

O Uso de Casos para Apoiar a Reflexão do Designer de IHC

Esse capítulo apresenta uma abordagem de raciocínio baseado em casos para apoiar a reflexão do designer durante o design de IHC. Primeiro, analisamos os conceitos utilizados no design de IHC para, então, propor o conteúdo de um caso de design de IHC e como os casos podem ser indexados e recuperados. Por fim, concluímos o capítulo tecendo algumas considerações sobre o processo de reflexão em ação consultando casos de design existentes.

3.1 Espaço de design de IHC

Antes de definirmos um caso de design de IHC, precisamos analisar o espaço de design de IHC. O nosso ponto de partida é uma interpretação do que é a interação humano-computador e quais são os elementos geralmente envolvidos nela. Elaboramos essa interpretação tendo como base uma situação típica de uso de um sistema computacional interativo, ilustrada na Figura 3.1.

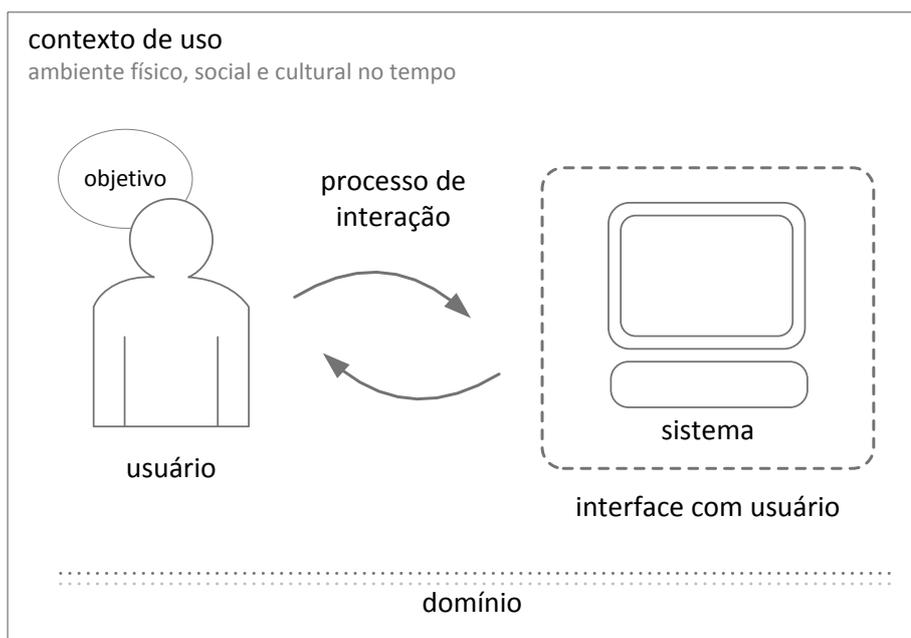


Figura 3.1: O processo de interação humano-computador.

Uma situação de uso pode ser caracterizada como sendo: um *usuário* engajado num processo de *interação* com a *interface* do *sistema*, buscando alcançar um *objetivo* em um *domínio* dentro de um *contexto de uso* (Sharp *et al.*, 2007; Barbosa e Silva, 2010).

3.1.1 Contexto

A interação entre um usuário e um sistema computacional ocorre dentro de um **contexto**. Em Computação, o contexto de uso tem sido investigado principalmente no âmbito de sistemas computacionais sensíveis ao contexto (Schilit *et al.*, 1994; Dey, 2001; Crowley *et al.*, 2002; Coutaz e Rey, 2002; Dourish, 2004; Fonseca e Lewis, 2009; Fonseca, 2010). Dey (2001) define o contexto de uso como sendo:

“qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio o usuário e a própria aplicação.” (p. 5)

Schilit e seus colegas (1994) relatam que:

“contexto engloba mais do que somente o local do usuário, porque outras coisas de interesse também são móveis e mudam. Contexto inclui nível de luminosidade, nível de ruído, conexão à rede, custos de comunicação, largura de banda de rede, e ainda a situação social” (p.85)

Essas definições nos permitem caracterizar o contexto como sendo *um conjunto de características de entidades relevantes à interação usuário-sistema*. As entidades relevantes geralmente discutidas em sistemas sensíveis ao contexto são: ambiente físico e sociocultural, usuário e sistema (também chamado de plataforma), todos situados no tempo. Nós preferimos tratar usuário e sistema como entidades separadas do contexto, para discuti-las adiante em maiores detalhes.

Em IHC, o contexto de uso é normalmente investigado em termos de ambiente físico, ambiente sociocultural, tempo e histórico da interação (Hackos e Redish, 1998; Beyer e Holtzblatt, 1998; Courage e Baxter, 2005; Sutcliffe *et al.*, 2006; Sharp *et al.*, 2007). É importante destacar que essas dimensões são apenas as mais comuns na literatura. Não pretendemos enumerar tudo o que é relevante no contexto de uso para a interação usuário-sistema. O designer de IHC continua

com a responsabilidade de julgar quais entidades do contexto e suas respectivas características são relevantes ou não para cada caso.

O ambiente físico costuma envolver:

- a *localização geográfica* onde a interação ocorre, tal como: país, região, estado, cidade, bairro e rua;
- o *ambiente de atividade*, tal como: trabalho, casa, universidade e trânsito;
- o *nível de luminosidade* que a tela do sistema será exposta, tal como: iluminação baixa, média ou alta;
- o *nível de ruído* do ambiente durante a interação, tal como: ruído baixo, médio ou alto;
- o *nível de atenção* que o usuário pode dedicar à interação no ambiente em questão, tal como: pouca atenção, mediana ou muita;

O ambiente sociocultural inclui normalmente:

- o *idioma* que os usuários compreendem e conseguem se expressar, tal como: Português, Inglês ou Espanhol;
- o *jargão* ou linguagem particular que os usuários de um grupo profissional ou sociocultural costumam utilizar para se comunicarem. Por exemplo, engenheiros e médicos costumam utilizar termos específicos para se comunicarem. Além disso, um jargão pode ser característico de um grupo ou até mesmo de uma pessoa, ou seja, ele pode ser de uso coletivo ou individual;
- como um grupo de pessoas costuma lidar com as *incertezas*. As pessoas podem, por exemplo, aceitar e lidar bem com as incertezas; tolerar as incertezas, apesar de não lidarem muito bem com elas; ou não tolerar as incertezas e não conseguirem lidar com elas;
- a *estrutura* de poder nos grupos sociais envolvidos. A estrutura de poder pode, por exemplo, ser bem definida e respeitada por todos; pode ser definida, apesar de nem sempre respeitadas; ou até não existir estrutura de poder estabelecida;
- a *interação entre pessoas* na situação atual. Por exemplo, é importante saber se as pessoas costumam trabalhar isoladas ou em grupo; ou se elas podem conversar e compartilhar material enquanto realizam suas atividades com o apoio do sistema.

- as *pressões sociais* que as pessoas costumam passar enquanto usam sistemas interativos. Dentre as mais comuns, podemos citar as pressões para: concluir as atividades (ou tarefas) rapidamente, não cometer erros e economizar recursos; e
- os *grupos socioculturais* que os usuários fazem parte. Por exemplo, os usuários podem ser ocidentais, orientais, brasileiros, portugueses, moçambicanos, angolanos, engenheiros, médicos, funcionários da empresa A ou funcionários do departamento D.

O **tempo** geralmente é representado por um *horário* (e.g. manhã, tarde ou noite, durante expediente, horário de almoço ou fora do expediente) e por uma *data* (e.g. dia útil, fim de semana, feriado, início das aulas, aniversário ou data comemorativa). Também pode ser representado por um ou mais intervalos de tempo, por um conjunto de datas, como em ocorrências sazonais (e.g., na primeira semana de aulas), ou relativo a algum evento.

O **histórico da interação** entre o usuário e o sistema geralmente se refere:

- a *sessões de interação* (e.g. sessão atual que começou algum tempo atrás, sessão anterior, ou primeira sessão de uso do sistema),
- a *tarefas executadas* anteriormente (e.g. quais tarefas, quando foram executadas e o número total de execuções),
- aos *conceitos do domínio manipulados* (e.g. quais conceitos foram manipulados por quais tarefas, quando foram manipulados e o número total de vezes), e
- aos *erros cometidos* pelo usuário (e.g. o tipo de erro, a tarefa afetada, o total de erros desse tipo cometidos).

As informações sobre o contexto, em particular o histórico da interação, são fundamentais para o desenvolvimento de interfaces com usuário adaptativas e adaptáveis (Kühme e Malinowski, 1993; Oppermann, 1994; Lieberman *et al.*, 2006). Nesses casos, o sistema precisa processar informações sobre o contexto para modificar adequadamente a interface e para oferecer ao usuário oportunidades adequadas para ele modificar a interface.

3.1.2 Domínio

Um **domínio** de conhecimento e atividade humana, ou simplesmente domínio, diz respeito a um conjunto de *conceitos e relações* que descrevem (parte de) o mundo, de acordo com uma determinada área de conhecimento (Sowa, 2000). Um domínio, portanto, auxilia as pessoas a compreenderem e a se referirem a coisas do mundo, principalmente quando o domínio é incorporado e estruturado por uma linguagem. Por isso, o domínio também costuma ser chamado de *universo do discurso*. Como as atividades humanas manipulam coisas do mundo, naturalmente as pessoas precisam conhecer e utilizar um domínio para se referirem ao que for utilizado durante suas atividades, e para se referirem às atividades propriamente ditas. Sendo assim, o processo de interação que apoia as atividades do usuário também se refere a um determinado domínio. Além disso, o próprio sistema deve ser capaz de processar os conceitos e relações pertencentes ao referido domínio para poder interagir com o usuário através da interface. Espera-se, portanto, que a interface represente o domínio referenciado durante a interação usuário-sistema.

Existe uma distinção básica entre os conceitos que compõem o domínio. Alguns conceitos são considerados como *tipos, categorias* ou *classes* que descrevem características de outros conceitos considerados como *instâncias, exemplares* ou *objetos* (Sowa, 2000). As características dos conceitos do domínio podem ser descritas por *atributos*. Cada atributo pode estar associado a um ou mais valores de algum tipo, possivelmente restrito a um conjunto específico de valores.

Os conceitos do domínio podem estar relacionados, de acordo com um tipo de relação e certas cardinalidades. Existem vários tipos de relações entre conceitos. Alguns tipos comuns são: a generalização (“é um”), agregação (“parte de”) e composição (“possui um”).

O papel do domínio na interação entre pessoas tem sido estudado em diferentes áreas, tais como a Semiótica e Linguística (Peirce, 1931-1958; Eco, 1976; Brown e Yule, 1983). Essas duas áreas em particular investigam os processos de significação e comunicação envolvendo pessoas, outros seres vivos e até artefatos artificiais como o computador. Durante processos de significação, os conceitos, significados ou ideias são codificados em e decodificados de alguma representação. Ou seja, os processos de significação envolvem criar e perceber associações entre conceitos (ou conteúdos) e representações (ou expressões). Já os processos

de comunicação estão relacionados ao modo como os conceitos e representações (geralmente organizados em sistemas de significação, Eco, 1976) são utilizados para alcançarem os propósitos dos interlocutores.

Nos termos da Semiótica, o domínio é representado por um conjunto de *signos*. Um signo é alguma coisa que possui algum significado para alguém (Peirce, 1931-1958), como, por exemplo, palavras, imagens e comportamentos. Sendo assim, conceitos e relações do domínio também são signos porque possuem representações com significados para alguém. Os conceitos ou relações que representam tipos ou classes são chamados de *signos-tipo*, enquanto aqueles que representam instâncias ou objetos são chamados de *signos-token* (Peirce, 1931-1958; Eco, 1976). Por exemplo, um livro pode fazer parte de um pedido de compra online. Ou seja, as classes “livro” e “pedido” podem estar associadas pelo tipo de relacionamento “parte de”. Enquanto que as instâncias “Dom Casmurro” e “pedido 53874” podem estar relacionadas pela instância de relacionamento < Dom Casmurro, parte de, pedido 53874>. No primeiro grupo temos signos-tipo, enquanto que no segundo temos signos-token.

Abordagens semióticas para o domínio têm ganhado importância em IHC. Recentemente, por exemplo, temos assistido a evolução da engenharia semiótica (de Souza, 2005a) como uma teoria de IHC que investiga os processos de significação e comunicação entre usuários e designers mediados por sistemas computacionais. Ou seja, de modo bem simplista, ela investiga, dentre outras coisas, como o domínio pode ser referenciado (e evoluir) durante a interação do usuário com sistema, mediada pela interface.

3.1.3 Usuário

O usuário e o sistema¹ são participantes diretos da interação. Portanto, o conhecimento sobre ambos é fundamental para elaborar um projeto de IHC adequado. O sistema deve ser projetado para atender às necessidades do usuário sempre respeitando as possibilidades e os limites do seu corpo, mente, emoções, cultura e

¹ A engenharia semiótica (de Souza, 2005a) considera o preposto do designer como a parte do sistema que participa do processo de interação. Esta seção considera o sistema como um todo.

posição social. O usuário não deveria fazer muito esforço para se adaptar ao sistema. Pelo contrário, o sistema é quem deveria poder ser adaptado ao usuário.

