

Além disso, essa especificação auxilia a implementação da interface, fornecendo ao programador orientações detalhadas sobre ela. Sem isso, os programadores podem ter de tomar algumas decisões relacionadas ao design de interface, sem que tenham participadoativamente do processo de análise e design e, consequentemente, sem estar adequadamente preparados para isso.

7.5 Projeto do Sistema de Ajuda⁶

Segundo de Souza (2005a), o poder explicativo do discurso do preposto do designer e, portanto, sua capacidade em negociar significados, são limitados por propriedades formais de linguagens computacionais. Uma das formas de maximizar a comunicabilidade do discurso do preposto do designer é enriquecer a forma e o conteúdo do sistema de ajuda.

O sistema de ajuda é na engenharia semiótica 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 façamos o projeto do sistema

⁶ Esta seção foi adaptada de Prates e Barbosa (2007), com permissão das autoras.

de ajuda para que o usuário possa entender melhor a solução do designer e fazer melhor uso do sistema, se recuperando facilmente de rupturas que porventura venham a ocorrer.

Com relação ao *conteúdo* do sistema de ajuda, podemos nos basear em compilações das dúvidas frequentes dos usuários ao utilizarem sistemas interativos, classificadas de acordo com o tipo de dúvida (Baecker *et al.*, 1995; Sellen e Nicol, 1990) (Tabela 7.7).

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

Tabela 7.7 Tipos de dúvidas frequentes dos usuários

tipo de dúvida	exemplo de pergunta
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?

A pergunta *informativa* **O que posso fazer com este programa?** pode ser respondida com os objetivos dos usuários que foram identificados nas personas, elaborados nos cenários e organizados no mapa de objetivos.

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 à interface. O nível conceitual pode ser

descrito no início do processo de desenvolvimento, mas 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 aos modelos de tarefas e de interação. 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 em que cena o usuário se encontra e quais precondições 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 conversas em direção a um ou mais objetivos, buscando identificar o objetivo atual do usuário e o que falta para ele alcançá-lo.

As perguntas *interpretativas, navegacionais e históricas* estão relacionadas principalmente ao modelo de interação, que representa os caminhos que o usuário pode seguir e que indicações o sistema fornece sobre em que ponto da interação ele se encontra a cada instante. A pergunta **O que está acontecendo agora?** está relacionada a falas do preposto e mecanismos 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. Tais perguntas dizem respeito ao sistema como um todo, e não a um objetivo específico. 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.

Com relação à *forma* como o sistema de ajuda é acessado, Silveira 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?” e “Cadê?”; veja Seção 10.2.2), adicionado de outras específicas para o entendimento da comunicação designer-usuário (e.g., “Para que serve isto?” e “Por que tenho que fazer isto?”).

7.6 Desafios de Design de Sistemas Adaptáveis e Adaptativos

Diaper (2003) afirma que, como descrições do mundo em IHC incluem pessoas em ambientes sociais e organizacionais complexos, a previsão acurada de qualquer futuro pós-design é praticamente impossível, ao menos porque as pessoas, com frequência, (1) adaptam seu comportamento às mudanças projetadas; (2) alteram outros aspectos do mundo para acomodar o que foi projetado; e (3) alteram e usam o que foi projetado de formas que não foram previstas pelos designers. Suchman (1987) discute a natureza situada da atividade do usuário, que não pode ser completamente prevista ou planejada *a priori*. E a engenharia semiótica levanta a necessidade de os designers buscarem ampliar a competência comunicativa do preposto do designer a fim de que os usuários façam uso criativo do sistema, permitindo que eles customizem ou estendam o sistema, ajustando-o ou reprojetando-o (de Souza 2005a; de Souza e Barbosa, 2006).

Para lidar com essa situação, diversas pesquisas vêm sendo realizadas no sentido de permitir aos usuários configurarem, adaptarem ou estenderem seus sistemas, denominadas **adaptáveis**, **customizáveis** ou **extensíveis** (Oppermann, 1994). Além dos desafios de engenharia de software envolvidos na oferta dessas possibilidades aos usuários, há desafios importantes no projeto de IHC, como as decisões sobre: quando e como comunicar aos usuários o que pode ser modificado ou estendido; como permitir que eles efetivamente realizem a modificação ou extensão; como comunicar-lhes as consequências da modificação ou extensão e ajudar-lhes a testá-la; como restringir o que eles podem modificar na interface para preservar a identidade do sistema (de Souza e Barbosa, 2006).

Além de permitir que os próprios usuários modifiquem o sistema, é possível também projetar sistemas que se adaptam automaticamente ao usuário, denominados **adaptativos** (Schneider-Hufschmidt *et al.*, 1993; Cypher, 1993; Oppermann, 1994; Lieberman, 2001). Nesse caso, é necessário tomar cuidado não apenas para comunicar adequadamente ao usuário as mudanças realizadas, mas também permitir que ele desfaça cada adaptação ou até mesmo impeça que o sistema faça novas adaptações daquele tipo no futuro. Se possível, esses sistemas devem aprender com o *feedback* do usuário, para melhor ajustar seus mecanismos de autoadaptação.

Sistemas adaptáveis e adaptativos trazem grandes desafios para todas as atividades de IHC. Em análise, os desafios envolvem identificar as necessidades e oportunidades de adaptação, manual ou automática, em diferentes contextos. Em design, envolvem conceber e modelar soluções flexíveis e “inteligentes” de modo que os usuários possam melhor se apropriar do sistema e estender a sua competência comunicativa, ampliando a gama de interpretações que o sistema pode gerar e, consequente-

mente, os comportamentos que pode apresentar. Finalmente, os desafios de avaliação se devem à inadequação, em geral, de avaliações pontuais em laboratório para julgar sistemas adaptáveis ou adaptativos. Para esse tipo de sistemas, estudos longitudinais em contextos de uso reais muitas vezes se fazem necessários.

Atividades

Para as atividades abaixo, considere uma agenda pessoal integrada com lista de afa-
zeres e correio eletrônico, que deva ser implementada para uso em dois dispositivos:
um computador desktop e um *smartphone*. Realize uma análise como indicado nos
Capítulos 5 e 6.

1. *Cenário de interação.* Para um mesmo cenário de análise, elabore cenários de interação alternativos. Que perguntas surgiram durante a elaboração da solução e que não haviam sido levantadas no momento da análise?
2. *Mapa de objetivos.* A partir da análise da atividade do usuário com o sistema escolhido, construa o mapa de objetivos correspondente. Há objetivos que precisam ser apoiados em apenas uma das plataformas? Quais, e por quê? Busque sistemas existentes que visam apoiar esses objetivos, e faça a sua “engenharia reversa”, identificando os objetivos que eles de fato apoiam. Há diferenças entre o resultado da análise do que os usuários precisam ou querem e aquilo que está projetado nesses sistemas? Quais objetivos latentes não são apoiados? E quais objetivos os sistemas apoiam que não surgiram na análise de necessidades e desejos dos usuários? Discuta as diferenças.
3. *Esquema conceitual de signos.* Elabore o esquema conceitual dos signos do sistema sendo projetado, incluindo as características de conteúdo, expressão, prevenção e recuperação de rupturas. Avalie a necessidade de signos voltados à customização do sistema e defina-os.
4. *Modelagem de tarefas.* Selecione alguns objetivos e faça a modelagem de tarefas correspondentes. Para um mesmo objetivo, elabore modelos de tarefas alternativos, pensando em diferentes características dos usuários e nos diferentes dispositivos considerados. Discuta as diferenças.
5. *Modelagem da interação.* Para os objetivos selecionados, represente a interação utilizando um diagrama MoLIC, um para cada dispositivo considerado. Elabore alternativas de solução distintas e compare suas representações. Contraste ainda um diagrama de interação elaborado seguindo uma abordagem *top-down*, e outro elaborado a partir da engenharia reversa de sistemas existentes. Com base na análise feita previamente, identifique os critérios de seleção dentre os modelos de interação elaborados.

6. *Elaboração de esboços de interface a partir de um modelo de interação.* Selecione uma das soluções modeladas e elabore esboços de interface alternativos para concretizar a solução de IHC: um conjunto de esboços a partir de um modelo de tarefas e outro a partir de um modelo de interação. Discuta as diferenças nas decisões tomadas em cada caso e nas soluções elaboradas.
7. *Sistema de ajuda.* Escolha um dos objetivos apoiados pelos esboços elaborados na atividade anterior e elabore o conteúdo de ajuda correspondente. Discuta as formas de acesso a esse conteúdo que podem ser fornecidas a partir da interface, indicando nos esboços esses pontos de acesso.
8. *Oportunidades de adaptação.* Considerando diferentes perfis de usuários, atividades e contextos de uso do sistema, identifique oportunidades de adaptação manual ou automática e reveja as soluções modeladas e esboçadas para possibilitar essas formas de adaptação. Discuta meios de comunicar aos usuários de que maneira eles podem adaptar o sistema ou ajustar as adaptações realizadas automaticamente pelo sistema.

8

Princípios e Diretrizes para o Design de IHC

Objetivos do Capítulo

- Apresentar princípios e diretrizes para o design de IHC, exemplificando seu uso.
 - Discutir os benefícios de se utilizar padrões de design de IHC e apresentar alguns modelos de documentação de padrões.
 - Descrever brevemente o uso de guias de estilo e apresentar uma estrutura para esse documento.
-

Este capítulo apresenta alguns princípios, diretrizes (Norman, 1988; Tognazzini, online; Nielsen, 1993; Shneiderman, 1998; Cooper, 1999) e padrões de design (Tidwell, 2005) comumente utilizados na construção das interfaces de usuário.

8.1 Introdução

A literatura de IHC está repleta de conjuntos de princípios, diretrizes (*guidelines*) e heurísticas. Princípios costumam representar objetivos gerais e de alto nível; diretrizes, regras gerais comumente observadas na prática; e padrões, soluções específicas a certos contextos bem delimitados, envolvendo certos usuários desempenhando determinadas tarefas (Mayhew, 1999). Entretanto, essa classificação nem sempre é utilizada precisamente, e alguns proponentes de diretrizes as intitulam de princípios. Os conjuntos mais conhecidos de princípios e diretrizes são os de Norman (1988), de Tognazzini (2003),¹ de Nielsen (1993)² e as regras de ouro de Shneiderman (1998).

Todos os pesquisadores e profissionais de IHC ressaltam que o uso de princípios e diretrizes jamais substitui as demais atividades de análise, design (conceitual e concreto) e avaliação. Embora princípios e diretrizes possam ser utilizados como auxílio ao design, elas não substituem um processo cuidadoso que inclui a busca pelo entendimento do problema, a elaboração de soluções candidatas e a avaliação dessas soluções alternativas.

Alguns conjuntos de diretrizes são desenvolvidos especificamente para certos ambientes de trabalho, como o Windows®, o MacOs® e o Gnome®, para certos dispositivos, como dispositivos móveis e televisão digital interativa, e certos domínios, como educação, governo eletrônico etc. Alguns conjuntos de diretrizes, como os voltados para certos ambientes de trabalho, visam motivar designers a seguir uma certa padronização para assegurar aparência e comportamento (*look and feel*) semelhantes com o que os desenvolvedores ou alguma outra organização propuseram. Entretanto, diretrizes frequentemente são recomendações genéricas e descontextualizadas da teoria ou base empírica que lhes deram origem. Além disso, muitas vezes duas ou mais diretrizes são conflitantes. A aplicação adequada de boa parte dos princípios e diretrizes depende, em alguma medida, do conhecimento do designer acerca do domínio do problema, dos usuários e das suas atividades nesse domínio. Sendo assim, cabe ao designer considerar cuidadosamente se e quais diretrizes são adequadas à sua situação de design, e como elas devem se manifestar na solução de IHC.

1 <http://www.asktog.com/basics/firstPrinciples.html>.

2 Nielsen (1993) denomina seu conjunto de diretrizes alternadamente de princípios, diretrizes e heurísticas.

Diretrizes são comumente utilizadas em listas de verificação (*checklists*), em que inspetores examinam uma interface para avaliar se ela está em conformidade com o conjunto selecionado de diretrizes (veja a Seção 9.2). No Brasil, temos como exemplo a ErgoList,³ lista de verificação desenvolvida com base em princípios de ergonomia.

8.2 Princípios e Diretrizes Gerais

Norman (1988) destaca a necessidade de projetarmos o sistema utilizando um modelo conceitual que o usuário possa apreender rapidamente e sem dificuldade. O modelo conceitual deve auxiliar a interpretar o relacionamento entre as ações e informações apresentadas pelo sistema e o conhecimento no mundo.

Segundo Norman (1988), o design deve facilitar: determinar quais ações são possíveis a cada momento, fazendo uso de restrições (*constraints*); tornar as coisas visíveis, incluindo o modelo conceitual do sistema, as ações alternativas e os resultados das ações; avaliar o estado corrente do sistema e seguir mapeamentos naturais entre as intenções e as ações requeridas, entre as ações e o efeito resultante, e entre a informação que está visível e a interpretação do estado do sistema.

Os princípios e as diretrizes comumente utilizados em IHC giram em torno dos seguintes tópicos: correspondência com as expectativas dos usuários; simplicidade nas estruturas das tarefas; equilíbrio entre controle e liberdade do usuário; consistência e padronização; promoção da eficiência do usuário; antecipação das necessidades do usuário; visibilidade e reconhecimento; conteúdo relevante e expressão adequada; e projeto para erros. As próximas subseções apresentam algumas diretrizes e ilustram como podem ser utilizadas no design da interação e da interface.

8.2.1 Correspondência com as Expectativas dos Usuários

Segundo Norman (1988), devemos *explorar os mapeamentos naturais*, seja entre as variáveis mentais e as físicas, seja entre as tarefas e os controles utilizados para manipular essas variáveis no mundo real e no sistema projetado (veja Seção 3.4). Devemos nos certificar de que o usuário consegue determinar os relacionamentos entre: intenções e ações possíveis; entre ações e seus efeitos no sistema; entre o estado real do sistema e o que é percebido pela visão, audição ou tato; entre o estado percebido do sistema e as necessidades, intenções e expectativas do usuário.

Por exemplo, ao projetar um sistema de comércio eletrônico, devemos examinar como as pessoas fazem suas compras em lojas físicas: uma pessoa entra em uma loja, escolhe um ou mais produtos (com ou sem ajuda de um vendedor), e somente precisa se identificar no momento de finalizar a compra e pagar pela mercadoria escolhida.

³ <http://www.labutil.inf.ufsc.br/ergolist/>.

Da mesma maneira, um sistema de comércio eletrônico deve permitir ao usuário buscar e selecionar os produtos anonimamente, e exigir sua identificação apenas no momento de finalizá-la (Figura 8.1a). Entretanto, ainda hoje, alguns sistemas na Web exigem que o usuário se identifique antes mesmo de encontrar um produto do seu interesse (Figura 8.1b).

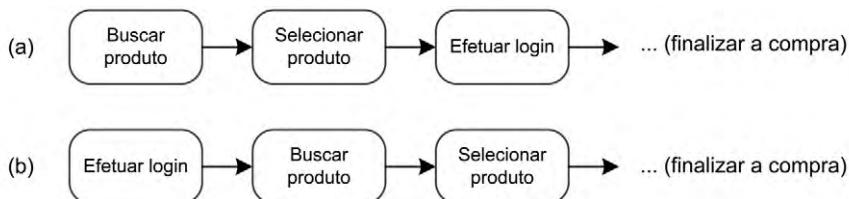


Figura 8.1 Sequências alternativas de cenas em sites de comércio eletrônico, solicitando a identificação do usuário em diferentes momentos da interação.

Shneiderman (1998) recomenda *estruturar o diálogo de forma a seguir uma linha de raciocínio e fornecer um fechamento*. Segundo ele, as sequências de ações devem ser organizadas em grupos com início, meio e fim. Segundo Nielsen (1993), o projetista deve seguir as convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica. Para isso, é importante entendermos as sequências de ações que são familiares aos usuários e, caso a solução se desvie do que lhes é familiar, deve ao menos refletir uma organização lógica que lhes seja plausível. Além disso, ele ressalta a importância de fornecer um *feedback* informativo na conclusão de um grupo de ações (veja Seção 8.2.7), para proporcionar aos usuários a satisfação de terem concluído uma tarefa, um sentimento de alívio, um sinal de que podem deixar de lado planos de contingência que tiverem formulado e uma indicação de que já podem se preparar para o novo grupo de ações.

Além dos aspectos estruturais, o designer deve projetar a interface utilizando o idioma do usuário, com palavras, expressões e conceitos que lhe são familiares, em vez de utilizar termos orientados ao sistema ou a desenvolvedores.

Tognazzini (2003) recomenda o *uso de metáforas* de forma cuidadosa, para permitir que os usuários identifiquem rapidamente sutilezas do modelo conceitual subjacente ao sistema. Boas metáforas são como histórias, criando imagens na mente. Elas devem evocar algo familiar, mas geralmente trazem também algo novo. Por exemplo, uma pasta num gerenciador de arquivos não tem a mesma limitação de número de itens de conteúdo que uma pasta física.

8.2.2 Simplicidade nas Estruturas das Tarefas

Norman (1988) recomenda *simplificar a estrutura das tarefas*, reduzindo a quantidade de planejamento e resolução de problemas que elas requerem. Tarefas desnecessariamente complexas podem ser reestruturadas, em geral utilizando inovações tecnológicas. Para simplificar a estrutura das tarefas, os designers podem seguir quatro abordagens tecnológicas: a) manter a tarefa a mesma, mas fornecendo diversas formas de apoio para que os usuários consigam aprender e realizar a tarefa; b) usar tecnologia para tornar visível o que seria invisível, melhorando o *feedback* e a capacidade de o usuário se manter no controle da tarefa; c) automatizar a tarefa ou parte dela, mantendo-a igual; e d) modificar a natureza da tarefa. Norman alerta para o perigo de a automação tirar controle demais do usuário, escravizando-o ou tornando-o tão confiante e dependente da tecnologia a ponto de reduzir ou até mesmo eliminar sua capacidade de trabalhar sem a automação.

8.2.3 Equilíbrio entre Controle e Liberdade do Usuário

Norman (1988), Nielsen (1993), Tognazzini (2003), Shneiderman (1998) e Cooper (1999) destacam a importância de manter o usuário no controle. Tognazzini (2003) lembra que o computador, a interface e o ambiente de trabalho “pertencem” ao usuário. Ele afirma que, quando deixamos o usuário “no comando”, ele aprende rapidamente e ganha um sentimento de maestria. Entretanto, ele ressalta a necessidade de buscar um equilíbrio, pois quando não há limites ou restrições os usuários podem se sentir perdidos ou angustiados com o excesso de opções. Devemos tentar reduzir o número de opções ou decisões que o usuário precisa tomar a cada instante. Norman (1988) recomenda *explorar o poder das restrições, tanto naturais como artificiais*, e projetar restrições para que o usuário sinta como se houvesse apenas uma coisa possível a fazer: a coisa “certa”, é claro.

Segundo Tognazzini (2003), os usuários não devem ficar presos num caminho de interação único para realizar uma atividade. O caminho mais rápido ou preferencial pode ser o de “menor resistência”, mas usuários que queiram explorar diferentes alternativas e cenários devem conseguir fazê-lo.

Sempre deve ser fornecida aos usuários uma “saída” clara e rápida, mas deve ser mais fácil se manter “no caminho” do que sair dele inadvertidamente (Figura 8.2). A decisão entre oferecer mais ou menos liberdade ao usuário varia com o seu perfil. Usuários mais inexperientes podem precisar de mais assistência e menos alternativas, ao passo que usuários experientes de interfaces de uso frequente devem poder comandá-la como melhor lhes convier.

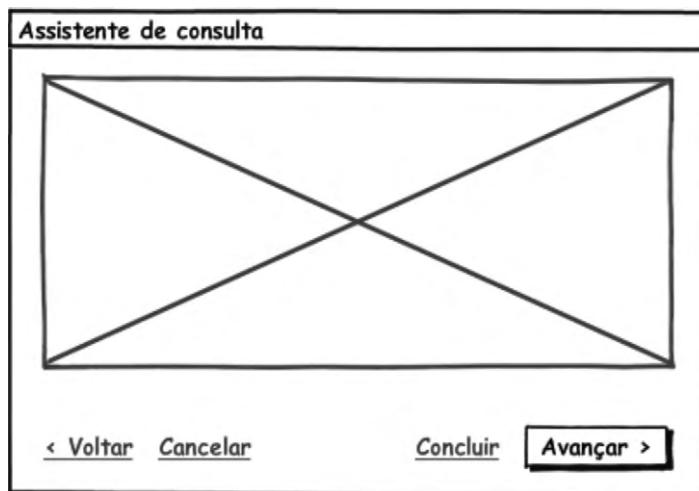


Figura 8.2 Tela de assistente com indicação clara do botão primário.

Cooper (1999) afirma que o software deve ser *maleável* e ter “jogo de cintura”. Para ele, isso significa aceitar permanecer em estados intermediários e realizar algumas ações fora de ordem ou antes que seus pré-requisitos tenham sido satisfeitos. Em outras palavras, não deve forçar os usuários a trabalharem de um único modo. Entretanto, deve sempre guardar um registro de tudo o que tiver sido feito para que o responsável por cada ação possa ser identificado.

Shneiderman (1998) recomenda *permitir que o usuário tenha controle local da interação*, ou seja, que o usuário inicie as ações, em vez de apenas reagir a ações do sistema. Ele se refere principalmente a usuários experientes, que costumam querer sentir que controlam o sistema e o sistema responde às suas ações, e não o contrário. Além dele, Nielsen (1993) e Tognazzini (2003) também destacam a importância de *permitir que o usuário cancele, desfaça e refaça suas ações*. Os usuários frequentemente escolhem funções do sistema por engano e precisam de uma “saída de emergência” claramente marcada para sair do estado indesejado sem ter de percorrer um diálogo extenso (Figura 8.3).

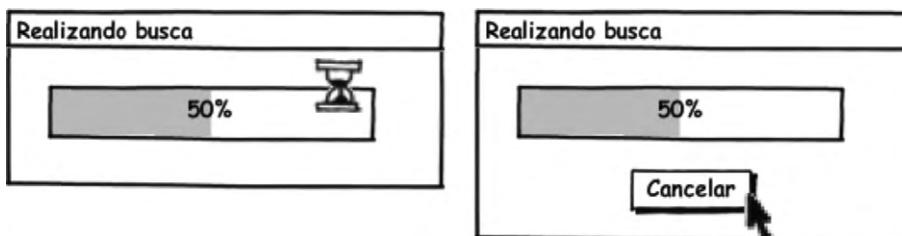


Figura 8.3 Telas de progresso sem e com opção de cancelar.

Permitir que o usuário desfaça suas ações reduz a sua ansiedade e o medo de errar, pois ele sabe que os erros podem ser desfeitos. Essa possibilidade também é importante para o *aprendizado por exploração*. Os usuários frequentemente exploram partes da interface para descobrir o que aconteceria se eles fizessem isso ou aquilo, e assim aprendem mais sobre o sistema e sobre seu próprio trabalho. Para isso, é importante que ações potencialmente perigosas possam ser desfeitas, pois podem ser acionadas equivocadamente pelos usuários. Se um usuário pedir para apagar um arquivo, por exemplo, o sistema deve obedecer e se preparar para desfazer a ação, caso solicitado.

Segundo Cooper (1999), a possibilidade de desfazer ações evita a necessidade de apresentar diversos diálogos pedindo confirmação das ações dos usuários. Usar diálogos de confirmação em excesso e de forma indiscriminada não apenas aumenta o tempo de realização das tarefas, mas também pode tornar a comunicação ineficiente, pois muitos usuários acabam prosseguindo a interação sem mesmo ler o conteúdo desses diálogos. Quando uma operação considerada perigosa não puder ser desfeita, devemos projetar medidas de segurança para que ela não seja acionada incidentalmente (Figura 8.4).

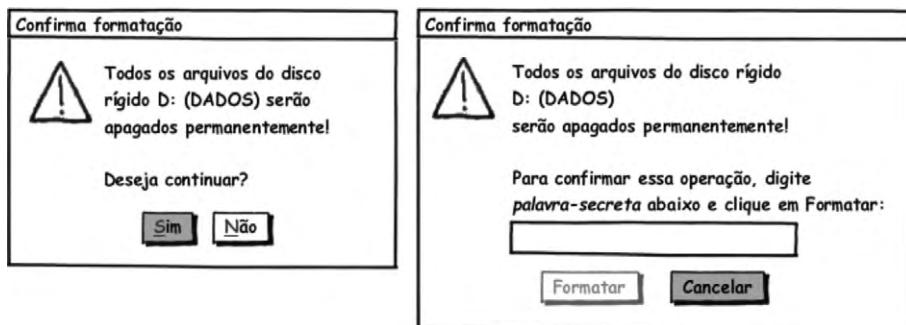


Figura 8.4 Diálogos de confirmação para uma operação que não pode ser desfeita. Qual é a mais segura?

Para aumentar o controle do usuário sobre a interação, o sistema também deve permitir que os usuários trabalhem de modo *flexível*, através de parâmetros configuráveis, por exemplo. Entretanto, devemos buscar um equilíbrio entre a gama de opções oferecidas ao usuário e sua capacidade de entender as consequências da combinação de parâmetros escolhida. O sistema não deve forçar o usuário a escolher o tempo todo um sem-número de opções para prosseguir rumo ao seu objetivo (Cooper, 1999). Daí a importância de escolher bons valores padrão (*defaults*) para quando não for necessário incomodar o usuário.

8.2.4 Consistência e Padronização

Para facilitar o aprendizado e uso de um sistema, Norman (1988) recomenda *assegurar a consistência da interface com o modelo conceitual embutido no sistema*. Isso requer que tudo sobre o produto (modelo de design e imagem do sistema — veja Seção 3.4 —, incluindo documentação e manuais de instrução) esteja consistente com e exemplifique a operação do modelo conceitual adequado.

Tanto Norman (1988) como Tognazzini (2003) acreditam que a consistência mais importante é com as expectativas dos usuários, como visto na seção anterior. Segundo eles, mesmo quando essa correspondência não é possível, ou seja, quando precisamos definir mapeamentos arbitrários, devemos *padronizar*. Norman, Tognazzini, Nielsen e Shneiderman recomendam padronizar as ações, os resultados das ações, o *layout* dos diálogos e as visualizações de informação. Ações relacionadas em situações semelhantes devem funcionar da mesma forma. Por exemplo, um botão Fechar não deve ser utilizado para cancelar um diálogo em algumas situações e para confirmá-lo em outras.

Os usuários não devem ter de se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. Por exemplo, utilizar rótulos Salvar e Gravar indiscriminadamente em um mesmo sistema pode confundir o usuário. A mesma terminologia deve ser utilizada em perguntas, menus e sistemas de ajuda. Nielsen (1993) recomenda ainda seguir as convenções da plataforma ou ambiente computacional. Segundo Cooper (1999), um sistema que se comporte de modo irregular ou instável não é confiável.

Tognazzini (2003) ressalta ainda que alguns elementos de interface exigem maior consistência do que outros. Em particular, elementos que não possuem correspondência visual na interface ou cuja operação é internalizada pelos usuários (e acionada “sem pensar”) não devem variar em diferentes partes do sistema. Por exemplo, sequências de teclas de atalho e o comportamento da roda do mouse para rolamento ou zoom devem ser padronizados em todo o sistema.

Em contrapartida, se dois elementos de interface possuem comportamento diferente, eles devem ter aparências distintas. Por exemplo, um elemento de interface utilizado para selecionar uma opção deve ser distinto de um elemento utilizado para disparar uma ação do sistema.

Tognazzini (2003) alerta, no entanto, que eventualmente queremos propositalmente tornar algo inconsistente, para que o usuário não possa atuar de forma automática e precise refletir sobre o que está fazendo. Esse é o caso, por exemplo, de diálogos de confirmação em que os botões Sim e Não são substituídos por rótulos

mais representativos dos efeitos de cada botão, e que não podem ser acionados pelo simples pressionamento das teclas S ou N.

8.2.5 Promovendo a Eficiência do Usuário

Tognazzini (2003) recomenda considerar sempre a *eficiência do usuário* em primeiro lugar, e não a do computador. As pessoas são mais custosas do que máquinas, e uma economia de tempo e esforço do usuário costumam trazer mais benefícios do que economias semelhantes de processamento ou armazenamento. Para isso, Tognazzini sugere *manter o usuário ocupado*. Toda vez que o usuário precisa esperar o sistema responder antes que possa continuar seu trabalho, há perda de produtividade e desperdício de dinheiro. Sendo assim, processamentos demorados não devem prender a interação, mas sim permitir que os usuários continuem seu trabalho com outras partes do sistema, deixando esses processos executando em *background*. Para isso, um bom projeto de arquitetura do software é essencial. Aliás, esse ponto reforça a necessidade de cooperação, comunicação e parceria constante entre os engenheiros e os designers de IHC durante todo o processo de desenvolvimento (veja a Seção 1.2).

Segundo Cooper (1999), o sistema deve ser *sensível ao que o usuário está fazendo* e não deve interrompê-lo desnecessariamente enquanto o usuário estiver trabalhando em algo. Por exemplo, mudar o estado de sistemas de comunicação (e.g., Windows® Live Messenger, Google® Talk) para ocupado enquanto o usuário estiver projetando uma apresentação.

O designer deve *proteger o trabalho dos usuários* (Tognazzini, 2003). Os usuários nunca devem perder o seu trabalho, seja por um erro seu, por uma falha na transmissão de rede, uma falha no fornecimento de energia para o computador ou qualquer outra razão.

O sistema deve se lembrar de tudo o que o usuário disse, para não perguntar de novo (Cooper, 1999), e se *manter informado sobre o usuário*. O sistema deve ser capaz de saber: se essa é a primeira vez em que o usuário acessou o sistema; onde o usuário está no sistema; para onde ele está indo (caso haja um caminho claro em direção a um objetivo); o que o usuário tem feito durante a sessão de uso atual; onde ele estava quando deixou o sistema na última sessão; e outras informações semelhantes que permitam poupar trabalho do usuário e melhorar sua experiência de uso do sistema. O usuário deve ser capaz de sair de um sistema numa máquina e acessá-lo a partir de outra, voltando exatamente ao ponto em que parou.

Para promover a eficiência de usuários frequentes, Nielsen (1993) e Shneiderman (1998) recomendam *fornecer atalhos e aceleradores*. À medida que a frequência de uso aumenta, aumenta também a vontade dos usuários de reduzir o número de

interações e acelerar o passo da interação. Teclas de atalho e comandos ocultos são bastante úteis a usuários experientes, e não prejudicam a interação dos usuários novatos. Exemplos de aceleradores são teclas de atalho para itens de menu (e.g., Ctrl+S para salvar) e acionamento de botões de comando em barras de ferramentas (e.g., botão para imprimir o documento atual utilizando a configuração default). Caso haja sequências de operações frequentes, o designer também pode fornecer meios mais sofisticados, como gravação de macro ou programação por demonstração (Cypher, 1993; Lieberman, 2001).

Para operações frequentes, o designer pode oferecer também a configuração de valores default, individualmente ou em grupo, formando perfis de execução dessas operações. A Figura 8.5 apresenta um diálogo de geração de arquivo PDF de um editor de apresentações que permite armazenar valores de parâmetros na forma de perfis de exportação.

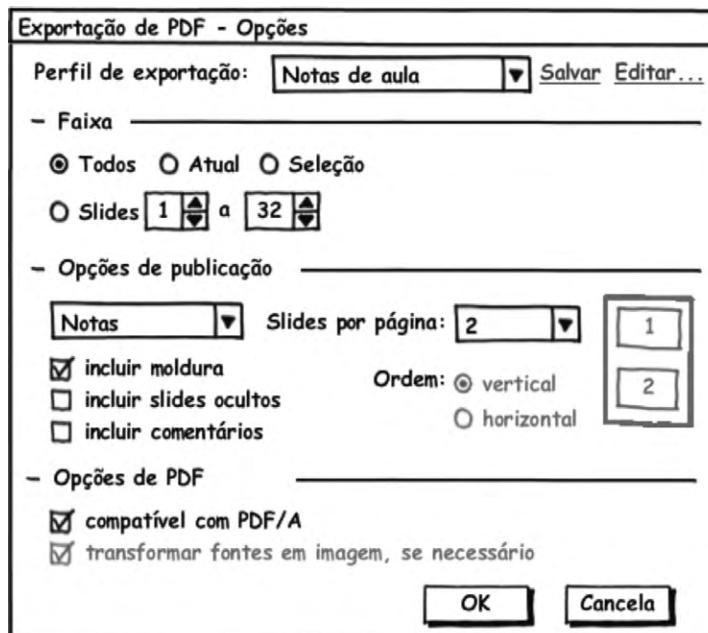


Figura 8.5 Diálogo com parâmetros configuráveis que podem ser armazenados em um perfil de exportação.

8.2.6 Antecipação

As aplicações devem tentar prever o que o usuário quer e precisa, em vez de esperar que os usuários busquem ou coletem informações ou invoquem ferramentas. O designer deve fornecer ao usuário todas as informações e ferramentas necessárias para cada passo do processo (Tognazzini, 2003).

Segundo Cooper (1999), o software deve *tomar iniciativa* e fornecer informações adicionais úteis, em vez de apenas responder precisamente a pergunta que o usuário tiver feito. Por exemplo, quando um usuário pergunta sobre o telefone de um restaurante, o software pode informar também seus dias e horários de funcionamento. Cooper acredita que, enquanto o software estiver à toa, deve *anticipar* e se preparar para situações que provavelmente acontecerão, de modo a responder mais rapidamente quando chegar a hora. Além disso, o software deve ser *observador* e se lembrar quais ações o usuário realiza em sequência, para tentar antever o próximo passo a cada momento e facilitar a sua execução.

Tognazzini destaca a importância de definir cuidadosamente os valores e a configuração padrão (defaults). Ele afirma que, de qualquer forma, os defaults devem ser facilmente substituídos por valores específicos mais adequados à situação atual. Se possível, campos contendo defaults devem vir já selecionados, de forma que os usuários possam editar seu conteúdo com novos valores rápida e facilmente. As pessoas tendem a aceitar os valores defaults para aquilo que elas não entendem ou assumem que o sistema já aprendeu sobre elas ou que seja a resposta “certa”. Por isso, a escolha deve ser cuidadosa. Considere cada alternativa apresentada na Figura 8.6. Ela é eficiente? É neutra? Ou induz a uma determinada opção?

<input type="checkbox"/> Não quero receber a newsletter semanal da Empresa <input checked="" type="checkbox"/> Quero receber a newsletter semanal da Empresa	Quer receber a newsletter semanal da Empresa? <input checked="" type="radio"/> sim <input type="radio"/> não
<input checked="" type="checkbox"/> Não quero receber a newsletter semanal da Empresa <input type="checkbox"/> Quero receber a newsletter semanal da Empresa	Quer receber a newsletter semanal da Empresa? <input type="radio"/> sim <input checked="" type="radio"/> não
Quer receber a newsletter semanal da Empresa? * <input type="radio"/> sim <input type="radio"/> não <small>* indica campo obrigatório</small>	

Figura 8.6 Ilustrações do uso (ou não uso) de valores default.

8.2.7 Visibilidade e Reconhecimento

Norman (1988) afirma que o designer deve *tornar as coisas visíveis: abreviar os golpos de execução e avaliação* (veja Seção 3.4). Antes de executar uma ação, é necessário tornar visível para os usuários o que é possível realizar e como as ações devem ser feitas. Para isso, a interface deve oferecer ações que correspondam a intenções do usuário.

Além disso, a interface não deve oferecer opções que não estejam disponíveis ou não façam sentido em um determinado momento da interação. Depois que o usuário realiza uma ação, a interface deve lhe fornecer indicações do estado do sistema que sejam prontamente percebidas e consistentes com o seu modelo mental, para que ele possa interpretá-las adequadamente e entender os efeitos da ação realizada.

Em outras palavras, o estado do sistema, os objetos, as ações e as opções devem estar atualizados e facilmente perceptíveis (Nielsen, 1993; Shneiderman, 1998; Tognazzini, 2003). O usuário não deve ter de se lembrar para que serve um elemento de interface cujo símbolo não é reconhecido diretamente. Também não deve ter de se lembrar de informações de uma parte da aplicação quando tiver passado para uma outra parte da aplicação. O sistema não deve exigir que o usuário memorize muitas informações ou comandos durante a interação, devido à limitação humana do processamento de informação na memória de curto prazo. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário.

Os usuários também não devem ter de procurar informações sobre o estado do sistema. Em vez disso, eles devem ser capazes de olhar rapidamente para o seu ambiente e obter pelo menos uma primeira aproximação desse estado. Segundo Tognazzini, os *mecanismos de status* são essenciais para fornecer a informação necessária para os usuários responderem adequadamente às mudanças ocorridas. O usuário poderá ter dificuldades em controlar o sistema adequadamente se não possuir informações suficientes sobre o seu estado atual. Entretanto, Cooper (1999) recomenda que o software não exagere nas mensagens de status. Em geral, as informações de status podem ser bem sutis. Por exemplo, o ícone de caixa de entrada do cliente de correio eletrônico pode aparecer vazio, meio cheio ou lotado, para refletir o número de mensagens não lidas.

Quando o usuário realiza uma ação, o sistema deve mantê-lo informado sobre o que ocorreu ou está ocorrendo, através de feedback (resposta do sistema) adequado e no tempo certo (Nielsen, 1993; Shneiderman, 1998; Tognazzini, 2003). Para ações frequentes e com resultado esperado, a resposta pode ser sutil, mas para ações infrequentes e com grandes consequências, a resposta deve ser mais substancial. A Figura 8.7 apresenta um *feedback* sutil como resultado de um cadastro bem-sucedido, e outro *feedback*, destacado, indicando uma falha.

Tognazzini (2003) afirma que, para reduzir a sensação de o programa não estar respondendo, o sistema deve: fornecer feedback visual/sonoro até 50 ms após um clique de botão; sinalizar que está ocupado (e.g., através de uma ampulheta, animada para indicar que o sistema não travou) quando o sistema realizar uma ação que leve entre 0,5 s e 2 s; apresentar, quando a ação levar mais do que 2 segundos, uma men-

sagem indicando a demora estimada e cada passo sendo realizado, juntamente com uma barra de progresso e opções de cancelamento, suspensão ou de execução em *background*. Quando uma operação demorada (mais do que 10 segundos) terminar, o sistema deve emitir um som e fornecer uma indicação visual destacada, para que usuários que tenham desviado sua atenção possam retomar o seu uso do sistema.

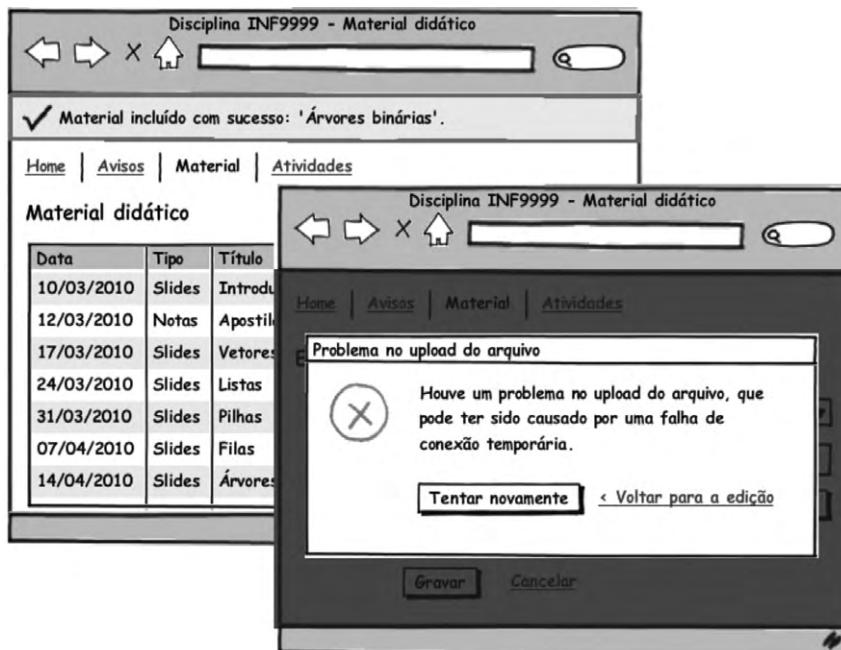


Figura 8.7 Feedback sutil e destacado.

O designer deve manter o usuário informado sobre o caminho que percorreu no sistema ou Web site até o ponto em que se encontra. O usuário não deve ser responsável por elaborar um mapa mental do que fez até o momento ou por onde passou no sistema. Além disso, sinalizações claras orientam a interação do usuário e lhe ajudam a navegar pela aplicação rapidamente, sempre cientes de onde estão (Tognazzini, 2003).

8.2.8 Conteúdo Relevante e Expressão Adequada

Segundo Reeves e Nass (1996), as pessoas dão tratamento humano para qualquer mídia ou tecnologia que apresente comportamento semelhante ao de uma pessoa, mesmo sabendo que isso é tolice e negando que tenham feito isso *a posteriori*. Em linha com o princípio de conversa cooperativa de Grice (1972), eles destacam que uma interação polida segue quatro máximas: qualidade, quantidade, relação (ou relevância) e modo (ou clareza).

A máxima da *qualidade* afirma que não devemos dizer nada que saibamos não ser verdade ou para o que não tenhamos evidências, ou seja, não devemos mentir ou especular. A máxima da *quantidade* diz respeito à quantidade de informação comunicada: a contribuição de uma fala deve ser tão informativa quanto necessário para os objetivos da conversa, e não mais. Uma constante dentre os profissionais de IHC é a busca pela simplicidade. Seguem o lema “menos é mais”. A máxima da *quantidade* está fortemente relacionada à simplicidade da interface. A máxima da *relação* ou *relevância* afirma que tudo o que for dito deve ter relação clara com os tópicos da conversa até o momento e ser relevante ao objetivo dos interlocutores. Finalmente, a máxima de *modo* ou *clareza* pede para evitar a prolixidade e ambiguidade, buscar a concisão e ordenar adequadamente a conversa.

Em linha com a máxima de quantidade, Nielsen (1993) defende o *projeto estético e minimalista*. Ele afirma que os diálogos não devem conter informações que sejam irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com as unidades relevantes de informação e reduz sua visibilidade relativa.

Cooper (1999) alerta que, quando o sistema sonega informação, ele obscurece seu comportamento. Para que os usuários gostem do software e tenham uma experiência agradável, ele sugere projetar uma interação respeitosa, generosa e prestativa. Ele considera uma boa interação mais importante do que a composição da interface concreta propriamente dita.

Tognazzini (2003) oferece uma série de recomendações relacionadas à *redação* em interfaces gráficas. As mensagens de instrução e ajuda devem ser concisas e informativas sobre problemas que ocorrerem. Os rótulos de menus e botões devem ser claros e livres de ambiguidade. Entretanto, nem sempre um termo mais preciso é melhor. Por exemplo, considere um editor de texto com os seguintes itens de menu: Inserir Quebra de Página, Acrescentar Nota de Rodapé e Construir Tabela. Embora precisos, eles trazem menos benefícios do que os itens: Inserir Quebra de Página, Inserir Nota de Rodapé e Inserir Tabela. Nessa segunda opção, os usuários conseguem varrer as opções disponíveis mais rapidamente, pois a variedade dos verbos utilizados aumenta o tempo que leva para decodificá-los, sem ser suficientemente informativa de modo a compensar o tempo despendido.

Além de cuidar do conteúdo, o designer deve se certificar de que o texto também seja *legível*. Para isso, deve ser apresentado com alto contraste e favorecer texto preto sobre fundo branco ou amarelo-claro, evitando fundos de cor cinza (Tognazzini, 2003). Os tamanhos de fonte devem ser suficientemente grandes para serem lidos em monitores de tamanho e resolução padrão. E os dados podem ser apresentados

em fonte maior ou com mais destaque do que rótulos e instruções, principalmente quando os dados forem numéricos. E, sempre que possível, o designer deve permitir que usuários com deficiências visuais aumentem o tamanho da fonte.

Ao *utilizar cores* para transmitir alguma informação através da interface, é necessário utilizar dicas secundárias claras para transmitir a mesma informação àqueles que não conseguem distinguir as cores utilizadas, seja por limitações físicas, como daltonismo, que afeta cerca de 10% da população masculina, ou por limitações do dispositivo utilizado (Tognazzini, 2003). Dicas secundárias incluem variações na escala em tons de cinza, diferentes ilustrações ou rótulos associados a cada cor apresentada.

Como visto na Seção 1.4, profissionais oriundos de diversas áreas devem colaborar em atividades de IHC. Dentre eles, os designers gráficos são os principais responsáveis pela identidade visual do sistema, incluindo o *layout*, que define a disposição espacial dos elementos de interface, e a escolha de fontes, formas, cores, texturas, imagens e outros símbolos não tipográficos (Mullet e Sano, 1995). Ao elaborar esboços de interface ou protótipos, deve-se buscar fazer uso de alguns princípios básicos, para que a atenção do usuário não se desvie para detalhes da interface que não sejam o foco do design naquele momento, mas que, por violarem princípios básicos de design gráfico, estejam lhe incomodando.

Mullet e Sano (1995) organizam princípios de design visual de interfaces em torno dos seguintes temas: elegância e simplicidade; escala, contraste e proporção; organização e estrutura visual; módulo e programa; imagem e representação; e estilo. Os esboços de interface se beneficiam principalmente dos princípios relacionados com a estrutura geral da interface, como, por exemplo, o uso de *grid* para alinhar os elementos e o uso de espaço para guiar a disposição de elementos conforme a direção natural de leitura dos usuários.

8.2.9 Projeto para Erros

Norman (1988) recomenda *projetar para o erro*, ou seja, assumir que qualquer erro potencial será cometido. O designer deve ajudar o usuário a se recuperar de um erro, informando-lhe sobre o que ocorreu, as consequências disso e como reverter os resultados indesejados. Como visto, os sistemas devem ser exploráveis, ou seja, deve ser fácil reverter as operações e difícil realizar ações irreversíveis. Além disso, Cooper (1999) recomenda não colocar controles de funções utilizadas com frequência adjacentes a controles perigosos ou que raramente são utilizados. Por exemplo, na Figura 8.8, um botão de inspeção de Propriedades está posicionado bem próximo ao botão

para Desabilitar a conexão de rede que, inclusive, efetua a operação sem pedir confirmação do usuário.

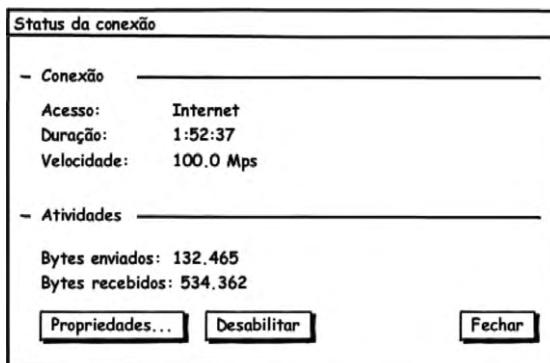


Figura 8.8 Posicionamento de um botão de uso frequente (Propriedades) próximo a um botão de uso infrequente e consequências graves (Desabilitar).

Nielsen (1993) e Shneiderman (1998) recomendam que o designer tente, em primeiro lugar, *evitar que os erros ocorram*, caso possível. Se um erro for cometido, o sistema deve ser capaz de detectá-lo e oferecer mecanismos simples e inteligíveis para tratá-lo. O designer deve *ajudar os usuários a reconhecerem, diagnosticarem e se recuperarem de erros* (Figura 8.9). As mensagens de erro devem ser expressas em linguagem simples e inteligível (sem códigos indecifráveis), indicar precisamente o problema e sugerir uma solução de forma construtiva.

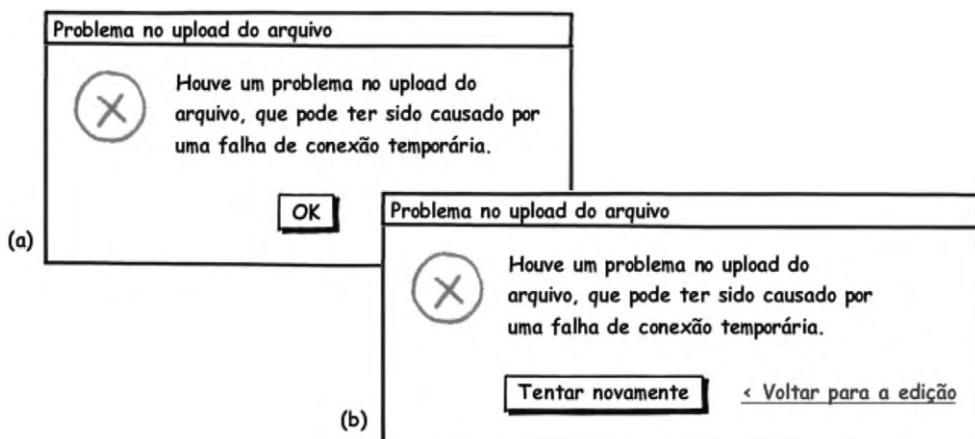


Figura 8.9 Exemplo de mensagens de erro (a) sem e (b) com oportunidade de recuperação e retomada de um caminho de interação produtivo.

Além de erros, também devemos apoiar os usuários a esclarecerem suas dúvidas durante a interação. Para isso, precisamos elaborar *ajuda e documentação* de alta qualidade. Tais informações devem ser facilmente encontradas, focadas na tarefa do usu-

ário, enumerar passos concretos a serem realizados e não ser muito extensas (veja Seção 7.5).

8.3 Padrões de Design de IHC

Padrões de design (*design patterns*) são descrições de melhores práticas num determinado domínio de design (Tidwell, 2006; Borchers, 2001). O conceito de padrões foi proposto na década de 1970 por Christopher Alexander (1979), no domínio de arquitetura e urbanismo, e apresentados à comunidade de Computação na década de 1990 na forma de padrões de arquitetura de software orientado a objetos utilizados na Engenharia de Software (Gamma *et al.*, 1995). Padrões capturam soluções comuns a certos interesses ou tensões de design (chamadas de “forças” na literatura de padrões). Alexander objetivava recriar o conhecimento compartilhado sobre soluções de design boas e adequadas tornando-o explícito, documentado, testado e gradualmente aperfeiçoado (Borchers, 2001).

Tidwell (2006) aponta como vantagens do uso de padrões, além da captura da sabedoria coletiva de designers experientes em IHC, o fornecimento de um vocabulário de design comum e divulgação de boas soluções para a comunidade de design. Ela alerta, no entanto, que padrões não são soluções prontas, nem regras ou heurísticas. Cada aplicação de um padrão difere ligeiramente uma da outra. Padrões de design em IHC também não são descrições passo a passo sobre como projetar uma interface inteira, mas sim descrições para problemas pontuais. O uso de padrões não substitui o processo criativo envolvido num projeto de design, nem assegura por si só a qualidade do produto final. Em outras palavras, padrões devem ser utilizados como recursos em um processo de design reflexivo (veja Seção 4.2), e não como soluções prontas.

Os padrões não são isolados; eles estão relacionados com outros padrões de diversas maneiras. Cada padrão pode ser descrito em maior ou menor nível de detalhes e pode ser adequado a apenas um certo contexto (Borchers, 2001). Um conjunto completo de padrões de design relacionados é chamado de linguagem de padrões (*pattern language*), que definem um extenso vocabulário de elementos utilizados em design, além de formas de articulá-los.

Borchers (2001) utiliza uma estrutura semelhante à de Alexander e sugere que os padrões sejam descritos através dos seguintes elementos:

- o *nome* do padrão, para transmitir em poucas palavras a ideia do padrão, de modo a facilitar sua memorização e a menção a ele em uma reflexão ou discussão durante o design;

- uma *avaliação* de sua validade (indicada por zero, um ou dois asteriscos), indicando o grau de confiança que os autores tinham no padrão;
- uma *imagem* como exemplo de aplicação do padrão;
- o *contexto* em que o padrão pode ser usado, com referência a padrões “maiores” que este ajuda a implementar;
- uma *breve descrição do problema*, um resumo da situação geral que o padrão endereça;
- uma *descrição detalhada do problema*, com base empírica e citando as forças conflitantes que o padrão almeja resolver ou equilibrar;
- a *solução central* do padrão, um conjunto claro mas relativamente genérico de instruções que possam ser aplicadas a diversas situações;
- um *diagrama* ilustrando a solução, geralmente um esboço gráfico da solução e seus principais constituintes;
- *referências a padrões “menores”*, recomendações do autor sobre como implementar e desdobrar ainda mais a solução representada no padrão atual.

Tidwell (2006), por sua vez, descreve seus padrões utilizando, além do título do padrão (e.g., “painéis colapsáveis”), as seguintes informações:

- *o que*, resumindo a solução em um parágrafo curto;
- *usar quando*, indicando as situações em que o padrão se aplica e as forças em atuação;
- *por que*, fornecendo dados que justificam a adequação do padrão à situação descrita;
- *como*, detalhando a solução e as formas como ela pode ser implementada;
- *exemplos*, incluindo diagramas e imagens de interfaces concretas que ilustram o uso do padrão.

Van Welie⁴ representa padrões de forma semelhante a Tidwell, através dos seguintes elementos: *problema*, *solução* (texto e imagem ilustrando uso real), *usar quando*, *como*, *por que*, *mais exemplos*, *implementação* e *literatura*.

O Exemplo 8.1 apresenta um padrão de design comumente descrito: o padrão de Acordeão, adaptado da biblioteca de van Welie. Os textos em itálico representam ligações com outros padrões, sejam da mesma biblioteca ou de outras bibliotecas, como a biblioteca de Tidwell.⁵

4 <http://www.welie.com/patterns/index.php>.

5 <http://designinginterfaces.com/>.

Exemplo 8.1 – Padrão de Design de Interface

Título: Acordeão

Problema: O usuário precisa encontrar um item dentre as opções de navegação.

Solução: Empilhar painéis vertical ou horizontalmente, abrindo um painel de cada vez, enquanto colapsa os demais.

Usar quando: Como mecanismo de navegação, sendo conceitualmente equivalente a *Guias* e uma solução alternativa a *Árvores de navegação*. Embora seja utilizado como parte de um *Assistente*, isso não é recomendado, pois apresenta pior qualidade de uso do que implementações tradicionais. Pode ser uma boa forma de implementar uma seção de *Perguntas Frequentes (FAQ)*, em que cada pergunta é aberta de uma vez. Um outro uso seria para gerenciar configurações de preferências. O número de painéis deve ser reduzido, em geral menor que dez.

Exemplo:

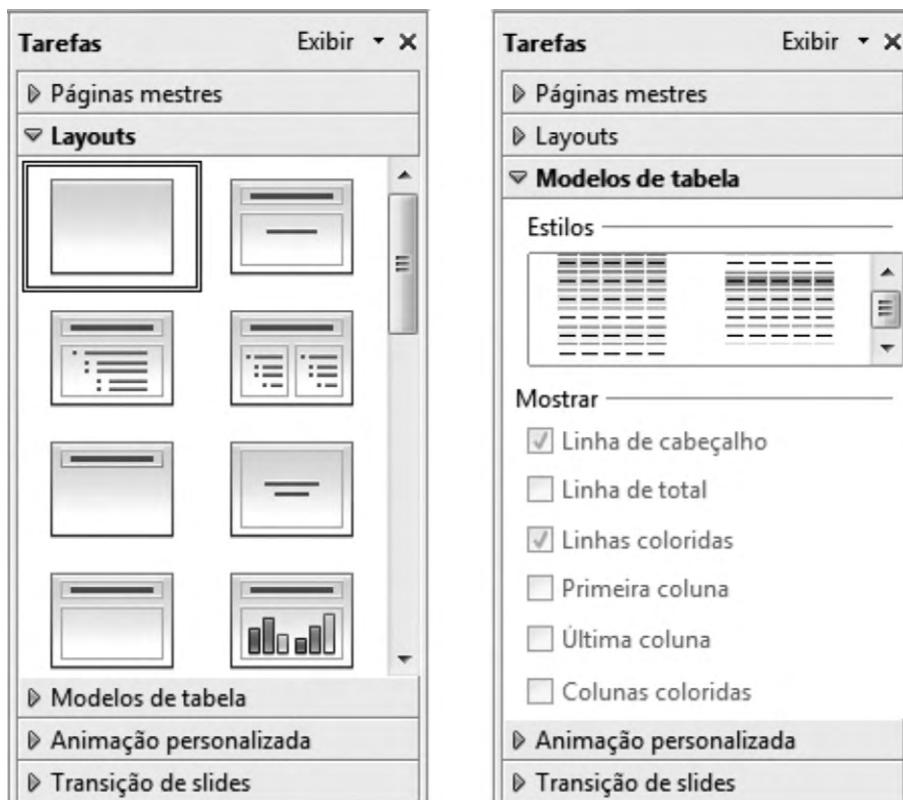


Figura 8.10 Exemplo do padrão de interface Acordeão, com o painel *Layouts* expandido (esq.) e com o painel *Modelos de tabela* expandido (dir.).

Como: Os painéis podem ser dispostos vertical ou horizontalmente. Apenas um painel é mantido aberto de cada vez. Quando mais do que um painel pode ser mantido aberto, trata-se do padrão *Painéis Colapsáveis*. Em geral, painéis verticais são destinados a submenus, enquanto painéis hori-

zontais revelam grandes áreas de conteúdo. Os seguintes cuidados devem ser tomados na implementação do padrão Acordeão:

- Anime a abertura dos painéis para fornecer aos usuários *feedback* sobre o que está acontecendo. A animação deve ser sutil e durar no máximo 250 ms.
- Permita que a navegação seja feita através das setas do teclado.
- Destaque o painel atual para que o usuário possa facilmente diferenciar o cabeçalho do painel aberto dos cabeçalhos dos painéis fechados.
- Certifique-se de que o tamanho total do Acordeão pode aumentar ou diminuir para acomodar o conteúdo adequadamente.

Por quê: Um acordeão é útil para comprimir muitos elementos num espaço de tela compacto. Os elementos podem ser propriedades, perguntas ou simples itens de navegação. A desvantagem óbvia é que os elementos dos outros painéis ficam ocultos.

8.4 Guias de Estilo

É comum, principalmente em projetos grandes, reunir os princípios e as diretrizes adotados em um documento intitulado **guia de estilo**. Trata-se de um registro das principais decisões de design tomadas, de forma que elas não se percam, isto é, sejam efetivamente incorporadas no produto final. Guias de estilo servem de ferramenta de comunicação entre os membros da equipe de design e também com a equipe de desenvolvimento. É importante que as decisões de design possam ser facilmente consultadas e reutilizadas nas discussões sobre extensões ou versões futuras do produto.

Um guia de estilo pode ser elaborado com diferentes escopos: plataforma (composição de dispositivo e sistema operacional), corporativo (para assegurar a padronização e consistência entre produtos de uma empresa), família de produtos e um produto específico (Mayhew, 1999).

Um guia de estilo deve incorporar decisões de design envolvendo os principais elementos e considerações de design de interface. Marcus (1992) considera os seguintes elementos:

- *layout*: proporção e *grids*; uso de metáforas espaciais; design gráfico de exibidores e ferramentas;
- tipografia e seu uso em diálogos, formulários e relatórios;
- simbolismo: clareza e consistência no design de ícones;
- cores: os dez mandamentos sobre o uso de cores;
- visualização de informação: design de gráficos, diagramas e mapas;
- design de telas e elementos de interface (*widgets*).

Uma estrutura comum de guia de estilo é a seguinte (Marcus, 1992; Mayhew, 1999):

1. Introdução
 - 1.1. Objetivo do guia de estilo
 - 1.2. Organização e conteúdo do guia de estilo
 - 1.3. Público-alvo do guia de estilos (programadores, gerentes, equipe de suporte)
 - 1.4. Como utilizar o guia (em produção e manutenção)
 - 1.5. Como manter o guia
2. Resultados de análise
 - 2.1. Descrição do ambiente de trabalho do usuário
3. Elementos de interface
 - 3.1. Disposição espacial e *grid*
 - 3.2. Janelas
 - 3.3. Tipografia
 - 3.4. Símbolos não tipográficos
 - 3.5. Cores
 - 3.6. Animações
4. Elementos de interação
 - 4.1. Estilos de interação
 - 4.2. Seleção de um estilo
 - 4.3. Aceleradores (teclas de atalho)
5. Elementos de ação
 - 5.1. Preenchimento de campos
 - 5.2. Seleção
 - 5.3. Ativação
6. Vocabulário e padrões
 - 6.1. Terminologia
 - 6.2. Tipos de tela (para tarefas comuns)
 - 6.3. Sequências de diálogos (e.g., para *feedback* ou confirmação de uma operação)

Mayhew (1999) sugere ainda que o guia de estilo inclua todos os produtos do levantamento de dados e análise das necessidades dos usuários, registrando o *design rationale*, ou seja, mantendo o rastreamento entre uma decisão de design e os elementos de discussão que culminaram naquela decisão.

Quando elaboramos um guia de estilo, não basta simplesmente produzi-lo. Deveremos comunicar adequadamente sua existência e importância para os demais designers e desenvolvedores, oferecer treinamento, facilitar o acesso ao documento como um todo ou a um tópico específico e investir em uma mudança na cultura de design

e desenvolvimento da equipe. Além disso, não devemos tratar o guia de estilo como um conjunto de regras, mas sim uma ferramenta prática de apoio ao trabalho e à criatividade. Em outras palavras, um guia de estilo deve ser utilizado como parte de um *processo reflexivo de design*, e não como um conjunto de soluções prontas ou fórmulas geradoras de soluções (veja Seção 4.2).

Atividades

Para as atividades abaixo, considere a agenda pessoal integrada com lista de afazeres e correio eletrônico utilizada no capítulo anterior, projetada para uso em dois dispositivos: um computador desktop e um *smartphone*. Retomando as atividades do capítulo anterior, examine os esboços produzidos.

1. *Uso de princípios e diretrizes.* Quais princípios e diretrizes foram aplicados nos esboços? Algum princípio ou diretriz foi violado? Como pode ser resolvido? Caso não seja resolvido, quais as possíveis consequências da violação?
2. *Uso de padrões.* Examine as bibliotecas de padrões citadas neste capítulo. Quais padrões se encaixam para a solução sendo projetada? Que adaptações precisam ser feitas?
3. *Guia de estilo.* A partir dos modelos e esboços elaborados, destaque os elementos que podem compor um guia de estilo para futuras funcionalidades do mesmo sistema ou para sistemas complementares produzidos para o mesmo cliente.

9

Planejamento da Avaliação de IHC

Objetivos do Capítulo

- Discutir a importância de avaliar a qualidade de uso de um sistema interativo.
 - Descrever o planejamento e a execução da avaliação de IHC envolvendo ou não usuários.
 - Caracterizar os objetivos de avaliação e contextos de projeto que auxiliam na escolha do método de avaliação a ser utilizado.
-

A avaliação de IHC é uma atividade fundamental em qualquer processo de desenvolvimento que busque produzir um sistema interativo com alta qualidade de uso. Ela orienta o avaliador a fazer um *julgamento de valor* sobre a qualidade de uso da solução de IHC e a *identificar problemas* na interação e na interface que prejudiquem a experiência particular do usuário durante o uso do sistema. Assim, é possível corrigir os problemas relacionados com a qualidade de uso antes de inserir o sistema interativo no cotidiano dos usuários, seja um sistema novo ou uma nova versão de algum sistema existente.

Com uma visão ampla do processo de desenvolvimento e dos critérios de qualidade desejáveis, este capítulo começa discutindo a importância de avaliar a qualidade de uso de um sistema interativo. Ele descreve o que pode ser avaliado, quando e onde uma avaliação pode ser realizada e quais tipos de dados são coletados e produzidos. Em seguida, descreve os tipos mais comuns de métodos de avaliação e as atividades básicas de uma avaliação de IHC. O capítulo conclui com uma breve apresentação do *framework* DECIDE, proposto para orientar o planejamento de uma avaliação de IHC (Sharp *et al.*, 2007).

9.1 Por que Avaliar?

Conhecer critérios de qualidade e seguir processos de fabricação que buscam criar produtos adequados a esses critérios nem sempre resultam em produtos de qualidade. É possível que algo passe despercebido durante a produção e acabe prejudicando a qualidade do produto final. Pode ser um problema de matéria-prima com defeito ou de má qualidade; pode ser um descuido na manipulação de materiais; pode acontecer um erro humano durante o processo de produção, e assim por diante. Em particular, quando estamos trabalhando com sistemas interativos, os problemas costumam ocorrer na coleta, interpretação, processamento e compartilhamento de dados entre os interessados no sistema (*stakeholders*), e até na fase de implementação (por exemplo, um programador pode cometer o equívoco de codificar uma informação errada sobre o domínio). Então, o que é possível fazer para entregar ao consumidor (i.e., usuário final) um produto de qualidade? Além de continuar seguindo processos de design e desenvolvimento comprometidos com a qualidade do produto final, também é preciso avaliar se o produto resultante desse processo atende aos critérios de qualidade desejados. A avaliação do produto final possibilita entregar um produto com uma garantia maior de qualidade. Para isso, se algum problema for encontrado durante a avaliação, ele deve ser corrigido antes de o produto chegar ao consumidor.

É difícil garantir a “qualidade total” de um produto, porque seria necessário avaliar o produto final em todas as situações de uso possíveis. Além de ser inviável prever

todas essas situações, o custo de tal avaliação seria alto demais, e exigiria muito tempo e esforço para sua realização.

Como qualquer produto, um sistema interativo deve ser avaliado sob a perspectiva de quem o concebe, de quem o constrói e de quem o utiliza. Na perspectiva de quem constrói, o objetivo principal da avaliação é verificar se o sistema funciona de acordo com a especificação de requisitos, ou seja, o foco está em verificar se o sistema recebe os dados de entrada, processa e fornece os dados de saída conforme especificado. Os critérios de qualidade avaliados nessa perspectiva estão relacionados à construção de sistemas interativos, tal como robustez e confiabilidade (Avizienis *et al.*, 2004). Esses critérios de qualidade normalmente abstraem o que existe e ocorre fora do sistema (através da interface com usuário durante a interação), e se concentram no que existe e ocorre *dentro* do sistema. Existem vários métodos de avaliação para verificar a qualidade de um sistema interativo do ponto de vista de quem o constrói (Pressman, 2002; Delamaro *et al.*, 2007), como testes de caixa-branca e caixa-preta, por exemplo. Essa avaliação costuma ser realizada por uma série de testes ao longo do processo de desenvolvimento, como, por exemplo, testes de unidade, de integração, de sistema e de operação. Mesmo depois de avaliado e confirmado que um sistema interativo funciona conforme sua especificação, ainda assim ele pode apresentar problemas relacionados ao uso, pois o que ocorre fora da interface durante a interação ainda não foi avaliado do ponto de vista do usuário. Em outras palavras, problemas de IHC ainda podem continuar existindo depois que os problemas na construção (das funcionalidades) do sistema forem identificados e resolvidos.

Nas perspectivas de quem concebe e de quem utiliza um sistema interativo, a avaliação tem por objetivo principal verificar se o sistema apoia adequadamente os usuários a atingirem seus objetivos em um contexto de uso. Nessa perspectiva, o que existe *dentro* do sistema só interessa à medida que determina o comportamento aparente deste (i.e., que emerge através da interface) e afeta a experiência vivenciada pelo usuário durante a interação. O foco passa a ser o que existe e ocorre da interface com usuário para *fora*. Os critérios de qualidade avaliados nessa perspectiva são relacionados ao uso, tais como: usabilidade, experiência do usuário, acessibilidade e comunicação, apresentados no Capítulo 2. Neste capítulo e no próximo, nos concentraremos em métodos de avaliação da qualidade de uso propostos na área de IHC.

Um sistema interativo é resultado de decisões de design tomadas com base na interpretação do designer sobre a situação atual, sobre sua identificação e interpretação das necessidades e oportunidades de melhoria, no seu conhecimento sobre soluções para problemas semelhantes ou relacionados e na sua criatividade para conceber novas soluções para o problema específico em questão. Desse modo, o processo de

interação usuário-sistema e o comportamento do sistema durante esse processo são regidos pela lógica definida pelo designer. Os usuários podem ou não compreender e concordar com a lógica do designer, podem ou não julgar a solução de IHC apropriada e melhor do que as soluções existentes e, quando tiverem escolha, podem ou não incorporá-la no seu dia a dia. Portanto, o designer não pode presumir que, se ele aprovar uma solução de interação e interface, o usuário também irá aprová-la e fazer uso dela no seu cotidiano. É importante lembrar que os usuários são pessoas diferentes dos designers e desenvolvedores. Eles muito provavelmente sentem, interpretam, pensam e vivem em culturas e sociedades diferentes, com costumes e gostos próprios, e possuem objetivos diferentes (e.g., usar *vs.* conceber *vs.* desenvolver um produto).

As diferenças entre quem concebe e quem utiliza uma solução IHC não podem ser desprezadas. É importante que a solução de IHC seja avaliada do ponto de vista dos usuários, preferencialmente com a participação deles durante a avaliação. Aquele que avalia a qualidade de uso defende o ponto de vista e os interesses dos usuários, atuando como uma espécie de advogado deles durante o processo de desenvolvimento (Prates e Barbosa, 2003). Sempre que possível, a avaliação de IHC deve ser conduzida por avaliadores que não participaram da concepção da solução, pois eles possuem melhores condições de analisar a solução sob um ponto de vista mais neutro, para defender os usuários e não o design concebido.

A qualidade de uso de um sistema interativo sempre será avaliada. Isso não pode ser ignorado. O usuário final sempre avalia o sistema durante sua experiência de uso, tecendo uma opinião sobre ele. É fundamental que a equipe de desenvolvimento assuma a responsabilidade de avaliar a qualidade de uso do sistema antes de ele ser entregue ao usuário. É preciso buscar um entendimento de como os usuários podem vir a utilizar o sistema e se vão apreciá-lo, principalmente quando ele for um sistema inovador resultante principalmente da criatividade do designer (Sharp *et al.*, 2007). Identificar e corrigir os problemas relacionados com a qualidade de uso antes de o sistema ser entregue ao usuário demonstra profissionalismo da equipe de desenvolvimento (Prates e Barbosa, 2003).

Dentre as razões para avaliarmos a qualidade de uso de sistemas computacionais interativos, Tognazzini (2000) destaca as seguintes:

- os problemas de IHC podem ser corrigidos antes e não depois de o produto ser lançado;
- a equipe de desenvolvimento pode se concentrar na solução de problemas reais, em vez de gastar tempo debatendo gostos e preferências particulares de cada membro da equipe a respeito do produto;

- os engenheiros sabem construir um sistema interativo, mas não sabem e não estão em uma posição adequada para discutir sobre a qualidade de uso. Quem será o advogado do usuário para defender seus interesses durante o processo de desenvolvimento?;
- o tempo para colocar o produto no mercado diminui, pois os problemas de IHC são corrigidos desde o início do processo de desenvolvimento, assim que aparecem, exigindo menos tempo e esforço para serem corrigidos;
- identificar e corrigir os problemas de IHC permitem entregar um produto mais robusto, ou seja, a próxima versão corretiva não precisa já começar a ser desenvolvida no momento do lançamento do produto no mercado.

Avaliar a qualidade de uso de sistemas interativos não representa apenas um aumento do custo de desenvolvimento, como alguns gerentes de projeto costumam pensar. O custo de avaliar a qualidade de uso não costuma ser alto quando comparado ao orçamento global de um projeto de desenvolvimento, e principalmente quando consideramos os benefícios significativos e importantes para o sistema (Rubin, 1994; Bias e Mayhew, 2005). A curto prazo, avaliar a qualidade de uso e corrigir os problemas identificados contribuem para aumentar a produtividade dos usuários, diminuir o número e a gravidade dos erros cometidos durante o uso, e aumentar a satisfação dos usuários. A médio e longo prazo, identificar e corrigir os problemas de IHC contribuem para diminuir o custo de treinamento e suporte, e para planejar versões futuras do sistema, pois chamam a atenção da equipe para partes do sistema que podem melhorar, além de indicar outras que podem ser mais exploradas. Um gerente de projeto não pode deixar de usufruir dos benefícios da avaliação da qualidade de uso pelo simples desconhecimento do que é, de quanto custa e de quais são seus benefícios. É possível equilibrar o custo da avaliação de IHC com os benefícios obtidos.

Para obter esses benefícios, uma avaliação de IHC não pode ser realizada simplesmente entregando (um protótipo de) o sistema para alguns usuários utilizarem e aguardando o relato espontâneo de problemas. Pelo contrário, avaliar a qualidade de uso requer um planejamento cuidadoso da avaliação para que não sejam desperdiçados tempo e dinheiro.

Ao planejar uma avaliação de IHC, o avaliador deve decidir o que, quando, onde e como avaliar, bem como os dados a serem coletados e produzidos, além do tipo de método utilizado. Essas questões são importantes para orientar a escolha do método de avaliação, sua execução e apresentação dos resultados. Nas próximas subseções, discutimos cada uma delas.

9.2 O que Avaliar?

A questão fundamental de uma avaliação de IHC é definir quais são os objetivos da avaliação, a quem eles interessam e por quê. Os objetivos de uma avaliação determinam quais *aspectos relacionados ao uso* do sistema devem ser investigados. Esses objetivos são motivados por requisições, reclamações ou comportamentos de qualquer interessado no sistema (*stakeholders*): usuários, designers, cliente (“dono do sistema”), desenvolvedores, departamento de marketing etc. Por exemplo, os usuários podem demonstrar desinteresse em utilizar o sistema ou fazer reclamações a respeito dele; o designer pode desejar comparar alternativas de design; o cliente pode querer verificar se a alta qualidade de uso é um diferencial do seu produto; os desenvolvedores podem querer examinar se a nova tecnologia empregada no desenvolvimento da interface agrada os usuários; o departamento de marketing pode querer lançar um produto que atenda a necessidades dos usuários ainda não exploradas pelos sistemas atuais; e assim por diante. Portanto, o avaliador deve estar atento a situações como essas para definir os objetivos de uma avaliação de IHC de acordo com os interesses dos *stakeholders*. A decisão sobre o que avaliar orienta o avaliador no planejamento, na execução e na apresentação dos resultados da avaliação de IHC.