Primeiro vamos abordar as características do usuário relevantes para a interação, e depois as características do sistema. Estamos interessados apenas na interação individual de um usuário com um sistema. Sendo assim, não consideramos a interação entre usuários através do sistema, como é o foco da área de Sistemas Colaborativos (ou *Computer Supported Cooperative Work*).

A literatura de IHC discute bastante sobre os aspectos do usuário que afetam sua interação com um sistema interativo (Norman e Draper, 1986; Norman, 1988; Nielsen, 1993; Norman, 2003; Dix *et al.*, 2003; Sears e Jacko, 2007; Sharp *et al.*, 2007; Barbosa e Silva, 2010). Os aspectos do usuário geralmente considerados durante o design de IHC são (Hackos e Redish, 1998; Courage e Baxter, 2005; Sutcliffe *et al.*, 2006; Cooper *et al.*, 2007):

- características pessoais (e.g. idade, profissão, papel),
- conhecimentos (e.g. sobre domínio, sobre as tarefas, sobre tecnologia),
- habilidades físicas, perceptivas e cognitivas (e.g. visão, audição, habilidade motora, cognição, alfabetismo),
- experiências (e.g. na execução da tarefas), e
- preferências (e.g. escreve o texto, depois formata, ou usa mais teclado, do que mouse).

As informações sobre o usuário são utilizadas basicamente *durante o processo de design* para apoiar as decisões do designer. Desse modo, a interação e a interface podem ser concebidas e construídas de forma adequada ao usuário que provavelmente vai utilizar o sistema. Além disso, as informações sobre o usuário também podem ser utilizadas *durante o processo de interação*. Isso permite adaptar ainda mais e melhor a interface do sistema às particularidades de cada usuário, acompanhando sua evolução com o tempo (Kühme e Malinowski, 1993; Oppermann, 1994; Lieberman *et al.*, 2006).

3.1.4 Sistema

A participação do sistema durante a interação é determinada pela sua capacidade de perceber, processar e responder adequadamente às ações do usuário na

interface (Dix *et al.*, 2003; Sears e Jacko, 2007; Sharp *et al.*, 2007). Por exemplo, se a conexão com a Internet falhar, o sistema poderia avisar ao usuário, tentar enviar os dados novamente, além de permitir o usuário salvar os dados para transmitir quando a conexão voltar a funcionar.

O comportamento do sistema durante a interação depende, dentre outros fatores, da *plataforma* de hardware e software (e.g. web, desktop, PDA, celular, Windows, MacOs e Linux), dos *dispositivos de entrada* (e.g. teclado, mouse, microfone) e *saída* (e.g. tela, autofalantes, impressora), e da *infraestrutura* disponível no momento (e.g. conexão de rede, largura de banda, espaço para armazenamento, nível de energia). Esses aspectos do sistema costumam ser investigados na área de sistemas sensíveis ao contexto para adaptar o comportamento do sistema e a interface com usuário adequadamente (Dieterich *et al.*, 1993; Calvary *et al.*, 2003; Fonseca, 2010).

Geralmente primeiro se identifica os requisitos funcionais e não funcionais do sistema, para então se projetar a interface com o usuário e a arquitetura do sistema que atendam aos requisitos especificados. Depois de todo o projeto pronto (e às vezes até depois do sistema desenvolvido), costuma-se especificar quais hardware, software e infraestruturas computacionais são necessários para executar o sistema projetado. Assim, na implantação do sistema pode ser necessário comprar algum hardware, software ou qualquer outro suporte necessário para executar o sistema desenvolvido.

No entanto, em qualquer atividade de design, o designer deve conhecer bem os materiais existentes com os quais é possível construir um artefato. Os materiais disponíveis impõem certas restrições de design. Portanto, nesse caso, conhecer os aspectos do sistema que podem afetar a interação significa apenas conhecer características dos materiais com os quais o sistema pode ser construído. Isso não corresponde a especificar os requisitos de hardware, software e infraestrutura. O objetivo aqui é ajudar o designer a pensar sobre o que é possível e adequado construir com determinados materiais disponíveis. O designer precisa estar ciente das possibilidades, limitações e comportamentos das tecnologias que ele vai utilizar no seu projeto. Sem esse conhecimento sobre as características do sistema não é possível projetar adequadamente como o sistema vai participar do processo de interação com usuário. Ainda assim, a decisão final sobre hardware, software e

infraestrutura computacional tende a ser tomada no final do processo de design e não o início. No início se investiga as possibilidades para determinar, no final, o que realmente será proposto como resultado de projeto.

Esse tipo de questão também ocorre em outras áreas, tal como: um arquiteto que precisa conhecer a resistência dos materiais para projetar um prédio; ou um designer industrial (ou de produto) que precisa consultar um químico para saber que materiais servem para armazenar determinado produto para o qual esteja projetando uma embalagem.

3.1.5 Objetivos

Os objetivos do usuário motivam e guiam sua interação com o sistema, influenciando suas ações sobre a interface e a interpretação das respectivas respostas do sistema. Antes de começar a interação, o usuário costuma possuir um desejo de alcançar determinado objetivo, mesmo que ainda não consiga defini-lo de forma concreta e detalhada. Não é raro os objetivos comecem bem abstratos na cabeça do usuário. Durante a interação, o usuário vai definindo seus objetivos de forma mais concreta e detalhada, por exemplo, conforme descrito pela teoria da ação de Norman (1986). Enquanto o usuário se esforça para alcançar um objetivo ou depois de alcançá-lo, é muito comum ele escolher outro objetivo de forma oportunista para continuar a interação, por exemplo, segundo as ações situadas de Suchman (1987). O processo de interação tende a continuar enquanto o usuário desejar alcançar um objetivo e existirem os recursos necessários.

Norman (1986, p. 37) define objetivo como “o estado que uma pessoa deseja alcançar”. O usuário interage com o sistema para receber um apoio computacional adequado e satisfatório que lhe permita chegar neste estado desejado. Um objetivo pode ser alcançado através de uma ou mais sequências de ações sobre a interface. Cabe ao usuário escolher qual sequência de ações ele prefere seguir em cada caso.

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”. Uma tarefa descreve os passos que usuário pode percorrer para atingir determinado objetivo, seguindo uma ou mais estraté-

gias. Em IHC, a definição de objetivos está fortemente relacionada com a definição das tarefas do usuário que o sistema deve apoiar (Diaper e Stanton, 2003). Os objetivos costumam ser definidos de forma mais abstrata, depois detalhados por tarefas que acabam definindo as ações sobre a interface de forma mais concreta. Na prática, não é tão simples separar objetivos de tarefas. Diaper e Stanton (2006, p. 27), por exemplo, argumentam que em muitos métodos de análise de tarefas “as descrições de estados psicológicos tal como os objetivos não são suficientemente diferentes das descrições do comportamento observado” durante a interação.

3.1.6 Interação

Quando o usuário estabelece um objetivo, ele precisa decidir quais atividades ou tarefas deve executar para alcançá-lo. Parte dessas tarefas pode ser apoiada por um sistema computacional. Neste sentido, utilizar um sistema computacional significa receber apoio para executar uma ou mais tarefas que contribuem para um ou mais objetivos estabelecidos. O usuário precisa, então, partir de uma definição mais abstrata de um objetivo, definir quais tarefas devem ser executadas com o apoio do sistema, e definir de forma mais concreta as ações que serão executadas sobre a interface. De acordo com as respostas do sistema através da interface, o usuário escolhe quais serão as próximas ações, tarefas e objetivos. Segundo Norman (1986), o processo de *interação* é responsável por mapear objetivos na mente do usuário em uma sequência de ações sobre a interface.

Existem outras interpretações para o processo de interação, conforme discutido por Sharp, Rogers e Preece (2007), e Barbosa e Silva (2010). Duas interpretações que se destacam na literatura consideram a interação como um processo de *operação* do sistema ou um processo de *comunicação* com o sistema (ou com o preposto do designer, nos termos da engenharia semiótica de de Souza, 2005a). No primeiro caso, a interação costuma ser descrita em termos de *objetivos*, *tarefas*, *ações* do usuário e do sistema, e *objetos* manipulados. Já no segundo, a interação geralmente é descrita em termos de *intenção comunicativa* (envolvendo a comunicação de significados desejando alcançar determinados efeitos nos interlocutores e no mundo), *sequência de falas*, cada *fala* do usuário e do sistema (de fato, do preposto do designer) propriamente dita e os *conceitos* (ou *signos*) utilizados.

3.1.7 Interface

A interface de um sistema interativo compreende toda a porção do sistema com a qual o usuário mantém contato físico (motor ou perceptivo) ou conceitual durante a interação (Moran, 1981). É através da interface que o usuário tem acesso ao sistema para usufruir o melhor que ele pode oferecer. Por isso, boa parte dos usuários acredita que o sistema é a interface com a qual entra em contato (Hix e Hartson, 1993).

Como a interface é o único meio de contato entre o usuário e o sistema, ela determina o que cada um pode “fazer” ou “falar” durante a interação, de que modo e em que ordem. Portanto, a interface também determina as possíveis interações entre o usuário e o sistema. As decisões de design sobre a interface afetam a interação, e vice-versa (Barbosa e Silva, 2010; Sharp *et al.*, 2007).

Sharp, Rogers e Preece (2007) discutem vários tipos de interfaces, como interfaces gráficas, via voz, por gestos e assim por diante. Destes, as interfaces gráficas têm sido uma das mais discutidas na literatura e utilizadas na prática por décadas. Por isso, vamos nos concentrar nesse tipo de interface nesta tese.

As interfaces gráficas costumam ser descritas em termos de *telas*, seus *wid-gets* e as *transições* entre telas (Dix *et al.*, 2003; Sharp *et al.*, 2007; Barbosa e Silva, 2010; Microsoft, 2010; Apple, 2010; Gnome, 2010). Um *widget* é um elemento de interface com o qual o usuário interage, tão como: um rótulo, um botão de comando, um link ou uma tabela. Dentre as propriedades da interface geralmente discutidas, aquela que costuma receber maior destaque é o layout. O layout envolve, por exemplo, a disposição, tamanho, fonte e cor dos *wid-gets* em cada tela.

Esses sete elementos do espaço de design de IHC estão fortemente inter-relacionados. Eles possuem papel significativo durante o uso e, conseqüentemente, durante a atividade de design de IHC. Vejamos como podemos organizá-los em termos de espaço de problema e de solução.

3.1.8 Espaço de problema e espaço de solução em IHC

Conforme discutido na Seção 2.2, o designer analisa a situação atual para definir um problema de design (isto é, identificar o que pode ser melhorado na situação atual). Assim, ele pode projetar uma solução de design (intervenção)

adequada. A Figura 3.2 ilustra como podemos caracterizar o espaço de problema e o espaço de solução em termos dos sete elementos de uma situação típica de uso discutidos anteriormente.

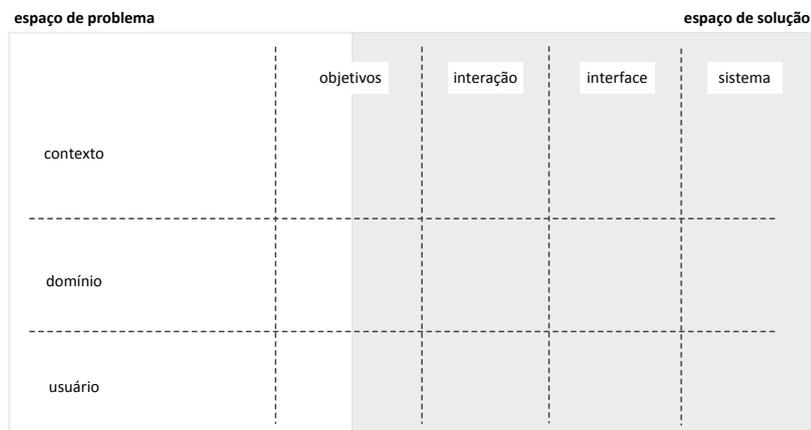


Figura 3.2: Espaço de problema e espaço de solução em IHC.

O *espaço de problema* envolve os elementos que existem antes do sistema interativo ser desenvolvido: o contexto em que o sistema será utilizado, o domínio, o usuário e seus objetivos. O *espaço de solução* envolve todos os elementos resultantes da atividade de design ou determinados por decisões do designer: a interação, a interface, o sistema e os objetivos do usuário apoiados pelo sistema. Apesar de a interação existir somente durante o uso do sistema, a definição da interface determina as possíveis interações que o usuário pode travar com o sistema. Os objetivos do usuário estão na fronteira entre o espaço de problema e de solução, pois eles são identificados durante a análise da situação atual, mas cabe ao designer decidir quais objetivos do usuário serão apoiados pelo sistema. Essa escolha do designer determina o que faz parte ou não da solução sendo concebida, e já é uma decisão de design.

Dependendo da abordagem de design de IHC utilizada, podemos empregar diferentes representações ou modelos. Se seguirmos uma abordagem em linha com a engenharia cognitiva (Norman, 1986), podemos representar o problema de design, por exemplo, usando personas (Cooper *et al.*, 2007) e cenários de problema (Rosson e Carroll, 2002; Carroll, 2000; Carroll, 1995); e representar a solução de design, por exemplo, usando um modelo de tarefas (Annett e Duncan, 1967; Card *et al.*, 1983; Paternò, 2000; Diaper e Stanton, 2006) e esboços de tela (Sharp *et al.*, 2007; Buxton, 2007; Snyder, 2003; Lin *et al.*, 2002; Bailey *et al.*, 2001; Landay e Myers, 1996), conforme ilustrado pela Figura 3.3.

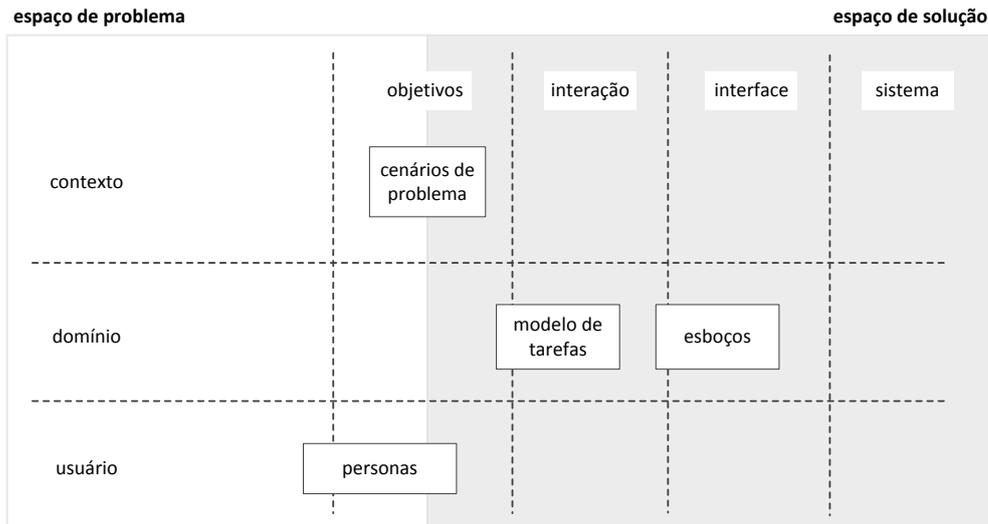


Figura 3.3: Exemplo de representações em linha com a engenharia cognitiva.

Por outro lado, se seguirmos uma abordagem em linha com a engenharia semiótica (de Souza, 2005a), podemos representar o problema de design usando, por exemplo, a parte de um esquema conceitual de signos que define o conteúdo (Paula, 2003; Barbosa e Paula, 2003; Barbosa e Silva, 2010), usando personas (Cooper *et al.*, 2007), cenários (Rosson e Carroll, 2002) e um mapa de objetivos (Barbosa e Silva, 2010); e representar a solução de design usando, por exemplo, o modelo de interação MoLIC (Paula, 2003; Barbosa e Paula, 2003; Barbosa e Silva, 2010), esboços de tela (Sharp *et al.*, 2007; Buxton, 2007; Snyder, 2003; Lin *et al.*, 2002; Bailey *et al.*, 2001; Landay e Myers, 1996) e a parte do diagrama conceitual que descreve a expressão dos signos (Barbosa e Silva, 2010), conforme ilustrado pela Figura 3.4

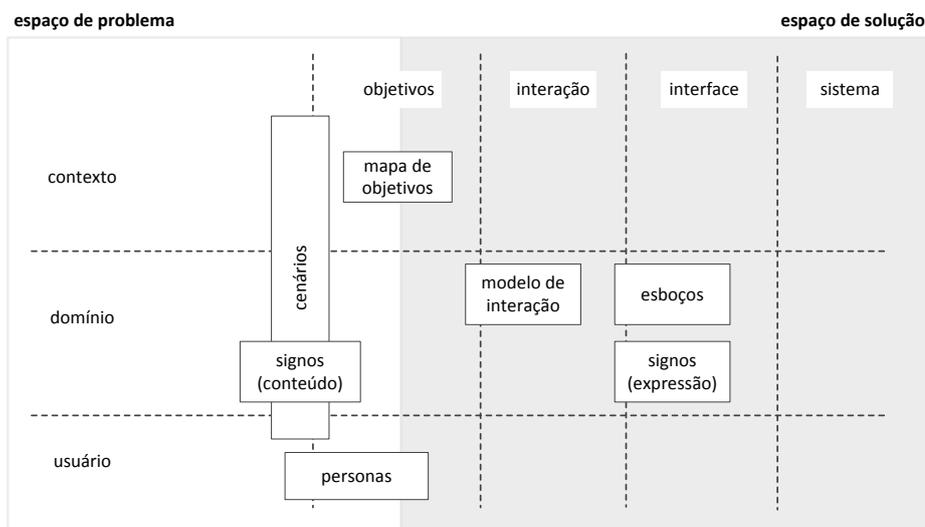


Figura 3.4: Exemplo de representações em linha com a engenharia semiótica.

É importante notar que estes são apenas exemplos de representações em linha com a engenharia cognitiva e engenharia semiótica. O designer pode escolher trabalhar com outros conjuntos de artefatos igualmente relevantes, coerentes e consistentes com essas abordagens teóricas de IHC.

3.2 Casos de design de IHC

Na abordagem de raciocínio baseado em casos, Kolodner e Leake (1996) definem um caso como sendo “uma parte de conhecimento contextualizado representando uma experiência que ensina uma lição fundamental para atingir os objetivos de quem raciocina sobre tal conhecimento” (p. 36). Essa definição sugere que um caso deve ter duas grandes partes:

- *um conteúdo*, responsável por registrar lições aprendidas; e
- *índices* que descrevem o contexto no qual essas lições podem ser ensinadas.

Vamos começar analisando o conteúdo dos casos de design. Segundo Kolodner e Leake (1996 p. 39), o conteúdo de qualquer caso deve conter espaços para o designer definir:

- *o problema*: descreve o estado do mundo onde o episódio registrado no caso ocorreu e, se apropriado, qual problema precisa ser resolvido naquele momento;
- *a solução*: descreve a solução para o problema especificado; e
- *a avaliação* (resultado ou *outcome*): relata uma apreciação do estado do mundo quando a solução foi implantada (ou inserida no mundo).

É interessante observarmos que essas três partes de um caso de design estão fortemente relacionadas com as três atividades básicas do processo de design discutidas na Seção 2.2: análise, projeto (design) e avaliação. Isso reforça a ideia de que um caso de design tem por objetivo registrar o aprendizado adquirido durante um processo de design específico.

Na proposta de Lee e seus colegas (2010) para apoiar o design de interface web, os casos de design são formado apenas pela solução, mais especificamente por uma representação da interface. Porém, uma solução de IHC certamente vai além da interface propriamente dita. Já na proposta de Kim e Yoon (2005), os

casos de design são representados pelos objetivos do usuário, pelo processo de interação (que eles também chamaram de tarefas) e interface. Nenhum deles registrou informações sobre o problema ou sobre a avaliação nos casos de design de IHC. Pela definição de casos de Kolodner e Leake (1996), esses trabalhos apresentam claramente limitações sobre o conteúdo dos casos.

3.2.1 Conteúdo dos casos de design de IHC

Uma vez organizado o espaço de problema e de solução em IHC, podemos definir qual será o conteúdo dos casos de design. Seguindo a proposta básica de Kolodner e Leake (1996), o **conteúdo de um caso** de design de IHC deve ter três partes que descrevem:

- o problema, com informações sobre contexto de uso, usuário e domínio;
- a solução, com informações sobre objetivos do usuário, interação, interface (e seus mecanismos de adaptação), sistema e sobre a concepção da solução; e
- a avaliação, com informações sobre a qualidade de uso da solução proposta (usabilidade, experiência do usuário, acessibilidade e comunicabilidade; Sharp et al., 2007; Barbosa e Silva, 2010).

A Figura 3.5 ilustra o modelo conceitual proposto para registrar o conteúdo dos casos de design de IHC. Um caso de design é composto por elementos que descrevem o problema, a solução ou a avaliação. Idealmente o designer deve registrar essas três partes de cada caso, apesar de ele também poder registrar apenas a solução.

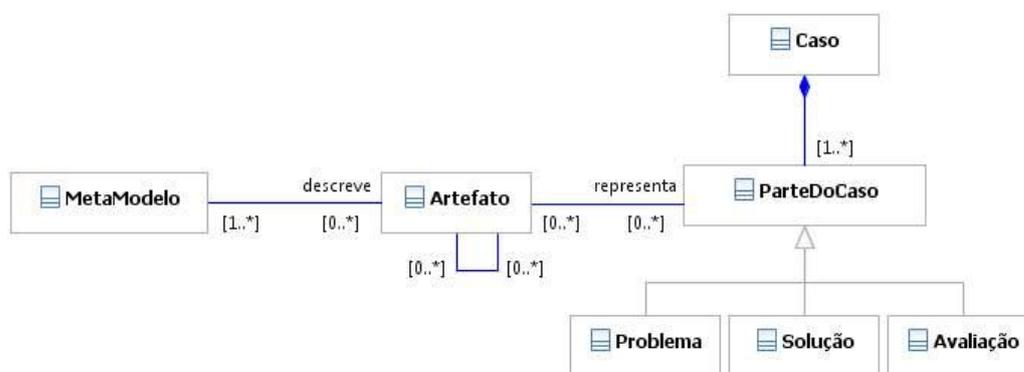


Figura 3.5: Modelo conceitual do conteúdo dos casos de design de IHC.

Como boa parte dos artefatos gerados durante o processo de design está representada em meio digital ou pode ser facilmente convertida para meio digital (por exemplo, digitalizando num *scanner* ou tirando uma foto), propomos utilizar esses próprios artefatos para descrever o conteúdo dos casos de design de IHC. Isso facilita a indexação e recuperação dos artefatos que representam casos de design, num sistema de raciocínio baseado em casos. Desse modo, o designer tende a empregar menos esforço para registrar seus casos de design, sem prejudicar sua leitura posterior. Além disso, o designer geralmente é capaz de ler artefatos produzidos por terceiros, seja porque está acostumado a utilizá-los ou pela sua formação.

Cada parte do caso pode estar associada a vários artefatos utilizados durante o processo de design de IHC. A parte que descreve o *problema* deve estar associada a artefatos que registram informações sobre o contexto de uso, o usuário e domínio. A parte que descreve a *solução* deve estar associada a artefatos que registram informações sobre os objetivos do usuário, a interação e a interface (e seus mecanismos de adaptação), o sistema e sobre o projeto. A parte que descreve a *avaliação* deve estar associada a artefatos que descrevam resultados de avaliações de IHC, tipicamente relatórios de avaliações formativas ou somativas (Hix e Hartson, 1993; Sharp *et al.*, 2007).

O designer está livre para definir casos de design de IHC usando os artefatos de sua preferência e cultura de design. Não existe nenhuma restrição nesse sentido no modelo conceitual proposto. Embora nossa proposta não imponha restrições sobre os artefatos utilizados em um caso, assumimos que uma base de casos estará alinhada com algum referencial teórico, ou seja, que os artefatos utilizados e a descrição dos casos estejam coerentes e consistentes com uma abordagem de IHC específica. A investigação sobre os efeitos de se utilizar múltiplas abordagens em uma mesma base de casos está fora do escopo deste trabalho.

Por exemplo, se o designer seguiu uma abordagem de design centrado na comunicação (Barbosa e Silva, 2010), espera-se que os artefatos escolhidos sejam apropriados para o design de um sistema com alta comunicabilidade. Nesse caso, não faz muito sentido utilizar um modelo KLM da família GOMS (Card *et al.*, 1983) com grande ênfase na eficiência no lugar de um diagrama de interação MoLIC (Paula, 2003; Barbosa e Paula, 2003; Barbosa e Silva, 2010). O contrário

também ocorre. Se o designer estiver seguindo uma abordagem cognitiva que prioriza a eficiência, não faz muito sentido deixar de usar um modelo como KLM para utilizar um diagrama de interação MoLIC.

Alguns artefatos podem ser descritos por um metamodelo específico. Por exemplo, o metamodelo do HTA (Annett e Duncan, 1967), do CTT (Paternò, 2000), do GOMS (Card *et al.*, 1983) ou do diagrama de objetivos (Barbosa e Silva, 2010) pode descrever um artefato que registre os objetivos do usuário. A Figura 3.6 apresenta um exemplo mais concreto: à esquerda temos a linguagem MoLIC (Paula, 2003) como um metamodelo que descreve um artefato (diagrama de interação), à direita. A relação entre artefato e metamodelo é análoga à relação entre instância (ou objeto) e classe, respectivamente. Esta relação só não foi definida como generalização (“é_um”) na Figura 3.5 porque podem existir artefatos que não tenham metamodelos definidos.