É possível avaliar diversos aspectos relacionados ao uso de acordo com os interesses dos *stakeholders*. Os principais aspectos avaliados são (Hix e Hartson, 1993; Rubin, 1994; Mack e Nielsen, 1994; Sharp *et al.*, 2007):

- apropriação de tecnologia pelos usuários, incluindo o sistema computacional a ser avaliado mas não se limitando a ele;
- ideias e alternativas de design;
- conformidade com um padrão;
- problemas na interação e na interface.

Avaliar a **apropriação de tecnologia** requer a participação dos usuários para permitir uma melhor compreensão sobre: o contexto em que o sistema avaliado se insere, quais os objetivos e as necessidades dos usuários, como os usuários costumam alcançá-los, em que grau as tecnologias disponíveis satisfazem suas necessidades e preferências e como elas afetam sua vida pessoal e profissional.

A avaliação desse aspecto pode ser realizada em diferentes momentos do processo de design. Pode consistir em um estudo exploratório para apoiar a atividade de análise. Nesse caso, costumamos avaliar sistemas interativos existentes ou outras formas de apoio às atividades dos usuários, buscando julgar se o uso da tecnologia atual é produtivo e identificar necessidades e oportunidades de intervenção. Já ao longo do design, permite à equipe de projeto julgar se existe um consenso entre a equipe de

design, os usuários e demais *stakeholders* sobre o que foi aprendido durante a elicitação e análise de requisitos e sobre a qualidade da solução sendo projetada. Para isso, é comum utilizarmos como insumo protótipos do sistema sendo projetado, em geral representados através de esboços de tela ou cenários de uso, conforme apresentado no Capítulo 7.

A avaliação da apropriação de tecnologia também permite compreender os efeitos da introdução de um sistema interativo novo ou reprojeto no cotidiano dos usuários. Para isso, devemos investigar como os usuários realizam suas atividades antes e depois da intervenção com o sistema, a fim de julgar se o (novo) sistema lhes oferece apoio computacional adequado, conforme definido por algum fator que possa ser observado ou medido. Podemos avaliar, por exemplo, se os usuários são capazes de atingir seus objetivos em menos tempo, com um número menor de erros ou sem a necessidade de treinamento prévio, se estão satisfeitos, se eles se sentem motivados a explorar e aprender novas funcionalidades do sistema, se o uso do sistema é agradável e outras opiniões sobre aspectos gerais ou específicos do sistema. Esse tipo de investigação também nos permite identificar motivos que levam os usuários a não incorporarem um sistema interativo (ou parte dele) no seu cotidiano.

A avaliação de **ideias e alternativas de design** busca comparar diferentes alternativas de solução de acordo com critérios relacionados com o uso e com a construção da interface com usuário. Por exemplo, com relação ao uso podemos avaliar a facilidade de aprendizado, o apoio à recuperação de erros ou o tipo de intervenção pretendido na vida dos usuários, enquanto com relação à construção podemos avaliar o custo e o tempo necessários para o desenvolvimento de cada alternativa e a infraestrutura de hardware necessária para executar cada proposta de solução. Os critérios de análise e dimensões de comparação de alternativas de design devem ser definidos de acordo com os resultados da análise da situação atual, isto é, o avaliador deve considerar o contexto de uso, os objetivos, as necessidades e as preferências dos usuários, como eles costumam satisfazê-los e por que o fazem assim. A avaliação (comparativa ou não) de ideias e alternativas de solução costuma ser realizada de forma rápida e informal durante a atividade de design como parte do ciclo iterativo de concepção da solução final. É comum utilizar protótipos de interface em vários níveis de detalhes nesse tipo de avaliação, mas também é possível comparar soluções de design de IHC prontas, ou seja, concretizadas em outros sistemas interativos existentes. Essa avaliação pode ser realizada com ou sem a participação dos usuários.

Avaliar a **conformidade com um padrão** é importante quando a solução de IHC precisa ter características específicas determinadas por padrões estabelecidos. Por exemplo, pode ser necessário que a solução de IHC esteja de acordo com os padrões

do W3C para acessibilidade. Assim, podemos assegurar que usuários com certas limitações físicas não encontrarão barreiras intransponíveis para acessar a interface do sistema e interagir com ele. Também podemos avaliar se uma solução de IHC segue os padrões do ambiente computacional em que será inserida, como, por exemplo, padrões estabelecidos pelos ambientes de trabalho GNOME® e KDE®, ou pelos sistemas operacionais Microsoft Windows® e MacOS®. Se esses padrões forem seguidos, os usuários acostumados com esses ambientes tendem a ter menos dificuldades para realizar operações básicas, tal como redimensionar e fechar uma janela, por exemplo.

Além disso, podemos verificar se a solução de IHC está em conformidade com padrões utilizados em domínios específicos, como correio e comércio eletrônico. Os usuários de sistemas desses domínios já estão acostumados a certos termos e a certas formas de realizar determinadas operações. Desse modo, se a solução de IHC estiver próxima das convenções adotadas no domínio do sistema, os usuários tendem a ter menos dificuldades para utilizá-lo porque já estão familiarizados com essas convenções. Uma organização também pode definir padrões (no sentido de normas) para seus sistemas, e exigir que as soluções de IHC propostas estejam de acordo com esses padrões. De maneira geral, verificar a conformidade com padrões contribui para a consistência e coerência entre as soluções de IHC que seguem esses padrões. A avaliação desse aspecto não exige a participação dos usuários.

Problemas na interação e na interface são os aspectos mais avaliados na área de IHC. Na avaliação desses aspectos, o avaliador pode contar ou não com a participação dos usuários para coletar dados relacionados ao uso de sistemas interativos. Ele analisa os dados coletados com objetivo de identificar problemas na interação e na interface que prejudiquem a qualidade de uso do sistema. Os problemas identificados costumam ser classificados de acordo com sua gravidade (grau de impacto nocivo), com a frequência em que tendem a ocorrer e com os fatores que compõem os critérios de qualidade de uso prejudicados — usabilidade, experiência do usuário, acessibilidade ou comunicabilidade. Por exemplo, um avaliador pode relatar e justificar um problema que prejudica a facilidade de recordação (fator de usabilidade) e outro que dificulta a localização na interface de onde o usuário pode expressar determinada intenção de comunicação durante o uso (etiqueta “Cadê?” no método de avaliação de comunicabilidade, apresentado na Seção 10.2.2).

Os objetivos de uma avaliação de IHC precisam ser detalhados em perguntas mais específicas para torná-los operacionais (Sharp *et al.*, 2007). A elaboração dessas perguntas deve considerar os usuários-alvo, suas atividades e os artefatos utilizados, que normalmente incluem uma representação ou protótipo da solução de IHC a ser

avaliada. A Tabela 9.1 apresenta exemplos de perguntas associadas aos objetivos de avaliação de IHC descritos anteriormente.

Tabela 9.1 Exemplos de perguntas que uma avaliação de IHC pode responder.

objetivos	exemplos de perguntas a serem respondidas
analisar a apropriação da tecnologia	De que maneira os usuários utilizam o sistema? Em que difere do planejado? Como o sistema interativo afeta o modo de as pessoas se comunicarem e relacionarem? Que variação houve no número de erros cometidos pelos usuários ao utilizarem o novo sistema? E no tempo que levam para atingir seus objetivos? E na sua satisfação com o sistema? O quanto os usuários consideraram o apoio computacional adequado para auxiliá-los na realização de suas atividades? O quanto eles são motivados a explorar novas funcionalidades? Quais são os pontos fortes e fracos do sistema, na opinião dos usuários? Quais objetivos dos usuários podem ser alcançados através do sistema? E quais não podem? Quais necessidades e desejos foram ou não atendidos? A tecnologia disponível pode oferecer maneiras mais interessantes ou eficientes de os usuários atingirem seus objetivos? O que é possível modificar no sistema interativo para adequá-lo melhor ao ambiente de trabalho? Por que os usuários não incorporaram o sistema no seu cotidiano?
comparar ideias e alternativas de design	Qual das alternativas é a mais eficiente? Mais fácil de aprender? Qual delas pode ser construída em menos tempo? De qual delas se espera que tenha um impacto negativo menor ao ser adotada? Qual delas torna mais evidente os diferenciais da solução projetada? Qual delas os usuários preferem? Por quê?
verificar a conformidade com um padrão	O sistema está de acordo com os padrões de acessibilidade do W3C? A interface segue o padrão do sistema operacional? E da empresa? Os termos na interface seguem convenções estabelecidas no domínio?
identificar problemas na interação e interface	<i>Considerando cada perfil de usuário esperado:</i> O usuário consegue operar o sistema? Ele atinge seu objetivo? Com quanta eficiência? Em quanto tempo? Após cometer quantos erros? Que parte da interface e da interação o deixa insatisfeito? Que parte da interface o desmotiva a explorar novas funcionalidades? Ele entende o que significa e para que serve cada elemento de interface? Ele vai entender o que deve fazer em seguida? Que problemas de IHC dificultam ou impedem o usuário de alcançar seus objetivos? Onde esses problemas se manifestam? Com que frequência tendem a ocorrer? Qual é a gravidade desses problemas? Quais barreiras o usuário encontra para atingir seus objetivos? Ele tem acesso a todas as informações oferecidas pelo sistema?

A avaliação de qualquer aspecto relacionado ao uso de sistemas computacionais interativos, principalmente os problemas na interação e na interface e a apropriação da tecnologia pelos usuários, também fornece insumos para elaborar material de apoio e de treinamento, tais como: tutoriais, instruções de uso e sistema de ajuda.

9.3 Quando Avaliar o Uso de um Sistema?

Os métodos de avaliação de IHC podem ser aplicados em diferentes momentos do processo de desenvolvimento, dependendo dos dados disponíveis sobre a solução de IHC sendo concebida. Desde o início da atividade de design, o designer explora ideias alternativas de intervenção na situação atual. Essas ideias são elaboradas e refinadas através de ciclos de (re)design e avaliação, até o designer chegar a uma solução de IHC que possa ser construída. A avaliação de IHC realizada durante a elaboração da solução, ou seja, *antes* de termos uma solução pronta, é chamada de avaliação *formativa* ou *construtiva*. A avaliação de IHC realizada *depois* de uma solução estar pronta é chamada de avaliação *somativa* ou *conclusiva* (Hix e Hartson, 1993; Sharp *et al.*, 2007).

A **avaliação formativa** é realizada ao longo de todo o processo de design para compreender e confirmar a compreensão sobre o que os usuários querem e precisam, e para confirmar se e em que grau a solução sendo concebida atende às necessidades dos usuários com a qualidade de uso esperada. Ela envolve principalmente analisar e comparar ideias e alternativas de design durante a elaboração da solução de IHC. A avaliação formativa permite identificar tão cedo quanto possível problemas na interação e na interface que possam prejudicar a qualidade de uso, quando os custos de correção ainda são baixos. Diversos artefatos que representam partes de uma solução de IHC podem servir de insumo para uma avaliação formativa, tais como: cenários de uso, esboços de tela, *storyboards*, modelagem da interação e protótipos do sistema em diferentes níveis de detalhe e fidelidade com a solução final, conforme apresentando no Capítulo 7.

A **avaliação somativa** é realizada ao final de um processo de design, quando existir uma solução (parcial ou completa) de interação e de interface pronta, de acordo com um escopo definido. A solução de IHC final pode ser representada por um protótipo de média ou alta fidelidade, ou até mesmo pelo sistema interativo implementado. A avaliação somativa julga a qualidade de uso de uma solução de IHC buscando evidências que indiquem que as metas de design foram alcançadas, ou seja, que o produto possui os níveis de qualidade de uso desejados.

No planejamento da avaliação, o avaliador deve identificar em que momento no processo de desenvolvimento a avaliação será realizada. Isso lhe permite saber quais

representações da solução de IHC estarão disponíveis ou se o próprio sistema pronto e funcionando estará acessível. Esse conhecimento auxilia na escolha do método de avaliação a ser empregado.

9.4 Onde Coletar Dados sobre Experiências de Uso?

A interação usuário-sistema afeta e é afetada pelo *contexto de uso*, que abrange o ambiente físico, social e cultural em que ela ocorre. Em particular, o usuário costuma utilizar outros artefatos em conjunto com o sistema e interagir com outras pessoas enquanto o utiliza. Por exemplo, o usuário pode fazer anotações em papel; ele tem a liberdade de organizar as informações de um modo particular nos diversos artefatos utilizados (no papel, sobre a mesa, em pastas, gavetas, armários etc.); ele pode consultar informações sobre o domínio que estejam fora do sistema; o telefone pode tocar no meio de uma operação ou alguém pode conversar com o usuário, distraindo sua atenção; o usuário pode passar por situações de pressão e cobrança maior no trabalho; a conexão com a Internet pode falhar; e assim por diante. Todos esses elementos e acontecimentos comuns num contexto real podem afetar o uso de um sistema interativo. Conhecer esses fatores é importante para avaliar a adequação do sistema ao ambiente real em que ele será utilizado.

As avaliações de IHC que envolvem a participação dos usuários podem ser realizadas em *contexto real de uso* ou em *laboratório*. A **avaliação em contexto**, que constitui uma forma de estudo de campo (veja Seção 5.5.6), aumenta as chances de verificarmos a qualidade de uso da solução de IHC perante um conjunto maior e mais diversificado de situações de uso. Apesar de não ser capaz de analisar todas as situações de uso possíveis, esse tipo de avaliação fornece dados de situações típicas de uso que não seriam percebidos em uma avaliação em laboratório. Ela permite entender melhor como os usuários se apropriam da tecnologia no seu cotidiano e quais problemas podem ocorrer em situações reais de uso. Todavia, é difícil controlar a execução de uma avaliação em contexto para assegurar que certos aspectos do sistema sejam analisados.

A **avaliação em laboratório** oferece um *controle* maior sobre as interferências do ambiente na interação usuário-sistema e facilita o *registro* de dados das experiências de uso com a solução de IHC avaliada. O laboratório é um ambiente preparado para proporcionar experiências de uso sem interrupções ou inconvenientes que podem ocorrer em um contexto real de uso e até mesmo atrapalhar certos aspectos da avaliação do sistema. Uma avaliação em laboratório permite comparar de forma consistente as experiências que diferentes usuários tiveram com o sistema. Sem as

interferências do contexto de uso, o usuário possui melhores condições de manter o foco nas tarefas sendo avaliadas.

Dependendo do método utilizado, uma sala de reunião com mesa e cadeiras pode ser um ambiente adequado para realizar uma avaliação. Esse é o caso dos métodos de grupo de foco e prototipação em papel. Outros métodos de avaliação são mais indicados para ambientes especiais que favoreçam a observação, como, por exemplo, o teste de usabilidade e o método de avaliação de comunicabilidade. Nesses casos, o laboratório costuma ser um ambiente projetado e construído para facilitar a observação e o registro de dados sobre experiências de uso. De qualquer forma, apesar de ser um ambiente artificial, o laboratório deve ser confortável para os participantes da avaliação, conforme visto na Seção 5.4.

Um ambiente de observação de uso costuma possuir duas salas: uma sala onde o usuário vai utilizar o sistema (*sala de uso*) e outra onde o avaliador vai observá-lo através de um vidro espelhado (*sala de observação*). Desse modo, quem está dentro da sala de uso não consegue ver o que as pessoas que estão na sala de observação estão fazendo. Pelo menos um avaliador fica na sala de observação, fazendo anotações e acompanhando a interação do participante através de um monitor clone, que reproduz tudo o que ocorre no monitor do participante. Um outro avaliador pode ficar atrás do participante como apoio, mas buscando não expressar opiniões ou fornecer instruções que prejudiquem ou invalidem a avaliação.

Em geral, apresentamos ao participante a sala de observação para que ele conheça o que fica do outro lado do vidro espelhado, entenda melhor o procedimento de avaliação e, assim, fique mais tranquilo. O objetivo do vidro espelhado é diminuir a interferência das ações dos observadores sobre o comportamento do participante. Desse modo, os avaliadores que estiverem na sala de observação durante a sessão podem fazer anotações livremente ou consultar algum material de apoio, sem com isso influenciar os resultados da avaliação. A Figura 9.1 apresenta um exemplo de *layout* de laboratório para observar a experiência de uso de um participante.

O registro dos dados observados pode ser feito de várias formas, orientado direta ou indiretamente pelo método de avaliação. A sala de uso deve ter microfones e câmeras de vídeo para gravar falas, gestos e expressões do participante, e um computador com software instalado para capturar um vídeo da interação do participante com o sistema e, ocasionalmente, registrar uma lista das teclas digitadas durante a experiência de uso.

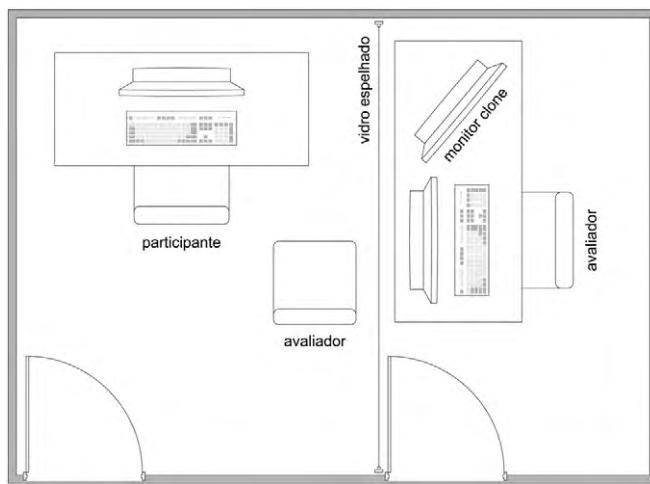


Figura 9.1 Exemplo de laboratório para observar um participante utilizando um sistema computacional interativo.

Outras configurações de laboratório são possíveis dependendo do sistema avaliado. Por exemplo, se for avaliado um sistema de TV digital interativa, a sala de uso deve ter um sofá e uma TV, em vez de uma mesa com computador pessoal e uma cadeira. Uma outra instalação consiste numa sala para diversos observadores, e uma sala de controle, por exemplo, para avaliações do tipo *Wizard of Oz*, nas quais o participante do teste pensa estar interagindo com o sistema, mas de fato é uma pessoa que está enviando as respostas “do sistema” para o seu terminal.

9.5 Que Tipos de Dados Coletar e Produzir?

Dependendo do tipo de avaliação, o avaliador pode coletar dados sobre a situação atual, uso atual da tecnologia, aspectos positivos e negativos identificados durante esse uso, necessidades e oportunidades de intervenção. A abrangência e o foco da coleta de dados devem ser definidos de acordo com os objetivos da avaliação. Por exemplo, se o objetivo for “verificar se determinado sistema (ou protótipo) satisfaz as necessidades dos usuários”, um avaliador poderia coletar dados sobre: o grau de satisfação dos usuários em relação ao sistema; se os usuários sentem necessidade de utilizar outros sistemas ou artefatos para realizarem suas atividades e por que são necessários; quais são os pontos fortes e fracos do sistema avaliado; o que os usuários gostariam que fosse melhorado ou estendido; e assim por diante. Se o objetivo for outro, como, por exemplo, “identificar problemas na interação e interface de determinado sistema (ou protótipo)”, o avaliador poderia coletar dados sobre: quantos usuários conseguiram concluir certas tarefas; quanto tempo foi necessário para concluir cada tarefa; quais erros foram cometidos, em que locais da interface e com que frequência;

quantos usuários sentiram necessidade de consultar material de apoio (tutoriais, manuais de uso, ajuda on-line etc.) e em quais momentos da interação o fizeram, quais eram as dúvidas e se foram esclarecidas e assim por diante.

Os dados coletados são interpretados e analisados de acordo com o método de avaliação escolhido, para produzirem resultados que atendam aos objetivos da avaliação, ou seja, que busquem responder as perguntas específicas elaboradas na definição da avaliação. Cada método descrito no próximo capítulo fornece ao avaliador orientações mais detalhadas sobre quais dados coletar, como analisá-los e quais dados produzir como resultado da avaliação.

Os dados coletados e produzidos em uma avaliação de IHC podem ser classificados de diferentes maneiras. As classificações mais comuns são: nominais, ordinais, de intervalo e de razão (Stevens, 1946; Kiess, 2002; Levine *et al.*, 2008); dados qualitativos e quantitativos (Hix e Hartson, 1993; Sharp *et al.*, 2007); dados subjetivos e objetivos (Meister e Rabideau, 1965). Como visto ao longo do próximo capítulo, cada método de avaliação de IHC privilegia dados e resultados de diferentes tipos.

Dados **nominais** ou categóricos representam conceitos na forma de rótulos ou categorias. Por exemplo, uma pessoa pode ser classificada conforme sua origem étnica em caucasiana, africana, hispânica, asiática ou outra. Podemos atribuir um número para dar nome à categoria (e.g., 1=caucasiano), mas esse número não transmite qualquer informação quantitativa, apenas serve como identificação dos dados, e poderia ser substituído por letra sem qualquer perda de informação.

Entre dados nominais não existe qualquer relação de ordem. Por exemplo, não se pode dizer que um dado nominal é maior ou menor, melhor ou pior, ou mais alto ou mais baixo do que outro. É possível dizer que os dados nominais são iguais ou diferentes, mas nem sempre se consegue caracterizar o *tipo* de diferença que existe entre eles. Exemplos de dados nominais são: atividades que um usuário realiza no sistema, sistemas semelhantes que o usuário já utilizou, formas de acesso à Internet que o usuário utiliza, idiomas que ele entende e nos quais sabe se expressar e problemas enfrentados durante uma sessão de avaliação.

Dados **ordinais**, como o próprio nome revela, representam conceitos com relações que definem algum tipo de ordem entre eles. Essencialmente, dados ordinais produzem um ranqueamento entre pessoas ou coisas, no qual alguém ou algo possui uma variável em maior quantidade ou intensidade do que outros (Kiess, 2002). Por exemplo, a relação entre o Web site que um usuário mais utiliza e o segundo site mais utilizado por ele. Embora possamos identificar se um dado ordinal é maior ou menor que outro, não podemos quantificar com precisão a sua diferença. Por exemplo, é possível que, por trás dos dados ordinais sobre os sites A, B e C, em ordem de frequência

de uso, esteja o fato de o usuário visitar o site A diversas vezes ao dia, o site B uma vez por dia e o site C a cada 15 ou mais dias. Em outras palavras, ao analisar dois ou mais dados ordinais, é possível dizer *qual* é maior ou melhor, mas não em *quanto*.

Se, além da relação de ordem entre os dados, há uma diferença de igual magnitude entre eles, tratam-se de dados de **intervalo** (ou intervalares). Eles representam períodos, faixas ou distâncias entre os dados ordinais, mas a origem da escala é arbitrária, ou seja, uma medida de intervalo não possui um *valor zero verdadeiro* para indicar a total ausência da variável medida (Kiess, 2002). Por exemplo, na escala de temperatura em graus Celsius, podemos afirmar que a diferença entre 20°C e 40°C é a mesma que entre 40°C e 60°C. No entanto, não podemos afirmar que a temperatura de 40°C é duas vezes maior do que a temperatura de 20°C. Já um dado de **razão** possui um valor zero verdadeiro, como, por exemplo, o tempo que uma pessoa leva para realizar uma tarefa. Se os participantes P1 e P2 levaram dois e seis minutos para realizar uma tarefa, respectivamente, faz sentido dizer que P2 levou o triplo do tempo de P1 para realizá-las. Também são dados de razão a frequência de acesso à Internet, o número de erros cometidos, o número de contatos que um usuário possui em uma comunidade virtual.

Os dados coletados com as escalas frequentemente utilizadas em questionários de IHC, como as escalas de Likert e os diferenciais semânticos (veja Seção 5.5.2), não são estritamente intervalares. No entanto, costumam ser considerados como dados de intervalo nas estatísticas produzidas, não apenas em IHC, mas também nas ciências comportamentais (Kiess, 2002).

Dados **qualitativos** representam conceitos que não são representados numericamente. Além dos dados nominais, também são dados qualitativos as respostas livres coletadas em questionários e entrevistas, tais como expectativas, explicações, críticas, sugestões e outros tipos de comentário.

Já os dados **quantitativos** representam numericamente uma quantidade, ou seja, uma grandeza resultante de uma contagem ou medição, tais como: o tempo e número de passos necessários para alcançar determinado objetivo; o número de erros cometidos durante uma sessão de uso; quantas vezes a ajuda on-line e o manual de uso foram consultados; e quantos usuários conseguiram alcançar o objetivo satisfatoriamente (Wixon e Wilson, 1997). Nessa classificação se encaixam os dados ordinais, intervalares e de razão.

Os dados quantitativos são utilizados com frequência para verificar hipóteses, possivelmente formuladas a partir de uma teoria ou de uma pesquisa qualitativa prévia. Por exemplo, a hipótese de que uma solução A é melhor do que uma solução B poderia ser verificada, dentre outras maneiras, contabilizando quantos usuários con-

seguem concluir certas tarefas em um tempo determinado utilizando cada solução. Nesse caso, obteríamos resultados do tipo: na solução A, 61% dos usuários concluíram as tarefas com sucesso, 23% deles concluíram metade das tarefas, e 16% não concluíram sequer a metade das tarefas. Já na solução B, os dados coletados foram 82%, 13% e 5%, respectivamente. Analisados isoladamente, esses números não explicam por que alguns usuários não concluíram as tarefas. Também não sugerem recomendações para tornar a solução A ainda mais adequada aos usuários e às suas tarefas. Apenas fornecem evidências de que a solução A é melhor do que a solução B e, caso os resultados sejam estatisticamente significativos, comprova a hipótese do estudo.

Diferente do foco na *contagem* e *medição* de quantidades realizadas na análise de dados quantitativos, a análise de dados qualitativos envolve principalmente a *interpretação* de conceitos por eles representados. Por isso, ao escolher trabalhar com dados qualitativos, um avaliador normalmente está interessado em *explorar* e *explicar* o que ocorreu (ou pode ocorrer) durante a interação usuário-sistema e como, em vez de testar uma hipótese.

A terceira classificação que utilizamos distingue dados objetivos e subjetivos. Dados **objetivos** podem ser medidos por instrumentos ou software, como, por exemplo, os termos que um participante utilizou para realizar uma determinada busca, as músicas que ele mais ouviu no último mês no seu computador e o tempo que ele levou para realizar uma tarefa numa sessão de teste. Já os dados **subjetivos** precisam ser explicitamente expressos pelos participantes da avaliação, como opiniões e preferências.

Nem todo uso de dados qualitativos consiste na realização de uma pesquisa qualitativa. A **pesquisa qualitativa** consiste em um conjunto de práticas interpretativas e materiais que tornam o mundo visível e transformam-no em uma série de representações, incluindo anotações em campo, entrevistas, conversas, fotografias, gravações e anotações pessoais. A pesquisa qualitativa envolve uma abordagem interpretativa e naturalista do mundo, através da coleção e análise de uma variedade de materiais empíricos que descrevem momentos e significados rotineiros e problemáticos nas vidas dos indivíduos. Esses materiais incluem: estudo de caso, experiência pessoal, introspecção, história de vida, entrevistas, artefatos, textos e produções culturais, textos observacionais, históricos, interacionais e visuais. Os pesquisadores estudam as coisas em seu ambiente natural, buscando interpretar os fenômenos em termos dos significados que as pessoas lhes dão. Eles utilizam uma ampla gama de práticas interpretativas inter-relacionadas, sempre buscando obter um melhor entendimento do assunto em questão (Denzin e Lincoln, 2005, p. 4).

9.6 Qual Tipo de Método de Avaliação Escolher?

Existem vários métodos para avaliar a qualidade de uso propostos na literatura. Cada método atende melhor a certos objetivos de avaliação, orienta explicita ou implicitamente quando e onde os dados devem ser coletados, como eles devem ser analisados, e quais critérios de qualidade de uso (usabilidade, experiência do usuário, acessibilidade ou comunicabilidade) sua análise privilegia. Os métodos de avaliação de IHC podem ser classificados em: métodos de investigação, de observação de uso e de inspeção.

Os métodos de **investigação** (*inquiry*) envolvem o uso de questionários, a realização de entrevistas, grupos de foco e estudos de campo, entre outros. Esses métodos permitem ao avaliador ter acesso, interpretar e analisar concepções, opiniões, expectativas e comportamentos do usuário relacionados com sistemas interativos. Em particular, permitem investigar alternativas de design, problemas que os usuários costumam enfrentar, como eles se apropriaram da tecnologia existente e quais são suas expectativas para futuras interações com tecnologias atuais e novas. São frequentemente utilizados em etapas iniciais do processo de design, para ratificar ou retificar o entendimento da situação atual e identificar necessidades e oportunidades de intervenção (i.e., em análise), bem como explorar formas alternativas de intervenção (i.e., em avaliação formativa). Também são utilizados para avaliar a introdução da nova tecnologia (i.e., em avaliação somativa). Em geral, os métodos de avaliação através de investigação não exigem que os usuários utilizem um sistema interativo durante a coleta de dados. No entanto, o uso de um sistema ou de materiais de apoio, como imagens, cenários e outros tipos de artefatos, pode contribuir para a investigação, ajudando os usuários a se lembrarem de suas experiências e expectativas e também a manterem o foco nas questões de interesse. Os dados obtidos através de investigação com usuários e demais *stakeholders* podem ser coletados através das técnicas discutidas no Capítulo 5, tais como: entrevista, questionário, grupo de foco, estudo de campo e investigação contextual.

Os métodos de **inspeção** permitem ao avaliador examinar (ou inspecionar) uma solução de IHC para tentar *antever* as possíveis consequências de certas decisões de design sobre as experiências de uso. Em outras palavras, tentar identificar problemas que os usuários podem vir a ter quando interagirem com o sistema. Esses métodos geralmente não envolvem diretamente usuários e, portanto, tratam de experiências de uso *potenciais*, e não reais. Além de permitir comparar designs alternativos e buscar problemas em soluções de IHC, os métodos de inspeção permitem ainda avaliar a conformidade com um padrão ou guia de estilo.

Ao inspecionar uma interface, os avaliadores tentam se colocar no lugar de um usuário com determinado perfil, com um certo conhecimento e experiência em algumas atividades. Mas existe um limite na empatia do avaliador; de fato, ele não é o usuário. Ele pode deixar de encontrar problemas que os usuários teriam, e também pode julgar como problemáticos pontos que não causariam dificuldades aos usuários. Além disso, um avaliador pode se concentrar mais em alguns aspectos de usabilidade do que em outros (Nielsen, 1993). Sempre que possível, devemos buscar dados de diferentes fontes e métodos de avaliação para fazer uma apreciação mais robusta do sistema em questão. Os métodos de avaliação através de inspeção, também denominados métodos analíticos, podem ser utilizados ao longo de todo o processo de design, à medida que modelos ou protótipos são elaborados. Costumam ser mais rápidos e ter um custo menor do que métodos que envolvem usuários. Alguns métodos de inspeção são descritos na Seção 10.1.

Os métodos de **observação** fornecem dados sobre situações em que os usuários realizam suas atividades, com ou sem apoio de sistemas interativos. Através do registro dos dados observados, esses métodos permitem identificar problemas *reais* que os usuários enfrentaram durante sua experiência de uso do sistema sendo avaliado. O avaliador pode observar os usuários em contexto ou em laboratório. A observação em contexto permite coletar uma gama mais ampla de dados mais ricos sobre a atuação dos usuários em seu ambiente de atividade (veja estudos de campo na Seção 5.5.6). Já a observação em laboratório costuma ser mais direcionada e simples, pois o avaliador tem controle sobre o ambiente. Alguns métodos de avaliação através de observação em laboratório, também denominados de métodos empíricos, são descritos na Seção 10.2.

Métodos de avaliação por inspeção costumam ser mais rápidos e de custo de execução mais baixo do que os métodos de investigação e de observação, pois eles não gastam tempo com recrutamento e sessões de coleta de opiniões ou de observação de usuários. Por exemplo, Salgado *et al.* (2006) relatam que uma avaliação por inspeção (avaliação heurística) gastou menos da metade do tempo do que uma avaliação com a participação dos usuários (avaliação de comunicabilidade). Entretanto, os resultados de uma avaliação por inspeção são baseados apenas na experiência do avaliador, com base em hipóteses sobre os usuários. Mesmo que o avaliador se coloque no lugar do usuário, ele *não é* o usuário. Não podemos ignorar os limites dessa empatia do avaliador. Apesar de ser necessário mais tempo para coletar e analisar dados empíricos de experiências de uso, os métodos de avaliação através de investigação e observação costumam fornecer resultados mais interessantes do que as previsões dos avaliadores. Isso se deve ao fato de que os usuários percorrem caminhos não previstos pelo

avaliador, de forma criativa e oportunista, proporcionando maior realidade, riqueza e diversidade nas experiências de uso analisadas.

Os objetivos da avaliação, detalhados por questões específicas, são os guias principais para o avaliador escolher os métodos de avaliação a serem utilizados. Se o objetivo da avaliação for encontrar problemas de IHC, o avaliador pode julgar mais adequado empregar um método por inspeção para cobrir (quase) toda a interface, e selecionar um pequeno número de partes importantes a serem avaliadas por um método de observação ou de investigação. Geralmente, ele selecionaria as partes cuja inspeção não forneceu resultados suficientemente confiáveis. Já avaliar a forma como os usuários se apropriam de tecnologia requer o emprego de um método de avaliação através de investigação ou de observação, por contar com a participação dos usuários. E, para avaliar a conformidade com um padrão, é mais adequado empregar um método de avaliação por inspeção, pois a participação dos usuários é desnecessária.

Além disso, o avaliador precisa escolher métodos de avaliação adequados ao prazo, orçamento, equipamentos, número de usuários acessíveis, número de avaliadores capacitados e experientes em cada método e demais recursos disponíveis. Ao planejar a avaliação, ele tem de verificar se poderá contar com a participação dos usuários para coletar e registrar dados sobre experiências de uso, ou se deverá inspecionar a interface para tentar prevê-las. Essas questões práticas são responsáveis por viabilizar a execução da avaliação de IHC planejada. Não adianta o avaliador planejar uma avaliação de IHC sem ter condições de executá-la. Por exemplo, o avaliador deve verificar se é possível ter acesso ao contexto de uso com o tempo e a liberdade necessários, para que possa observá-lo adequadamente e coletar e registrar livremente os dados necessários.

9.7 Como Avaliar?

Os métodos de avaliação de IHC possuem as seguintes atividades básicas: preparação, coleta de dados, interpretação, consolidação e relato dos resultados. Caso a avaliação encontre problemas ou oportunidades de melhoria, também é planejado um reprojeto do sistema.

9.7.1 Por Onde Começar?

Como primeiro passo para preparar uma avaliação, o avaliador deve **aprender** sobre a situação atual, que inclui o domínio do problema, os papéis e perfis dos usuários, seus objetivos e atividades, e o contexto em que o sistema é ou será utilizado. O avaliador também deve conhecer as interfaces dos sistemas complementares ou semelhantes com os quais os usuários estejam acostumados a utilizar, além de, é claro, a interfa-

ce do próprio sistema ou protótipo a ser avaliado. Sempre que possível, o avaliador deve buscar saber quais são os comportamentos e as dificuldades típicos dos usuários durante o uso de sistemas interativos semelhantes. O Capítulo 5 apresenta algumas técnicas de análise que podem auxiliar os avaliadores nesse aprendizado.

Além de necessário para planejar a avaliação adequadamente, esse entendimento contribui para a coleta e análise dos dados. No caso de uma avaliação por inspeção, ajuda os avaliadores a se colocarem no lugar dos usuários ao tentarem prever problemas na interface e na interação. No caso de uma avaliação através de investigação ou de observação, ajuda os avaliadores a compreenderem certos dados fornecidos pelos usuários.

9.7.2 Preparação

Apesar de alguns equivocadamente considerarem-na burocrática, a atividade de preparação é fundamental para a condução adequada de uma avaliação que forneça resultados úteis e confiáveis. Ela não pode ser negligenciada, pois pode acarretar em desperdício de tempo, dinheiro e outros recursos, envolvendo avaliadores, usuários e demais interessados na avaliação.

Os **objetivos** da avaliação são definidos com base em requisições, reclamações ou comportamentos dos *stakeholders* do sistema. Se o avaliador conhece o que será avaliado, ele tem melhores condições de compreender as motivações dos interessados e de ajudá-los a definir adequadamente os objetivos da avaliação. Os objetivos devem ser detalhados através de **questões mais específicas** que a avaliação deverá responder, conforme apresentado na Seção 9.2.