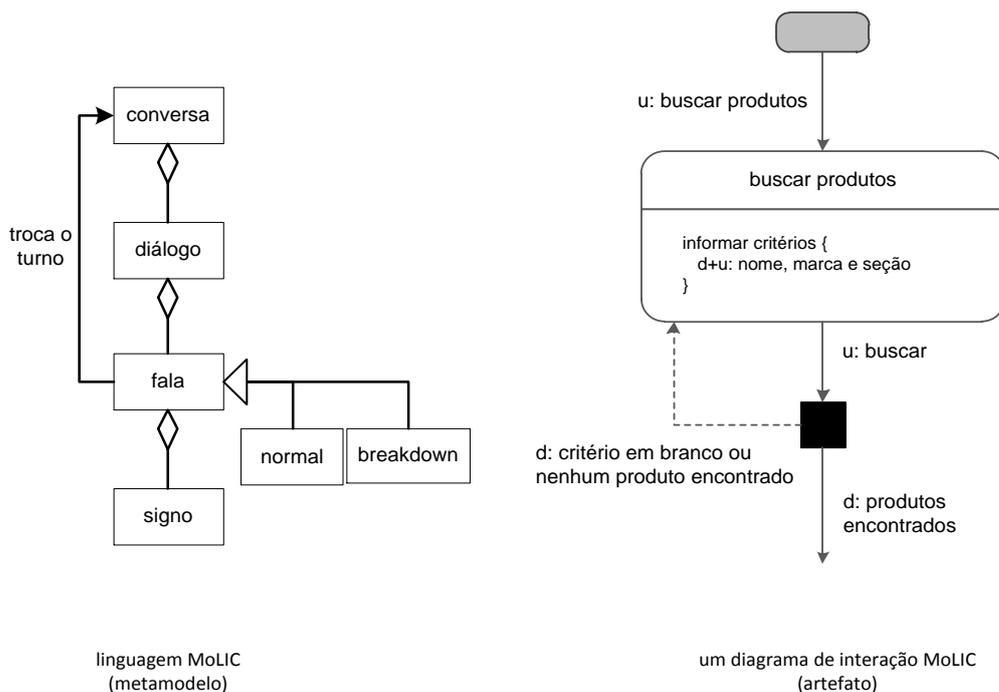


Figura 3.6: Exemplo do metamodelo da linguagem MoLIC e um artefato correspondente.

Um artefato pode estar relacionado a outros artefatos. Por exemplo, diagrama de interação pode estar relacionado a um ou mais esboços de tela (que representam a interface). A relação entre artefatos também serve para registrar a evolução das ideias durante o projeto, seja seguindo um raciocínio convergente (onde um artefato dá origem a outro através de algum tipo de refinamento) ou divergente (onde cada artefato dá origem a um ou mais artefatos). (A Figura 2.3 pode ajudar

a perceber este tipo de relação que marca a evolução de ideias durante o processo de design.).

Compreendidas as partes que compõem os casos de design de IHC, quanta informação podemos colocar em cada caso? Um caso de design de IHC pode descrever um sistema completo ou uma parte bem pequena dele, quase como descrever tudo relacionado a apenas um *widget* de interface. Entretanto, esses são os limites superior e inferior, e muito provavelmente o ideal é um meio termo entre os dois.

Nós recomendamos cadastrar casos de design de IHC relacionados a apenas um objetivo principal, ou seja, recomendamos que um caso de design inclua toda a interação e interface necessárias para o usuário alcançar um objetivo principal, bem como as demais informações necessárias para compreensão do objetivo principal. Se o usuário precisar atingir outros objetivos auxiliares (ou instrumentais, nos termos de Barbosa e Silva, 2010) para alcançar um objetivo principal, os objetivos auxiliares e as demais informações relacionadas também podem ser incluídos no caso que descreve o objetivo principal. Por exemplo, o objetivo de “consultar produto” é um objetivo auxiliar do objetivo “comprar produtos”. O caso de design que descreve o objetivo “comprar produtos” também pode incluir o objetivo “consultar produto” e as demais representações relacionadas com este objetivo. Sendo assim, podemos definir um caso de design de IHC do seguinte modo:

DEFINIÇÃO DE CASO DE DESIGN DE IHC

Um caso de design de IHC é **um conjunto de artefatos** produzidos durante a atividade de design de IHC que descrevem um problema, uma solução e a respectiva avaliação dessa solução; geralmente com escopo de um objetivo principal do usuário.

Exemplo de caso de design de IHC

A Tabela 3.1 apresenta *parte* dos artefatos do caso de design “comprar produtos”. Este caso de design de IHC foi definido por engenharia reversa do site do supermercado Zona Sul.² Por restrição de espaço, incluímos apenas o objetivo principal “comprar produtos” e omitimos todos os objetivos auxiliares, como “efetuar login”, por exemplo. O caso “comprar produtos” é descrito por um cenário (Carroll, 1995; Carroll, 2000) com informações sobre contexto de uso e objetivos do usuário, uma persona (Cooper et al., 2007) com informações sobre o usuário e seus objetivos, um diagrama de entidade e relacionamento (Chen, 1976) com informações sobre o domínio, um mapa de objetivos (Barbosa e Silva, 2010), um diagrama de interação MoLIC (Barbosa e Paula, 2003) com informações sobre a interação, e imagens de tela com informações sobre a interface, que deixam implícitas informações sobre a plataforma utilizada. Como não avaliamos a interface nessa engenharia reversa e não possuímos nenhum relatório de avaliação deste site, este exemplo de caso de design de IHC não possui uma parte de avaliação. O Apêndice D apresenta outros exemplos de casos de design de IHC construídos por engenharia reversa.

² www.zonasul.com.br

Tabela 3.1: Caso de design do objetivo comprar produtos em um supermercado online.

caso de design de IHC - comprar produtos**problema****cenário**

Clara convidou amigos para jantar em sua casa no próximo fim de semana. Ela decidiu fazer uma receita especial de família que não fazia há muito tempo. Quando ela chega em casa quarta-feira à noite, ela procura nas suas coisas a receita desejada. Como ela tem pouco tempo para ir ao supermercado, resolve encomendar o que precisa pelo site do seu supermercado preferido. Ela procura no site os itens da receita e coloca no carrinho de compras. Aproveitando a oportunidade, ela acrescenta também algumas coisas que estão faltando em casa. Ao concluir a compra, ela informa o endereço de entrega e o horário que pretende receber o pedido, indica a forma de pagamento e confirma o pedido.

persona

Clara é uma mulher jovem com a vida muito atarefada. Faz pouco tempo que ela é casada e ainda não tem filhos. Ela prefere ir ao supermercado para escolher os produtos pessoalmente com cuidado, pois considera importante para saúde escolher bem o que consome. Ela gosta de observar o tamanho, cor e textura das coisas frescas. Os produtos embalados costumam receber atenção especial, pois ela geralmente lê as informações disponíveis: data de validade, informações nutricionais, local de fabricação, etc. Quando não tem tempo de ir ao supermercado, ela então recorre à praticidade de fazer compras pela internet. Não é sempre que isso acontece, mas já precisou fazer compras online algumas vezes.

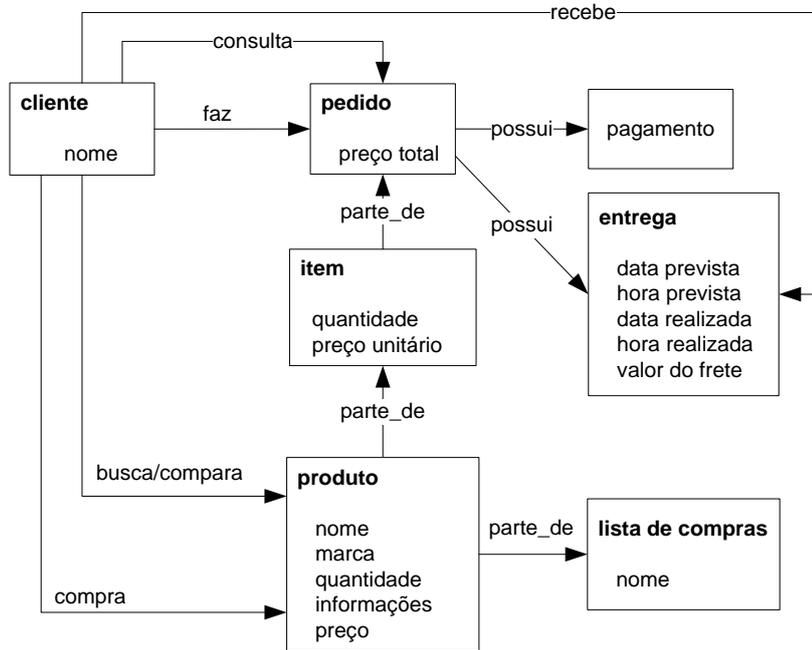


Ela está acostumada a usar sistemas computacionais para coisas básicas, como ler e escrever e-mails, ler notícias e usar serviços bancários pela internet. Para atividades mais avançadas, costuma pedir ajuda a amigos que sabem mais do que ela, e em último caso tenta aprender sozinha consultando tutoriais e ajuda online. No site do seu supermercado preferido, Clara costuma encontrar dificuldades para comparar informações sobre produtos, como faz facilmente no supermercado físico. Ela reclama muito por perder o contato com o produto e com sua embalagem. Por isso, ela prefere comprar apenas os produtos que já conhece.

continua na próxima página

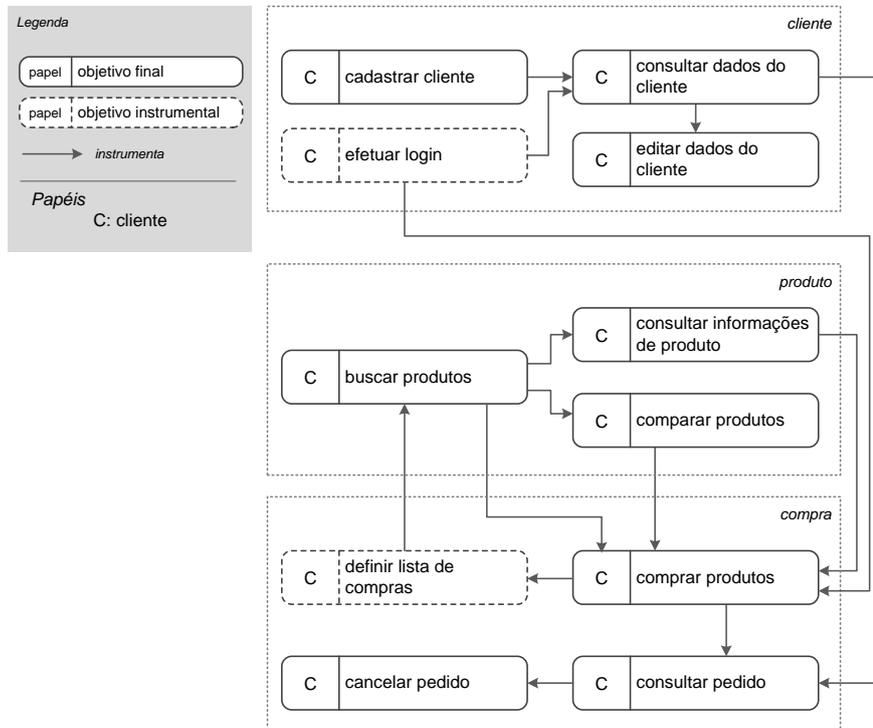
caso de design de IHC - comprar produtos

modelo conceitual (diagrama de entidade e relacionamento)



solução

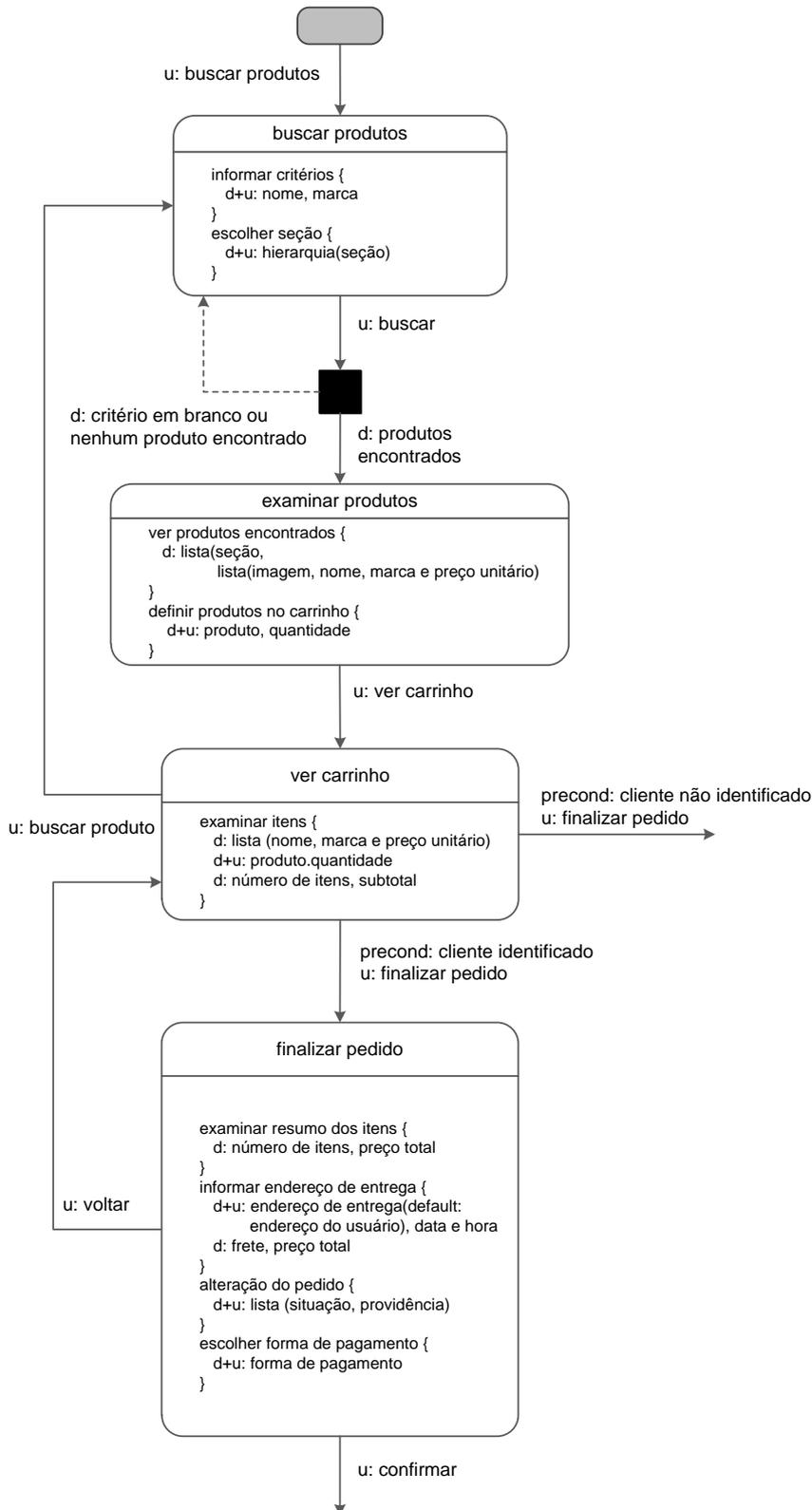
objetivos (mapa de objetivos)



continua na próxima página

caso de design de IHC - comprar produtos

interação (diagrama de interação MoLIC)



PUC-Rio - Certificação Digital Nº 0611946/CA

continua na próxima página

caso de design de IHC - comprar produtos

interface (imagens de tela)



Uma vez cadastrado um conjunto de casos de design de IHC, como eles poderiam ser indexados para serem recuperados posteriormente?

3.2.2 Indexação de casos de design de IHC

O designer pode utilizar um sistema computacional de raciocínio baseado em casos para organizar um repertório de casos que serão consultados posteriormente. Uma forma de facilitar a recuperação desses casos é classificá-los com um conjunto de índices que podem ser processados computacionalmente. Segundo Kolodner e Leake (1996 p.40), os **índices** de um caso são combinações de descritores importantes, responsáveis por distinguir um caso dos demais. Eles serão utilizados para referenciar os casos que o designer deseja consultar em dado momento. Como indexar casos de design registrados em diferentes tipos de artefatos?

Uma forma de fazer isso é **processar automaticamente** cada artefato para indexá-los. A proposta de Kim e Yoon (2005) segue esta abordagem. Eles propuseram uma espécie de indexação automática do *Operation and Control Diagram* (Yoon *et al.*, 1996) para apoiar o design de interfaces em geral com um sistema de raciocínio baseado em casos. Nessa proposta, o designer precisa fazer seu projeto de IHC usando o OCD, ou reescrever a sua solução neste diagrama específico. Feito isso, os algoritmos propostos se encarregam de processar automaticamente e recuperar os casos de acordo com uma gramática de consulta. O diagrama OCD e a gramática de consulta referenciam informações sobre os objetivos do usuário, a interação e a interface. Portanto, na proposta de Yoon, os casos de design de IHC não são indexados por outras características do problema (e.g usuário e contexto) e nem da avaliação, reduzindo assim o poder de expressão do designer na indexação e recuperação de casos.

Abordagens como essa são difíceis de serem escaláveis para outros artefatos, principalmente quando os artefatos não possuem um metamodelo associado (Figura 3.5). No mínimo seria necessário um esforço de desenvolvimento (ou esforço equivalente de *end user development* - Lieberman *et al.*, 2006) para tornar o sistema de raciocínio baseado em casos capaz de processar automaticamente um novo artefato quando o designer desejar.

Outra forma é através de **indexação manual**. Aqui cabe ao designer indexar manualmente os artefatos produzidos, enquanto o sistema é responsável por processar automaticamente os índices definidos pelo designer na recuperação dos casos desejados. Lee e seus colegas (2010) seguem esta abordagem para apoiar o design de interfaces Web. Eles indexaram os casos de design com metadados fa-

cetados (Naaman *et al.*, 2004) relacionados ao layout das páginas, tais como: cor de fundo, fonte, número de colunas e número de imagens.

Apesar de a segunda abordagem exigir esforço do designer na indexação dos casos, o designer está livre para escolher o artefato que desejar utilizar na descrição do caso e para indexá-los conforme desejar. Desse modo, o designer pode construir uma base de casos que acomode diferentes artefatos de design capazes de registrar o mesmo conceito. Por exemplo, é possível construir uma base de casos que acomode um cenário de interação, um modelo de tarefas e um diagrama de interação para representar a interação usuário-sistema. Além disso, esta segunda abordagem facilita o designer modificar e estender o conjunto de descritores do caso para acomodar diferentes culturas de design. Por exemplo, ele pode indexar um caso com alguma característica específica dos usuários que ainda não foi considerada.

Conjugar essas duas abordagens é bem promissor, pois podemos aproveitar as vantagens de ambas. Quando for utilizado um artefato com metamodelo definido, o sistema poderia criar índices automaticamente a partir do artefato. Além disso, o designer também poderia indexar casos de design para artefatos não previstos ou estender o conjunto inicial de índices previsto para descrever particularidades dos casos.

Independente da abordagem seguida, um bom ponto de partida para a construção de um sistema de raciocínio baseado em casos é definir um conjunto de índices que pode ser utilizado automática ou manualmente. Os índices dos casos devem **destacar algumas características, aspectos ou dimensões** que podem servir como **critérios de comparação entre diferentes tipos de artefatos**. O conjunto de índices de um artefato servirá, portanto, como uma abstração das particularidades do artefato. Essa abstração permite compará-lo com artefatos de diferentes tipos que representam conceitos semelhantes. Uma das abstrações mais relevantes diz respeito ao processo de interação. Aqui estamos considerando a interação simplesmente como uma sequência de passos do usuário e do sistema. Esses passos de interação podem ser compreendidos como tarefas ou ações representados em modelos de tarefas como GOMS (Card *et al.*, 1983), HTA (Annett e Duncan, 1967) e CTT (Paternò, 2000), ou ainda como falas e diálogos representados

em um diagrama de interação MoLIC (Paula, 2003; Barbosa e Paula, 2003; Silva, 2005).

Com base no espaço de design de IHC discutido na Seção 3.1, propomos nesta tese um conjunto de descritores para indexar artefatos que descrevem o problema, a solução e a respectiva avaliação de um caso de design de IHC. Tais descritores podem ser utilizados numa abordagem automática, manual ou uma conjugação das duas. Para tornar a recuperação de casos mais **flexível**, rica e, potencialmente, precisa, ampliamos os aspectos do problema, da solução e da avaliação considerados nas indexações de casos propostas por (Lee *et al.*, 2010) e por (Kim e Yoon, 2005). A nossa proposta de indexação vai além de objetivos, interação e interface. Também consideramos outras características desses conceitos não considerados em tais trabalhos relacionados. Mesmo nos esforçando para ser o mais abrangente possível, certamente deixamos de considerar alguns aspectos que podem ser relevantes para casos de design específicos. Por isso, é fundamental que o mecanismo de indexação proposto seja **extensível**. Assim, o designer poderá acrescentar índices que considerar relevantes para os casos de design que ele tiver acesso.

A Figura 3.7 apresenta o modelo conceitual proposto para indexar os casos de design de IHC usando descritores.

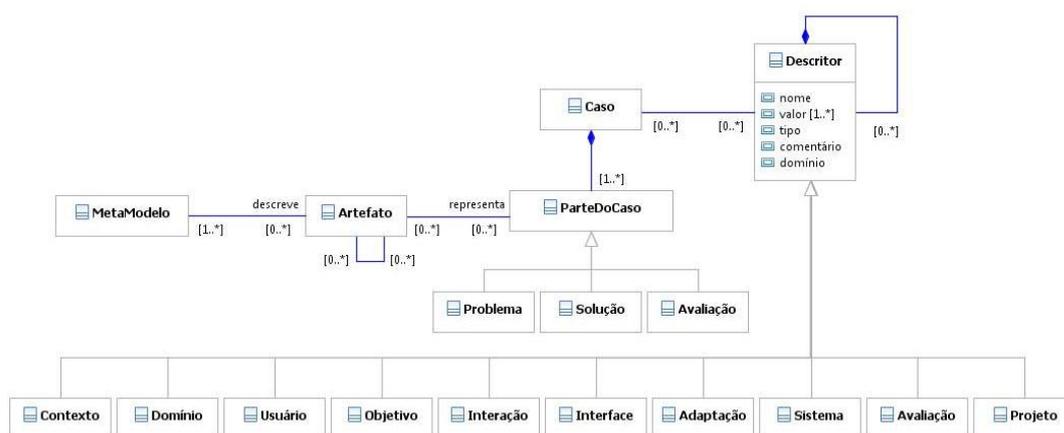


Figura 3.7: Modelo conceitual com conteúdo e índices dos casos de design de IHC.

Um caso de design pode estar associado a zero ou mais descritores. Um **descritor** possui nome, um ou mais valores, tipo, domínio e comentário. Os descritores de tipo simples são basicamente texto, número, data e hora. Os descritores de tipo composto são basicamente tuplas formadas por outros descritores. O do-

mínio define os possíveis valores que um descritor pode assumir e as relações entre esses valores.

A flexibilidade da nossa proposta advém da possibilidade de o designer empregar alguns descritores dentre um conjunto maior que aborda características do problema, da solução e da avaliação. Já a capacidade de extensão advém da possibilidade de o designer adicionar novos descritores ao conjunto de descritores proposto, e novos valores de descritores. Vejamos o conjunto de descritores propostos, bem como exemplos de possíveis valores.

Os descritores que abordam o **problema** de design estão relacionados com o contexto de uso, o domínio e o usuário. Esses descritores são apresentados nas Tabelas 3.2, 3.3 e 3.4 com exemplos de possíveis valores. Muitos desses valores foram definidos pensando no domínio de supermercado. É importante destacar que os valores apresentados aqui são apenas exemplos. O designer tem a liberdade de usar ou não esses valores, bem como propor novos valores conforme julgar adequado. Esses descritores foram propostos com base nos conceitos normalmente investigados na análise do problema durante o processo de design de IHC, conforme discutido na Seção 3.2.

Contexto de uso

A Tabela 3.2 exemplifica um conjunto de descritores sobre o contexto de uso. Nesses exemplos, consideramos o ambiente físico e sociocultural, e o tempo em que a interação pode ocorrer.

Tabela 3.2: Características do contexto normalmente consideradas no design de IHC.

Contexto		
dimensão	descritor	exemplos de possíveis valores
ambiente físico	local	<ul style="list-style-type: none"> ▪ país, região, estado, cidade, bairro, rua ▪ trabalho, casa, universidade, trânsito
	nível de atenção	<ul style="list-style-type: none"> ▪ alto, médio, baixo
	nível de luminosidade	<ul style="list-style-type: none"> ▪ alto, médio, baixo
	nível de ruído	<ul style="list-style-type: none"> ▪ alto, médio, baixo

ambiente sociocultural	idioma	<ul style="list-style-type: none"> ▪ Português, Inglês, Espanhol
	jargão (linguagem particular)	<ul style="list-style-type: none"> ▪ não utiliza ▪ uso coletivo em <área de conhecimento> ▪ uso individual
	incertezas	<ul style="list-style-type: none"> ▪ aceitáveis ▪ toleráveis ▪ inaceitáveis ▪ as pessoas lidam bem com incertezas ▪ as pessoas não lidam muito bem com incertezas ▪ as pessoas não sabem lidar com incertezas
	estrutura de poder	<ul style="list-style-type: none"> ▪ definida ▪ não definida ▪ todos respeitam ▪ nem todos respeitam ▪ poucos respeitam
	interação entre pessoas	<ul style="list-style-type: none"> ▪ trabalham isoladas ▪ trabalham em grupo ▪ podem/costumam conversar ▪ podem/costumam trabalhar em grupo ▪ podem/costumam compartilhar material
	pressões sociais	<ul style="list-style-type: none"> ▪ concluir rapidamente as tarefas ▪ não cometer erros ▪ economizar recursos ▪ não significativa
	grupo sociocultural	<ul style="list-style-type: none"> ▪ ocidental, oriental ▪ brasileiros, portugueses, moçambicanos, angolanos ▪ engenheiros, artistas, professores, médicos ▪ funcionários da empresa A, funcionários do departamento D, funcionários públicos, profissionais liberais
tempo	horário	<ul style="list-style-type: none"> ▪ manhã, tarde, noite ▪ expediente, almoço, fora do expediente

Domínio

A Tabela 3.3 apresenta um conjunto de descritores sobre o domínio. Nesses exemplos de descritores, consideramos os conceitos e as relações entre conceitos referenciados durante a interação.

Tabela 3.3: Características do domínio normalmente consideradas no design de IHC.