Raramente avaliamos o sistema inteiro. Em vez disso, precisamos definir o **escopo** da avaliação, delimitando quais partes da interface, caminhos de interação, tarefas e perfis de usuário devem fazer parte da avaliação. Essa delimitação é feita de acordo com os objetivos e as questões que a avaliação pretende responder. Na escolha das tarefas a serem investigadas, o avaliador pode considerar as tarefas mais importantes para os usuários, as que apresentam mais problemas durante sua realização usando a tecnologia atual, as que foram mais difíceis de projetar, as que motivaram a produção dos sistema ou as que são o diferencial do sistema com relação a sistemas semelhantes ou complementares. Além disso, o avaliador deve considerar o prazo e os recursos disponíveis. Uma sessão de teste com usuários, por exemplo, costuma durar em torno de uma hora.

O avaliador **escolhe um ou mais métodos** de acordo com os objetivos da avaliação, dos recursos disponíveis e do acesso aos usuários e ao contexto de uso, conforme apresentado na Seção 9.6. Caso seja escolhido um método de avaliação que envolva

usuários, o avaliador deve também **escolher o perfil e o número de participantes**, com base nos perfis de usuários, nos objetivos e no escopo da avaliação. Por exemplo, se o objetivo for verificar como usuários novatos aprendem a realizar determinadas tarefas utilizando o sistema, o avaliador deve recrutar usuários inexperientes no uso do sistema e na realização das tarefas em questão.

Os avaliadores devem buscar participantes que representem o público-alvo do sistema avaliado, ou seja, que possuam características semelhantes aos usuários típicos. A definição dos perfis de participantes pode considerar fatores como: idade, sexo, formação acadêmica, grau de conhecimento sobre o domínio, nível de experiência na realização das tarefas e nível de experiência no uso do sistema avaliado e de sistemas semelhantes, por exemplo. Sempre que possível e pertinente, o avaliador deve buscar equilibrar o número de homens e mulheres.

Dumas e Redish (1999) relatam que uma avaliação de IHC em geral envolve de cinco a 12 usuários. Nielsen (2000), por sua vez, afirma que bastam cinco usuários para encontrarmos a maioria dos problemas na interface (85%, segundo o seu experimento), alcançando uma boa relação custo–benefício. Caso seja necessário obter resultados estatisticamente significativos, a amostra de usuários deve ser suficientemente grande e representativa. Entretanto, o tempo e outros recursos necessários para a coleta e análise de dados de muitos usuários pode inviabilizar uma abordagem estatística. Por isso, uma avaliação de IHC com frequência pretende apenas obter indícios sobre a qualidade de uso do sistema e sobre como aumentá-la. Em outras palavras, mesmo quando os resultados não são estatisticamente significativos, eles podem ser úteis para o reprojeto do sistema avaliado.

O planejamento de uma avaliação envolve **questões práticas**, que incluem alojar pessoal, recursos e equipamentos e preparar o material de apoio. Uma avaliação com usuários requer também a preparação do ambiente de teste, a realização de um teste-piloto e o recrutamento dos participantes. Ela envolve ainda **questões éticas**, apresentadas na Seção 5.4.

De acordo com os métodos de avaliação escolhidos, o avaliador deve **alocar pessoal, recursos e equipamentos** necessários. Pode ser preciso alojar outros avaliadores que auxiliem na coleta, análise e divulgação dos resultados. Caso os avaliadores tenham pouca experiência com o método selecionado, eles podem precisar de treinamento. Equipamentos para o registro de dados, como câmeras, máquinas fotográficas e gravadores de áudio são comumente utilizados na coleta de dados, e equipamentos mais sofisticados podem requerer a contratação de profissionais especializados. Diversos softwares podem apoiar as diferentes atividades de avaliação, desde a captura de dados até a análise e divulgação dos resultados.

Antes de começar a coletar dados, o avaliador deve **preparar e imprimir o material de apoio** necessário. No caso de avaliações que envolvam participantes, esse material costuma incluir:

- termo de consentimento, de acordo com os cuidados éticos necessários (veja Seção 5.4);
- questionário pré-teste (ou roteiro de entrevista estruturada) para coletar informações dos participantes que podem influenciar a interação usuário-sistema, tais como: características pessoais, experiências anteriores com tecnologia e conhecimento sobre o domínio;
- roteiro de entrevista pós-teste para coletar informações sobre a opinião e os sentimentos do participante decorrentes da experiência de uso observada;
- instruções e cenários para orientar os participantes sobre as tarefas a serem realizadas;
- roteiro de acompanhamento da observação, de modo a facilitar a captura de dados e anotações.

Para assegurar a validade do estudo, é importante que todos os participantes recebam o mesmo material, as mesmas informações e orientações.

Quando for elaborar os cenários de tarefa, o avaliador deve estar atento ao tempo necessário para os participantes realizarem as tarefas descritas. O tempo estimado para um participante realizar cada tarefa não deveria ultrapassar 20 minutos. Se for necessário mais tempo, o participante deve poder fazer uma pausa rápida para evitar que sua fadiga física ou mental interfira no resultado da avaliação. Podemos sugerir, por exemplo, que o participante desvie o olhar para um ponto distante depois de passar algum tempo olhando somente para a tela, ou faça um alongamento das mãos e dos braços depois de usar o teclado e o mouse durante algum tempo. Cada participante normalmente despende cerca de uma hora numa sessão de avaliação. Mais do que isso seria cansativo para ele (Sharp *et al.*, 2007). Caso o avaliador tenha dificuldade de planejar tarefas curtas, ele pode revisar o escopo da avaliação.

O formulário de acompanhamento das sessões de observação facilita a anotação de dados importantes para análise da interação, tais como: falas espontâneas do usuário durante o uso, se o usuário conseguiu ou não concluir uma tarefa, se foi necessário ajudá-lo em uma tarefa para prosseguir com a tarefa seguinte etc. Além de observar e registrar dados durante uma sessão de interação, o avaliador pode usar entrevistas curtas antes e depois da execução das tarefas para coletar outros dados importantes.

O avaliador deve **preparar todo ambiente, hardware e software** necessário para o uso do sistema avaliado e a captura de dados. Todas as instalações, configurações

e demais procedimentos de preparação para a sessão de avaliação devem ser concluídos antes de receber cada participante. Devemos configurar e testar câmeras de vídeo, gravadores de áudio, monitores-clone para o registro e acompanhamento das ações do participante e softwares que capturem diversos dados como, por exemplo, um filme da interação do usuário com o sistema, as teclas digitadas e os movimentos e as ações com o mouse. O laboratório deve estar limpo e arrumado para receber os participantes. Quando os participantes forem utilizar o sistema, ele deve estar funcionando perfeitamente.

Concluído todo o planejamento da avaliação, é muito importante que o avaliador **realize um teste-piloto**. O objetivo desse teste é avaliar o próprio planejamento, e analisar se a avaliação, tal como planejada, produz os dados necessários para responder a questões e objetivos do estudo. O avaliador deve conduzir o teste-piloto como se fosse uma sessão normal de avaliação. Dessa forma, ele tem oportunidade de verificar se a linguagem nas explicações e nos materiais fornecidos é clara e objetiva, e se esses materiais contêm informações adequadas e suficientes para orientar o participante durante a avaliação. O teste-piloto também permite ao avaliador verificar se o sistema a ser avaliado, os equipamentos e os softwares para registrar os dados estão funcionando corretamente, e se o tempo necessário para executar as tarefas solicitadas é adequado. Mas o teste-piloto não se limita à atividade de coleta de dados. O avaliador deve fazer uma breve análise dos dados coletados, para se certificar de que eles permitirão responder as questões de avaliação. A escolha de quais dados devem ser coletados deve ser cuidadosa. Devemos registrar apenas dados que possam contribuir com os objetivos da avaliação. Além do desperdício de recursos, registrar dados desnecessários pode tornar a sessão de uso mais artificial e constrangedora para os participantes. Por exemplo, gravar a imagem dos participantes em vídeo pode intimidá-los desnecessariamente, caso o vídeo não seja analisado.

Se algum problema for detectado no teste-piloto, o planejamento da avaliação e o material de apoio devem ser corrigidos. Caso sejam feitas muitas mudanças ou o avaliador sinta necessidade, outros testes-pilotos podem ser executados, até que tudo esteja pronto para a realização da avaliação definitiva. Os dados do teste-piloto devem ser descartados, pois podem estar contaminados por problemas que ocorreram durante o piloto ou não serem compatíveis ou válidos considerando o planejamento revisado da avaliação.

Caso haja um número reduzido de pessoas com o perfil desejado disponíveis para participar dos testes, o avaliador pode convidar para o teste-piloto uma pessoa com perfil um pouco diferente do desejado. Para o teste-piloto, não há muito problema em convidar um colega de trabalho, um amigo, ou uma pessoa mais próxima

do avaliador, pois os dados coletados serão descartados. Entretanto, esse participante não deve ter se envolvido no planejamento e na preparação da avaliação.

Com o planejamento da avaliação concluído, o avaliador recruta os participantes para as sessões de avaliação. Para **recrutar participantes** com os perfis especificados, o avaliador pode utilizar questionários ou entrevistas curtas a fim de conferir se uma pessoa possui o perfil desejado. Ao convidar os participantes selecionados, o avaliador deve esclarecer brevemente quais são os objetivos da avaliação, como e onde ela será realizada, e quanto tempo será requerido do participante. Desde o primeiro contato com o participante, o avaliador deve estar aberto e disposto a esclarecer suas eventuais dúvidas sobre a avaliação. Confirmada a disponibilidade e o interesse do participante, o avaliador deve agendar com ele a data, o horário e o local para a realização da avaliação.

Sempre que possível, o avaliador deve evitar selecionar seus conhecidos, amigos, alunos, subordinados ou pessoas com as quais possua alguma relação próxima, para evitar que a relação entre ele e o participante influencie ou contamine os dados coletados. Por exemplo, um amigo e um aluno podem não relatar problemas que perceberam na interface por se sentirem criticando o trabalho do avaliador amigo ou professor. De qualquer modo, o avaliador pode atenuar a interferência das relações entre eles esclarecendo que o objetivo da avaliação é encontrar problemas na interface e que a opinião dos participantes é importante para melhorá-la.

9.7.3 Coleta de Dados

A coleta de dados deve ocorrer conforme o planejamento realizado e o método de avaliação selecionado. No caso de métodos de avaliação por inspeção, essa atividade envolve apenas os avaliadores, que utilizam o material preparado e seguem o procedimento prescrito pelo método selecionado. Os avaliadores examinam a interface para identificar discrepâncias com um padrão ou para tentar *prever* as experiências de uso que os usuários terão com a solução de IHC avaliada (veja Seção 10.1).

Em avaliações por investigação e por observação, essa atividade costuma ter por objetivo registrar as experiências vivenciadas pelos usuários durante a interação com o sistema ou protótipo sendo avaliado. Os métodos empíricos contam com a participação dos usuários para *relatar* experiências de uso vivenciadas ou *permitir a observação* de experiências reais de uso com a solução de IHC avaliada.

Ao receber os participantes, o avaliador deve ser cordial e deixá-los bem à vontade. Ele costuma estabelecer uma conversa de aquecimento ou “quebra-gelo” com os participantes, sobre tópicos gerais como: o clima, notícias recentes e como foi o trajeto até o local da avaliação. Se a avaliação ocorrer em laboratório, o avaliador deve

apresentar todo o ambiente de teste aos participantes, inclusive a sala de observação. Sempre que possível, o avaliador deve oferecer água, café, oportunidade para ir ao toalete ou algo mais de que os participantes precisem. Tudo isso tem por objetivo dar oportunidade e tempo para o participante se acostumar com o ambiente, reduzindo sua ansiedade antes de iniciar a sessão de coleta de dados.

Depois dessa conversa inicial, o avaliador deve explicar ao participante os objetivos do estudo, o sistema de interesse, o procedimento da avaliação (por exemplo, apresentar uma visão geral das atividades que o participante é convidado a realizar) e os cuidados éticos sendo tomados (veja Seção 5.4). Se surgir alguma dúvida, ela deve ser sanada imediatamente pelo avaliador. Depois, o avaliador entrega o termo de consentimento em duas vias, já assinadas por ele próprio, e aguarda enquanto o participante lê e assina o termo, caso aceite participar da avaliação. Uma via do termo de consentimento fica com o avaliador e a outra com o participante.

Em seguida, o participante responde o questionário pré-teste. O avaliador pode tornar esse momento mais natural se ele ler as perguntas do questionário para o participante e registrar as respostas fornecidas, como numa entrevista estruturada. Isso pode ajudar o participante a se sentir mais à vontade durante a avaliação.

Respondido o questionário, é iniciada a sessão de observação. Nesse momento, o software e os equipamentos que registram os dados da interação devem ser ativados. O avaliador apresenta o sistema a ser avaliado e pode permitir que o participante explore-o livremente por alguns minutos, caso seja o seu primeiro contato com o sistema. Depois da exploração livre, o avaliador entrega ao participante as instruções e os cenários das tarefas a serem realizadas. Esclarecidas as eventuais dúvidas, o participante passa a realizar as tarefas solicitadas.

É recomendável que pelo menos dois avaliadores trabalhem na coleta de dados: um para acompanhar o participante mais de perto na sala de uso do sistema e outro mais distante da sala de observação. Enquanto o participante utiliza o sistema, os avaliadores devem anotar qualquer acontecimento que possa ser relevante para a interpretação dos dados coletados e sobre eventuais dúvidas a serem esclarecidas na entrevista pós-teste. Eles devem evitar interferir na atividade do usuário. O participante não deve ser interrompido nem questionado enquanto realiza as tarefas solicitadas.

Se for adequado, o avaliador pode pedir que o participante relate em voz alta o que ele está pensando e fazendo durante a interação. Essa técnica é conhecida como *think aloud* (Ericsson e Simon, 1993). Apesar de ser uma técnica útil e relativamente simples para o avaliador ter acesso ao que se passa na mente do participante, ela depende muito do participante. Certas pessoas podem se distrair e parar de falar enquanto realizam alguma atividade. Outras podem gastar mais tempo e esforço

relatando o que estão pensando e fazendo, do que realmente executando as tarefas solicitadas. Se não for aplicada de forma adequada, essa técnica pode interferir nos resultados da avaliação, como, por exemplo, no número de erros cometidos, pois o usuário pode pensar mais antes de fazer, ou no tempo de execução das tarefas, pois o participante pode gastar mais tempo pensando e falando. Portanto, a técnica de *think aloud* deve ser utilizada cuidadosamente.

Caso o participante já tenha gasto muito tempo interagindo sem conseguir concluir uma tarefa, demonstre não ter condições de concluir a tarefa, ou passe por uma situação de constrangimento ou emoções desagradáveis, os avaliadores podem intervir na interação, sugerindo ao participante que passe para a próxima tarefa solicitada, ou até mesmo abandone as tarefas restantes. Na transição entre tarefas, se for o caso, o avaliador pode interagir com o sistema a fim de prepará-lo para a execução da próxima tarefa. Além disso, o participante pode pedir para fazer uma pausa, sempre que quiser.

Terminada a sessão de interação, os avaliadores devem conduzir a entrevista pós-teste. Essa entrevista é uma oportunidade para coletar a opinião do participante sobre a experiência de uso que acabou de vivenciar e esclarecer eventuais dúvidas sobre seu comportamento e suas intenções, percepções e interpretações durante a execução das tarefas.

9.7.4 Interpretação

Na atividade de **interpretação**, o avaliador analisa o material registrado para atribuir significado aos dados coletados. A interpretação do avaliador deve ser orientada pelo método de avaliação selecionado e pelo que foi definido durante a atividade de preparação da avaliação. Cada método de avaliação costuma apontar os focos de análise (i.e., quais dados devem ser analisados sob quais perspectivas de análise) e os tipos de interpretações mais frequentes. Por exemplo, o método de avaliação heurística enfatiza a análise de um conjunto de heurísticas, enquanto o método de avaliação de comunicabilidade investiga problemas na recepção da metamensagem associados a etiquetas de rupturas de comunicação. Em geral, o avaliador se concentra inicialmente na interpretação dos dados coletados de cada participante individualmente, buscando respostas às questões específicas definidas no planejamento da avaliação. Esse tipo de análise também é conhecido como análise *intrassujeito* ou *intraparticipante* (Nicolaci-da-Costa, 1994; Nicolaci-da-Costa *et al.*, 2004).

A interpretação ou análise dos dados coletados pode ser feita de forma *automática* ou *manual*, dependendo do tipo de cada dado. Alguns aspectos de uma solução de IHC podem ser avaliados automaticamente por um programa computacional, de

forma mais rápida e sistemática do que um ser humano poderia realizar manualmente. A análise automática dos dados tem sido utilizada para avaliar a navegação por sites na Web, verificando, por exemplo, a existência de “links quebrados” e alguns critérios de acessibilidade (e.g., imagens que não possuem descrição textual associada). O DaSilva¹ é um dos primeiros programas brasileiros que avaliam a acessibilidade de Web sites. Ele analisa o código HTML do endereço indicado para detectar partes da interface que violem regras de acessibilidade.

Também é possível verificar automaticamente se o uso do sistema avaliado ocorre conforme o esperado. Para isso, na etapa de preparação o avaliador representa em algum modelo os caminhos de interação oferecidos pelo designer para o usuário alcançar seus objetivos (por exemplo, através de um modelo de tarefas ou de interação). Na etapa de coleta de dados, um programa monitora e registra a interação do usuário com o sistema enquanto ele tenta atingir os objetivos desejados. Desse modo, o avaliador pode comparar automaticamente o que ele esperava que ocorresse e o que de fato ocorre durante a interação. Dentre outros resultados, essa comparação permite calcular quantas vezes os usuários conseguiram ou não alcançar seus objetivos da forma prevista pelo designer, bem como identificar em que partes da interação ocorreram problemas que dificultaram ou impediram os usuários de alcançarem seus objetivos (Lecerof e Paternò, 1998). Essa é uma forma de avaliação baseada em modelos e registros de interação.

Os critérios de qualidade de uso que podem ser analisados automaticamente por programas computacionais ainda são muito limitados, pois exigem uma definição *a priori* de regras definindo o que é “bom” e “ruim”, e o que é “certo” e “errado”. A análise realizada por um avaliador humano ainda é fundamental para verificar a qualidade de uso, porque é difícil codificar num programa toda a visão que o avaliador pode adquirir sobre domínio, usuário, atividades e contexto, bem como sua capacidade de análise, principalmente diante de situações imprevistas. No entanto, para aspectos pontuais da avaliação, as análises automáticas podem agilizar parte do processo.

9.7.5 Consolidação e Relato dos Resultados

Uma vez concluída a interpretação individual dos dados coletados, seja das previsões dos avaliadores ou das observações das experiências de uso dos participantes, os resultados individuais são **consolidados** e analisados em conjunto, em uma análise denominada de *intersujeito* ou *interparticipante* (Nicolaci-da-Costa, 1994; Nicolaci-da-Costa *et al.*, 2004). Nessa atividade, os avaliadores buscam recorrências nos resultados de acordo com o método selecionado. As **recorrências** são importantes porque,

1 <http://www.dasilva.org.br/>.

ao expressarem resultados comuns a vários participantes de um grupo, permitem fazer uma distinção entre características representativas do grupo e as idiossincrasias de participantes individuais.

Na consolidação dos resultados, os avaliadores devem novamente endereçar as questões que motivaram o estudo, buscando respondê-las ou justificar por que alguma resposta não foi encontrada. Mesmo no caso de avaliações empíricas, a **generalização** dos resultados exige muito cuidado. Eles sempre são fortemente influenciados pelo ambiente de avaliação e pelas características, preferências, interesses e necessidades dos participantes individuais.

Os métodos de avaliação podem apresentar diferentes resultados. Por exemplo, quando o objetivo é identificar problemas em soluções de IHC, diferentes métodos podem revelar tipos diversos de problemas que prejudiquem a qualidade de uso, seja por causa de diferenças na coleta de dados ou na sua análise. Outros fatores também influenciam os resultados da avaliação: o conhecimento e a experiência dos avaliadores, o tempo disponível para a avaliação, a quantidade e qualidade dos dados disponíveis, e assim por diante.

Os resultados de uma avaliação de IHC normalmente indicam tendências de problemas, e não uma certeza de que eles vão ocorrer durante o uso do sistema. De modo semelhante, caso não sejam encontrados problemas durante a avaliação, também não podemos afirmar categoricamente que o sistema tenha alta qualidade de uso, apenas que um estudo não revelou problemas num determinado escopo do sistema que foi avaliado com base em um certo planejamento.

Finalmente, os avaliadores devem **relatar os resultados** consolidados, que costumam incluir:

- os objetivos e escopo da avaliação;
- a forma como a avaliação foi realizada (método de avaliação empregado);
- o número e o perfil de usuários e avaliadores que participaram da avaliação;
- um sumário dos dados coletados, incluindo tabelas e gráficos;
- um relato da interpretação e análise dos dados;
- uma lista dos problemas encontrados;
- um planejamento para o reprojeto do sistema.

9.8 O Framework DECIDE

Sharp, Rogers e Preece (2007) propõem um *framework* chamado DECIDE para orientar o planejamento, a execução e a análise de uma avaliação de IHC. As atividades do

framework são interligadas e executadas iterativamente, à medida que o avaliador articula os objetivos da avaliação, os dados e recursos disponíveis. Então, quando o avaliador descobre uma necessidade de modificar os rumos da avaliação por algum motivo, as demais atividades são afetadas. Por exemplo, se o avaliador não conseguir permissão para visitar o ambiente de uso de um sistema, ele não pode aplicar um método de avaliação que coleta dados sobre o uso do sistema em contexto. Nesse caso, provavelmente seus objetivos precisarão ser revistos. As atividades do *framework* DECIDE são descritas a seguir.

D	Determinar os objetivos da avaliação de IHC. O avaliador deve determinar os objetivos gerais da avaliação e identificar por que e para quem tais objetivos são importantes. O restante do planejamento da avaliação, sua execução e a apresentação dos resultados serão orientados por esses objetivos.
E	Explorar perguntas a serem respondidas com a avaliação. Para cada objetivo definido, o avaliador deve elaborar perguntas específicas a serem respondidas durante avaliação. Essas perguntas são responsáveis por operacionalizar a investigação e o julgamento de valor a serem realizados. Elas devem considerar o perfil dos usuários-alvo e suas atividades.
C	Escolher (<i>Choose</i>) os métodos de avaliação a serem utilizados. O avaliador deve escolher os métodos mais adequados para responder as perguntas e atingir os objetivos esperados, considerando também o prazo, o orçamento, os equipamentos disponíveis e o grau de conhecimento e experiência dos avaliadores.
I	Identificar e administrar as questões práticas da avaliação. Existem muitas questões práticas envolvidas numa avaliação de IHC, como, por exemplo, o recrutamento dos usuários que participarão da avaliação, a preparação e o uso dos equipamentos necessários, os prazos e o orçamento disponíveis, além da mão-de-obra necessária para conduzir a avaliação.
D	Decidir como lidar com as questões éticas. Sempre que usuários são envolvidos numa avaliação, o avaliador deve tomar os cuidados éticos necessários (veja Seção 5.4). Os participantes da avaliação devem ser respeitados e não podem ser prejudicados diretamente ou indiretamente, nem durante os experimentos, nem após a divulgação dos resultados da avaliação.
E	Avaliar (<i>Evaluate</i>), interpretar e apresentar os dados. O avaliador precisa estar atento a alguns aspectos da avaliação realizada antes de tirar conclusões e divulgar resultados. Ele deve considerar: o grau de <i>confiabilidade</i> dos dados (i.e., semelhança dos resultados obtidos quando emprega mais de uma vez o mesmo método de avaliação nas mesmas circunstâncias; a <i>validade interna</i> do estudo (i.e., se o método de avaliação mede o que deveria medir, se o faz com rigor e evita que os dados sejam distorcidos); a <i>validade externa</i> do estudo (i.e., até que ponto os resultados podem ser generalizados ou transferidos a um outro contexto semelhante); e a <i>validade ecológica</i> do estudo (i.e., o quanto os materiais, métodos e ambiente de estudo se assemelham à situação real investigada).

Atividades

1. *Por que avaliar o uso de software?* Imagine que você seja apresentado a um produtor de software que lhe conta sobre seus planos para produzir uma nova versão de um sistema comercial. Quais argumentos você elaboraria para convencê-lo a realizar uma avaliação de IHC?
2. *Planejamento da avaliação de IHC.* Escolha um software de sua preferência e defina dois objetivos de avaliação (por exemplo, dois objetivos citados na Tabela 9.1). Planeje duas avaliações de IHC do software escolhido, uma para cada objetivo definido. Em cada planejamento, realize cada tarefa de preparação da avaliação e relate sua execução. Compare os dois planejamentos.

10

Métodos de Avaliação de IHC

Objetivos do Capítulo

- Apresentar os métodos de avaliação de IHC por inspeção: avaliação heurística, percurso cognitivo e inspeção semiótica.
 - Apresentar os métodos de avaliação de IHC por observação: teste de usabilidade, avaliação de comunicabilidade e prototipação em papel.
 - Comparar os métodos de avaliação de acordo com o que é avaliado, quando a avaliação é realizada, e qual tipo de resultado é produzido.
-

Este capítulo descreve os seguintes métodos de avaliação de IHC: avaliação heurística (Nielsen, 1994a), percurso cognitivo (Wharton *et al.*, 2004), inspeção semiótica (de Souza *et al.*, 2006; Prates e Barbosa, 2007), teste de usabilidade (Rubin, 1994), avaliação de comunicabilidade (Prates *et al.*, 2000a; de Souza, 2005a; Prates e Barbosa, 2007) e prototipação em papel (Snyder, 2003). Esses métodos são comparados na Seção 10.3, de acordo com o que é avaliado, quando a avaliação é realizada, e qual tipo de resultado é produzido.

10.1 Avaliação de IHC através de Inspeção

Como apresentado na Seção 9.6, os métodos de inspeção permitem ao avaliador examinar (ou inspecionar) uma solução de IHC para tentar **antever** as possíveis consequências de certas decisões de design. Esses métodos não envolvem diretamente os usuários, portanto, tratam de experiências de uso **potenciais**, e não reais. Ao inspecionar uma interface, os avaliadores tentam se colocar no lugar de um usuário com determinado perfil, com um certo conhecimento e experiência em algumas atividades, para então tentar identificar problemas que os usuários podem vir a ter quando interagirem com o sistema, e quais formas de apoio o sistema oferece para ajudá-los a contornarem esses problemas. Nas próximas subseções apresentamos três métodos de avaliação por inspeção: a avaliação heurística, o percurso cognitivo e a inspeção semiótica.

10.1.1 Avaliação Heurística

A **avaliação heurística** é um método de avaliação de IHC criado para encontrar problemas de usabilidade durante um processo de design iterativo (Nielsen e Molich, 1990; Nielsen, 1993; Nielsen, 1994a). Esse método de avaliação orienta os avaliadores a inspecionar sistematicamente a interface em busca de problemas que prejudiquem a usabilidade. Por ser um método de inspeção, a avaliação heurística foi proposta como uma alternativa de avaliação rápida e de baixo custo, quando comparada a métodos empíricos.

A avaliação heurística tem como base um conjunto de diretrizes de usabilidade, que descrevem características desejáveis da interação e da interface, chamadas por Nielsen de heurísticas. Essas heurísticas resultam da análise de mais de 240 problemas de usabilidade realizada ao longo de vários anos por experientes especialistas em IHC (Nielsen, 1994b). Nielsen (1993) descreve um conjunto inicial de heurísticas a serem utilizadas em seu método de avaliação heurística (p. 30):

- *visibilidade do estado do sistema:* o sistema deve sempre manter os usuários informados sobre o que está acontecendo através de feedback (resposta às ações do usuário) adequado e no tempo certo;
- *correspondência entre o sistema e o mundo real:* o sistema deve utilizar palavras, expressões e conceitos que são familiares aos usuários, em vez de utilizar termos orientados ao sistema ou jargão dos desenvolvedores. O designer deve seguir as convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica, conforme esperado pelos usuários;
- *controle e liberdade do usuário:* os usuários frequentemente realizam ações equivocadas no sistema e precisam de uma “saída de emergência” claramente marcada para sair do estado indesejado sem ter de percorrer um diálogo extenso. A interface deve permitir que o usuário desfaça e refaça suas ações;
- *consistência e padronização:* os usuários não devem ter de se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. O designer deve seguir as convenções da plataforma ou do ambiente computacional;
- *reconhecimento em vez de memorização:* o designer deve tornar os objetos, as ações e opções visíveis. O usuário não deve ter de se lembrar para que serve um elemento de interface cujo símbolo não é reconhecido diretamente; nem deve ter de se lembrar de informação de uma parte da aplicação quando tiver passado para uma outra parte dela. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário;
- *flexibilidade e eficiência de uso:* aceleradores — imperceptíveis aos usuários novatos — podem tornar a interação do usuário mais rápida e eficiente, permitindo que o sistema consiga servir igualmente bem os usuários experientes e inexperientes. Exemplos de aceleradores são botões de comando em barras de ferramentas ou teclas de atalho para acionar itens de menu ou botões de comando. Além disso, o designer pode oferecer mecanismos para os usuários customizarem ações frequentes;
- *projeto estético e minimalista:* a interface não deve conter informação que seja irrelevante ou raramente necessária. Cada unidade extra de informação em uma interface reduz sua visibilidade relativa, pois compete com as demais unidades de informação pela atenção do usuário;
- *prevenção de erros:* melhor do que uma boa mensagem de erro é um projeto cuidadoso que evite que um problema ocorra, caso isso seja possível;
- *ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros:* as mensagens de erro devem ser expressas em linguagem simples (sem códigos indecifráveis), indicar precisamente o problema e sugerir uma solução de forma construtiva;

- *ajuda e documentação:* embora seja melhor que um sistema possa ser utilizado sem documentação, é necessário oferecer ajuda e documentação de alta qualidade. Tais informações devem ser facilmente encontradas, focadas na tarefa do usuário, enumerar passos concretos a serem realizados e não ser muito extensas.

Esse é um conjunto inicial, que pode ser expandido para incluir novas diretrizes conforme os avaliadores julgarem necessário. Por exemplo, há diretrizes específicas para certos estilos de interação (e.g., Web, WIMP, manipulação direta, interfaces via voz, realidade virtual) e para certos domínios de aplicação (e.g., comércio eletrônico, sistemas colaborativos, educação a distância).

Nielsen (1992) realizou um experimento com 19 avaliadores realizando individualmente uma avaliação heurística num sistema de atendimento eletrônico. Naquele estudo, alguns problemas foram descobertos por todos os avaliadores, outros foram encontrados por um número pequeno de avaliadores, e um número substancial de problemas foi encontrado por apenas um avaliador. Com base no estudo, Nielsen recomenda que uma avaliação heurística envolva de três a cinco avaliadores. Algumas atividades devem ser realizadas por cada avaliador (individualmente), enquanto em outras eles devem trabalhar em conjunto. A Tabela 10.1 apresenta as atividades envolvidas em uma avaliação heurística.

Tabela 10.1 Atividades do método de avaliação heurística

avaliação heurística	
atividade	tarefa
Preparação	<p><i>Todos os avaliadores:</i></p> <ul style="list-style-type: none"> ■ aprendem sobre a situação atual: usuários, domínio etc. ■ selecionam as partes da interface que devem ser avaliadas
Coleta de dados	<p><i>Cada avaliador, individualmente:</i></p> <ul style="list-style-type: none"> ■ inspeciona a interface para identificar violações das heurísticas ■ lista os problemas encontrados pela inspeção, indicando local, gravidade, justificativa e recomendações de solução
Interpretação	
Consolidação dos resultados	<p><i>Todos os avaliadores:</i></p> <ul style="list-style-type: none"> ■ revisam os problemas encontrados, julgando sua relevância, gravidade, justificativa e recomendações de solução
Relato dos resultados	<ul style="list-style-type: none"> ■ geram um relatório consolidado

Na atividade de *preparação*, os avaliadores organizam as telas do sistema ou protótipo a ser avaliado, conforme o escopo definido para a avaliação (veja Seção 9.7.2), e a lista de heurísticas ou diretrizes que devem ser consideradas. A solução de IHC avaliada pode ser o próprio sistema funcionando, bem como protótipos executáveis e não

executáveis em vários níveis de fidelidade e detalhes, inclusive protótipos desenhados em papel. Por isso, a avaliação heurística pode ser executada durante todo o processo de design de IHC, desde que haja alguma representação da interface proposta.

Em seguida, os avaliadores prosseguem com a *coleta* e a *interpretação* dos dados (veja Seções 9.7.3 e 9.7.4). Cada avaliador deve inspecionar individualmente cada tela selecionada e cada um de seus elementos, com o objetivo de identificar se as diretrizes foram respeitadas ou violadas. Cada violação de diretriz é considerada um problema potencial de IHC. O avaliador deve percorrer a interface pelo menos duas vezes: uma para ganhar uma visão de conjunto e outra para examinar cuidadosamente cada elemento de cada tela. Ele pode adotar uma estratégia de avaliação por diretriz ou por tela. No primeiro caso, o avaliador seleciona uma diretriz e percorre toda a interface avaliando-a, e, em seguida, repete o procedimento com a próxima diretriz, até esgotar o conjunto de diretrizes. No segundo caso, o avaliador seleciona uma tela, avalia-a considerando todas as diretrizes e, em seguida, repete o procedimento com a próxima tela, até percorrer toda a interface. Também é possível combinar essas duas estratégias de inspeção.

Para cada problema identificado, o avaliador deve anotar: qual diretriz foi violada, em que local o problema foi encontrado (em que tela e envolvendo quais elementos de interface), qual a gravidade do problema e uma justificativa de por que aquilo é um problema. Também é interessante anotar ideias de soluções alternativas que possam resolver os problemas encontrados.

O **local** em que cada problema foi encontrado indica quais partes da interface devem ser modificadas. O problema pode ser *pontual*, em um único local na interface; *ocasional*, em dois ou mais locais na interface, casualmente; ou *sistemático*, na estrutura geral da interface. Além disso, o problema pode ser causado pela *ausência* de algum elemento na interface, que deveria ser incluído.

Cada avaliador deve julgar a **severidade** (ou gravidade) dos problemas encontrados, para facilitar a análise de custo/benefício da correção dos problemas e priorização dos esforços de correção ou reprojeto. Segundo Nielsen (1994a), o julgamento da severidade de um problema de usabilidade envolve três fatores:

- a *frequência* com que o problema ocorre: é um problema comum ou raro?
- o *impacto* do problema, se ocorrer: será fácil ou difícil para os usuários superarem o problema?
- a *persistência* do problema: o problema ocorre apenas uma vez e será superado pelos usuários, ou atrapalhará os usuários repetidas vezes?

Para facilitar a compreensão e comparação do julgamento dos problemas encontrados, Nielsen (1994a) sugere a seguinte escala de severidade:

- 1: *problema cosmético* – não precisa ser consertado a menos que haja tempo no cronograma do projeto;
- 2: *problema pequeno* – o conserto deste problema pode receber baixa prioridade;
- 3: *problema grande* – importante de ser consertado e deve receber alta prioridade. Esse tipo de problema prejudica fatores de usabilidade tidos como importantes para o projeto (por exemplo, são exigidos muitos passos de interação para alcançar um objetivo que deveria ser atingido de forma eficiente);
- 4: *problema catastrófico* – é extremamente importante consertá-lo antes de se lançar o produto. Se mantido, o problema provavelmente impedirá que o usuário realize suas tarefas e alcance seus objetivos.

Uma sessão de inspeção da interface na avaliação heurística costuma durar em torno de uma ou duas horas. Caso a interface seja muito complexa, podemos realizar mais de uma sessão de inspeção para diferentes partes da interface, mas não devemos realizar sessões longas, pois o desempenho do avaliador diminui muito com o passar do tempo, e ele deixa de produzir dados de qualidade.

Depois que todas as inspeções individuais tenham sido realizadas, os avaliadores devem se reunir para *consolidar os resultados* (veja Seção 9.7.5). Nessa atividade, cada avaliador compartilha sua lista de problemas com os demais avaliadores, para que todos adquiram uma visão abrangente dos problemas encontrados na interface avaliada. Em seguida, eles realizam um novo julgamento, no qual cada avaliador pode atribuir um novo grau de severidade para cada problema. Caso um avaliador discorde de que algum item seja de fato um problema, pode atribuir a ele um grau de severidade zero. Considerando os novos julgamentos, os avaliadores conversam e entram em acordo sobre o grau de severidade final de cada problema e decidem quais problemas e sugestões de solução devem fazer parte do relatório consolidado.

Depois que a equipe de avaliadores adquire uma visão mais abrangente, algumas vezes é necessário unir problemas encontrados por diferentes avaliadores ou até pelo mesmo avaliador, seja porque relatam exatamente o mesmo problema ou porque relatam casos particulares ou partes de um problema maior.

O *relato dos resultados* de uma avaliação heurística geralmente contém (veja Seção 9.7.5):

- os objetivos da avaliação;
- o escopo da avaliação;

- uma breve descrição do método de avaliação heurística;
- o conjunto de diretrizes utilizado;
- o número e o perfil dos avaliadores;
- lista de problemas encontrados, indicando, para cada um:
 - local onde ocorre;
 - descrição do problema;
 - diretriz(es) violada(s);
 - severidade do problema;
 - sugestões de solução.

Exemplo 10.1 – Avaliação Heurística

Considere o seguinte fragmento de tela de login de um Web site de livraria:¹



The screenshot shows a login page with the following elements:

- Login**: The title of the form.
- Para fazer o login digite seu email ou seu CPF /CNPJ (caso tenha se cadastrado como Pessoa Física ou Jurídica, respectivamente) depois digite sua Senha e clique em Confirmar.**: A descriptive text above the input fields.
- Email:** An input field for entering an email address.
- ou CPF/CNPJ:** An input field for entering a CPF or CNPJ number.
- Senha:** An input field for entering a password.
- :Confirmar:** A button labeled "Confirmar" used for password verification.
- Cadastre-se**: A secondary action button.
- Esqueci Senha**: A secondary action button.

O trecho de relatório a seguir ilustra a descrição das violações resultante da avaliação heurística. Observe que alguns problemas constituem violação de mais de uma heurística.

- *Visibilidade do estado do sistema, prevenção de erros.* O elemento secundário *Cadastre-se* tem mais destaque do que o elemento *Confirmar*. Isso pode levar o usuário a acionar o botão errado ou se perguntar se entrou corretamente na tela de login, e até mesmo voltar para a página anterior e repetir a operação de acesso a essa página.
 - Local: abaixo do formulário, apenas nessa tela.
 - Severidade: 3 (problema grande), pois o usuário pode acreditar que precisa se cadastrar a cada compra, ou que o sistema está com defeito, e com isso pode desistir de efetuar a compra através desse site.
 - Recomendação: destacar o botão primário (*Confirmar*) e reduzir a ênfase dos botões secundários (*Cadastre-se* e *Esqueci senha*). Considere modificar os botões secundários para links, mais afastados do botão primário do formulário.
- *Controle e liberdade do usuário.* Os usuários não têm a opção, através do Web site, de voltar à página anterior. Para isso, precisam utilizar o botão de volta do próprio navegador.

1 <http://www.livrariagalileu.com.br/>.

- Local: ausência de um botão de volta em todos os formulários do site.
- Severidade: 2 (problema pequeno). O usuário está acostumado a utilizar o botão de volta do navegador em outros sites, e perceberá que pode fazer isso sem perder o que tenha feito no site (e.g., itens colocados no carrinho de compras).
- Recomendação: incluir um botão *Voltar* como botão secundário do formulário.
- *Consistência e padronização, prevenção de erros.* Os campos de preenchimento alternativo ("Email:" e "ou CPF/CNPJ:") não estão claramente marcados, como de costume, por botões de opção (*radio buttons*). Como os usuários costumam seguir dicas visuais melhor do que instruções textuais, muitos preencherão os dois campos.
 - Local: formulário de login, campos "Email:" e "ou CPF/CNPJ:".
 - Severidade: 2 (problema pequeno). Apesar de ineficiente, o preenchimento dos dois campos não impede o usuário de efetuar o login.
 - Recomendação: identificar os campos alternativos por botões de opção, que devem ser automaticamente selecionados quando o usuário inicia a digitação no campo correspondente.
- *Flexibilidade e eficiência de uso, consistência e padronização.* O usuário não tem a opção de pedir para o sistema se lembrar do seu e-mail ou mesmo manter seu login ativo, como ocorre em boa parte dos sites de comércio eletrônico.
 - Local: formulário de login, ausência de botões de seleção (*checkboxes*).
 - Severidade: 2 (problema pequeno) para usuários ocasionais; 3 (problema grande) para usuários frequentes, que provavelmente darão preferência a Web sites que se lembram "deles".
 - Recomendação: oferecer um checkbox *Lembrar dos meus dados* e/ou um checkbox *Mantar meu login ativo por 15 dias*.

10.1.2 Percurso Cognitivo

O **percurso cognitivo** (*cognitive walkthrough*) é um método de avaliação de IHC por inspeção cujo principal objetivo é avaliar a *facilidade de aprendizado* de um sistema interativo, através da exploração da sua interface (Wharton *et al.*, 1994). Esse método foi motivado pela preferência de muitas pessoas em “aprenderem fazendo”, em vez de aprenderem através de treinamentos, leitura de manuais etc. Para julgar a facilidade de aprendizado do sistema, o método considera principalmente a correspondência entre o modelo conceitual dos usuários e a imagem do sistema, no que tange à conceptualização da tarefa, ao vocabulário utilizado e à resposta do sistema a cada ação realizada.

O percurso cognitivo guia a inspeção da interface pelas tarefas do usuário. Nesse método, o avaliador percorre a interface inspecionando as ações projetadas para um usuário concluir cada tarefa utilizando o sistema. Para cada ação, o avaliador tenta se colocar no papel de um usuário e detalha como seria sua interação com o sistema naquele momento. Em um bom projeto de IHC, esperamos que a própria interface guie os usuários pela sequência de ações esperada (projetada pelo designer) para realizar

suas tarefas. Caso isso não aconteça, o método levanta hipóteses sobre as possíveis causas dos problemas encontrados e busca fornecer sugestões de reprojeto. Cabe ao avaliador formular hipóteses sobre o sucesso ou insucesso da interação a cada passo. Para isso, ele avalia o processo de interação segundo a visão da engenharia cognitiva, conforme descrito na Seção 3.4. Ele verifica se a imagem do sistema apoia as tarefas de forma compatível com o modelo conceitual que os usuários de determinado perfil possuem e o modo como realizam tais tarefas.

A Tabela 10.2 apresenta as atividades propostas pelo método de percurso cognitivo. O percurso cognitivo pode ser realizado por um ou mais avaliadores. Se houver mais de um avaliador, todos devem realizar todas as atividades em conjunto.

Tabela 10.2 Atividades do método de percurso cognitivo

percurso cognitivo	
atividade	tarefa
Preparação	<ul style="list-style-type: none"> ■ identificar os perfis de usuários ■ definir quais tarefas farão parte da avaliação ■ descrever as ações necessárias para realizar cada tarefa ■ obter uma representação da interface, executável ou não
Coleta de dados	<ul style="list-style-type: none"> ■ percorrer a interface de acordo com a sequência de ações necessárias para realizar cada tarefa
Interpretação	<ul style="list-style-type: none"> ■ para cada ação enumerada, analisar se o usuário executaria a ação corretamente, respondendo e justificando a resposta às seguintes perguntas: <ul style="list-style-type: none"> – O usuário vai tentar atingir o efeito correto? (Vai formular a intenção correta?) – O usuário vai notar que a ação correta está disponível? – O usuário vai associar a ação correta com o efeito que está tentando atingir? – Se a ação for executada corretamente, o usuário vai perceber que está progredindo na direção de concluir a tarefa? ■ relatar uma história aceitável sobre o sucesso ou falha em realizar cada ação que compõe a tarefa
Consolidação dos resultados	<ul style="list-style-type: none"> ■ sintetizar resultados sobre: <ul style="list-style-type: none"> – o que o usuário precisa saber <i>a priori</i> para realizar as tarefas – o que o usuário deve aprender enquanto realiza as tarefas – sugestões de correções para os problemas encontrados
Relato dos resultados	<ul style="list-style-type: none"> ■ gerar um relatório consolidado com os problemas encontrados e sugestões de correção

Na atividade de *preparação*, o avaliador organiza os objetos do estudo e prepara o material de apoio (veja Seção 9.7.2). Os objetos do estudo são: a lista de tarefas in-

vestigadas e a sequência das ações que, na visão do designer da solução, um usuário com o perfil especificado deveria executar para concluir a tarefa. O material de apoio inclui a lista de perguntas do método e a descrição do perfil de usuários, incluindo o seu conhecimento e experiência no domínio investigado, nas tarefas e no uso de tecnologias e sistemas semelhantes.

As tarefas a serem avaliadas podem estar representadas por um modelo de tarefas (conforme apresentado na Seção 6.4), um protótipo em papel, um protótipo funcional ou um sistema pronto. Quanto mais próxima e fiel for a representação da interface da solução final, melhores serão as condições de o avaliador prever a facilidade que o usuário terá para aprender a realizar as tarefas em questão.

Nas atividades de *coleta e interpretação dos dados* (veja Seções 9.7.3 e 9.7.4), o avaliador simula, na (representação da) interface, a execução das tarefas que fazem parte do escopo de avaliação. Para cada tarefa, o avaliador examina cada passo, analisando se e por que um usuário com o perfil especificado escolheria a ação “correta” ou perceberia que o efeito correto foi alcançado. Para a avaliação de cada passo, o avaliador responde as seguintes perguntas:

- *o usuário tentaria atingir o efeito correto?* A formulação da intenção do usuário seria a esperada (pelo designer do sistema)? Um usuário tem mais chance de formular a intenção correta se: a ação faz parte da tarefa tal como concebida pelo usuário; o usuário tem experiência em utilizar o sistema avaliado ou sistemas semelhantes; ou o sistema fornece uma instrução ou solicita que o usuário realize a ação;
- *o usuário perceberia que a ação correta está disponível?* Um usuário normalmente sabe que uma ação está disponível se: tem experiência em utilizar o sistema avaliado ou sistemas semelhantes; ou se percebe na interface uma representação da ação desejada (por exemplo, em um item de menu, link ou botão de comando);
- *o usuário conseguiria associar a ação correta com o efeito que está tentando atingir?* O usuário costuma saber qual ação é adequada para o efeito esperado se: tem experiência em utilizar o sistema avaliado ou sistemas semelhantes; se a interface comunica essa associação entre a ação e o efeito esperado; ou se nenhuma outra ação parece adequada (i.e., por eliminação);
- *se a ação correta for realizada, o usuário perceberia que está progredindo para concluir a tarefa?* O usuário geralmente sabe que está avançando na direção da conclusão da tarefa se: tem experiência em utilizar o sistema avaliado ou sistemas semelhantes; ou as respostas do sistema estão de acordo com o efeito esperado.

Considerando a tarefa, o perfil dos usuários e a interface, o avaliador deve relatar histórias de sucesso ou de insucesso ao responder essas perguntas. Todas as perguntas devem ser respondidas para cada ação. Mesmo que a resposta a uma pergunta seja negativa, isto é, indique que o usuário não conseguiria avançar, o avaliador deve, após registrar seu relato de insucesso, supor que a resposta poderia ser positiva e então prosseguir respondendo à pergunta seguinte, até que todas as perguntas tenham sido respondidas para aquela ação. Em seguida, o procedimento se repete para a próxima ação, e assim sucessivamente, até concluir a inspeção de todas as ações que compõem a tarefa sendo avaliada.

As perguntas do método auxiliam o avaliador a identificar as ações que apresentam problemas, ou seja, que prejudicam ou impedem que o usuário aprenda a interagir com a interface para concluir sua tarefa. Também ajudam-no a justificar os problemas encontrados.

Na atividade de *consolidação dos resultados*, os avaliadores analisam as histórias de sucesso e insucesso sobre a realização das tarefas para sintetizar resultados sobre:

- o conhecimento que os usuários devem possuir *a priori* para serem capazes de executar as tarefas analisadas;
- o conhecimento que os usuários deveriam aprender enquanto realizam as tarefas analisadas;
- as sugestões de correções na interface.

As quatro perguntas que guiam a análise definem classes de problemas de IHC para os quais os avaliadores podem sugerir tipos de correção. Os tipos de correção previstos são os seguintes:

- Se na pergunta “*O usuário tentaria alcançar o efeito desejado?*” for relatada uma história de insucesso, ou seja, se o usuário não tentar fazer a coisa certa, há pelo menos três soluções possíveis:
 - eliminar a ação, combinando-a com outras ações ou deixar o sistema executá-la sozinho;
 - fornecer uma instrução ou indicação de que a ação precisa ser realizada;
 - modificar alguma parte da tarefa para que o usuário entenda a necessidade dessa ação.
- Se na pergunta “*O usuário saberá que a ação correta está disponível?*” for relatada uma história de insucesso, ou seja, se o usuário formula a intenção correta mas não sabe que a ação está disponível na interface, a solução pode ser tornar a ação mais evidente. Por exemplo, acrescentar um item de menu

ou um botão na interface para ativar a mesma ação associada a um conjunto de teclas.

- Se na questão “*O usuário conseguirá associar a ação correta com o efeito que está tentando atingir?*” for relatada uma história de insucesso, ou seja, se o usuário não for capaz de mapear seu objetivo nas ações disponíveis na interface, pode ser necessário renomear as ações e reescrever as instruções da interface. Com um vocabulário mais familiar, o usuário tem maiores chances de realizar o mapeamento dos seus objetivos nas ações disponíveis na interface.
- Se a ação correta for realizada e a resposta para a pergunta “*O usuário perceberá que está progredindo em direção à conclusão da tarefa?*” for relatada por uma história de insucesso, ou seja, se o usuário não for capaz de perceber que está caminhando para concluir a tarefa, as respostas (feedbacks) do sistema devem ser destacadas ou expressas mais claramente. Idealmente, as respostas do sistema devem deixar claro *o que ocorreu e o que é possível fazer* em seguida para concluir a tarefa do usuário.

Caso seja esperado que uma mesma tarefa precise ser realizada por usuários de diferentes perfis, a avaliação por percurso cognitivo deve ser realizada diversas vezes, uma para cada perfil de usuário.

O *relato dos resultados* do percurso cognitivo costuma conter (veja Seção 9.7.5):

- os objetivos e escopo da avaliação;
- uma breve descrição do método de percurso cognitivo, incluindo as perguntas que devem ser respondidas;
- o número e o perfil de avaliadores;
- a descrição das tarefas analisadas.

Para cada tarefa analisada, o relatório deve conter:

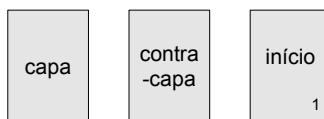
- um resumo do conhecimento que os usuários devem ter *a priori* para serem capazes de executar a tarefa;
- um resumo do conhecimento que os usuários deveriam aprender enquanto realizam a tarefa;
- lista de problemas encontrados, indicando:
 - a ação que o usuário deveria executar;
 - local na interface onde ocorreu o problema;
 - descrição e justificativa do problema;
 - sugestões de solução.

Exemplo 10.2 – Percurso Cognitivo

Perfil de usuário: aluno de graduação em projeto final

Sistema: Microsoft® Word

Tarefa: formatar o relatório conforme modelo requerido pela universidade, com capa, contracapa, e numeração a partir da terceira página, ali iniciando com o número 1, conforme o exemplo a seguir:

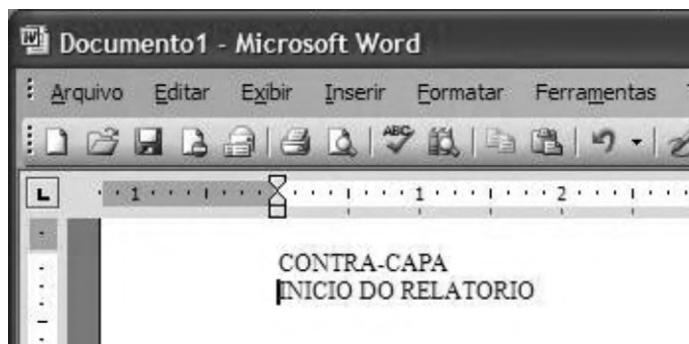


Passos necessários para realizar a tarefa:

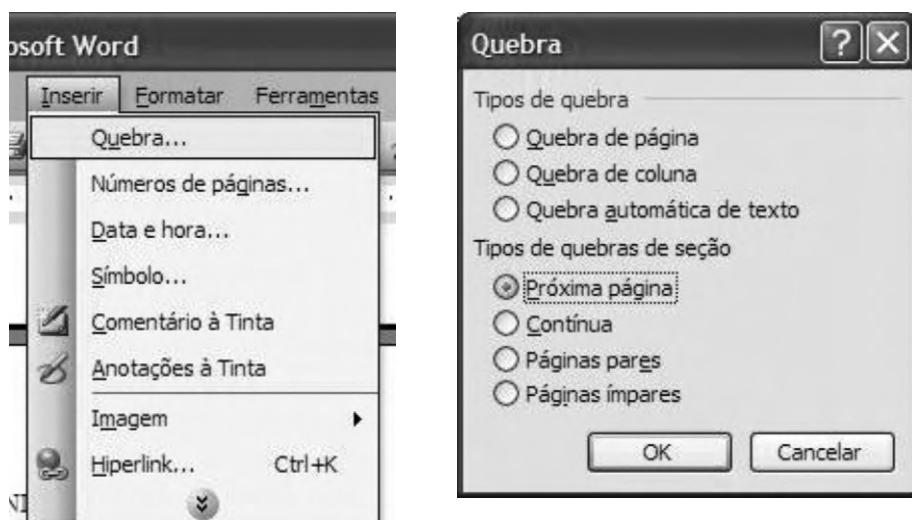
1. mover o cursor para o início do texto que deverá ficar na terceira página
2. inserir uma quebra de seção para “Próxima página”
3. exibir cabeçalho e rodapé
4. ir para o rodapé
5. desvincular a seção atual da seção anterior
6. inserir número de página
7. formatar número de página para iniciar com 1
8. alinhar parágrafo à direita
9. fechar cabeçalho e rodapé

Percorso cognitivo (parcial)

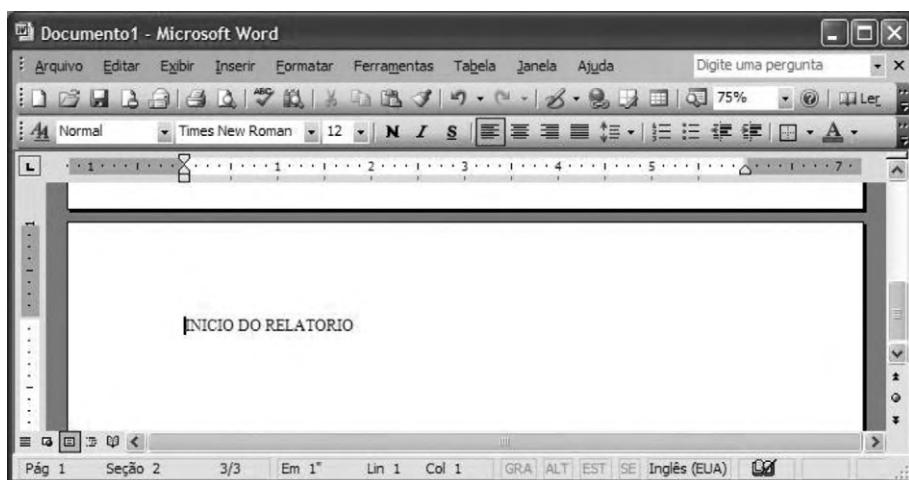
1. Mover o cursor para o início da terceira página
 - *O usuário tentaria atingir o efeito correto?*
Sim, pois é a partir daquele ponto que deseja numerar o documento.
Ou não, caso acredite que pode posicionar o cursor em qualquer ponto daquela página.
 - *O usuário perceberia que a ação correta está disponível?*
Sim, por aprendizado prévio básico sobre como navegar por um documento com o mouse ou o teclado.
 - *O usuário conseguiria associar a ação correta com o efeito que está tentando atingir?*
Sim, pois sabe os efeitos de utilizar o mouse e o teclado para mover o cursor de texto.
 - *Se a ação correta for realizada, o usuário perceberia que está progredindo para concluir a tarefa?*
Sim, pois o cursor de texto indica sua nova posição.



2. Inserir uma quebra de seção para “Próxima página”

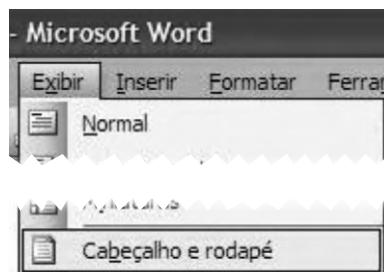


- *O usuário tentaria atingir o efeito correto?*
Não, pois não conhece o conceito de seção.
- *O usuário perceberia que a ação correta está disponível?*
Sim. Se ele souber que deve inserir uma quebra de seção, vai procurar no menu Inserir, e verá o item Quebra...
- *O usuário conseguiria associar a ação correta com o efeito que está tentando atingir?*
Sim, pois os rótulos do item de menu e das opções do diálogo deixam claro o seu efeito.

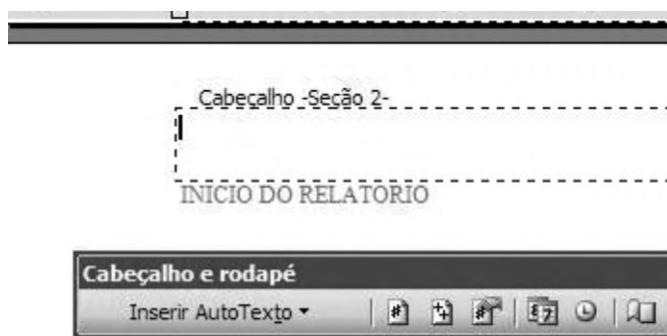


- *Se a ação correta for realizada, o usuário perceberia que está progredindo para concluir a tarefa?*
Sim, ele percebe que houve a quebra de página, e a barra de status indica que ele agora está na Seção 2.

3. Exibir cabeçalho e rodapé



- *O usuário tentaria atingir o efeito correto?*
Não, pois acredita que não precisa editar o rodapé para numerar as páginas.
- *O usuário perceberia que a ação correta está disponível?*
Sim. Se ele souber que deve exibir o rodapé, vai procurar no menu Exibir, e verá o item Cabeçalho e rodapé.
Ou não. Caso ele acredite que tem de "editar" o rodapé, ele buscará no menu Editar e nada encontrará.
- *O usuário conseguiria associar a ação correta com o efeito que está tentando atingir?*
Sim, pois o rótulo do item de menu deixa claro o seu efeito.



- *Se a ação correta for realizada, o usuário perceberia que está progredindo para concluir a tarefa?*

Sim, ele percebe que agora está editando o cabeçalho da Seção 2. Aparece uma moldura com o rótulo *Cabeçalho - Seção 2* e também uma barra de ferramentas intitulada *Cabeçalho e rodapé*.

(O restante da avaliação por percurso cognitivo fica como exercício para o leitor.)

10.1.3 Método de Inspeção Semiótica

Fundamentado na engenharia semiótica, o método de **inspeção semiótica** avalia a comunicabilidade de uma solução de IHC por meio de inspeção (de Souza *et al.*, 2006; Prates e Barbosa, 2007; de Souza e Leitão, 2009; Seção 3.8). O objetivo da inspeção semiótica é avaliar a qualidade da *emissão* da metacomunicação do designer codificada na interface. Portanto, não é necessário envolver usuários nessa avaliação.

Conforme discutido na Seção 3.8, a engenharia semiótica classifica os signos codificados na interface em três tipos: estáticos, dinâmicos e metalinguísticos (de Souza e Leitão, 2009; de Souza, 2005a; Seção 3.8.1). Essa classificação orienta o trabalho do avaliador durante a inspeção semiótica. Para cada tipo de signo, o avaliador inspeciona a interface, incluindo a documentação disponível para o usuário (por exemplo, a ajuda on-line e manuais de uso), interpretando os signos daquele tipo codificados no sistema com objetivo de reconstruir a metamensagem do designer. Dessa forma, o avaliador tem três versões da metamensagem reconstruída, uma para cada tipo de signo. Em seguida, o avaliador contrasta e compara as três metamensagens reconstruídas, e por fim faz um julgamento de valor sobre a comunicabilidade do sistema interativo.

Assim como ocorre nos demais métodos de avaliação por inspeção, os resultados fornecidos pela inspeção semiótica dependem fortemente da interpretação do avalia-

dor sobre os signos codificados na interface. A Tabela 10.3 apresenta as atividades do método de inspeção semiótica.

Tabela 10.3 Atividades do método de inspeção semiótica

inspeção semiótica	
atividade	tarefa
Preparação	<ul style="list-style-type: none"> ▪ identificar os perfis de usuários ▪ identificar os objetivos apoiados pelo sistema ▪ definir as partes da interface que serão avaliadas ▪ escrever cenários de interação para guiar a avaliação
Coleta de dados	<ul style="list-style-type: none"> ▪ inspecionar a interface simulando a interação descrita pelo cenário de interação
Interpretação	<ul style="list-style-type: none"> ▪ analisar os signos metalingüísticos e reconstruir a metamensagem correspondente ▪ analisar os signos estáticos e reconstruir a metamensagem correspondente ▪ analisar os signos dinâmicos e reconstruir a metamensagem correspondente
Consolidação dos resultados	<ul style="list-style-type: none"> ▪ contrastar e comparar as metamensagens reconstruídas nas análises de cada tipo de signo ▪ julgar os problemas de comunicabilidade encontrados
Relato dos resultados	<ul style="list-style-type: none"> ▪ relatar a avaliação da comunicabilidade da solução de IHC, sob o ponto de vista do emissor da metamensagem

Na atividade de *preparação*, o avaliador deve identificar os perfis dos usuários a quem o sistema se destina e os objetivos que o sistema apoia, para então definir o escopo da avaliação (veja Seção 9.7.2). Conhecendo os perfis dos usuários e definido o escopo da avaliação, o avaliador deve elaborar cenários de interação (Carroll, 2000; Rossen e Carroll, 2002; Seção 7.2) para guiar a inspeção da interface e sua interpretação dos signos nela codificados. Os cenários de interação são ferramentas importantes para definir um contexto de uso e um conjunto de objetivos (ou intenções de comunicação) que os usuários desejam alcançar utilizando o sistema. Essas informações fornecem ao avaliador melhores condições para identificar, interpretar e analisar os signos codificados na interface.

No método de inspeção semiótica, o avaliador realiza em conjunto as atividades de coleta de dados sobre experiências de uso e de interpretação. Nessas atividades, ele inspeciona a interface para identificar, interpretar e analisar os signos metalingüísticos, estáticos e dinâmicos nela codificados. Dependendo do tipo de signo analisado no momento, o avaliador concentra sua inspeção em diferentes partes da interface. Por exemplo, a análise dos signos metalingüísticos requer a inspeção do sistema de ajuda on-line, das mensagens de erro e das explicações presentes na interface. Já a

análise dos signos estáticos requer a inspeção dos elementos da interface em determinado instante no tempo. O método de inspeção semiótica apresenta melhores resultados se a inspeção for realizada sobre a versão final do sistema interativo, pois a representação mais concreta dos signos na interface influencia fortemente sua interpretação (seja pelo avaliador ou por usuários), e a análise dos signos dinâmicos é mais fácil, acurada e precisa durante o uso de uma versão executável do sistema.

À medida que o avaliador identifica e interpreta os três tipos de signos codificados na interface, ele deve prosseguir sua análise reconstruindo iterativamente uma metamensagem do designer para cada tipo de signo analisado. A paráfrase da metamensagem deve ser usada como um modelo (*template*) a ser preenchido. Ela é reproduzida a seguir, com destaque em partes que devem ser completadas durante a inspeção semiótica (de Souza, 2005a, p. 25):

Este é o meu entendimento, como designer, de *quem você, usuário, é*, do que aprendi que você *quer ou precisa fazer*, de *que maneiras prefere fazer, e por quê*. Este, portanto, é o sistema que projetei para você, e esta é *a forma como você pode ou deve utilizá-lo* para alcançar uma gama de objetivos que se encaixam nesta visão.

Essa paráfrase serve de base para a elaboração de um conjunto de perguntas que guiam a reconstrução da metamensagem durante a análise dos três tipos de signos. Tais perguntas auxiliam o avaliador a interpretar as expectativas do designer para as situações de uso do sistema, e interpretar a solução de IHC correspondente proposta por ele. Assim, a reflexão do avaliador pode ser guiada pelas seguintes perguntas (adaptado de Souza, 2005a; de Souza e Leitão, 2009, p. 26):

- [quem você, usuário, é] A quem a mensagem do designer está endereçada (i.e., para o designer, quem são os usuários do sistema)? Quais os perfis desses destinatários (i.e., quais são suas características, valores e crenças)?
- [quer ou precisa fazer] Na visão do designer, o que os usuários vão querer comunicar ao sistema (i.e., quais são os desejos e necessidades dos usuários, o que eles querem ou precisam fazer com apoio do sistema)? Por quê?
- [de que maneiras prefere fazer] Como, onde e quando o designer espera que os usuários se engajem nessa comunicação (i.e., utilizem o sistema para realizar o que querem ou precisam fazer)? Por quê?
- [Este, portanto, é o sistema que projetei para você] O que o designer está comunicando? Que conteúdo e expressão está utilizando nessa comunicação? Qual é a sua visão de design?
- [a forma como você pode ou deve utilizá-lo] Como essa metacomunicação privilegia certos desejos e necessidades dos usuários, em detrimento a

outros? Como essa metacomunicação indica diferentes estratégias de comunicação que o usuário pode seguir ao se comunicar com o preposto do designer? Como a comunicação do usuário com o preposto do designer é facilitada em certos contextos, em detrimento a outros? Por quê?

- [alcançar uma gama de objetivos] Que efeito(s) o designer espera que sua comunicação cause? Que objetivos ele espera que o usuário alcance por meio dessa comunicação?

Essas perguntas são respondidas durante a análise de cada tipo de signo, com o objetivo de reconstruir o trecho correspondente da metamensagem do designer. Vale lembrar que a análise dos signos se limita aos cenários de interação elaborados com base no escopo da avaliação. Portanto, a metamensagem reconstruída é parcial, ou seja, não corresponde a toda a metamensagem do designer sobre o sistema avaliado. De qualquer forma, sejam quais forem os trechos de metamensagem reconstruídos, mesmo que incompletos, eles devem ser analisados em conjunto visando julgar se são consistentes e coerentes entre si.

O Exemplo 10.3 apresenta um cenário de interação definido para avaliar o cadastramento de material didático em um sistema de apoio acadêmico.

Exemplo 10.3 – Cenário de interação (parcial) utilizado em uma inspeção semiótica

Visando avaliar o sistema Moodle,² de apoio ao ensino a distância ou presencial, foi definido o seguinte cenário:

Lucas, professor de Introdução ao Cálculo, utiliza o Moodle para divulgar o seu material didático para os alunos. Esse material inclui slides, listas de exercícios e provas aplicadas em períodos anteriores, e fica armazenado em arquivos de diversos tipos: slides, animações, documentos textuais e planilhas. Está começando um novo período, e ele precisa cadastrar todo esse material no Moodle.

Passado um mês de aula, Lucas decide substituir parte do material cadastrado, a fim de fazer pequenas correções, e incluir mais alguns exemplos.

Mesmo dentro do escopo definido pelos cenários de interação, é comum existirem lacunas nas reconstruções da metamensagem. Nesse caso, o avaliador deve prever suas consequências. Por exemplo, se não houver signos metalingüísticos que expliquem determinado signo estático, pode ser o caso de o usuário não conseguir interpretá-lo e assim deixar de fazer uso (adequado) dele.

A partir dos cenários de interação elaborados com base no objetivo e escopo da avaliação, o avaliador inspeciona os *signos metalingüísticos*: a ajuda on-line, os manuais do usuário e demais formas de documentação do sistema e os materiais de divul-

² <http://moodle.org>.

gação. Os signos metalinguísticos são os primeiros a serem analisados na inspeção semiótica, pois expressam e explicam explicitamente outras partes da metamensagem do designer. Eles comunicam aos usuários os significados dos signos estáticos, dinâmicos e outros signos metalinguísticos, e como todos esses signos podem ser utilizados durante a interação. Normalmente eles são encontrados por toda a interface em instruções, explicações, avisos e mensagens de erros, mas se concentram na ajuda on-line, manuais do usuário e em materiais de divulgação do sistema. O resultado da inspeção e análise dos signos metalinguísticos é a reconstrução de trechos da metamensagem do designer de acordo com o que foi aprendido nesse passo.

O Exemplo 10.4 ilustra um trecho da metamensagem reconstruída com base em signos metalinguísticos de um sistema de apoio acadêmico, conforme o cenário de interação definido anteriormente.

Exemplo 10.4 – Análise (parcial) dos signos metalinguísticos

O sistema de ajuda on-line do Moodle apresenta o seguinte trecho de ajuda sobre o gerenciamento dos arquivos associados a um curso:

Arquivos

A área de arquivos pode incluir conteúdo em PDF, HTML, multimídia, editor de texto, apresentações ou qualquer outro conteúdo digital para incluir em uma atividade, recurso, seção do curso, link ou download direto.

O link de arquivos apresenta uma lista de arquivos e pastas, dependendo do papel do usuário. A lista conterá o nome, tamanho, data da última modificação e ações que podem modificar o item.

Para visualizar um arquivo, clique em seu nome. Seu navegador vai exibi-lo ou efetuar o download para o seu computador.

Ferramentas

Mover, cancelar, criar arquivo ZIP

É possível mover, cancelar completamente ou arquivar em ZIP um ou mais itens. Primeiro, selecione os itens na lista marcando as caixas à sua esquerda. Então utilize o menu “Com arquivos escolhidos” até a ação que deseja realizar.

Criar um diretório

O botão “Criar um diretório” se encontra abaixo da lista. A estrutura de arquivos inicial para um curso é simples. Módulos no Moodle podem criar seu próprio diretório. Em geral, um professor pode criar um ou mais diretórios em qualquer lugar da área de “arquivos”. Esses diretórios podem ser vistos ao acrescentar uma imagem ou recurso de dentro de um curso.

Enviar um arquivo

Na parte de baixo de todas as telas de arquivo há um botão “Enviar um arquivo”. Isso permitirá o envio de um único arquivo. Enviar um arquivo com o mesmo nome de um arquivo existente sobrescreverá automaticamente o arquivo existente sem um aviso.

DICA: Ao criar primeiro um arquivo ZIP com um grupo ou diretório de arquivos, você pode enviar esse arquivo e o Moodle lhe dará um link para descompactá-lo. A ação de descompactação criará os arquivos e diretórios na seção de arquivos administrativos.

Com base nesse trecho, é possível reconstruir parte da metamensagem do designer, como a seguir:

Eu acredito que você trabalha com diversos tipos de arquivo, cada qual identificado pelo seu nome, tamanho e data de última modificação. Às vezes você quer organizar os arquivos, e para isso prefere utilizar diretórios, por entre os quais você quer mover um ou mais arquivos de uma só vez. Ainda para manter os arquivos de um curso organizados, você quer poder excluí-los ou compactá-los em um arquivo ZIP.

Como você costuma atualizar os arquivos com novas versões, mantendo o mesmo nome, eu tornei muito fácil substituir a versão anterior, bastando para isso enviar o novo arquivo com o mesmo nome do arquivo existente. Acredito que você será cuidadoso e, portanto, não vou pedir confirmação antes de efetuar essa sobreposição.

Como você costuma enviar vários arquivos de uma só vez, para poupar o seu tempo eu permito que você envie um arquivo ZIP contendo toda a estrutura de diretórios e arquivos que você organizou, e dentro do Moodle descompacte-os nos diretórios apropriados.

Tendo reconstruído a metamensagem com base nos signos metalingüísticos, o avaliador prossegue então para a *inspeção e análise dos signos estáticos*. Ele inspeciona a parte da interface que corresponde ao cenário de interação avaliado, buscando identificar e interpretar os signos estáticos nela codificados. Os signos estáticos expressam o *estado do sistema* em determinado instante (veja Seção 3.8.1). Eles são representados pelos elementos presentes nas telas da interface (ou equivalentes em interfaces não visuais), como rótulos, imagens, caixas de texto, botões, menus etc., bem como a disposição (*layout*), tamanho, cor, fonte e demais características desses elementos de interface. A análise dos signos estáticos deve considerar apenas os elementos de interface apresentados em cada tela num instante de tempo, sem examinar o comportamento do sistema, nem as relações temporais e causais entre os elementos de interface. Para concluir a análise dos signos estáticos, o avaliador deve reconstruir um novo trecho da metamensagem do designer, também guiado pelas perguntas utilizadas anteriormente (Exemplo 10.5). Esse trecho deve ser elaborado separadamente daquele reconstruído com base nos signos metalingüísticos, a fim de que o avaliador possa compará-los somente após a inspeção de todos os tipos de signo.

Exemplo 10.5 – Análise (parcial) dos signos estáticos

Considere a seguinte tela de seleção de um arquivo do sistema Moodle, para associá-lo a um tópico do curso:



Figura 10.1 Tela para seleção de arquivo no Moodle.

Uma possível reconstrução (parcial) da mentamensagem do designer com base nos signos estáticos é a seguinte (entre colchetes são apresentadas as evidências que apoiam cada afirmação):

Eu acredito que você seja um professor que organiza seu material didático em diversos arquivos [apresentação em tabela], e que toda vez que deve selecionar um arquivo pode aproveitar para reorganizar um ou mais arquivos [vários botões e links além do link *Escolher*]. No entanto, acredito que você não registre tantos arquivos a ponto de precisar ordená-los de diferentes maneiras [ausência de opções de ordenação].