Domínio		
dimensão	descriptor	exemplos de possíveis valores
definição	conceito	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < nome [: atributos] > <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < cliente: nome > ▪ < produto: nome, marca, quantidade, informações, preço > ▪ < pagamento >
	relação	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < [card. mín. do conceito 1, card. máx. do conceito 1,] conceito 1, [nome,] relação 1, conceito 2, [card. mín. do conceito 2, card. máx. do conceito 2] [, [nome,] relação N, conceito N, [card. mín. do conceito N, card. máx. do conceito N]] > ▪ relação: (algo) está_associado (outro algo), (algo) é_um (outro algo), (algo) é_instância_de (outro algo), (algo) é_parte_de (um todo), (recipiente) contém (um conteúdo), (objeto) éfeito_de (substância), (algo) causa (um efeito), (agente) usa (objeto), (produtor) produz (produto), (local) abriga (evento) <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < 0,n, cliente, busca, produto, 0,n > ▪ < 0,n, item, é_parte_de, pedido, 1,1 >

Usuário

A Tabela 3.4 apresenta um conjunto de descritores sobre o usuário. Nesses exemplos de descritores, consideramos características pessoais, conhecimento, experiência, habilidades físicas, perceptivas e cognitivas dos usuários, bem como suas preferências sobre a forma de realizar as tarefas.

Tabela 3.4: Características do usuário normalmente consideradas no design de IHC.

Usuário		
dimensão	descritor	exemplos de possíveis valores
características pessoais	idade	<ul style="list-style-type: none"> ▪ criança, jovem, adulto, idoso
	profissão	<ul style="list-style-type: none"> ▪ professor, administrador, médico
	papel	<ul style="list-style-type: none"> ▪ aluno, professor, funcionário do suporte, coordenador, reitor
conhecimento	sobre domínio	<ul style="list-style-type: none"> ▪ baixo, mediano, alto
	sobre tarefa	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < [conhecimento], qual tarefa, [quais formas de realizar] > ▪ conhecimento: baixo, médio, alto <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < conhece, efetuar login > ▪ < não conhece, comparar produtos >
experiência	com tecnologia	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < [experiência], qual tecnologia, [quais formas de realizar] > ▪ experiência: não conhece, conhece, costuma utilizar pouco, costuma utilizar, costuma utilizar muito <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < costuma utilizar, email > ▪ < conhece, mensagens SMS de celular > ▪ < não conhece, chat por vídeo >
habilidades físicas, perceptivas e cognitivas	visão	<ul style="list-style-type: none"> ▪ normal, limitada, daltônico, sem visão
	audição	<ul style="list-style-type: none"> ▪ normal, limitada, sem audição
	habilidade motora	<ul style="list-style-type: none"> ▪ normal, movimento dos braços e mãos limitados, sem movimento nos braços e mãos ▪ movimentos precisos, movimentos imprecisos
	cognição	<ul style="list-style-type: none"> ▪ normal, dificuldades de aprendizado, deficiência mental
	alfabetismo	<ul style="list-style-type: none"> ▪ alfabetizado, analfabeto funcional, analfabeto pleno
preferências sobre a forma de realizar as tarefas	preferências	<ul style="list-style-type: none"> ▪ escreve o texto, depois formata ▪ usa mais teclado, do que mouse ▪ organização de arquivos em diretórios

Os descritores que abordam a **solução** de design estão relacionados com os objetivos do usuário, a interação, a interface, os mecanismos de adaptação da interface, o sistema e o projeto propriamente dito. Esses descritores são apresentados nas Tabelas 3.5, 3.6, 3.7, 3.9, 3.10, 3.11 e 3.12 com exemplos de possíveis valores. Novamente, muitos desses exemplos foram definidos pensando no domínio de supermercado. Dependendo das particularidades do domínio e do caso sendo descrito, pode ser necessário utilizar outros valores para os descritores citados ou mesmo definir novos descritores. Esses descritores foram propostos com base nos conceitos normalmente investigados na síntese da solução durante o processo de design de IHC, conforme discutido na Seção 3.2. Todavia, precisamos destacar alguns pontos a respeito dos descritores de objetivo do usuário, de interação e de mecanismos de adaptação da interface.

Os objetivos do usuário e os passos de interação poderiam ser descritos por um texto não estruturado, mas isso dificultaria o processamento da recuperação de casos. Uma solução mais interessante é estruturá-los de algum modo. Fillmore (1968) estrutura sentenças em linguagem natural na forma de *cases*. Os *cases* de Fillmore são uma tupla na forma verbo (case1, case2, ...), onde os cases podem ser de diferentes tipos: *Agentive*, *Dative*, *Objective*, *Instrumental*, *Factitive*, *Locative*, *Experiencer* e *Benefactive*. Por exemplo, o objetivo de “imprimir um documento” pode ser estruturado no case de Fillmore: imprimir [Objective(documento), Instrumental(impressora)]. Já o objetivo de “remover uma música de uma lista de reprodução” pode ser estruturado no case: remover[Objective(arquivo de música), Locative(lista de reprodução)]. Nesses exemplos, imprimir e reproduzir são verbos, documento e arquivo de músicas são objetos que sofrem a ação, impressora é um instrumento e lista de reprodução é um local onde se armazena referência a um arquivo de música. Uma tupla estruturada com base nos cases de Fillmore seria uma construção mais elaborada da estrutura sintática básica que estamos acostumados: sujeito + verbo + objeto.

Os *cases* de Fillmore têm sido aplicados na análise de variabilidade (Liaskos *et al.*, 2006; Liaskos, 2008; Silva *et al.*, 2008; Silva e Barbosa, 2008), onde os *cases* são considerados como *slots* (lacunas) que podem assumir valores do domínio (algo como parâmetros de um procedimento em linguagens de programação). Silva e Barbosa (2008) exploraram diferentes formas de preencher um *slot* nave-

gando por relacionamentos entre os conceitos do domínio. Esse mecanismo de navegar pelos conceitos do domínio é muito promissor para auxiliar na recuperação de casos de design semelhantes.

Inspirados nos *cases* de Fillmore (1968) e nas suas aplicações em análise de variabilidade propostas por (Silva *et al.*, 2008), propomos estruturar os objetivos do usuário na forma <verbo, [slots], [pré-condição], [pós-condição], [frequência]>, e os passos de interação do seguinte modo <agente, verbo, [slots]>. Os verbos podem ser definidos livremente pelo designer, fazer parte de uma taxonomia proposta por ele, ou ainda pertencer a uma taxonomia proposta por outros, como aquela proposta por (González-Calleros *et al.*, 2009).

Objetivos do usuário

A Tabela 3.5 apresenta um conjunto de descritores sobre os objetivos do usuário. Nesses exemplos de descritores, consideramos a definição dos objetivos e as relações entre eles.

Tabela 3.5: Características dos objetivos do usuário normalmente consideradas no design de IHC.

Objetivos		
dimensão	descritor	exemplos de possíveis valores
definição	objetivo	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ <verbo, [slots], [pré-condição], [pós-condição], [frequência]> ▪ verbo: buscar, ver (lista de), examinar (detalhes de), analisar, comparar, cadastrar (ou criar ou postar), alterar, remover, confirmar e controlar ▪ slot: conceito do domínio ▪ frequência: <quantidade de> 'vezes por' <período>, onde: <ul style="list-style-type: none"> - quantidade: poucas algumas muitas n^o de vezes - período: dia semana mês ano anos dia útil fim de semana feriado início das aulas aniversário data comemorativa <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ <comprar, [O(produtos)], usuário identificado, , algumas vezes por mês> ▪ <cadastrar, [F(cliente)], , , uma vez por anos> ▪ <identificar, [O(cliente)], usuário não identificado, usuário identificado, poucas vezes ao mês>

relações entre objetivos	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < objetivos, relação, objetivos > ▪ relação: hierarquia, sequencial, independente de ordem, alternativa, concorrente <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < <consultar, [O(cliente)]>, ativação, <editar, [F(cliente)]> > ▪ < <cadastrar, [F(cliente)]>, sequencial, <consultar, [O(cliente)]> > ▪ < <identificar, [O(cliente)]>, ativação, <consultar, [O(cliente)]>, <comprar, [O(produto)]> >
--------------------------	--

Interação

A Tabela 3.6 apresenta um conjunto de descritores sobre a interação. Aqui consideramos basicamente a interação como uma sequência de passos, que podem ser mapeados para tarefas, ações do usuário sobre a interface ou fala na conversa usuário-sistema (preposto do designer, nos termos da engenharia semiótica de de Souza, 2005a).

Tabela 3.6: Características da interação normalmente consideradas no design de IHC.

Interação		
dimensão	descritor	exemplos de possíveis valores
tarefa, ação ou fala	passo	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < agente, verbo [, slots] > ▪ agente: usuário, designer ou sistema. ▪ verbo: buscar, ver (lista de), examinar (detalhes de), analisar, comparar, cadastrar (ou criar ou postar), alterar, remover, confirmar e controlar ▪ slot: conceito do domínio <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < u, buscar, [O(produto)] > ▪ < u, informar, [O(produto.nome), O(produto.marca)] > ▪ < u, escolher, [O(seção)] > ▪ < u, confirmar >

sequência de tarefas, ação ou fala	sequência de passos	<p><i>tupla --</i></p> <ul style="list-style-type: none"> ▪ < passo 1, relação 1, passo 2 [, relação N, passo N] > ▪ relação: hierarquia, sequencial, independente de ordem, alternativa, concorrente, ativação <p><i>exemplo de tuplas --</i></p> <ul style="list-style-type: none"> ▪ < <u, buscar, [O(produto)]>, sequencial, <u, informar, [O(produto.nome), O(produto.marca)]>, alternativo, <u, escolher, [O(seção)]>, sequencial, <u, confirmar >
suporte oferecido	suporte oferecido	<ul style="list-style-type: none"> ▪ assistente (wizard), ▪ estimula exploração, ▪ solicita confirmação, ▪ visão geral primeiro e detalhes sob demanda

Interface

A Tabela 3.7 apresenta um conjunto de descritores sobre a interface. Nesses exemplos de descritores, consideramos o mapeamento da interação para interface, os *widgets*, estilos de interação e padrões de interface utilizados.

Tabela 3.7: Características da interface normalmente consideradas no design de IHC.

Interface		
dimensão	descritores	exemplos de possíveis valores
layout	mapeamento da interação em interface	<ul style="list-style-type: none"> ▪ cada passo de interação em uma tela diferente ▪ notificação dos erros na mesma tela em que ocorrem ▪ notificação de erros com destaque ▪ notificação de sucesso sutis ▪ separar opções simples de avançadas
	widget	<ul style="list-style-type: none"> ▪ caixa de texto, botão de comando, link, lista simples, lista hierárquica, etc.
	estilo de interação	<ul style="list-style-type: none"> ▪ linguagem de comando, linguagem natural, menus, formulário, manipulação direta
	padrão de interface	<ul style="list-style-type: none"> ▪ acordeão, barra de progresso, grades de página

Adaptação

Até aqui, tratamos as características da interação humano-computador como “estáveis”, ou seja, como sistemas que não se modificam ao longo da interação. No entanto, parte dos sistemas computacionais interativos incluem alguma forma de adaptação. Uma **adaptação** da interface diz respeito a qualquer modificação que possa ocorrer na interface durante o tempo de interação, seja uma modificação realizada pelo usuário ou pelo próprio sistema. Existem alguns mecanismos, métodos ou técnicas para modificar a interface durante o uso. Por exemplo, o usuário pode criar estilos de formatação ou uma macro num editor de texto, ou ainda o próprio sistema editor de texto pode mudar seu comportamento normal e transformar “1 - <espaço>” em um mecanismo de autonumeração. Dieterich e seus colegas (1993) investigaram o que está envolvido na adaptação da interface durante o uso, seja uma adaptação realizada pelo usuário ou pelo sistema. De Souza e Barbosa (de Souza, 2005a; de Souza e Barbosa, 2006) propuseram uma abordagem semiótica para os mecanismos de adaptação realizados pelo usuário. De Souza (2005a) considera a interface com usuário como uma linguagem que permite a comunicação entre o usuário e o preposto do designer mediado pelo sistema. Nesse sentido, elas analisaram os mecanismos de adaptação da interface em termos de modificações léxicas, sintáticas e semânticas da linguagem de interface.

Bueno, Silva e Barbosa (Bueno, 2008, Silva, Bueno e Barbosa, 2009; Silva, Barbosa e Leite, 2009) fazem uma revisão mais recente do que está envolvido na adaptação da interface durante o uso, com ênfase em adaptações realizadas pelo usuário. Bueno (2008), em particular, relaciona mecanismos (técnicas ou procedimentos) de adaptação realizados pelo usuário com os efeitos na linguagem de interface em termos de modificações léxicas, sintáticas e semânticas, propostas por de Souza e Barbosa.

Compreender a interface com usuário como uma linguagem e as adaptações durante o uso como modificações nessa linguagem é um ferramental analítico poderoso e muito útil. Entretanto, a cultura de design de IHC não está acostumada a projetar diretamente uma linguagem no sentido literal, em termos de projetar um vocabulário e uma gramática. Esse costuma ser o ponto de vista do sistema a respeito da interação. Muito antes de passar por algo parecido com uma linguagem processável por computador, os designers costumam pensar no processo de intera-

ção do ponto de vista do usuário, considerando, por exemplo, os objetivos do usuário, sequência de passos e como expressá-los na interface. De Souza (2005a) discute sobre as correspondências e diferenças entre os pontos de vista (ou dimensões semióticas) do usuário e do sistema sobre a interface.

Para facilitar a reflexão do designer de IHC sobre os mecanismos de adaptação, propomos enunciar as possíveis modificações na interface em termos de objetivos, passos de interação, expressão dos passos na interface e conceitos do domínio (ou signos). Assim, o designer pode continuar pensando sobre os mesmos tipos de conceitos utilizados no projeto de IHC para projetar mecanismos de adaptação na interface. Em outras palavras, o designer continua se mantendo do ponto de vista do usuário ao projetar a interface. Essa proposta está em linha com a proposta de Koch (2000) para adaptações em hipermídia adaptativa: adaptações de conteúdo, navegação e apresentação. A Tabela 3.8 apresenta uma correspondência entre as modificações na linguagem de interface e modificações no processo de interação do ponto de vista do usuário, conforme previsto pelo designer.

Tabela 3.8: Correspondência entre modificações na linguagem de interface e no processo de interação.

modificação em termos de linguagem de interface	modificação em termos de objetivos, passos de interação, interface e conceitos do domínio
léxico + sintaxe + semântica	objetivo
sintaxe	sequência de passos
léxico + sintaxe + semântica	conteúdo do passo
léxico + sintaxe	expressão do passo na interface
léxico + sintaxe + semântica	conceitos (ou signos) do domínio manipulados

Os descritores de mecanismos de adaptação da interface propostos na Tabela 3.9 são um aprimoramento dos trabalhos de Bueno, Silva e Barbosa (Silva, Bueno e Barbosa, 2009; Silva, Barbosa e Leite, 2009; Bueno, 2008). Em particular, aprimoramos a forma de enumerar o que pode ser modificado durante o uso (descriptor modificação). Os descritores de adaptação envolvem quem realiza a adaptação, o que é adaptado, em função de que, como adaptar e a persistência da adaptação.

Tabela 3.9: Características de adaptações da interface normalmente consideradas no design de IHC.

Adaptação		
dimensão	descritor	exemplos de possíveis valores
quem realiza	agente	<ul style="list-style-type: none"> ▪ usuário ou sistema ▪ qualquer usuário, usuários específicos definidos pelo designer, usuários específicos definidos por um administrador
o que adaptar	modificação	<p><i>objetivo --</i></p> <ul style="list-style-type: none"> ▪ adicionar/remover objetivos ▪ modificar objetivos existentes: <ul style="list-style-type: none"> – modificar verbo – adicionar/remover slots – modificar um slot: <ul style="list-style-type: none"> - signo-tipo - cardinalidade máxima - cardinalidade mínima - possíveis valores - ordem dos possíveis valores - valor padrão – modificar pré e pós-condição – modificar frequência em que o usuário busca atingir o objetivo <p><i>sequência de passos de interação --</i></p> <ul style="list-style-type: none"> ▪ modificar ordem dos passos ▪ modificar relação entre os passos ▪ adicionar/remover passos para de um objetivo <p><i>passos de interação --</i></p> <ul style="list-style-type: none"> ▪ modificar passo: <ul style="list-style-type: none"> – modificar verbo – adicionar/remover slots – modificar um slot: <ul style="list-style-type: none"> - signo-tipo - cardinalidade máxima - cardinalidade mínima - possíveis valores - ordem dos possíveis valores - valor padrão – modificar pré e pós-condição – modificar frequência em que o usuário busca atingir o objetivo

		<p><i>expressão de um passo na interface considerando o tempo --</i></p> <ul style="list-style-type: none"> ▪ modificar sequência de verbo e slots na expressão de um passo <p><i>expressão de um passo na interface considerando o espaço --</i></p> <ul style="list-style-type: none"> ▪ modificar sequência e forma de expressão do verbo e dos conceitos num passo ▪ modificar layout da representação na tela <p><i>conceitos do domínio / signos --</i></p> <ul style="list-style-type: none"> ▪ compor novos tipos de conceitos ▪ compor novas instâncias de conceitos ▪ definir a representação de tipos de conceitos ▪ definir a representação de instâncias de conceitos
em função de que	motivo	<ul style="list-style-type: none"> ▪ mudança nas características do <i>contexto</i> ▪ mudança nas características dos <i>objetivos</i> ▪ mudança de <i>objeto manipulado</i> ▪ mudança de <i>usuário</i> ▪ mudança das <i>características do usuário</i> ▪ mudança de <i>dispositivo</i> ▪ mudança nas características do <i>sistema</i> ▪ mudança no <i>histórico de interação</i>, tais como: tarefas executadas, instâncias de conceitos manipuladas, erros cometidos
como adaptar	<p>técnica de adaptação realizada pelo sistema</p> <p>técnica de adaptação realizada pelo usuário</p>	<ul style="list-style-type: none"> ▪ reconhecimento de planos ▪ observação do contexto ▪ geração automática de texto ▪ sistemas de recomendação ▪ parametrização; ▪ composição declarativa; ▪ composição procedimental concreta; ▪ composição procedimental generalizada; ▪ redação utilizando linguagem de script, linguagem específica da aplicação ou de domínio; ou ▪ redação utilizando linguagem de propósito geral
persiste quanto tempo	persistência	<ul style="list-style-type: none"> ▪ durante uma tarefa (ou enquanto uma tela estiver aberta) ▪ na sessão atual ▪ em todas as sessões até a próxima adaptação

Sistema

A Tabela 3.10 apresenta um conjunto de descritores sobre o sistema. Nesses exemplos de descritores, consideramos a plataforma, a infraestrutura, e os dispositivos de entrada e saída disponíveis durante a interação do usuário com o sistema.

Tabela 3.10: Características do sistema normalmente consideradas no design de IHC.

Sistema		
dimensão	descritor	exemplos de possíveis valores
plataforma	plataforma (hardware + sistema operacional)	<ul style="list-style-type: none"> ▪ web, desktop, PDA, celular ▪ Windows, MacOS, Linux
dispositivos	de entrada	<ul style="list-style-type: none"> ▪ teclado, mouse, microfone, câmera, joystick, caneta, tela sensível ao toque
	de saída	<ul style="list-style-type: none"> ▪ tela, autofalantes, fones de ouvido, impressora
infraestrutura	conexão de rede	<ul style="list-style-type: none"> ▪ 3G, Wi-Fi da universidade, de casa e aeroporto, com fio
	largura de banda	<ul style="list-style-type: none"> ▪ 56 Kbps, 256 Kbps, 512 Kbps, 1Mbps
	espaço para armazenamento	<ul style="list-style-type: none"> ▪ $x \geq 100\text{GB}$, $5\text{GB} < x < 100\text{GB}$, $5\text{GB} \geq x$
	nível de energia	<ul style="list-style-type: none"> ▪ 30%, 50%, 75%, 100% ou na tomada

Projeto

A Tabela 3.11 apresenta um conjunto de descritores sobre o projeto. Nesses exemplos de descritores, consideramos características do projeto em que o caso atual foi elaborado, tais como autor e data.

Tabela 3.11: Características do projeto normalmente consideradas no design de IHC.

Projeto		
dimensão	descritor	exemplos de possíveis valores
definição	nome	<ul style="list-style-type: none"> ▪ Zona Sul Atende, Princesa Online
	autor	<ul style="list-style-type: none"> ▪ AWM Interactive, Marlin
	cliente	<ul style="list-style-type: none"> ▪ Zona Sul, Princesa
	data	<ul style="list-style-type: none"> ▪ 20/10/2002, 03/05/2006

Avaliação

Os descritores que abordam a **avaliação** do caso estão relacionados a avaliações formativas e somativas realizadas anteriormente. A Tabela 3.12 apresenta exemplos de descritores de avaliação acompanhados de possíveis valores. Esses descritores são geralmente encontrados nos relatórios de avaliação de IHC (Barbosa e Silva, 2010).

Tabela 3.12: Características da avaliação normalmente consideradas no design de IHC.

Avaliação		
dimensão	descritor	exemplos de possíveis valores
resultados de avaliação por inspeção	princípios de design favorecidos	<ul style="list-style-type: none"> ▪ elegância, simplicidade, escala, contraste, proporção, organização e estrutura visual, módulo, representação e estilo (Mullet e Sano, 1995) ▪ correspondência com as expectativas dos usuários, simplicidade nas estruturas das tarefas, equilíbrio no controle e liberdade do usuário, consistência e padronização, promover eficiência do usuário, antecipação, visibilidade e reconhecimento, conteúdo relevante, expressão adequada, projeto para erros (Barbosa e Silva, 2010)
	princípios de design prejudicados	<ul style="list-style-type: none"> ▪ idem
	critérios de qualidade favorecidos	<ul style="list-style-type: none"> ▪ facilidade de aprendizado, facilidade de recordação, eficiência, segurança no uso, satisfação do usuário ▪ motivação, diversão, desafio, cansaço, frustração ▪ comunicabilidade, acessibilidade
	critérios de qualidade prejudicados	<ul style="list-style-type: none"> ▪ idem
	gravidade dos erros previstos	<ul style="list-style-type: none"> ▪ problema cosmético, pequeno, grande e catastrófico
resultados de avaliação com a participação de usuários	percentagem de usuários que alcançaram o objetivo principal	<ul style="list-style-type: none"> ▪ 89%, 77%, 52 %
	média de erros cometidos durante o objetivo principal	<ul style="list-style-type: none"> ▪ 3 erros, 5 erros, 0 erros

	tempo médio para alcançar o objetivo principal	▪ 0:56 , 2:12, 3:44
	gravidade dos erros cometidos	▪ problema cosmético, pequeno, grande e catastrófico
treinamento	previsão de custo de treinamento	▪ baixo, médio, alto

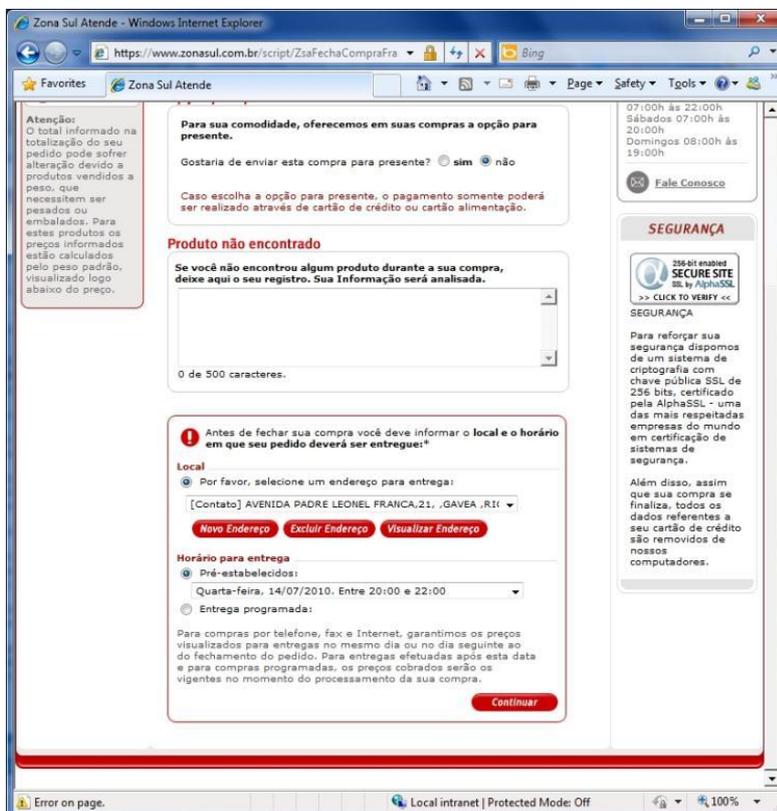
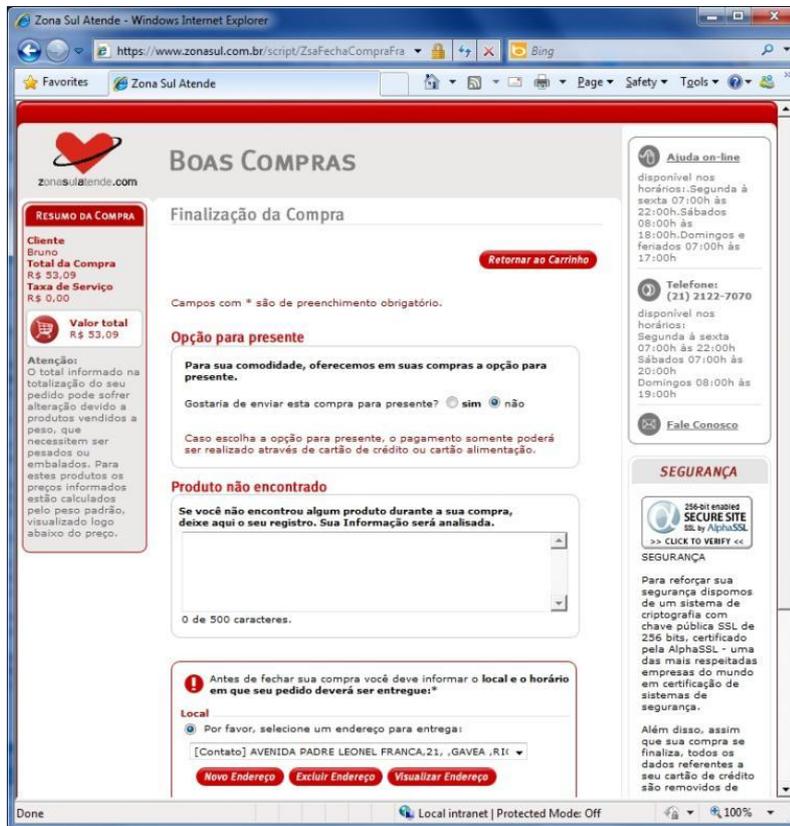
Alguns dos descritores propostos poderiam ser extraídos automaticamente de representações utilizadas no processo de design de IHC. Outros, no entanto, precisam ser definidos manualmente pelo designer.