Para identificar se um arquivo é o desejado, você precisa apenas examinar o nome, tamanho e data de registro do arquivo [colunas da tabela]. Se identificar que o arquivo desejado ainda não foi registrado no sistema, quer registrá-lo logo a partir daqui [botão *Enviar um arquivo*], para não perder tempo dando voltas no sistema.

Você é cuidadoso, e costuma examinar se o arquivo foi registrado corretamente [link no nome do arquivo]. Mesmo após efetuar o envio de um arquivo, você pode decidir modificar o seu nome [link *Renomear*], para identificar mais claramente o seu conteúdo.

Você gosta de organizar seus arquivos hierarquicamente em pastas [botão *Criar um diretório*], pois está acostumado a organizá-los assim em seu sistema operacional [padrão do Windows Explorer]. Além disso, você quer poder manipular diversos arquivos de uma vez [checkbox ao lado de cada arquivo, botões *Selecionar tudo* e *Anular todas as seleções* e combo *Com arquivos escolhidos...*], para agilizar o seu trabalho.

Na *análise dos signos dinâmicos*, o avaliador deve inspecionar o processo de interação que o usuário pode vivenciar através da interface. Com base nos cenários de interação, o avaliador navega pela interface para identificar os signos dinâmicos evidenciados pelas relações temporais e causais entre outros signos. Os signos dinâmicos são percebidos através de *modificações na interface* que comuniqueem ao usuário o comportamento do sistema em decorrência de ações do usuário (e.g., clicar no mouse, tecer *enter*, mudar o foco de um campo de formulário para outro), de eventos externos (e.g., receber um novo e-mail, a conexão com a Internet falhar etc.) ou do passar do tempo. Por exemplo, signos dinâmicos são geralmente representados por animações, abrir e fechar diálogos, transições entre telas, ou modificações nos elementos de uma

tela (e.g., habilitar um botão, atualizar um texto ou imagem, modificar o *layout* de alguns elementos de interface etc.). A conclusão da análise dos signos dinâmicos deve ser registrada pelo avaliador com uma nova reconstrução da metamensagem pelo designer, também guiado pelas perguntas apresentadas anteriormente (Exemplo 10.6).

Exemplo 10.6 – Análise (parcial) dos signos dinâmicos

Considere, a partir da tela apresentada na figura anterior, esta sequência de telas para enviar um arquivo através do Moodle:

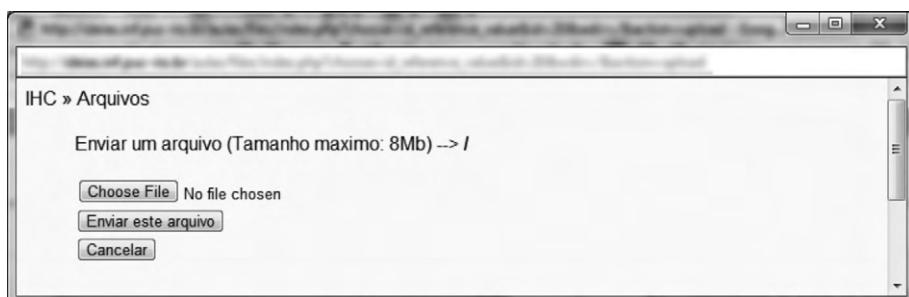


Figura 10.2 Tela para envio de um arquivo (acionada clicando-se em *Enviar um arquivo* na Figura 10.1).

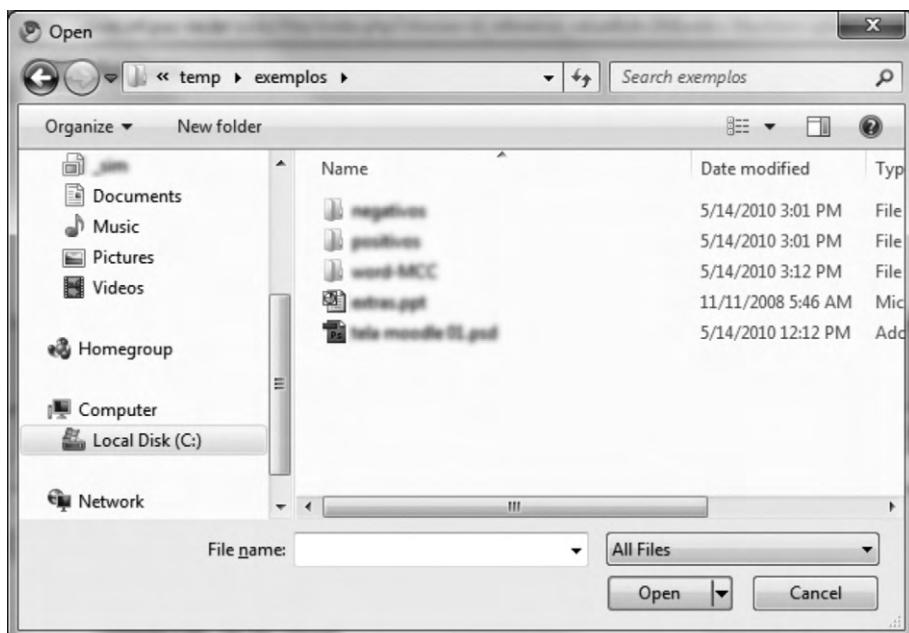


Figura 10.3 Tela para envio de um arquivo (acionada clicando-se em *Choose file* na figura anterior).

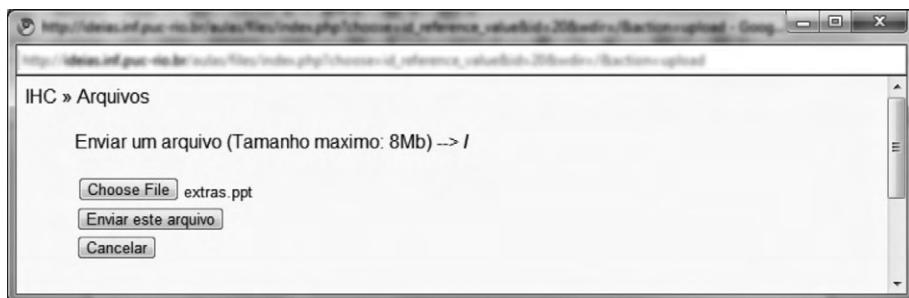


Figura 10.4 Tela para envio de um arquivo (acionada clicando-se em um arquivo e em *Open* na figura anterior).

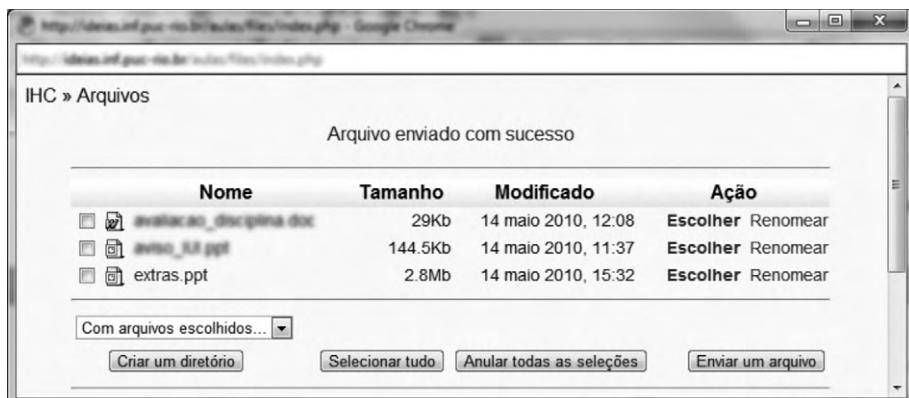


Figura 10.5 Tela de gerenciamento de arquivos, após acionar *Enviar este arquivo* na figura anterior.

Com base nessa sequência, é possível reconstruir um trecho da metamensagem do designer como a seguir:

Acredito que você gosta de ser informado sobre o que pode fazer com o sistema passo a passo, mesmo que isso seja um pouco ineficiente. Caso haja alguma restrição sobre uma ação, você quer ser informado antes de realizá-la. Portanto, projetei o sistema para lhe informar o tamanho máximo do arquivo que você pode enviar, antes de permitir que você localize o arquivo. Como você privilegia uma atuação cuidadosa sobre a eficiência, antes de efetuar o envio propriamente dito, o sistema pede para você confirmar o arquivo localizado. Finalmente, como você deve querer confirmar que o envio foi feito, projetei o sistema para lhe informar sobre o resultado do envio [mensagem Arquivo enviado com sucesso], além de incluir o arquivo na lista de arquivos registrados, em ordem alfabética.

Sempre que não houver signos metalinguísticos que expliquem algum signo estático ou dinâmico, o avaliador deve julgar se os signos sem explicações podem ser compreendidos (ou inferidos) pelo usuário com a experiência de uso.

Os signos metalinguísticos, estáticos e dinâmicos têm poder de expressão diferente, pois pertencem a vários sistemas de significação. Por exemplo, os signos metalinguísticos presentes na ajuda on-line descrevem e fornecem explicações sobre os demais signos codificados na interface em linguagem natural, possivelmente complementada por imagens e animações. Já os signos estáticos em interfaces visuais costumam utilizar ícones, botões e menus para possibilitar a interação do usuário com o sistema. Sendo assim, não é raro haver diferenças nas metamensagens reconstruídas durante a análise de cada um dos três tipos de signos. Para que a comunicação designer-usuário seja bem-sucedida, essas metamensagens não podem ser *contraditórias* ou *inconsistentes* umas com as outras. A consistência e a regularidade são importantes para criar e evocar padrões de significação que são familiares aos usuários.

Também é preciso ter um cuidado especial com ambiguidades entre as metamensagens reconstruídas, porque elas podem atrapalhar a interação. Por exemplo, em um sistema que possui uma biblioteca de arquivos, como um reproduutor de música ou gerenciador de fotos, o significado do comando Excluir pode ser ambíguo para o usuário. Ele pode significar excluir o arquivo apenas da biblioteca do programa ou excluir o arquivo da biblioteca do programa e também do local de armazenamento. Em casos como esse, é fundamental o designer comunicar ao usuário o significado do comando Excluir codificado no sistema (de Souza *et al.*, 2006; Prates e Barbosa, 2007). No sistema Moodle, a ação Cancelar completamente (uma das opções da combo Com arquivos escolhidos... da Figura 10.5) tem o efeito de remover o(s) arquivo(s) selecionado(s).

Na atividade de *consolidação dos resultados* (veja Seção 9.7.5), o avaliador deve *contrastar* e *comparar* as metamensagens reconstruídas durante a análise dos signos metalinguísticos, estáticos e dinâmicos. Desse modo, ele revisa as três metamensagens reconstruídas, procurando intencionalmente por significados contraditórios, inconsistentes ou ambíguos para os signos que as compõem (Exemplo 10.7).

Exemplo 10.7 – Comparação parcial entre as metamensagens reconstruídas a partir dos signos metalinguísticos, estáticos e dinâmicos

Nos exemplos anteriores, podemos observar que as metamensagens reconstruídas a partir dos signos metalinguísticos e estáticos indicam que o designer considera que usuário privilegia a eficiência, permitindo o acesso a diversas funções diferentes quando o usuário entra na tela para selecionar um arquivo. Já na metamensagem reconstruída a partir dos signos dinâmicos, o designer parece considerar que a eficiência não é tão importante. Essa discrepância pode indicar uma falta de compreensão do designer sobre o perfil dos seus usuários ou um descuido no momento de projetar os passos necessários para alcançar um determinado objetivo. Esse tipo de inconsistência nas metamensagens pode causar um impacto negativo na interação do usuário, e deve ser evitada.

Para motivar e auxiliar a comparação das três metamensagens, o método de inspeção semiótica sugere as cinco perguntas a seguir (de Souza *et al.*, 2006):

1. O usuário poderia interpretar este signo ou esta mensagem diferente do previsto pelo designer? Como? Por quê?
2. Essa outra interpretação ainda seria consistente com a intenção de design?
3. A interpretação que estou (como avaliador) fazendo no momento me lembra de outras que já fiz em momentos anteriores da avaliação? Quais? Por quê?
4. É possível formar classes de signos estáticos e dinâmicos a partir das análises realizadas? Quais?
5. Existem signos estáticos ou dinâmicos que estão aparentemente mal classificados de acordo com as classes propostas em 4? Isso poderia causar problemas de comunicação com o sistema? Como?

Se o avaliador desejar, ele pode realizar outras perguntas durante a comparação das metamensagens reconstruídas. O conjunto de perguntas sugerido serve apenas como um guia útil para proporcionar uma inspeção semiótica mais produtiva.

Depois de contrastar e comparar as três metamensagens reconstruídas, o avaliador deve elaborar uma *versão unificada da metamensagem*. Por fim, o avaliador deve realizar um *julgamento dos problemas de comunicabilidade* identificados (de Souza *et al.*, 2006). Esses problemas podem atrapalhar os usuários de terem acesso à metamensagem do designer e de interagirem com o sistema de forma produtiva. Eles basicamente ocorrem quando há lacunas, inconsistências, contradições ou ambiguidades nas metamensagens reconstruídas a partir da inspeção e análise de cada tipo de signo.

Na atividade de *relato dos resultados*, o avaliador deve (de Souza *et al.*, 2006; veja Seção 9.7.5):

- fazer uma breve descrição do método para auxiliar o leitor a compreender como os resultados foram obtidos;
- descrever os critérios utilizados para selecionar as partes da interface inspecionadas;
- para cada um dos três tipos de signos inspecionados, fornecer:
 - identificação de signos relevantes (listar e justificar a sua relevância);
 - identificação das classes de signos utilizadas;
 - uma versão revisada da metamensagem do designer.
- redigir a apresentação e explicação do julgamento do avaliador sobre os problemas de comunicabilidade encontrados, que possam dificultar ou impedir

os usuários de entenderem a metamensagem ou interagirem com o sistema de forma produtiva.

O método de inspeção semiótica não exige mais de um avaliador. Se houver mais de um avaliador, eles devem trabalhar em conjunto em todas as atividades. Caso o sistema avaliado possua mais de um perfil de usuário, cada avaliador pode ficar responsável por inspecionar a interface sob o ponto de vista de um dos perfis.

10.2 Avaliação de IHC através de Observação

Como apresentado na Seção 9.6, os métodos de observação permitem ao avaliador coletar dados sobre situações em que os participantes realizam suas atividades, com ou sem apoio de tecnologia computacional. O registro e a análise desses dados permitem identificar **problemas reais** que os participantes enfrentaram, e não apenas problemas potenciais previstos pelo avaliador como em uma avaliação por inspeção. Nas próximas subseções são apresentados os seguintes métodos de avaliação por observação: teste de usabilidade, método de avaliação de comunicabilidade e prototipação em papel.

10.2.1 Teste de Usabilidade

O **teste de usabilidade** visa a avaliar a usabilidade de um sistema interativo a partir de experiências de uso dos seus usuários-alvo (Rubin, 1994; Rubin e Chisnell, 2008). Os objetivos da avaliação determinam quais critérios de usabilidade devem ser medidos. Esses critérios são geralmente explorados por perguntas específicas associadas a algum dado mensurável, que com frequência pode ser objetivamente capturado durante a interação do usuário com o sistema. Por exemplo, caso o objetivo do estudo seja avaliar a facilidade de aprendizado de um determinado sistema, um teste de usabilidade poderá fornecer respostas para perguntas como: “Quantos erros os usuários cometem nas primeiras sessões de uso?”, “Quantos usuários conseguiram completar com sucesso determinadas tarefas?” e “Quantas vezes os usuários consultaram a ajuda on-line ou o manual de usuário?”.

Para realizar as medições desejadas, um grupo de usuários é convidado a realizar um conjunto de tarefas usando o sistema num ambiente controlado, como um laboratório. Durante as experiências de uso observadas, são registrados vários dados sobre o desempenho dos participantes na realização das tarefas e suas opiniões e sentimentos decorrentes de suas experiências de uso. A Tabela 10.4 apresenta as atividades do teste de usabilidade.

Tabela 10.4 Atividades do teste de usabilidade

teste de usabilidade	
atividade	tarefa
Preparação	<ul style="list-style-type: none"> ■ definir tarefas para os participantes executarem ■ definir o perfil dos participantes e recrutá-los ■ preparar material para observar e registrar o uso ■ executar um teste-piloto
Coleta de dados	<ul style="list-style-type: none"> ■ observar e registrar a performance e a opinião dos participantes durante sessões de uso controladas
Interpretação	<ul style="list-style-type: none"> ■ reunir, contabilizar e sumarizar os dados coletados dos participantes
Consolidação dos resultados	
Relato dos resultados	<ul style="list-style-type: none"> ■ relatar a performance e a opinião dos participantes

Na atividade de *preparação*, são realizadas as atividades comuns aos métodos de avaliação por observação (veja Seção 9.7.2). Em particular, são definidas as tarefas que os participantes vão realizar e os dados a serem coletados.

A *coleta de dados* inclui o questionário pré-teste, a sessão de observação e a entrevista pós-teste (veja Seção 9.7.3). Durante as sessões de observação, são coletados diferentes tipos de dados. Por exemplo, para cada tarefa, realizada por cada participante, é possível medir: o grau de sucesso da execução, o total de erros cometidos, quantos erros de cada tipo ocorreram, quanto tempo foi necessário para concluir-la, o número de vezes que a ajuda on-line foi consultada, o grau de satisfação do usuário, e assim por diante. Além disso, também são coletados: anotações do avaliador, vídeos de interação, registro das teclas digitadas e áudio com os comentários dos participantes.

Nas atividades de *interpretação* e *consolidação*, os dados dos participantes devem ser organizados de modo a evidenciar as relações entre eles (veja Seções 9.7.4 e 9.7.5). Historicamente, o teste de usabilidade tem sido empregado para obter sobretudo resultados quantitativos, tais como: testar hipóteses, descobrir tendências, comparar soluções alternativas e verificar se o sistema atingiu as metas de usabilidade definidas no início do projeto. Para esse tipo de análise, geralmente utilizamos tabelas e gráficos, e calculamos médias, porcentagens e qualquer outro indicador relevante. Por exemplo, considere uma meta de usabilidade definindo que uma determinada tarefa deve ser realizada em até cinco minutos. Um teste de usabilidade permite avaliar o grau em que essa meta foi atingida, através do número de usuários que concluíram a tarefa dentro do tempo desejado, do tempo médio despendido por eles e do desvio padrão correspondente. Se as metas de usabilidade não forem atingidas, as conclusões que o

avaliador pode tirar desse tipo de resultado costumam ser bem gerais, como: a parte do sistema relacionada com a tarefa avaliada não é tão eficiente quanto desejado.

Mais recentemente, o teste de usabilidade também tem sido empregado para fornecer resultados qualitativos (Rubin e Chisnell, 2008; Kuniavsky, 2003). Para Rubin e Chisnell (2008), a análise dos dados coletados também deve identificar a origem dos problemas na interação que prejudicaram o desempenho mensurado. Um trecho de interação em que ocorreu um problema pode estar associado a uma parte do áudio com comentários do participante, a um conjunto de teclas digitadas e a certas anotações do avaliador. A análise conjunta desses dados pode revelar aspectos que não seriam identificados através da análise de um único tipo de dado. Para cada problema observado, o avaliador deve analisar todos os dados coletados de modo a interpretar quais características, partes e comportamentos da interface podem tê-lo causado e assim elaborar possíveis explicações sobre o problema. Essas explicações constituem um resultado qualitativo importante para justificar as recomendações do avaliador no reprojeto da interface e da interação.

Na consolidação dos dados coletados, Kuniavsky (2003) também recomenda que o avaliador categorize os problemas encontrados durante a interação de todos os participantes, descrevendo cada categoria de problema, em que partes da interface ela ocorre e os impactos imediatos na usabilidade do sistema avaliado. Tanto quanto possível, o avaliador deve explicar suas hipóteses sobre as possíveis causas de cada classe de problemas com base nas suas interpretações do que ocorreu e fornecer sugestões de melhorias na interface e interação.

O *relato dos resultados* do teste de usabilidade deve descrever (veja Seção 9.7.5):

- os objetivos e escopo da avaliação;
- uma breve descrição do método de teste de usabilidade;
- o número e o perfil dos avaliadores e dos participantes;
- as tarefas executadas pelos participantes;
- tabelas e gráficos que sumarizam as medições realizadas;
- uma lista de problemas encontrados, indicando, para cada problema:
 - local onde ocorreu;
 - descrição e justificativa;
 - discussão, indicando os fatores de usabilidade prejudicados;
 - sugestões de solução.

Exemplo 10.8 – Resultados de um teste de usabilidade

Um teste de usabilidade foi projetado para avaliar o desempenho dos usuários na inclusão de um arquivo associado a um tópico de aula em dois sistemas: no Moodle (denominado sistema A) e em outro sistema Web desenvolvido pelo grupo que realizou a avaliação (sistema B).

O perfil dos participantes do teste era de professores que não conheciam nenhum dos sistemas. Foram recrutados 12 professores, dentre os quais seis homens e seis mulheres, todos professores de disciplinas de ciências exatas e de computação. Cada usuário deveria utilizar os dois sistemas.

Para que a ordem em que eles fossem expostos ao sistema não interferisse nos resultados do teste, metade dos participantes foi exposta ao sistema A e depois ao sistema B (grupo AB), e a outra metade na ordem inversa (grupo BA).

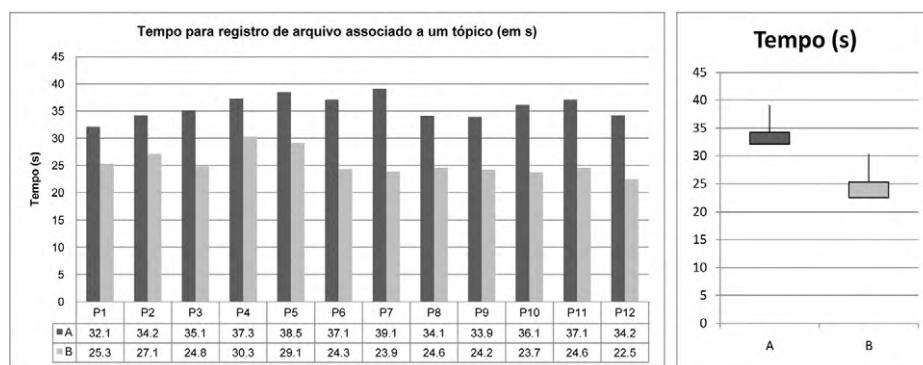
Os dados coletados foram: tempo para conclusão da tarefa; número de erros cometidos; número de acessos ao sistema de ajuda; número de usuários que não conseguiram realizar a tarefa; número de vezes que os usuários se desviaram do caminho mais eficiente.

Para assegurar a validade dos dados coletados, foi solicitado a todos os usuários associar o mesmo arquivo, localizado no mesmo diretório, sob condições semelhantes de conexão com o servidor. A memória cache do navegador era esvaziada entre cada sessão de teste.

A hipótese nula e a hipótese alternativa são:

- H_0 : Não há diferença entre o tempo de realização da atividade nos sistemas A e B.
- H_1 : Existe diferença entre o tempo de realização da atividade nos sistemas A e B.

Os gráficos a seguir apresentam os valores coletados para o tempo de realização da tarefa:



Considerando $\alpha = 0,05$, um teste t em pares rejeita a hipótese nula ($t = 3,22 > 3,1058$). Sendo assim, podemos concluir que, para a atividade de inclusão de um arquivo associado a um tópico de aula por usuários iniciantes, a eficiência dos usuários é maior no sistema B que no sistema A (Levine *et al.*, 2008).

10.2.2 Método de Avaliação de Comunicabilidade

O método de **avaliação de comunicabilidade** visa apreciar a qualidade da comunicação da metamensagem do designer para os usuários (Prates *et al.*, 2000a; de Souza,

2005a; Prates e Barbosa, 2007; de Souza e Leitão, 2009). Assim como o método de inspeção semiótica, o método de avaliação de comunicabilidade tem como fundamentação teórica a engenharia semiótica, apresentada na Seção 3.8. Esses dois métodos avaliam a comunicabilidade a partir de diferentes pontos de vista: enquanto a inspeção semiótica avalia a qualidade da *emissão* da metacomunicação do designer, o método de avaliação de comunicabilidade avalia a qualidade da *recepção* dessa metacomunicação.

Representantes dos usuários são convidados a realizar um conjunto de tarefas utilizando o sistema em um ambiente controlado, como um laboratório. Essas experiências de uso são observadas e registradas, principalmente em vídeos de interação. Os avaliadores analisam cada registro de experiências de uso para compreender como foi a interação de cada usuário com o sistema sendo avaliado. O foco dessa análise abrange os prováveis caminhos de interpretação dos usuários, suas intenções de comunicação e, principalmente, as rupturas de comunicação que ocorreram durante a interação. Como resultado, os avaliadores identificam problemas na comunicação da metamensagem do designer e na comunicação do usuário com o sistema, e também ajudam a informar ao designer as causas desses problemas. A avaliação de comunicabilidade é um método qualitativo que privilegia a análise em profundidade. Desse modo, o número de participantes normalmente é pequeno, variando entre cinco e dez participantes. A Tabela 10.5 apresenta as atividades do método de avaliação de comunicabilidade.

Tabela 10.5 Atividades do método de avaliação de comunicabilidade

avaliação de comunicabilidade	
atividade	tarefa
Preparação	<ul style="list-style-type: none"> ▪ inspecionar os signos estáticos, dinâmicos e metalinguísticos ▪ definir tarefas para os participantes executarem ▪ definir o perfil dos participantes e recrutá-los ▪ preparar material para observar e registrar o uso ▪ executar um teste-piloto
Coleta de dados	<ul style="list-style-type: none"> ▪ observar e registrar sessões de uso em laboratório ▪ gravar o vídeo da interação de cada participante
Interpretação	<ul style="list-style-type: none"> ▪ etiquetar cada vídeo de interação individualmente
Consolidação dos resultados	<ul style="list-style-type: none"> ▪ interpretar as etiquetagens de todos os vídeos de interação ▪ elaborar perfil semiótico
Relato dos resultados	<ul style="list-style-type: none"> ▪ relatar a avaliação da comunicabilidade da solução de IHC, sob o ponto de vista do receptor da metamensagem

Na atividade de *preparação* (veja Seção 9.7.2), recomenda-se realizar uma breve inspeção dos signos estáticos, dinâmicos e metalingüísticos da interface, caso não tenha sido feita uma inspeção semiótica completa. Essa inspeção orienta a definição dos cenários de tarefas que os participantes deverão realizar e a elaboração do material de apoio. Ao preparar o ambiente de avaliação, o avaliador deve configurar e testar cuidadosamente o software de gravação do vídeo de interação, com tudo o que aparece na tela do usuário durante a interação, bem como as teclas digitadas. Esse vídeo é o material básico e fundamental para as atividades de interpretação e consolidação dos resultados nesse método de avaliação.

A *coleta de dados* inclui o questionário pré-teste, a sessão de observação e a entrevista pós-teste (veja Seção 9.7.3). O principal resultado da coleta de dados é um conjunto dos vídeos de interação capturados de cada sessão (um vídeo por participante, ou um vídeo de cada tarefa), acompanhados de anotações dos avaliadores e demais registros sobre o que ocorreu durante essas experiências de uso e sobre o que os usuários disseram na entrevista pós-teste.

Na atividade de *interpretação* (veja Seção 9.7.4), o avaliador faz a **etiquetagem** dos vídeos. Ele assiste a cada vídeo de interação repetidas vezes para identificar *rupturas de comunicação*, ou seja, momentos da interação em que o usuário demonstra não ter entendido a metacomunicação do designer, ou momentos em que o usuário encontra dificuldades de expressar sua intenção de comunicação na interface. As rupturas de comunicação encontradas nos vídeos de interação devem ser categorizadas por uma expressão de comunicabilidade que coloca “palavras na boca do usuário”, tais como: “Cadê?” e “Epa!”. Dessa forma, associar uma expressão de comunicabilidade a uma sequência de interação permite ao avaliador presumir o que o usuário poderia ter dito (ou de fato disse) naquele momento. Por exemplo, se o usuário procura na interface como executar determinada ação, o avaliador associa essa ruptura de comunicação com a etiqueta “Cadê?”, como se o usuário estivesse dizendo isso em voz alta naquele momento. Existem 13 etiquetas para categorizar rupturas de comunicação no método de avaliação de comunicabilidade. São elas (Prates *et al.*, 2000a; de Souza, 2005a; Prates e Barbosa, 2007; de Souza e Leitão, 2009): *Cadê? E agora? O que é isto? Epa! Onde estou? Ué, o que houve? Por que não funciona? Assim não dá. Vai de outro jeito. Não, obrigado! Pra mim está bom. Socorro! e Desisto.*

A etiqueta “**Cadê?**” é usada quando o usuário deseja expressar sua intenção de comunicação, mas não consegue expressá-la com os signos codificados na interface. Por exemplo, o usuário pode saber que o sistema permite executar determinada ação, mas não encontra como acioná-la na interface. É semelhante ao usuário saber o que dizer, mas não encontrar palavras para tal. Essa etiqueta pode indicar uma escolha

inadequada de organização ou expressão dos signos de interface. Presumindo-se que o usuário saiba qual ação deseja realizar na interface, um sintoma típico dessa ruptura de comunicação ocorre quando o usuário navega por vários elementos de interface. Ele abre e fecha menus e diálogos, e varre os elementos da interface com o cursor do mouse. Depois de navegar pela interface e encontrar o elemento que deseja, o usuário pode atuar sobre ele com os dispositivos de entrada (e.g., clica com o mouse, pressiona as teclas do teclado etc.) ou simplesmente examinar os elementos de interface a fim de obter a informação desejada. Em geral, o usuário inicia a busca guiado pela sua interpretação dos signos codificados na interface. Por exemplo, ele abre inicialmente o item de menu que interpreta como sendo mais próximo ao desejado, depois o segundo mais próximo, e assim sucessivamente. Quando essa busca temática não funciona, ele pode passar para uma busca mais aleatória e exaustiva, percorrendo toda a interface. Quanto menor for o tempo e os passos necessários para o usuário encontrar o que deseja, menor será a gravidade dessa ruptura.

A etiqueta “**E agora?**” é empregada quando o usuário não sabe o que fazer em determinado momento para concluir a tarefa, e procura descobrir qual deve ser o seu próximo passo. Em geral, essa ruptura de comunicação ocorre quando, na interpretação do usuário, os signos da interface aos quais ele tem acesso no momento não contribuem para avançar em direção ao alcance do seu objetivo. Como o usuário não consegue formular a próxima intenção de comunicação, o sintoma típico é navegar pelos elementos da interface de forma sequencial ou aleatória para tentar obter alguma dica que lhe permita formular uma intenção e identificar o próximo passo a ser executado. Nesse caso, costuma ser difícil definir alguma relação entre o passo anterior e o seguinte, pois o usuário parece estar perdido. Ele abre e fecha diálogos, varre menus e barras de ferramentas, e lê sistematicamente as dicas, instruções e avisos em busca de uma orientação sobre qual deve ser o seu próximo passo.

Embora os sintomas associados às etiquetas “E agora?” e “Cadê?” sejam parecidos, eles representam rupturas diferentes. No caso de “Cadê?”, o usuário *sabe* o que quer fazer. Porém, no caso da etiqueta “E agora?”, o usuário *não sabe* o que deve fazer para concluir a tarefa. Sempre que os avaliadores perceberem esses sintomas durante uma observação de uso e estiverem em dúvida sobre qual ruptura de comunicação realmente ocorreu, eles devem esclarecer essas dúvidas na entrevista pós-teste.

A etiqueta “**O que é isto?**” é usada quando o usuário não consegue interpretar o significado dos signos estáticos e dinâmicos codificados na interface. Pode indicar o uso de um código expressivo inadequado, que não seja familiar ao usuário. O sintoma típico é o usuário navegar pela interface procurando por alguma dica, aviso ou explicação que explique o significado codificado dos signos não compreendidos por

ele. Por exemplo, o usuário pode parar o cursor sobre ícones e botões de comando esperando ver uma dica explicativa, ou pode acionar um menu ou botão de comando apenas para verificar os efeitos dessa ação.

Os sintomas da etiqueta “O que é isto?” podem se manifestar também durante uma ruptura etiquetada como “Cadê?” ou “E agora?”. A intenção de comunicação do usuário diferencia esses tipos de rupturas. Se o usuário estiver apenas explorando a interface para aprender os significados nela codificados, tratam-se de casos isolados de “O que é isto?”. Caso contrário, pode ser uma combinação de “O que é isto?” com um “Cadê?” (caso o usuário saiba o que está procurando) ou com um “E agora?” (caso o usuário ainda não saiba o que procurar).

A etiqueta “**Epa!**” representa uma situação em que o usuário cometeu um equívoco, percebe o engano rapidamente e busca desfazer os resultados da ação indesejada. Essa etiqueta pode indicar uma ambiguidade na expressão do signo que o usuário utilizou e o levou ao equívoco. A recuperação de um equívoco pode ser rápida, como cancelar logo um diálogo acionado por engano, sem mesmo interagir com ele, possivelmente causado pela semelhança entre dois itens de menu. Também pode ser o acionamento de um comando desfazer (*undo*) imediatamente após realizar a ação equivocada. Pode ainda exigir um caminho de interação maior e mais complexo, como editar um registro que acabou de criar porque percebeu que digitou algo errado. Quanto maior o esforço e tempo necessários para desfazer o engano cometido, maior será a gravidade dessa ruptura de comunicação.

A etiqueta “**Onde estou?**” é utilizada quando o usuário tenta dizer algo que o sistema é capaz de “entender” (i.e., reagir adequadamente) em um outro contexto, diferente do atual. Sintomas comuns dessa ruptura de comunicação ocorrem quando o usuário tenta ativar ações desabilitadas (e.g., tentar acionar um botão de comando que esteja desabilitado momentaneamente) ou interagir com signos que são apenas de exibição (e.g., tentar editar um texto em modo de pré-visualização ou em uma caixa de texto desativada). Essas operações poderiam ser realizadas sobre esses mesmos elementos de interface em outros contextos de uso. Nesses casos, o usuário demonstra estar confuso em relação ao contexto atual e sobre o que é possível fazer no momento, pois sua interpretação dos signos de interface não corresponde aos significados nela codificados para aquele contexto.

A etiqueta “**Ué, o que houve?**” é usada quando o usuário não percebe ou não comprehende as respostas do sistema decorrentes de uma ação ou evento anterior. Essa etiqueta pode indicar uma ambiguidade na expressão do signo que o designer utilizou para comunicar a resposta do sistema ou falta de familiaridade do usuário com essa expressão. Nesse caso, é comum o usuário repetir a operação realizada.

Também é possível perceber essa ruptura de comunicação quando as ações posteriores do usuário são inconsistentes com as respostas do sistema.

A etiqueta “**Por que não funciona?**” representa uma situação na qual o usuário esperava obter determinados resultados do sistema e não entende por que o sistema produziu os resultados diferentes do esperado. Nesse caso, o usuário percebe os resultados do sistema em decorrência de suas ações, mas não se conforma em obter resultados diferentes do esperado. Ele acredita ter feito as coisas certas, então costuma repetir suas ações com a esperança de identificar o problema que gerou resultados inesperados para poder corrigi-lo. Essa etiqueta pode indicar uma falta de correspondência entre a visão do designer e a expectativa do usuário sobre os efeitos de uma ação do usuário na interface ou sobre como um objetivo pode ser alcançado.

A repetição de ações do usuário pode ser etiquetada como “Ué, o que houve?” ou “Por que não funciona?”. A diferença entre essas rupturas de comunicação depende do que o usuário percebeu e compreendeu das respostas do sistema. Na etiqueta “Ué, o que houve?”, o usuário nem chega a perceber ou compreender as respostas do sistema. Já na etiqueta “Por que não funciona?”, o usuário percebeu e compreendeu as respostas do sistema, mas não se conformou com o resultado encontrado.

A etiqueta “**Assim não dá**” é usada quando o usuário interrompe e abandona um caminho de interação com vários passos por considerá-lo improutivo. Quando o usuário percebe estar engajado num caminho de interação que não contribui para a conclusão da tarefa, ele costuma interrompê-lo subitamente, desfazer as ações realizadas nesse caminho improutivo, e iniciar um caminho diferente para concluir sua tarefa. Essa etiqueta pode indicar uma ambiguidade na expressão de uma sequência de signos utilizados pelo usuário e pelo preposto do designer.

As rupturas de comunicação representadas por “Assim não dá” e “Epa!” se assemelham pelo abandono de caminhos de interação. No primeiro caso, o usuário abandona uma sequência de ações geralmente longa, com custo maior de recuperar um caminho produtivo. No segundo, o usuário abandona rapidamente uma ação isolada, com um custo menor de recuperar um caminho produtivo.

A etiqueta “**Vai de outro jeito**” é usada quando o usuário não conhece o caminho de interação preferido pelo designer (geralmente mais curto e simples) ou não consegue percorrê-lo, e então é *obrigado* a seguir por um outro caminho de interação. Essa etiqueta pode indicar uma falta de correspondência entre a visão do designer e a expectativa do usuário sobre como um objetivo do usuário pode ser alcançado. Por exemplo, num editor de texto, o usuário pode formatar individualmente cada parágrafo por desconhecer que o sistema oferece estilos que podem ser aplicados a diversos parágrafos, de forma consistente. Ou ele tenta utilizar estilos, não obtém o

resultado esperado e então prossegue para a formatação manual. Para o avaliador conhecer o caminho preferencial do designer, ele pode consultar a ajuda on-line, a documentação do sistema, e, se possível, o próprio designer.

A etiqueta “**Não, obrigado!**” é utilizada quando o usuário *decide* seguir por um caminho não preferido pelo designer, mesmo conhecendo o caminho preferido e sabendo percorrê-lo. Ela pode indicar uma falta de compreensão do designer sobre as formas preferenciais de o usuário alcançar um objetivo ou idiossincrasias do usuário que revelam a necessidade de mecanismos de customização (de Souza, 2005a). O avaliador pode perceber que o usuário *conhece* o caminho preferido pelo designer se o usuário o percorre pelo menos uma vez ou se ele o menciona explicitamente na entrevista pós-teste. Num editor de textos, por exemplo, o usuário pode dispensar a operação de numeração automática que já conhece por achar mais simples inserir os números manualmente.

A diferença entre as etiquetas “Não, obrigado!” e “Vai de outro jeito” depende de o usuário estar ou não ciente dos caminhos de interação oferecidos e preferenciais. No primeiro caso, o usuário *conhece* o caminho preferido pelo designer, mas *decide* seguir por outro. No segundo, o usuário *não conhece* o caminho preferido pelo designer, e por isso *tem de* percorrer um outro.

A etiqueta “**Pra mim está bom**” é usada quando o usuário equivocadamente acredita que concluiu a tarefa, sem, no entanto, tê-la concluído com sucesso. Nesse caso, o usuário tipicamente dá por encerrada a tarefa, e relata na entrevista pós-teste que a concluiu com sucesso. Esse equívoco é geralmente causado por uma resposta do sistema com conteúdo ou expressão inadequados.

A etiqueta “**Socorro!**” é usada quando o usuário consulta a ajuda on-line ou outras fontes de informação e explicação (o manual do usuário, os avaliadores etc.) para concluir as tarefas. Isso ocorre porque o usuário não consegue interpretar os signos estáticos e dinâmicos codificados na interface, e precisa recorrer aos signos metalinguísticos, que descrevem todos os signos e explicam como utilizá-los.

A etiqueta “**Desisto**” é usada quando o usuário explicitamente admite não conseguir concluir uma tarefa (ou subtarefa) e desiste de continuar tentando. O sintoma típico é o usuário abandonar o cenário de tarefa atual sem tê-la concluído e passar para o próximo cenário de tarefa. Nas etiquetas “Desisto” e “Para mim está bom”, o usuário interrompe a interação antes de concluir a tarefa com sucesso. A diferença é que, no primeiro caso, ele sabe que não concluiu a tarefa, e no segundo, acredita erroneamente que concluiu a tarefa.

As entrevistas pré e pós-teste, as anotações dos avaliadores e os demais registros obtidos durante as sessões de interação auxiliam o avaliador na etiquetagem dos ví-

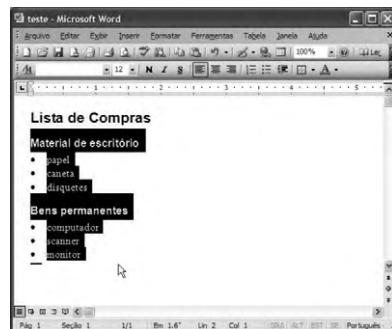
deos de interação. Em particular, esses dados são importantes quando o avaliador fica em dúvida sobre qual etiqueta utilizar, ou qual ruptura de comunicação ocorreu em uma determinada parte do vídeo de interação. Por exemplo, na entrevista pós-teste, o avaliador pode perguntar ao participante qual caminho ele acredita que seria o preferido pelo designer e, assim, distinguir se ocorreu uma ruptura de comunicação indicada pela etiqueta “Não, obrigado!” ou pela etiqueta “Vai de outro jeito”.

No final da etiquetagem, o avaliador terá uma lista de etiquetas para cada vídeo de interação. Cada etiqueta deve estar associada a um trecho do vídeo e pode estar acompanhada de anotações do avaliador (Exemplo 10.9).

Exemplo 10.9 – Etiquetagem de um vídeo de interação

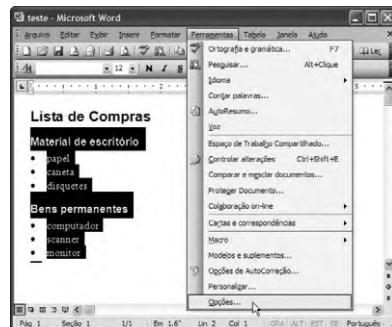
Considere a tarefa de ordenar uma lista estruturada de compras no Microsoft® Word. A sequência de telas a seguir ilustra instantâneos da interação do usuário com o editor de texto, juntamente com anotações sobre as ações do usuário e etiquetas de comunicabilidade associadas aos momentos interpretados como rupturas de comunicação.

Usuário seleciona os itens que deseja ordenar, esperando que o editor preserve os agrupamentos definidos pelos títulos.



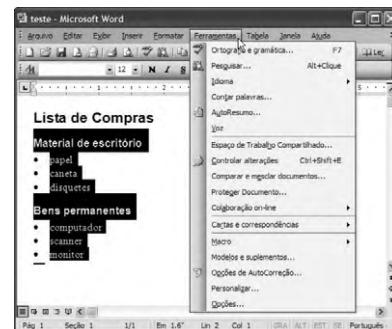
Decide procurar em *Ferramentas > Opções*.

Etiqueta: *Cadê...?* (temático)



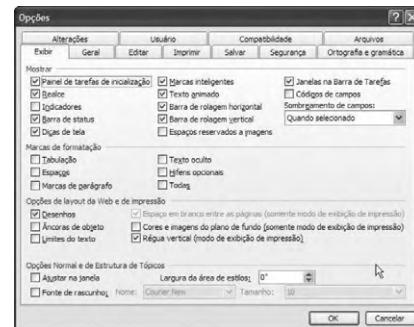
Ele procura uma opção de ordenação no menu *Ferramentas*, mas não a encontra.

Etiqueta: *Cadê...?* (temático)



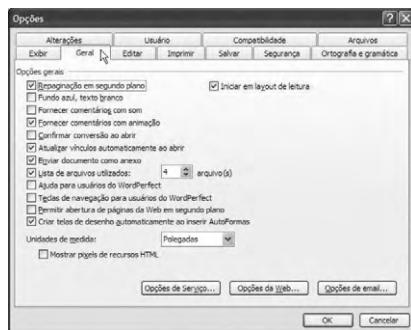
Examina a guia *Exibir*.

Etiqueta: *Cadê...?* (temático)



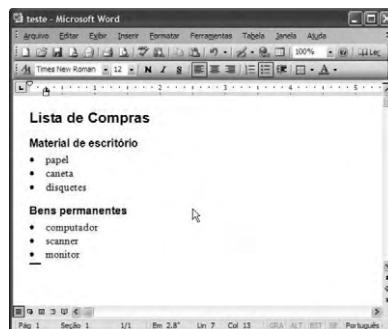
Examina também a guia *Geral*, em vão.

Etiqueta: *Cadê...?* (temático)



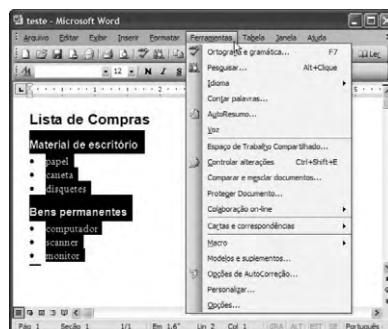
Para fechar o menu, o usuário clica na área em branco do documento, acidentalmente desfaz a seleção dos itens que quer ordenar e rapidamente os seleciona de novo.

Etiqueta: *Epa!*



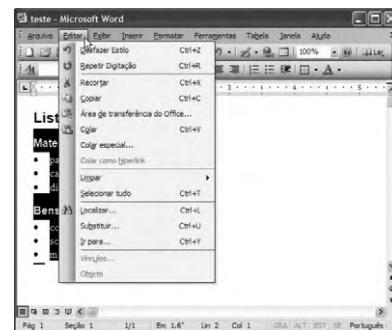
Inconformado, o usuário acessa novamente o menu *Ferramentas*.

Etiqueta: *Por que não funciona? (Por que não está aqui?)*



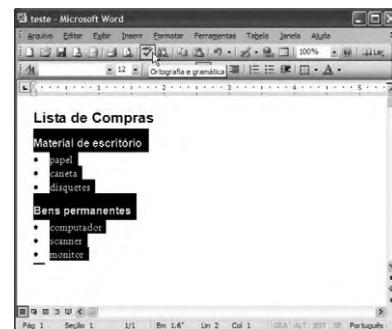
Ele decide explorar agora o menu *Editar*.

Etiqueta: *Cadê...?* (temático)



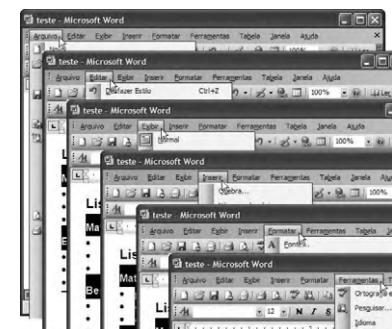
Ele examina a barra de ferramentas, mantendo o mouse sobre alguns elementos para aguardar a dica correspondente.

Etiqueta: *O que é isto?*



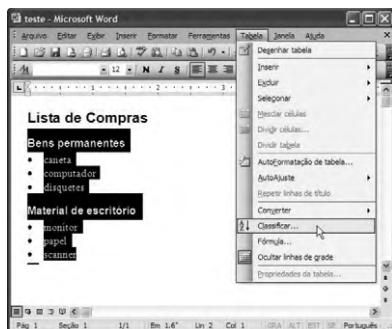
Determinado a encontrar a ferramenta de ordenação, o usuário examina cada menu, sequencialmente.

Etiqueta: *Cadê...?* (sequencial)



Finalmente, encontra um item *Classificar...* no menu *Tabela*. Ele o aciona e confirma a janela de diálogo apresentada pelo sistema.

Não há ruptura e, portanto, não há etiqueta.



Satisfeita, ele declara a tarefa concluída com sucesso. Entretanto, o editor não respeitou a estrutura dos itens ao ordená-los.

Etiqueta: *Pra mim está bom.*



Na atividade de *consolidação dos resultados*, o avaliador interpreta o significado do conjunto de todas as etiquetas nos vídeos de interação, ou seja, ele julga a qualidade da comunicação da metamensagem em função das rupturas de comunicação observadas do ponto de quem a recebe. Para atribuir significado às etiquetas, o avaliador deve considerar os seguintes fatores (de Souza, 2005a, p. 137):

- a frequência e o contexto em que ocorre cada etiqueta (por participante, por tarefa, ou em toda a interação), que auxiliam o avaliador a identificar problemas recorrentes ou sistemáticos na metacomunicação;
- sequências de etiquetas (por participante, por tarefa, ou em toda a interação), que podem indicar uma ruptura comunicativa de maior alcance, envolvendo diferentes signos de interface e requerendo mais tempo ou esforço para o usuário se recuperar e retomar um caminho de interpretação produtivo;
- o nível dos problemas relacionados aos objetivos dos usuários (operacional, tático ou estratégico);
- outras ontologias ou classes de problemas de IHC oriundas de outras teorias, abordagens e técnicas que podem enriquecer a interpretação do avaliador.

A frequência com que as etiquetas ocorrem tende a mudar conforme o usuário ganha experiência de uso (Prates *et al.*, 2000b). Por exemplo, o número de etiquetas “Cadê?” e “O que é isto?” costuma diminuir à medida que o usuário aprende a utilizar o sistema (de Souza, 2005a).

O contexto em que as etiquetas ocorrem pode apontar inconsistências entre os caminhos da interpretação do usuário e do designer causadas por problemas na di-

ferenciação de contextos de interação. Tais inconsistências podem revelar a oportunidade ou necessidade de permitir que os usuários se expressem ou realizem ações num contexto em que atualmente isso não é permitido.

A sequência de etiquetas, principalmente quando se repete algumas vezes, fornece ao avaliador informações relevantes para identificar problemas nos caminhos interpretativos do usuário. Por exemplo, uma sequência das etiquetas “Cadê?”, “Assim não dá” e “Desisto” indica um sério problema de metacomunicação. Nesse caso, o usuário procurou expressar uma intenção correta, e somente depois de vários passos percebeu que estava num caminho improdutivo e acabou desistindo de expressar sua intenção na interface.

Os problemas nesse processo de interpretação do usuário podem ser classificados em três níveis: operacional, tático e estratégico (de Souza, 2005a, p. 125; de Souza *et al.*, 2000). Problemas **operacionais** ocorrem na expressão de uma fala individual do usuário, ou na execução de uma ação. Os problemas **táticos** ocorrem na expressão de uma sequência de falas, ou na execução de uma sequência de ações do usuário, visando alcançar determinado objetivo. Já os problemas **estratégicos** ocorrem na própria definição dos objetivos do usuário. Problemas operacionais podem causar problemas táticos e estratégicos, assim como problemas táticos podem causar problemas estratégicos. Por exemplo, se o usuário encontra um problema operacional por não conseguir expressar uma intenção de comunicação (etiqueta “Cadê?”), ele também terá um problema tático por ter dificuldade de iniciar ou continuar uma sequência de falas (ou ações) para alcançar determinado objetivo, e possivelmente ter um problema estratégico por considerar que o sistema não oferece suporte ao objetivo em questão, quando, na verdade, ele o faz. Os problemas operacionais costumam ser relativamente mais fáceis de resolver do que problemas táticos e estratégicos. Já os problemas estratégicos costumam ser mais graves porque indicam falhas completas na comunicação designer-usuário.

A interpretação da etiquetagem dos vídeos auxilia o avaliador a decidir se houve problemas na recepção da metamensagem (i.e., se existem problemas de comunicabilidade no sistema avaliado). Caso existam, o avaliador terá condições de dizer não apenas *quais* são os problemas, mas também *por que* eles ocorreram. Já a ausência de etiquetas evidencia que os participantes receberam a metamensagem corretamente durante a avaliação.

Quando a comunicação do usuário com o preposto do designer consegue obter um efeito coerente e consistente com a intenção do usuário, dizemos que a comunicação usuário-sistema foi bem-sucedida. Entretanto, se o efeito obtido for incoerente ou inconsistente com a intenção do usuário, então ocorreram falhas na comunicação,

sejam elas percebidas ou não pelo usuário. As seguintes categorias de ruptura na comunicação ajudam a explicar essas falhas:³

- o usuário não consegue *expressar* o significado pretendido;
- o usuário *escolhe o modo errado de expressar* o significado pretendido;
- o usuário não consegue *interpretar* o que o sistema expressa;
- o usuário *escolhe a interpretação errada* para o que o sistema expressa;
- o usuário não consegue sequer *formular uma intenção de comunicação*.

A Tabela 10.6 apresenta a classificação das falhas de comunicação entre o usuário e o preposto do designer, de acordo com as etiquetas de rupturas de comunicação (de Souza, 2005a, p. 138; de Souza e Leitão, 2009, p. 43–46).

Tabela 10.6 Classificação das falhas de comunicação usuário–sistema representadas pelas etiquetas do método de avaliação de comunicabilidade (adaptada de Souza, 2005a, p.138 ; de Souza e Leitão, 2009, p. 43–46).

Falhas de comunicação completas: efeito obtido é inconsistente com a intenção comunicativa do usuário		
aspecto semiótico	característica específica	etiqueta
O usuário termina uma semiose malsucedida, mas não inicia outra para obter o resultado esperado,	porque, mesmo percebendo que não obteve o resultado esperado, não possui mais recursos, capacidade ou vontade de continuar tentando.	Desisto.
	porque não percebe que não obteve o resultado esperado.	Para mim está bom...
Falhas de comunicação parciais: o efeito obtido é somente parte do efeito pretendido de acordo com a intenção do usuário		
aspecto semiótico	característica específica	etiqueta
O usuário abandona uma semiose antes de obter o resultado esperado, e inicia outra com o mesmo propósito,	porque, embora entenda a solução de IHC proposta, prefere seguir por outro caminho no momento.	Não, obrigado.
	porque não entende a solução de IHC proposta.	Vai de outro jeito.

3 http://www.id-book.com/casestudy_14-1_2.htm.

Falhas de comunicação temporárias: o efeito parcial do processo de interpretação (semiose) e de comunicação (interação) do usuário é inconsistente e incoerente com sua intenção de comunicação		
aspecto semiótico	característica específica	etiqueta
O usuário interrompe temporariamente sua semiose,	porque não encontra uma expressão apropriada para sua intenção de comunicação.	Cadê?
	porque não percebe ou não entende a expressão do sistema (preposto do designer).	Ué, o que houve?
	porque não consegue formular sua próxima intenção de comunicação.	E agora?
O usuário percebe que seu ato comunicativo não foi bem-sucedido,	porque percebeu que havia “falado” algo no contexto errado.	Onde estou?
	porque percebeu que havia “falado” algo errado.	Epa!
	porque não obteve o resultado esperado depois de conversar com o sistema (preposto do designer) por algum tempo, alternando vários turnos de fala com ele.	Assim não dá.
O usuário procura compreender o ato comunicativo do sistema (preposto do designer)	através da metacomunicação implícita.	O que é isto?
	através da metacomunicação explícita.	Socorro!
	testando várias hipóteses sobre o significado do que o sistema comunicou.	Por que não funciona?

Depois da interpretação das etiquetas, o avaliador deve elaborar o **perfil semiótico** do sistema avaliado para identificar e explicar seus problemas de comunicabilidade, bem como informar seu reprojeto da interface de modo a corrigi-los. O perfil semiótico é elaborado através da reconstrução da metamensagem do designer tal como ela foi recebida pelo usuário. A paráfrase da metamensagem deve ser usada como um modelo (*template*) a ser preenchido (de Souza, 2005a, p.25):

Este é o meu entendimento, como designer, de **quem você, usuário, é**, do que aprendi que você **quer ou precisa fazer**, de **que maneiras prefere fazer**, e **por quê**. Este, portanto, é o sistema que projetei para você, e esta é **a forma como você pode ou deve utilizá-lo** para alcançar uma gama de objetivos que se encaixam nesta visão.

Essa paráfrase permite definir um conjunto de perguntas que guiam a reconstrução da metamensagem. A etiquetagem dos vídeos e as respectivas interpretações do ava-

liador fornecem ao avaliador evidências para responder as seguintes perguntas (de Souza e Leitão, 2009, p. 47):

- Quem o designer pensa ser o usuário do produto por ele projetado? Ou seja, quem são os usuários destinatários da metamensagem do designer? Quais são seus perfis, incluindo características e valores? Responder essa pergunta ajuda a julgar a correspondência (ou falta de correspondência) entre os usuários presumidos pelo designer e aqueles que utilizam o sistema.
- Quais são os desejos e as necessidades dos usuários, na visão do designer? Como a metacomunicação do designer privilegia certos desejos e necessidades em detrimento a outros? Responder essa pergunta ajuda a julgar a correspondência (ou falta de correspondência) entre o que o designer quis comunicar com o seu design e o que os usuários entendem e fazem com ele.
- Na visão do designer, de que maneiras os usuários preferem fazer o que desejam e precisam, onde, quando, e por quê? Os usuários podem escolher diferentes formas de comunicação com o sistema? Responder essa pergunta ajuda o designer a justificar os sistemas de significação utilizados e julgar se suas decisões são compatíveis com a visão de mundo dos usuários.
- Qual foi o sistema que o designer projetou para os usuários, e como eles devem utilizá-lo? Quão bem a expressão e o conteúdo da metacomunicação estão sendo transmitidos aos usuários?
- Qual é a visão de design? Quão bem a lógica de design (*design rationale*) é compreendida (e aceita) pelos usuários?

Conforme o avaliador responde essas perguntas, ele pode comparar o que o designer pretendia comunicar com as evidências de como os usuários interpretaram o que foi comunicado. Um sistema com boa comunicabilidade significa que o designer conseguiu comunicar bem a metamensagem para o usuário, através da interface do sistema.

Na atividade de *relato dos resultados*, o avaliador deve descrever (veja Seção 9.7.5):

- os objetivos da avaliação;
- uma breve descrição do método para auxiliar o leitor a compreender como os resultados foram obtidos;
- o número e o perfil dos avaliadores e dos participantes;
- as tarefas executadas pelos participantes;

- o resultado da etiquetagem, em geral contabilizando as etiquetas por usuário e tarefa;
- os problemas de comunicabilidade encontrados;
- sugestões de melhorias;
- o perfil semiótico do sistema.

10.2.3 Prototipação em Papel

O método de **prototipação em papel** (Snyder, 2003) avalia a usabilidade de um design de IHC representado em papel, através de simulações de uso com a participação de potenciais usuários. Simular o uso em papel é um modo rápido e barato de identificar problemas de usabilidade antes mesmo de construir uma solução de IHC executável. Sendo assim, esse método é uma opção interessante para uma avaliação formativa junto aos usuários, principalmente para comparar alternativas de design. Ele permite avaliar facilmente soluções parciais, que não cobrem toda a interface com usuário, e soluções de baixa e média fidelidade, que ainda não definem todos os detalhes da interface.

Com tudo preparado, o avaliador convida usuários para executarem algumas tarefas com apoio do sistema simulado em papel. Durante a simulação, os usuários falam, fazem gestos ou escrevem para manifestar como desejam interagir com o sistema. Um avaliador atua como “computador” para simular em papel a execução do sistema e expressar suas reações em resposta às ações do usuário. Essas experiências de uso simuladas permitem identificar as partes da interface que funcionam bem e aquelas que apresentam problemas de usabilidade. A Tabela 10.7 sumariza as atividades do método de prototipação em papel.

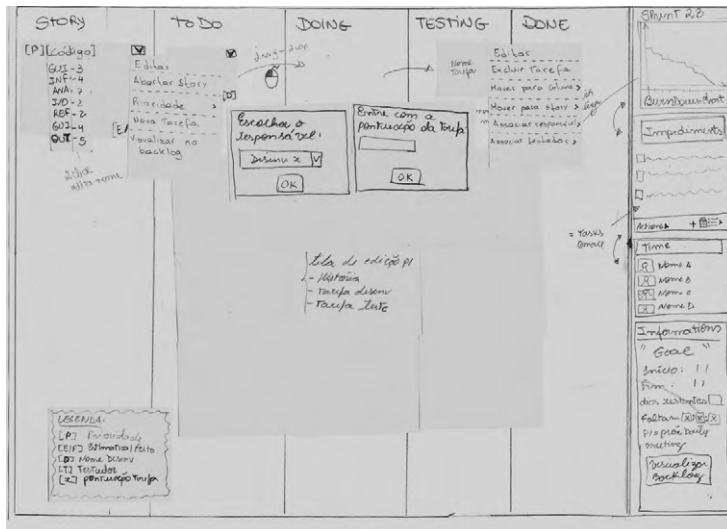
Na atividade de *preparação*, além do material de apoio elaborado na maioria das avaliações com usuários (veja Seção 9.7.2), o avaliador deve elaborar protótipos em papel. Representamos as telas do sistema em papel, em geral desenhadas à mão livre e sem nos preocuparmos com detalhes de interface que não sejam relevantes para a avaliação. A intenção é representar e destacar os elementos principais da interface com os quais o usuário vai interagir durante a simulação da interação. Além das telas estáticas, o avaliador também deve preparar outros pedaços de papel com partes da interface que se modificam durante a interação, como menus, dicas sobre elementos de interface, itens de alguma lista, resultados de busca e diálogos, por exemplo. O que for possível prever deve ser preparado antes das simulações de uso. O que não for possível será desenhado no papel durante as simulações.

Tabela 10.7 Atividades do método de prototipação em papel

prototipação em papel	
atividade	tarefa
Preparação	<ul style="list-style-type: none"> ■ definir tarefas para os participantes executarem ■ definir o perfil dos participantes e recrutá-los ■ criar protótipos em papel da interface para executar as tarefas ■ executar um teste-piloto
Coleta de dados	<ul style="list-style-type: none"> ■ cada usuário deve executar as tarefas propostas interagindo com os protótipos em papel, mediado pelo avaliador
Interpretação	<ul style="list-style-type: none"> ■ avaliador deve <ul style="list-style-type: none"> – listar os problemas encontrados – refinar os protótipos em papel para resolver os problemas mais simples
Consolidação dos resultados	<ul style="list-style-type: none"> ■ priorizar a correção dos problemas não resolvidos ■ sugerir correções
Relato dos resultados	<ul style="list-style-type: none"> ■ relatar os problemas encontrados e sugestões de correção

O Exemplo 10.10 apresenta um esboço de tela utilizado em uma sessão de prototipação em papel. Observe que há anotações em pedaços de papel adesivo que serão manipulados durante a sessão de avaliação.

Exemplo 10.10 – Esboço de tela utilizado em uma sessão de prototipação em papel⁴



⁴ Imagem extraída do Projeto Orientado de Marcela Câmara e Eduardo Ribeiro (2010).

Em seguida, o avaliador prepara o ambiente em que a simulação vai ocorrer, tipicamente uma sala de reunião com mesa e cadeiras, e prepara os equipamentos necessários para registrar os dados, como gravador de voz e câmera de vídeo.

A *coleta de dados* (veja Seção 9.7.3) na prototipação em papel deve ser realizada por pelo menos dois avaliadores: um responsável por simular o comportamento do sistema e outro por observar a experiência de uso. O responsável por simular o sistema busca compreender as ações do usuário sobre o protótipo em papel (e possivelmente as intenções que motivaram tais ações), e modifica a interface conforme o comportamento planejado para o sistema, sem, no entanto, fornecer explicações ou orientações para o usuário. Tudo o que for necessário informar ao usuário deve estar representado na interface do sistema.

No início da sessão, o responsável por simular o comportamento do sistema apresenta o protótipo em papel e explica como estão representados os elementos de interface (*widgets*) e como os participantes podem “interagir” com eles. Por exemplo, os avaliadores podem mostrar o que é um item de menu, um botão de comando ou uma *combobox* e dizer que é possível “clicar” sobre eles (com um dedo, uma caneta ou algum outro instrumento). Depois de apresentar a interface, os avaliadores entregam os cenários ao participante e explicam as tarefas a serem executadas.

O participante, então, começa a interagir com o protótipo em papel da interface do sistema. Para iniciar uma tarefa, o participante pode querer navegar pelos itens de menu. Ele pode manifestar isso de várias formas, tal como dizer em voz alta ou colocar o dedo sobre um item do menu principal. Como resposta a essa ação, o avaliador responsável por simular o comportamento do sistema deve apresentar um pedaço de papel com os subitens do menu principal indicado. Quando o usuário escolher qual menu acionar, ele indica isso para o avaliador, que, por sua vez, modifica a interface com usuário simulando outro comportamento correspondente à ação do usuário, seja abrindo um diálogo sobre a tela atual, substituindo a tela atual por outra etc.

Sempre que necessário, os avaliadores podem modificar ligeiramente a interface para solucionar algum problema simples de usabilidade, como, por exemplo, colocar um botão de comando numa tela de modo a facilitar o acesso a uma operação. Se o problema de usabilidade for mais complexo, os avaliadores podem sugerir que o participante passe para a próxima tarefa solicitada.

Durante a simulação da interação, o observador deve ficar atento a diversos aspectos: partes da interface que funcionaram bem e que funcionaram mal, quais tarefas foram concluídas com sucesso, quais erros foram cometidos, quais comentários foram feitos e quaisquer outros dados que lhe auxiliem a apreciar a qualidade da interface. Como nos demais métodos de observação, depois de terminar a execução das

tarefas os avaliadores podem realizar uma entrevista pós-teste para colher a opinião do participante sobre o protótipo da interface e sugestões de melhorias.

Após cada experiência de uso observada, os avaliadores se reúnem para realizar a atividade de *interpretação* (veja Seção 9.7.4). As anotações dos avaliadores sobre a experiência de uso, as entrevistas pré e pós-teste, e possivelmente o áudio e o vídeo gravados são analisados a fim de identificar problemas de usabilidade no protótipo de interface avaliado. O resultado dessa análise é uma lista de problemas na interface que devem ser corrigidos, além de indicações de partes do sistema que podem ser aperfeiçoadas. Os problemas fáceis de resolver podem ser resolvidos antes da execução da próxima simulação de uso com outro participante. Dessa forma, o protótipo em papel da interface com usuário pode ser aprimorado por ciclos sucessivos de avaliação e reprojeto.

Como durante a simulação de uso existe pelo menos um avaliador responsável pela observação, ele pode começar a interpretar os dados da experiência de uso enquanto observa a atuação do usuário. Portanto, na prática não existe uma separação clara de quando termina a atividade de coleta de dados e quando começa a atividade de interpretação. Pelo menos algumas partes dessas duas atividades são conduzidas em conjunto na prototipação em papel.

Na atividade de *consolidação dos resultados*, os avaliadores verificam quais problemas não puderam ser resolvidos no reprojeto rápido do protótipo de interface (veja Seção 9.7.5). Eles priorizam a correção dos problemas com base na gravidade (o quanto prejudicaram a interação) e frequência em que ocorreram. Por fim, os avaliadores sugerem propostas de correção desses problemas ou de caminhos que podem ser explorados para melhorar a interface.

No *relato dos resultados*, os avaliadores devem comunicar aos interessados:

- os objetivos da avaliação;
- uma breve descrição do método de prototipação em papel;
- o número e o perfil de avaliadores e dos participantes;
- as tarefas executadas pelos participantes;
- uma lista de problemas de usabilidade corrigidos durante os ciclos de avaliação e reprojeto, indicando:
 - local onde ocorreu;
 - fatores de usabilidade prejudicados;
 - descrição e justificativa do problema;
 - correção realizada no protótipo em papel;
 - indicação se o problema voltou a ocorrer depois da correção;

- uma lista dos problemas de usabilidade ainda não corrigidos, indicando:
 - local onde ocorreu;
 - fatores de usabilidade prejudicados;
 - descrição e justificativa do problema;
 - prioridade para correção;
 - sugestões de correção;
- indicações de partes do sistema que podem ser mais bem elaboradas.

10.3 Resumo Comparativo dos Métodos de Avaliação

Os métodos de avaliação apresentados neste capítulo propõem modos próprios de coleta, análise e julgamento do valor de dados relacionados ao uso de sistemas computacionais interativos. Nesta seção comparamos algumas características desses métodos para auxiliar o avaliador na escolha do método mais adequado em cada caso. Eles serão comparados de acordo com *o que* e *quando* se avalia, bem como o *tipo de resultado* produzido.

Cada método de avaliação de IHC é mais adequado para avaliar alguns desses aspectos relacionados ao uso do sistema (*o que avaliar*; Tabela 10.8).

Tabela 10.8 Aspectos geralmente avaliados através de cada método

		apropriação de tecnologia	alternativas de design	conformidade com padrão	problemas de IHC
investigação	entrevistas	+++	+	-	++
	questionários	++	+	-	++
	grupos de foco	++	+++	-	+++
inspeção	avaliação heurística	-	+++	+++	+++
	percurso cognitivo	+	++	-	+++
	inspeção semiótica	-	++	+	+++
observação	estudo de campo	+++	+	-	+++
	teste de usabilidade	+++	++	-	+++
	aval. de comunicabilidade	+++	++	-	+++
	prototipação em papel	+	+++	-	+++

A classificação varia de um método mais adequado (“+++”) até um método inadequado (“-”) para se avaliar o referido aspecto. Para avaliar a forma como os usuários

se apropriam dos sistemas computacionais interativos, utilizamos entrevistas, estudos de campo, testes de usabilidade e avaliação de comunicabilidade. Para comparar e avaliar ideias e alternativas de design, costumamos utilizar grupos de foco, avaliação heurística e prototipação em papel. Já para avaliar a conformidade com um padrão, utilizamos uma avaliação por inspeção, como a avaliação heurística, considerando algum conjunto de recomendações, guia de estilo ou norma. Como era de se esperar, os problemas na interação e na interface são avaliados por todos os métodos de avaliação analisados.

A avaliação de IHC pode ser realizada em diferentes momentos no processo de design e desenvolvimento. Podemos comparar os métodos de avaliação em função da necessidade de existir uma solução de IHC pronta e funcionando. A Tabela 10.9 compara a aplicação formativa (durante o processo de design, antes de uma solução concluída) e somativa ou conclusiva (com uma solução concluída) dos métodos de avaliação apresentados. Os métodos classificados com “++” são empregados mais frequentemente no tipo de avaliação referido. Entrevistas, questionários, grupos de foco, avaliação heurística e de percurso cognitivo podem ser utilizados para avaliação formativa e somativa com benefícios semelhantes. A inspeção semiótica, o estudo de campo, o teste de usabilidade e a avaliação de comunicabilidade são mais apropriados para avaliação somativa, apesar de também poderem ser aplicados para certos tipos de avaliação formativa. Já a prototipação em papel é mais apropriada para a avaliação formativa, por explorar ideias em papel e não numa solução de IHC executável.

Tabela 10.9 Quando cada método de avaliação costuma ser utilizado

		avaliação formativa	avaliação somativa
investigação	entrevistas	++	++
	questionários	++	++
	grupos de foco	++	++
inspeção	avaliação heurística	++	+
	percurso cognitivo	++	+
	inspeção semiótica	+	++
observação	estudo de campo	+	++
	teste de usabilidade	+	++
	aval. de comunicabilidade	+	++
	prototipação em papel	++	+

Os métodos de avaliação de IHC normalmente produzem em alguma medida tanto resultados quantitativos quanto resultados qualitativos. Entretanto, cada método costuma privilegiar um desses dois tipos de resultados. A Tabela 10.10 sumariza os tipos de dados produzidos pelos métodos de avaliação apresentados. Desses métodos, questionários e testes de usabilidade costumam fornecer mais resultados quantitativos do que qualitativos. Os demais métodos de avaliação costumam fornecer mais resultados qualitativos. O avaliador deve escolher um método de avaliação de IHC que forneça resultados adequados ao objetivo da avaliação, sejam eles resultados quantitativos, qualitativos, ou ambos. Por exemplo, se o avaliador estiver interessado em compreender as causas dos problemas na interface, os métodos que fornecem resultados qualitativos tendem a oferecer explicações melhores.

Tabela 10.10 Tipos de dados produzidos por método de avaliação

		quantitativos	qualitativos
investigação	entrevistas	++	+++
	questionários	+++	++
	grupos de foco	++	+++
inspeção	avaliação heurística	+	+++
	percurso cognitivo	+	+++
	inspeção semiótica	+	+++
observação	estudo de campo	++	+++
	teste de usabilidade	+++	++
	aval. de comunicabilidade	+	+++
	prototipação em papel	+	+++

Se houver pouco tempo disponível para executar uma avaliação de IHC, é interessante optar por métodos de avaliação por inspeção, pois eles costumam ser executados mais rapidamente do que os métodos que envolvem usuários. Entretanto, sempre que possível, devemos executar pelo menos uma avaliação empírica, pois através das sessões de interação os usuários sempre revelam aspectos que os avaliadores não conseguiriam prever. É importante oferecermos e aproveitarmos a oportunidade para usuários contribuírem diretamente com o julgamento de valor do sistema.

Idealmente, a avaliação de IHC deveria ser realizada ao longo de todo o processo de design e de desenvolvimento. Dessa forma, seria possível avaliar e corrigir

a solução de IHC conforme ela vai sendo concebida, construída e mantida. Como cada método de avaliação possui características próprias, a execução de métodos de avaliação de IHC complementares é uma prática promissora para o desenvolvimento de sistemas interativos. Desse modo, os resultados da avaliação seriam mais ricos, o reprojeto da solução de IHC seria mais bem informado e, assim, haveria melhores condições de aumentar a qualidade de uso do sistema.

Atividades

1. *Comparação de Métodos de Avaliação de IHC por Inspeção.* Escolha um software de sua preferência e realize avaliações de IHC utilizando métodos de inspeção distintos: uma avaliação heurística, uma avaliação por percurso cognitivo e uma avaliação por inspeção semiótica. Compare o trabalho necessário para executá-las (pessoas envolvidas, tempo de execução, material necessário etc.) e os resultados obtidos (que tipos de conclusões é possível tirar com cada avaliação, qual a relação dessas conclusões com o tipo de avaliação etc.).
2. *Comparação de Métodos de Avaliação de IHC por Observação.* Escolha um software de sua preferência e realize avaliações de IHC utilizando métodos de observação distintos: testes de usabilidade, avaliação de comunicabilidade e prototipação em papel. Compare o trabalho necessário para executá-las (pessoas envolvidas, tempo de execução, material necessário etc.) e os resultados obtidos (que tipos de conclusões é possível tirar com cada avaliação, qual a relação dessas conclusões com o tipo de avaliação etc.).
3. *Comparação de Métodos de Avaliação de IHC Pautados na Engenharia Semiótica.* Escolha um software de sua preferência e realize avaliações de IHC utilizando os métodos de inspeção semiótica e de avaliação de comunicabilidade. Compare as metamensagens reconstruídas pela avaliação e analise as conclusões sobre a emissão e a recepção da metacomunicação.

Referências Bibliográficas

- ACM, *The ACM Code of Ethics and Professional Conduct*. Disponível em <http://www.acm.org/about/code-of-ethics>, 1992.¹
- Alexander, C. *The Timeless Way of Building*. Oxford University Press, 1979.
- Annett, J. & Duncan, K. D. “Task analysis and training design”. *Occupational Psychology*, **41**, pp. 211–221, 1967.
- Annett, J. “Hierarchical Task Analysis”. In: D. Diaper & N. Stanton (eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 67–82, 2003.
- Armitage, J. “Are agile methods good for design?” *Interactions*, **11** (1), pp. 14–23, 2004.
- Aureliano, V. C. O. *eXtreme communication-centered design: um processo ágil para o projeto da interação humano-computador*. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2007.
- Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C. “Basic Concepts and Taxonomy of Dependable and Secure Computing”. *IEEE Transactions on Dependable and Secure Computing* **1** (1); Los Alamitos, CA: IEEE Computer Society, pp. 11–33, 2004.

¹ O acesso mais recente às referências on-line foi realizado em junho de 2010.

- Barbosa, S. D. J. & Paula, M. G. "Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering". In: J. Jorge, N. J. Nunes & J. Falcão e Cunha (eds.) *Interactive Systems Design, Specification, and Verification – 10th International Workshop, DSV-IS 2003*, Funchal, Ilha da Madeira, Portugal, Lecture Notes in Computer Science, **2844**, pp. 16–33, 2003.
- Barbosa, S.D.J.; Silveira, M.S.; Paula, M.G.; Breitman, K. "Supporting a Shared Understanding of Communication-Oriented Concerns in Human-Computer Interaction: a Lexicon-based Approach". *Proceedings of EHCI-DSVIS'04: The 9th IFIP Working Conference on Engineering for Human-Computer Interaction and the 11th International Workshop on Design, Specification and Verification of Interactive Systems*, pp. 271–288, 2004.
- Beck, K. *eXtreme Programming Explained: Embrace Change*. Reading, MA: Addison-Wesley, 2000.
- Bertelsen, O.W. & Bødker, S. "Activity Theory". In: J.M. Carroll (ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, CA: Morgan Kaufmann, pp. 291–324, 2003.
- Bevan, N. "Extending Quality in Use to Provide a Framework for Usability Measurement". *Human-Computer Interaction* **10**, pp. 13–22, 2009.
- Beyer, H. & Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann, 1998.
- Bias, R. & Mayhew, D. *Cost-Justifying Usability*. San Francisco, CA: Morgan Kaufmann Publishers, 2005.
- Blomqvist, S. "Towards a Model for Bridging Agile Development and User-Centered Design". In: A. Seffah, J. Gulliksen & M.C. Desmarais (eds.), *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*. Springer, pp. 219–244, 2005.
- Bødker, S., "Creating Conditions for Participation: Conflicts and Resources in Systems Development". *Human-Computer Interaction*, vol. 11, pp. 215–236, 1996.
- Boehm, B.W. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice Hall, 1981.
- Borchers, J. *A Pattern Approach to Interaction Design*. West Sussex, England: John Wiley & Sons, 2001.
- Button, G. "Studies of Work in Human-Computer Interaction". In: J.M. Carroll (ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, CA: Morgan Kaufmann, pp. 357–380, 2003.
- Buxton, B. *Sketching User Experiences: getting the design right and the right design*. San Francisco, CA: Morgan Kaufmann, 2007.

- Card, S.; Moran, T.P.; Newell, A. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.
- Carroll, J. M. *Making Use: Scenario-Based Design of Human-Computer Interactions*. Cambridge, MA: MIT Press, 2000.
- Carroll, J.M. (ed.) *Scenario-based design: envisioning work and technology in system development*. New York, NY: John Wiley & Sons, 1995.
- Carroll, J.M.; Mack, R.L.; Robertson, S.P.; Rosson, M.B. "Binding Objects to Scenarios of Use". *International Journal of Human-Computer Studies* 41, pp. 243-276. Academic Press, 1994.
- Carroll, J.M. & Rosson, M.B. "Deliberated evolution: Stalking the View Matcher in design space". *Human-Computer Interaction* 6, 3/4, pp. 281-318, 1991.
- Cavano, J. P. & McCall, J. A. "A framework for the measurement of software quality." *Proceedings of the Software Quality Assurance Workshop on Functional and Performance Issues*, pp. 133-139, 1978.
- Conselho Nacional de Saúde. Resolução nº 196/96 sobre pesquisas envolvendo seres humanos. Disponível em <http://conselho.saude.gov.br/docs/Resolucoes/Reso196.doc>, 1996.
- Constantine, L. & Lockwood, L. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Reading, MA: Addison-Wesley, 1999.
- Cooper, A. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Sams Publishing, 1999.
- Cooper, A.; Reimann, R.; Cronin, D. *About Face 3: The Essentials of Interaction Design*. New York, NY: John Wiley & Sons, 2007.
- Courage, C. & Baxter, K. *Understanding your users: a practical guide to user requirements, methods, tools, and techniques*. San Francisco, CA: Morgan Kaufmann Publishers, 2005.
- Cypher, A. (ed.) *Watch What I Do: Programming by Demonstration*. Cambridge, MA: The MIT Press, 1993.
- Danesi, M, & Perron, P. *Analyzing Cultures*. Bloomington, IN: Indiana University Press, 1999.
- de Souza, C.S. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, MA: The MIT Press, 2005a.
- de Souza, C.S. "Semiotic engineering: Bringing designers and users together at interaction time". *Interacting with Computers* 17 (3), pp. 317-341, 2005b.

- de Souza, C.S. & Barbosa, S. D. J. "A Semiotic Framing of End-User Extension and Customization". In: H. Lieberman, F. Paterno & V. Wulf (eds.), *End User Development*. Springer, 2005.
- de Souza, C.S. & Leitão, C.F. *Semiotic Engineering Methods for Scientific Research in HCI*. In: J.M. Carroll (ed.) *Synthesis Lectures on Human-Centered Informatics*. Princeton, NJ: Morgan & Claypool Publishers, 2009.
- de Souza, C.S.; Leitão, C.F.; Prates, R.O.; da Silva, E.J. "The Semiotic Inspection Method". *Anais do VII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais, IHC 2006*, pp. 148-157, 2006.
- de Souza, C. S.; Prates, R. O.; Carey, T. "Missing and declining affordances: Are these appropriate concepts?". *Journal of the Brazilian Computer Society*, SBC. Porto Alegre, RS, v. 7, n. 1, p. 26-34, 2000.
- Delamaro, M.E.; Maldonado, J.C.; Jino, M. *Introdução ao teste de software*. Elsevier, 2007.
- Denzin, N.K. & Lincoln, Y.S. "Introduction: The discipline and practice of qualitative research". In: N.K. Denzin & Y.S. Lincoln (eds.) *The Sage Handbook of Qualitative Research, 3rd edition*. Thousand Oaks, CA: Sage Publications, pp. 1-45, 2005.
- Dey, A.K. "Understanding and Using Context". *Personal and Ubiquitous Computing*, 5 (1), fevereiro de 2001, pp. 4-7, 2001.
- Diaper, D. "Understanding Task Analysis for Human-Computer Interaction". In: D. Diaper & N. Stanton (eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 5-47, 2003.
- Diaper, D. & Stanton, N. *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.
- Dourish, P. & Button, G. "On 'technomethodology': foundational relationships between ethnomethodology and system design". *Human-Computer Interaction*, 13 (4), pp. 395-432, 1998.
- Draper, S. *The Hawthorne, Pygmalion, Placebo and other effects of expectation: some notes*. Disponível em <http://www.psy.gla.ac.uk/~steve/hawth.html>, 2009.
- Dumas, J.S. & Redish, J.C. *A Practical Guide to Usability Testing*, edição revisada. Exeter, UK: Intellect, 1999.
- Eco, U. *A Theory of Semiotics*. Bloomington, IN: Indiana University Press, 1976.
- Engeström, Y. *Learning by expanding: An activity-theoretical approach to developmental research*. Helsinki, Finlândia: Orienta-Konsultit Oy, 1987.
- Ericsson, K.A. & Simon, H.A. *Protocol Analysis: Verbal Reports as Data*, edição revisada. Cambridge, MA: The MIT Press, 1993.

- Ferre, X. "Integration of Usability Techniques into the Software Development Process." *Proceedings of the Bridging the Gaps Between Software Engineering and Human-Computer Interaction Workshop at ICSE 2003*, pp. 28–35, 2003.
- Ferre, X.; Juristo, N.; Moreno, A. M. "Improving Software Engineering Practice with HCI Aspects". *Software Engineering Research and Applications. Lecture Notes in Computer Science*, **3026**, pp. 349–363, 2004.
- Ferre, X.; Juristo, N.; Moreno, A. M. "Which, When and How Usability Techniques and Activities Should Be Integrated" In: A. Seffah, J. Gulliksen & M. C. Desmarais (eds.), *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*. Springer, pp. 173–200, 2005.
- Fitts, P. M. "The information capacity of the human motor system in controlling amplitude of movement". *Journal of Experimental Psychology*, **47**, pp. 381–391, 1954.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.M. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1995.
- Garfinkel, H. *Studies in ethnomethodology*. Englewood Cliffs, NJ: Prentice Hall, 1967.
- Gay, G. & Hembrooke, H. *Activity-Centered Design: An Ecological Approach to Designing Smart Tools and Usable Systems*. Cambridge, MA: The MIT Press, 2004.
- Gibson, J.J. "The theory of affordances". In: R. Shaw e J. Bransford (eds.), *Perceiving, Acting and Knowing*. Hillsdale, NJ: Erlbaum, 1977.
- Gibson, J.J. *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin, 1979.
- González-Calleros, J.M.; Guerrero-García, J.; Vanderdonckt, J.; Muñoz-Arteaga, J. "Towards Canonical Task Types for User Interface Design". *Proceedings of the 2009 Latin American Web Congress, LA-WEB 2009*, IEEE Computer Society, pp. 63–70, 2009.
- Gould, J.D. & Lewis, C. "Design for Usability: Key Principles and What Designers Think". *Communications of the ACM*, **28** (3), pp. 300–311, março de 1985.
- Gulliksen, J.; Göransson, B.; Boivie, I.; Blomkvist, S.; Persson, J.; Cajander, Å. "Key Principles for User-Centred Systems Design". In: A. Seffah, J. Gulliksen & M. C. Desmarais (eds.), *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*. Springer, pp. 17– 33, 2005.
- Hackos, J.T. & Redish, J.C. *User and task analysis for interface design*. New York, NY, John Wiley & Sons, 1998.
- Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong e Verplank. *ACM SIGCHI Curricula for Human-Compu-*

- ter Interaction. ACM SIGCHI Report, ACM, NY. Disponível em <http://old.sigchi.org/cdg>, 1992.
- Hick, W. E. "On the rate of gain of information". *Quarterly Journal of Experimental Psychology* 4, pp. 11–26, 1952.
- Hix, D. & Hartson, H. *Developing User Interfaces: Ensuring Usability Through Product and Process*. New York, NY: John Wiley & Sons, 1993.
- Hollan, J.; Hutchins, E.; Kirsh, D. "Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research". *ACM Transactions on Computer-Human Interaction*, 7 (2), pp. 174–196, junho de 2000.
- Holtzblatt, K.; Wendell, J.B.; Wood, S. *Rapid Contextual Design: a How-to Guide to Key Techniques of User-Centered Design*. San Francisco, CA: Morgan Kaufmann Publishers, 2001.
- Hoover, S.P.; Rinderle, J.R.; Finger, S. "Models and abstractions in design". *Design Studies* 12 (4), pp. 237–245, outubro de 2001.
- Hyman, R. "Stimulus information as a determinant of reaction time". *Journal of Experimental Psychology* 45, pp. 188–196, 1953.
- IEEE. *The IEEE Code of Ethics*. Disponível em <http://www.ieee.org/web/aboutus/ethics/code.html>, 2006.
- ISO 9241-11: *Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on Usability*. ISO, 1998.
- ISO/IEC 9126: *Software engineering – Product quality*. ISO, 1991.
- Jakobson, R. "Linguistics and poetics". In: T. A. Sebeok (ed.), *Style in Language*. Cambridge, MA: The MIT Press, pp. 350–377, 1960.
- John, B.E. "Information Processing and Skilled Behavior". In: J.M. Carroll (ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, CA: Morgan Kaufmann, pp. 55–101, 2003.
- John, B. E.; Bass, L.; Sanchez-Segura, M. I.; Adams, R. J. "Bringing Usability Concerns to the Design of Software Architecture". *Proceedings of EHCI-DSVIS 2004, Lecture Notes in Computer Science*, 3425, pp. 1–19, 2004.
- John, B.E. & Gray, W.D. "CPM-GOMS: an analysis method for tasks with parallel activities", *Proceedings of ACM CHI'95*, pp. 393–394, 1995.
- Johnson, D.G. *Computer Ethics*, 3^a edição. Englewood Cliffs, NJ: Prentice Hall, 2001.
- Jokela, T. & Abrahamsson, P. "Usability assessment of an extreme programming project: close co-operation with the customer does not equal to good usability". *Lecture Notes in Computer Science*, 3009, pp. 393–407, 2004.

- Kammersgaard, J. "Four different perspectives on Human–Computer Interaction". *International Journal of Man-Machine Studies* 28, pp. 343–362, 1988.
- Kaptelinin, V. "Activity theory: Implications for human-computer interaction". In: B. A. Nardi (ed.), *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA: The MIT Press, pp. 103–116, 1996.
- Kieras, D. *Using the Keystroke-Level Model to Estimate Execution Times*, <ftp://www.eecs.umich.edu/people/kieras/GOMS/KLM.pdf>, 1993–2001.
- Kieras, D. "GOMS Models for Task Analysis". In: D. Diaper & N. Stanton (eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 83–116, 2003.
- Kiess, H.O. *Statistical Concepts for the Behavioral Sciences*, 3^a edição. Boston, MA: Allyn & Bacon, 2002.
- Korpela, M. *Activity Analysis and Development in a Nutshell*. Disponível em <http://www.uku.fi/tike/actad/nutshell.html>, 1999.
- Kruger, C. & Cross, N. "Solution-driven versus problem-driven design: strategies and outcomes", *Design Studies* 27 (5), pp. 527–548, setembro de 2006.
- Kuniavsky, M. *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco, CA: Morgan Kaufmann Publishers, 2003.
- Lawson, B. *How Designers Think: The Design Process Demystified*, 4^a edição. Oxford, UK: Architectural Press, 2006.
- Lazar, J. (ed.) *Universal Usability: Designing Information Systems for Diverse User Populations*. New York, NY: John Wiley & Sons, 2007.
- Lazar, J.; Feng, J. H.; Hochheiser, H. *Research Methods in Human-Computer Interaction*. New York, NY: John Wiley & Sons, 2010.
- Lecerof, A. & Paternò, F. "Automatic Support for Usability Evaluation", *IEEE Transactions on Software Engineering* 24 (10), pp. 863–888, outubro de 1998.
- Leontiev, A.N. *Activity, Consciousness, and Personality*. Hillsdale: Prentice-Hall, 1978.
- Levine, D.M.; Stephan, D.F.; Krehbiel, T.C.; Berenson, M.L. *Estatística – Teoria e Aplicações*, 5^a edição (tradução). Rio de Janeiro, RJ: LTC, 2008.
- Lieberman, H. (ed.) *Your Wish is My Command: Programming by Example*. San Francisco, CA: Morgan Kaufmann Publishers, 2001.
- Likert, R. "A Technique for the Measurement of Attitudes". *Archives of Psychology* 140, pp. 1–55, 1932.
- Löwgren, J. & Stolterman, E. *Thoughtful Interaction Design: A Design Perspective on Information Technology*. Cambridge, MA: The MIT Press, 2004.

- Mack, R. e Nielsen, J. (eds.) *Usability Inspection Methods*. New York, NY: John Wiley & Sons, 1994.
- MacKenzie, I. S. *Fitts' law as a performance model in human-computer interaction*. Doctoral dissertation. University of Toronto: Toronto, Canada. Disponível em <http://www.yorku.ca/mack/phd.html>, 1991.
- Marcus, A. *Graphic Design for Electronic Documents and User Interfaces*. New York, NY: The ACM Press, 1992.
- Mayhew, D. *The Usability Engineering Lifecycle: a practitioner's handbook for user interface design*. San Francisco, CA: Morgan Kaufmann, 1999.
- McCall, J.; Richards, P.; Walters, G. *Factors in Software Quality*, 3 NTIS AD-A049-015, 015, 055, novembro de 1977.
- Meister, D. & Rabideau, G.F. *Human Factors Evaluation in System Development*. New York, NY: Wiley, 1965.
- Melo, A. M. & Baranauskas C. C. "Design e avaliação de tecnologia Web acessível". In: *Jornada de Atualização em Informática, Anais do XXV Congresso da SBC*, pp. 1500–1544, 2005.
- Melo, A. M. & Baranauskas, M. C. C. "Design inclusivo de sistemas de informação na Web. *Anais do VII Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2006*", pp. 167–212, 2006.
- Miller, G.A. "The Magical Number Seven, Plus or Minus Two". *The Psychological Review* **63** (2), pp. 81–97, 1956.
- Moran, T. "The Command Language Grammars: a representation for the user interface of interactive computer systems". *International Journal of Man-Machine Studies* **15**, pp. 3–50, 1981.
- Moran, T. & Carroll, J.M. *Design rationale: concepts, techniques, and use*. Routledge, 1996.
- Mullet, K. & Sano, D. *Designing Visual Interfaces: communication-oriented techniques*. Sun Press, 1995.
- Newell, A. & Simon, H.A. *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall, 1972.
- Nicolaci-da-Costa A. M. "A análise de discurso em questão". *Psicologia: Teoria e Pesquisa*, **10** (2), pp. 317–331, 1994.
- Nicolaci-da-Costa, A.M.; Leitão, C.F.; Dias, D.R. "Como conhecer usuários através do Método de Explicitação do Discurso Subjacente (MEDS)". *Anais do VI Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2004*, pp. 49–59, 2004.
- Nielsen, J. *Why You Only Need to Test with 5 Users*, Alertbox. Disponível em <http://www.useit.com/alertbox/20000319.html>, 2000.

- Nielsen, J. "Heuristic Evaluation". In: R. Mack & J. Nielsen (eds.), *Usability Inspection Methods*. New York, NY: John Wiley & Sons, pp. 25–62, 1994a.
- Nielsen, J. "Enhancing the explanatory power of usability heuristics". *Proceedings of ACM CHI'94*, pp. 152–158, 1994b.
- Nielsen, J. *Usability Engineering*. New York, NY: Academic Press, 1993.
- Nielsen, J. "Finding usability problems through heuristic evaluation". *Proceedings of ACM CHI'92*, pp. 373–380, 1992.
- Nielsen, J. e Molich, R. "Heuristic evaluation of user interfaces". *Proceedings of ACM CHI'90*, pp. 249–256, 1990.
- Norman, D.A. *Psychology of Everyday Things*. Basic Books, 1988.
- Norman, D.A. *Emotional Design: why we love (or hate) everyday things*. Basic Books, 2003.
- Norman, D.A., "Cognitive artifacts". In: J. M. Carroll (ed.), *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge, MA: Cambridge University Press, pp. 17–38, 1991.
- Norman, D.A. "Cognitive Engineering". In: D.A. Norman e S.W. Draper (eds.), *User-Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 31–61, 1986.
- Norman, D.A. & Draper, S.W. *User Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum, 1986.
- Oppermann, R. (ed.) *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.
- Palmer, S. & Rock, I. "Rethinking perceptual organization: The role of uniform connectedness". *Psychonomic Bulletin & Review* 1, pp. 29–35, 1994.
- Palmer, S. "Common region: a new principle of perceptual grouping". *Cognitive Psychology* 24, pp. 436–447, 1992.
- Paternò, F. *Model-Based Design and Evaluation of Interactive Applications*. London, UK: Springer-Verlag, 1999.
- Paula, M. G. *Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: construção de um modelo de interação*. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2003.
- Peirce, C.S. In: N. Houser & C. Kloesel (eds.) *The Essential Peirce: Selected Philosophical Writings*, vols. 1–2. Bloomington, IN: Indiana University Press, 1992–1998.
- Perry, M. "Distributed Cognition". In: J.M. Carroll (ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, CA: Morgan Kaufmann, pp. 193–223, 2003.

- Prates, R.O. & Barbosa, S.D.J. "Introdução à Teoria e Prática da Interação Humano Computador fundamentada na Engenharia Semiótica". In: T. Kowaltowski& K. Breitman (orgs.), *Atualizações em informática 2007*. XXVII Congresso da Sociedade Brasileira de Computação, JAI/SBC, 2007.
- Prates, R.O. & Barbosa, S.D.J. "Avaliação de Interfaces de Usuário – Conceitos e Métodos". *Jornadas de Atualização em Informática*, XXIII Congresso da SBC, pp. 245–293, 2003.
- Prates, R. O.; de Souza, C. S.; Barbosa, S. D. J. "A method for evaluating the communicability of user interfaces". *ACM Interactions* 7 (1), New York, NY: ACM Press, pp. 31–38, 2000.
- Prates, R.O.; Barbosa, S.D.J.; de Souza, C.S. "A case study for evaluating interface design through communicability". *Proceedings of the ACM International Conference on Designing Interactive Systems, DIS 2000*, pp. 308–317, 2000.
- Preece, J.; Sharp, H.; Rogers, Y. *Interaction design: beyond human-computer interaction*. New York, NY: John Wiley & Sons, 2002.
- Pressman, R.S. *Engenharia de Software, 5ª edição*. Rio de Janeiro: McGraw-Hill, 2002.
- Pruitt, J. & Adlin, T. *The Persona Lifecycle: keeping people in mind throughout product design*. San Francisco, CA: Morgan Kaufmann Publishers, 2006.
- Reason, J. *Human Error*. Cambridge, MA: Cambridge University Press, 1990.
- Reeves, B. & Nass, C. *The media equation: How people treat computers, television, and new media like real people and places*. New York, NY: Cambridge University Press/CSLI, 1996.
- Rosson, M.B. & Carroll, J.M. *Scenario-Based Development of Human-Computer Interaction*. San Francisco, CA: Morgan Kaufmann Publishers, 2002.
- Rubin, J. *Handbook of Usability Testing*. New York, NY: John Wiley & Sons, 1994.
- Rubin, J. & Chisnell, D. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2ª edição. Indianapolis, IN: Wiley Publishing, Inc., 2008.
- Sacks, H.; Schegloff, E.A.; Jefferson, G. "A Simplest Systematics for the Organization of Turn-Taking for Conversation Conversation," *Language*, 50 (4), pp. 696–735, 1974.
- Salgado, L.C.C.; Bim, S.A.; de Souza, C.S. "Comparação entre os métodos de avaliação de base cognitiva e semiótica". *Anais do Simpósio sobre Fatores Humanos em Sistemas Computacionais, IHC 2006*, pp. 158–167, 2006.
- Santaella, L. *A Teoria Geral dos Signos: Como as linguagens significam as coisas*. São Paulo: Editora Pioneira, 2000.
- Schön, D. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1983.

- Schön, D.A. & Bennett, J. "Reflective conversation with materials". In: T. Winograd (ed.), *Bringing Design to Software*. New York, NY: Addison-Wesley, pp. 171–184, 1996.
- Schegloff, E.A. "Notes on a Conversational Practice: Formulating Place". In: D. N. Sudnow (ed.), *Studies in Social Interaction*, New York, NY: MacMillan, pp. 75–119, 1972.
- Schegloff, E.A. & Sacks, H. "Opening Up Closings", *Semiotica* VIII (4), pp. 289–327, 1973.
- Schneider-Hufschmidt, M.; Kühme, T.; Malinowski, U. (eds.) *Adaptive User Interfaces: Principles and Practice*. Amsterdam, The Netherlands: North Holland, 1993.
- Schwaber, K. & Beedle, M. *Agile Software Development with Scrum*. Englewood Cliffs, NJ: Prentice Hall, 2002.
- Seffah, A.; Gulliksen, J.; Desmarais, M.C. (eds.) *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*. Springer, 2005.
- Seidman, I. *Interviewing as Qualitative Research: a guide for researchers in Education and the Social Sciences*. New York, NY: Teachers College Press, 1998.
- Sharp, H.; Rogers, Y.; Preece, J. *Interaction design: beyond human-computer interaction*, 2^a edição. New York, NY: John Wiley & Sons, 2007.
- Shneiderman, B. *Designing the User Interface*, 3^a ed. Reading, MA: Addison Wesley, 1998.
- Silva, B. S. *MoLIC Segunda Edição: revisão de uma linguagem para modelagem da interação humano-computador*. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2005.
- Silva, B.S.; Netto, O.A.M.; Barbosa, S.D.J. "Promoting a Separation of Concerns via Closely-Related Interaction and Presentation Models". *Proceedings of the Second Latin American Conference on Human-Computer Interaction*, CLIHC 2005, Cuernavaca, Mexico, pp. 170–181, 2005.
- Silveira, M.S.; Prates, R.O. "Uma Proposta da Comunidade para o Ensino de IHC no Brasil". Anais do XV Workshop sobre Educação em Computação, WEI, organizado junto ao XXVII Congresso da SBC, Rio de Janeiro, 2007.
- Silveira, M.S. *Metacomunicação Designer-Usuário na Interação Humano-Computador: Design e Construção do Sistema de Ajuda*. Tese de Doutorado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2002.
- Silveira, M.S.; Barbosa, S.D.J.; de Souza, C.S. "Model-Based Design of Online Help Systems". In: R. Jacob, Q. Limbourg & J. Vanderdonckt (eds.) *Computer-Aided Design of User Interfaces IV*. Kluwer Academic Publishers, pp. 29–42, 2004.
- Simon, H. *The Sciences of the Artificial*. New York, NY: ACM Press, 1981.
- Snyder, C. *Paper Prototyping: the fast and easy way to design and refine user interfaces*. San Francisco, CA: Morgan Kaufmann, 2003.
- Soares, L.F.G.S. & Barbosa, S.D.J. *Programando em NCL*. Campus-Elsevier, 2009.

- Spencer, D. *Card Sorting: designing usable categories*. Brooklyn, NY: Rosenfeld Media, 2009.
- Stephanidis, C. (ed.) *User Interfaces for All: Concepts, Methods, and Tools*. Mahwah, NJ: Lawrence Erlbaum Associates, 2001.
- Stevens, S. S. "On the Theory of Scales of Measurement". *Science* **103** (2684), pp. 677–680, 1946.
- Suchman, L. A. *Plans and Situated Actions: The problem of human-machine communication*. New York, NY: Cambridge University Press, 1987.
- Tidwell, J. *Designing Interfaces: Patterns for Effective Interaction Design*. Sebastopol, CA: O'Reilly Media, Inc., 2005.
- Tognazzini, B. *How user testing saves money*, Ask Tog. Disponível em <http://www.asktog.com/columns/037TestOrElse.html>, 2000.
- Tognazzini, B. *A Quiz Designed to Give You Fitts*, Ask Tog. Disponível em <http://www.asktog.com/columns/022DesignedToGiveFitts.html>, 1999.
- Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*, editado por M. Cole, V. John-Steiner e S. Scribner. Harvard, MA: Harvard University Press, 1978.
- WAI: Web Accessibility Initiative do W3C. Disponível em <http://www.w3.org/WAI>.
- Ware, C. "Design as Applied Perception". In: J.M. Carroll (ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, CA: Morgan Kaufmann, pp. 11–26, 2003.
- Wharton, C.; Rieman, J.; Lewis, C.; Polson, P. "The Cognitive Walkthrough Method: A Practitioner's Guide". In: R. Mack & J. Nielsen (eds.) *Usability Inspection Methods*. New York, NY: John Wiley & Sons, pp. 105–140, 1994.
- Wixon, D. & Wilson, C. "The usability engineering framework for product design and evaluation". In: M.G. Helander, T.K. Landauer, P.V. Prabhu (eds.), *Handbook of Human-Computer Interaction*. Elsevier, Amsterdam, pp. 653–688, 1997.
- Wroblewski, L. *Web Form Design: Filling in the Blanks*. Brooklyn, NY: Rosenfeld Media, 2008.

Índice Remissivo

A

- ação
 - na teoria da atividade 69
 - planejada 61
 - situada 62
 - teoria da 56
- acessibilidade 28, 32
- adaptação 260
- affordance* 26
- afinidade
 - diagrama de 157
- ajuda
 - design do sistema de 257
- análise
 - atividade de 93, 132
 - da conversação 63
- anonimato 140
- aprendizado
 - facilidade de 29
 - na teoria da atividade 73
- arquitetura de sistemas computacionais 11
- artefato
 - cognitivo 56
 - de metacomunicação 78
 - intelectual 83
 - socialmente construído 69
- atividade
 - de análise 132
 - de avaliação 303
 - mediada 71
 - motivada 70
 - teoria da atividade 69
- avaliação
 - atividade de 95, 303
 - dificuldade de (engenharia cognitiva) 54
 - framework DECIDE 313
 - golfo de 56
 - método
 - avaliação de comunicabilidade 344
 - avaliação heurística 316
 - inspeção semiótica 330

percurso cognitivo 322
 prototipação em papel 358
 teste de usabilidade 341
 motivação 286
 na teoria da ação 58
 objetivos 290
 onde realizar 295
 por inspeção 316
 por observação 341
 quando realizar 294
 tipos de método 301

B

brainstorming 155

C

características humanas 11
card sorting 159
 cenário 183
 de interação 210
 design baseado em 112
 tipos de 113
 ciclo de vida
 em estrela 103
 classificação de cartões 159
 cliente
 visão do 7
 cognição
 arquitetura 76
 artefato cognitivo 56
 distribuída 75
 e cultura 76
 engenharia cognitiva 53
 materialização 76
 sistema cognitivo 49
 comunicabilidade 28, 38, 79
 avaliação de 344
 método de inspeção semiótica 330
 comunicação
 design centrado na 117

do preposto do designer 234
 processo de 80
 tecnologias de informação e 2
ConcurTaskTrees 203
 conectividade (Gestalt) 51
 consentimento 140
 termo de 141
 consistência 270
 contexto de uso 10
 contextual
 design 111
 investigação 111, 166
 continuidade
 princípio gestáltico 51
 contradição
 teoria da atividade 72
 controle
 dificuldade de 54
 e liberdade do usuário 267

conversa
 com os materiais 97
 interação como 212
 tópico 212
 conversação
 análise da 63
 cultura
 em cognição distribuída 76
 na teoria da atividade 69

D

dados
 tipos de 134, 298
 deficiência
 e acessibilidade 33
 desenvolvimento
 abordagens de 9
 processo de 11
 visões de 7
 design
 atividade de 92, 208
 baseado em cenários 112
 centrado na comunicação 117
 centrado no usuário 88, 100

contextual 111
dirigido por objetivos 114
dirigido por problema vs. solução 99
espaço de 84
modelo de 60
padrões de 279
perspectivas de 96
princípios de 265
processos de 99
destino comum (Gestalt) 51
diagrama
 de afinidade 157
diálogo
 em MoLIC 240

E

eficácia 29
eficiência 29, 30, 271
 antecipação 272
engenharia
 cognitiva 53
 de software 8
 de usabilidade
 ciclo de vida de Nielsen 104
 semiótica 77
entrevista 145
 roteiro 146
 tipos de 145
epistêmica
 ferramenta 87
erro
 captura de 224
 projetar para o 277
espaço de design de IHC 84
especificação de ações
 na teoria da ação 57
estado do sistema 53
estrela
 ciclo de vida em 103
estudo em IHC
 benefícios 13
 objetos de 10
estudo-piloto 133

ética
 em IHC 138
etnometodologia 67, 68
execução
 golfo de 56
 na teoria da ação 57
experiência do usuário 28, 31
externalização
 na teoria da atividade 70

F

facilidade de aprendizado 29
facilidade de memorização 30
fecho (Gestalt) 51
ferramenta
 epistêmica 87
 perspectiva de 23
Fitts (lei de) 45
foco
 em uma conversa 212
formulário 246

G

Gestalt 50
golfos
 de avaliação e execução 56, 273
GOMS 196
grupos de foco 154
guia de estilo 282

H

Hick-Hyman (lei de) 45

I

imagem do sistema

na engenharia cognitiva 61
 informação e comunicação
 tecnologias de 2
 inspeção
 avaliação por 301
 intenção
 comunicativa 78
 na teoria da ação 57
 interação humano-computador 8
 e engenharia de software 121
 e métodos ágeis 127
 interação usuário-sistema 20
 definição 20
 elementos envolvidos em 18
 modelagem da 229
 perspectivas de 21
 interface 25
 design da 243
 internalização
 na teoria da atividade 70
 interparticipante (análise) 149
 interpretação
 na teoria da ação 58
 intervenção (síntese) 94
 intraparticipante (análise) 150
 investigação
 avaliação por 301
 iterativo
 processo de design 99

K

KLM 198

L

lei de Fitts 45
 lei de Hick-Hyman 45
 Likert, escala de 152
 linguagem
 de comando 243
 interativa única 85

natural 244

M

manipulação direta 247
 mapeamento 54
 mapeamentos naturais 265
 mediação 71
 cultural 69
 mensagem 85
 menus 245
 metacomunicação designer-usuário 78
 métodos ágeis e IHC 127
 MHP 48
 mídia
 perspectiva de 24
 modelo
 de design (engenharia cognitiva) 60
 de interação 229
 de interface 249
 de tarefas 225
 do usuário (engenharia cognitiva) 61
 tarefas 191
 MoLIC 229
 multidisciplinar
 área 12, 123
 equipe 12, 101, 124

O

objetivos
 de análise 133
 design dirigido por 114
 mapa de 214
 teoria da ação 57
 observação
 avaliação por 302
 operações
 na teoria da atividade 69
 orientação a objetos
 na teoria da atividade 72

P

- padronização 270
- par adjacente 64
- parceiro do discurso
 - perspectiva de 23
- percepção
 - cores 52
 - sistema perceptivo 48
 - teoria da ação 58
- perguntas
 - em entrevistas 145
 - em questionários 152
- personas 176
- perspectivas de interação 21
- perturbação
 - na teoria da atividade 72
- pesquisa qualitativa 300
- piloto
 - estudo 133
- plano
 - ação planejada 61
- prevenção
 - apoiada 223
 - ativa 223
 - passiva 223
- princípios
 - na teoria da atividade 71
- privacidade 140
- problema
 - design dirigido pelo 99
- processador humano de informação
 - 48
- processos
 - de desenvolvimento 11
 - de design de IHC 99
- prototipação 251
 - em papel (avaliação) 358
- proximidade (Gestalt) 51
- psicologia aplicada 47

Q

- qualidade
 - de uso 14. (veja usabilidade, comunicabilidade, acessibilidade, experiência de usuário)
- acessibilidade 32
- comunicabilidade 38
- experiência do usuário 28
- usabilidade 28
 - em Computação 8
- questionário 150

R

- racionalismo técnico 96
- receptor 85
- reconhecimento 273
- recuperação
 - apoiada 223
- reflexão em ação 87, 96
- região comum (Gestalt) 51
- requisitos 132
 - técnicas de levantamento de 144
- ruptura comunicativa 223, 235

S

- satisfação do usuário 29, 31
- segurança no uso 31
- semiose 80
- semiótica
 - engenharia 77
 - inspeção 330
- sequência colateral 65
- significação
 - processo de 80
- signo 80, 213
 - conteúdo 218
 - dinâmico 86, 336
 - esquema conceitual 218
 - estático 85, 335

- expressão 256
- metalinguístico 86, 334
- simetria (Gestalt) 51
- similaridade (Gestalt) 51
- simplicidade 267
- síntese
 - em design 93
- sistema
 - perspectiva de 21
- sistema computacional
 - arquitetura de 11
- sistemas computacionais interativos 2
- solução
 - design dirigido pela 99
- stakeholders* 7, 136

T

- tarefa
 - análise de 191
 - CTT 203
 - GOMS 196
 - HTA 192
 - KLM 198
 - análise hierárquica de (HTA) 192
 - modelo adaptado de 225
- tecnologias de informação e comunicação 2
- teoria
 - da ação 56
 - da atividade 69
 - em IHC 73
- TICs 2
- tópico
 - de conversa 212, 230
- triangulação 133

- uso
 - contexto de 10
 - qualidade de 14
- usuário
 - modelo do 61
 - perfil de 174
 - primário 136
 - visão do 7

V

- variáveis
 - físicas 53
 - psicológicas 53
- visibilidade 273

W

- widget* 256
- WIMP 248
- wireframes* 251

U

- usabilidade 28
- engenharia de (Mayhew) 109
- engenharia de (Nielsen) 104