Exemplo de indexação de um caso de design de IHC

A Tabela 3.13 apresenta um exemplo de descritores para parte da interface do supermercado Zona Sul. Ela apresenta a parte da interface que permite finalizar uma compra de supermercado, indicando se a compra é um presente, o que fazer quando um produto escolhido não for encontrado no estoque e a definição do endereço de entrega. Estes descritores foram elaborados com base numa engenharia reversa do site, e, portanto, podem não corresponder com o aprendizado naquele processo de design.

Tabela 3.13: Exemplo de descritores de um caso de design sobre supermercado online.

Interface Zona Sul



Descritores

interface

mapeamento da interação em interface	diálogos em diferentes telas
widget	link, menu, caixa de texto, rótulo, imagem, botão
estilo de interação	menus e formulário

contexto

local	casa	
nível de atenção	alto ou médio	
incertezas	são inaceitáveis	os produtos solicitados devem ser entregues no dia e horário marcado, sem alteração de quantidade ou de marca do produto não solicitada pelo usuário
evento isolado	dia útil ou toda primeira semana do mês	
interação entre pessoas	podem conversar	sem mediação do sistema
horário	qualquer ou noite	

usuário

idade	adulto	
conhecimentos sobre domínio	alto ou mediano	
conhecimento sobre tarefas	<alto, comprar produto> <médio, comprar de produto>	
experiência com tecnologia	<costuma usar, internet no computador> <costuma utilizar, sistemas de e-mail> < costuma utilizar, sites de notícias> < costuma utilizar, sites de banco>	
preferência	prefere que a entrega seja na sua casa	o endereço do usuário é o valor default de endereço de entrega

domínio

conceito	<cliente> <pedido: preço total> <item: quantidade, preço unitário> <produto: nome, marca, quantidade, informações, preço> <pagamento> <entrega: data prevista, hora prevista, data realizada, hora realizada, valor do frete>
relação	<cliente, faz, pedido> <pedido, possui, pagamento> <pedido, possui, entrega> <produto, é parte de, item> <item, é parte de, pedido>

objetivo	
objetivo	<cadastrar, [F(cliente)], , , geralmente uma vez> <identificar, [O(cliente)], usuário não identificado, usuário identificado, poucas vezes ao mês> <consultar, [O(cliente)], usuário identificado, , , poucas vezes ao ano> <editar, [F(cliente)], usuário identificado, , , poucas vezes ao ano> <buscar, [O(produto)], , , muitas vezes por mês> <comprar, [O(produtos)], usuário identificado, , , algumas vezes por mês>
interação	
passo: tarefa, ação ou fala	<d, apresentar, [O(número de itens), O(preço total)]> <u, definir, [O(entrega)]>
relações	< <u, finalizar, [O(pedido), O(carrinho)]>, sequencial, <d, apresentar, [O(número de itens), O(preço total)]> , sequencial, <u, definir, [F(entrega)]> , sequencial, <u, definir, [F(alteração do pedido)]>, sequencial, <u, definir, [O(forma de pagamento)]> >
adaptação	
agente	usuário
modificação	valor padrão do passo de interação <u, definir, [F(endereço de entrega)]>
motivo	mudou de usuário/usuário identificado
técnica	parametrização
persistência	até a próxima adaptação
avaliação	
princípio de design favorecido	organização e estrutura visual contraste
critério de qualidade favorecido	eficiência por recuperar endereço do usuário
sistema	
plataforma	web, desktop
dispositivo de entrada	teclado, mouse
dispositivo de saída	monitor, impressora
largura de banda	maior ou igual a 256 kbps
projeto	
projeto	Zona Sul Atende
autor	Marlin
cliente	Zona Sul

Considerando que tenhamos uma base de casos de IHC indexada pelos descritores propostos, como poderíamos recuperar casos relevantes?

3.2.3 Recuperação de casos de design de IHC

Computadores costumam ser melhores do que seres humanos para lidar com tarefas sistemáticas e muito repetitivas. Uma tarefa desse tipo é varrer um grande conjunto de coisas em busca de algo que seja de interesse no momento, considerando, é claro, que seja possível processar computacionalmente o que se deseja. Um dos desafios de sistemas de raciocínio baseado em casos é varrer uma base de casos de design e recuperar aqueles que sejam relevantes para o designer no momento. Para definir o que é relevante, precisamos esclarecer primeiro os objetivos pelos quais o designer procura casos na base.

Nosso foco é apoiar a reflexão em ação. Logo, precisamos lembrar a função de casos de design existentes nesse processo de reflexão. Schön (1983) sugere que o designer pode fazer analogias entre o caso atual e casos que já ocorreram no passado, sempre atentando para as semelhanças e, principalmente, as diferenças entre eles. Então, um sistema computacional que recuperasse casos semelhantes ao caso atual tem grandes chances de ser útil para o designer. Como poderia ser esse mecanismo de busca de casos semelhantes quando se trata do design de IHC?

Kim e Yoon (2005) buscam casos semelhantes usando algoritmos que calculam a similaridade entre grafos. Nessa proposta, o designer especifica um caso de design de IHC usando uma gramática que representa uma estrutura das tarefas (ou sequência de passos de interação). O sistema, então, busca e retorna tarefas (ou soluções de interação) com estruturas semelhantes, bem como os objetivos e interfaces associados. Não existe uma forma de referenciar diretamente características da interface, tal como algum tipo de *widget* utilizado. A única forma de ter acesso e referenciar a interface é navegar pelo conjunto de interfaces recuperado via tarefas.

No sistema de raciocínio baseado em casos proposto por Lee e seus colegas (2010), o designer já começa a interagir com uma galeria de sites Web. Se ele desejar, pode ver algum site em detalhes ou buscar sites similares de acordo com *uma dimensão de similaridade*. Ele pode dizer, por exemplo, que deseja ver sites com cor de fundo semelhante ou com número de colunas semelhante. Aqui, a semelhança entre casos é calculada pela semelhança entre as dimensões desejadas e os metadados dos sites. Esse princípio básico é muito útil para a nossa base de casos.

Então, nesta tese propomos recuperar casos de design que possuam descritores semelhantes aos definidos como critério de busca. A pergunta básica que vai guiar a recuperação de casos é a seguinte: **Quais casos possuem descritores semelhantes?** Os descritores utilizados na recuperação de casos serão um subconjunto dos sugeridos na Seção 3.2.2, além de qualquer outro descritor que o designer desejar incluir na sua base de casos. Uma vez encontrado um caso, todos os seus artefatos e os respectivos descritores devem se tornar disponíveis.

Esse modo de consulta à base de casos assume que o designer esteja interessado em casos com certas **características indicadas por descritores**. O designer pode partir de um caso que tenha em mãos e buscar outros casos que contenham descritores semelhantes, ou ser capaz de escolher um conjunto de descritores diretamente do modelo conceitual de casos sem ainda ter em mãos um caso atual.

Um conjunto de descritores pode ser definido por extenso, indicando valor por valor, ou de forma intencional, indicando características dos descritores de interesse. Quando o designer definir cada descritor individualmente, ele busca encontrar casos de design com exatamente aqueles descritores. Por exemplo, o designer pode pesquisar casos de design de sistemas com tela sensível ao toque (descritor dispositivo de entrada) ou pesquisar casos de design de interfaces com menus (descritor estilo de interação). Quando o designer define os descritores de forma intencional, ele pretende encontrar casos de design com descritores semelhantes. Por exemplo, o designer pode pesquisar casos de design elaborados entre 2008 e 2010 (um intervalo de possíveis valores do descritor data de projeto) ou pesquisar casos de design que tratam da ação comprar (um verbo dentre os possíveis valores do descritor objetivo).

O modelo conceitual dos descritores (Figura 3.7) prevê que os possíveis valores de um descritor e as relações entre eles podem estar organizados em um **domínio de valores**. Através da navegação por esse domínio de valores, um sistema computacional pode processar um critério de busca que referencia atributos e relações dos descritores definidos intencionalmente. Existem várias formas de se navegar pelo domínio de valores dos descritores em busca de casos de design similares. As formas mais comuns procuram descritores por:

- atributos:
 - um termo;

- um termo com atributo A;
- relações de generalização ou especialização (“é_um”), tais como:
 - irmãos de um termo T: (P, é_um, ?) e (T, é_um, P);
 - pais de um termo T: (T, é_um, ?);
 - filhos de um termo T: (?, é_um, T);
- relações metonímicas (Eco, 1984), tais como:
 - partes do termo T: (?, parte_de, T);
 - produtor de T: (?, produz, T);
 - localizado em T: (?, localizado_em, T).

A navegação entre conceitos do domínio por relações de generalização/especialização e por relações metonímicas já foi utilizada com propósitos semelhantes. Barbosa (1999) busca inferir o significado de novas expressões do usuário na interface através de metáforas e metonímias. Barbosa, Breitman e seus colegas (Barbosa *et al.*, 2007; Breitman *et al.*, 2007) investigam como modelos conceituais podem ser enriquecidos com relações metonímicas e metafóricas. Silva e seus colegas (Silva *et al.*, 2008; Silva e Barbosa, 2008) instigam como as relações de metáfora e metonímia pode contribuir com a análise de variabilidade dos objetivos do usuário.

Também é possível definir critérios de busca mais complexos utilizando os operadores “e”, “ou” e “não”. Por exemplo, a pergunta “O que se pode fazer a respeito de produtos além de comprar?” poderia ser respondida por uma pesquisa por descritores de objetivos: < ? , [O(produtos) | (O(produto), parte_de, ?)], _ , _ , _> e (não (< comprar , [O(produtos)], _ , _ , _>)); onde um objetivo é expresso na forma <verbo, [slots], [pré-condição], [pós-condição], [frequência]>. Se considerarmos que os slots dos objetivos podem ser preenchidos por conceitos do domínio apresentados na Figura 3.8, podemos navegar pelas relações entre os conceitos em busca de uma relação semelhante a (O(produto), parte_de, ?). Isso pode retornar casos associados aos objetivos “consultar informações de produtos” e “comparar produtos” pela variação do verbo, e “consultar pedidos” e “definir lista de compras” pela relação de “parte_de”.

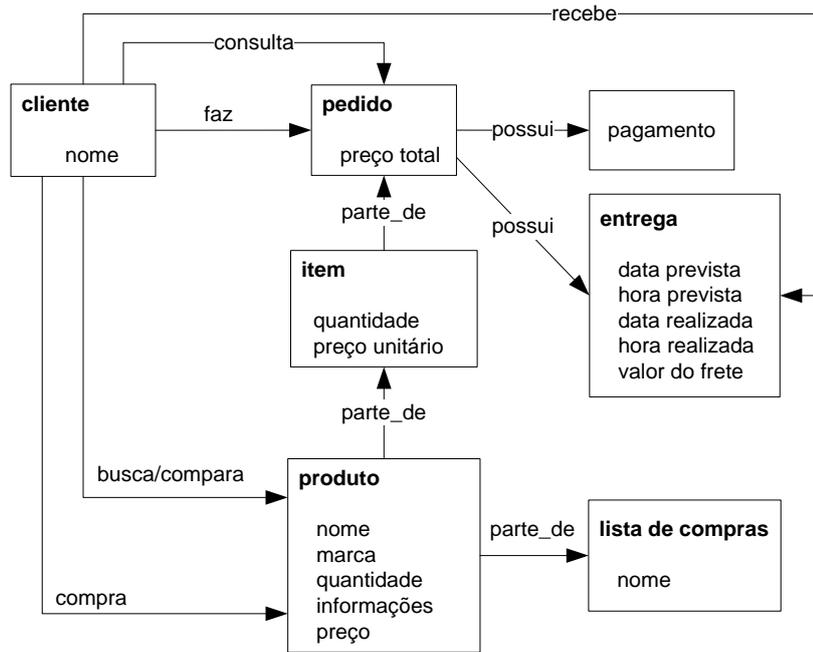


Figura 3.8. Exemplo de parte do modelo conceitual de supermercado online.

Pensando no domínio de supermercados, o designer poderia, por exemplo, pesquisar casos de design com base nas seguintes perguntas:

Quais **casos** possuem...

linguagem natural

usando descritores de casos de design

...domínio de supermercado?

- **projeto.nome** contendo *supermercado* ou
- **domínio.conceito** contendo *supermercado*

...objetivos do usuário com verbos comprar ou vender?

- **objetivo** na forma <comprar,_,_,_,_> ou
- **objetivo** na forma <vender,_,_,_,_>

...objetivos que manipulam produto, pedido ou pagamento?

- **objetivo** na forma <_,O(prodoto),_,_,_> ou
- **objetivo** na forma <_,O(pedido),_,_,_> ou
- **objetivo** na forma <_,O(pagamento),_,_,_>

Quando o designer tem acesso a projetos semelhantes, ele pode consultar e comparar seus respectivos objetivos, interações e interfaces, bem como seus problemas de design e avaliações. As semelhanças nessa situação podem variar desde outros projetos para sites de supermercados, passando por sites que vendem produtos específicos como eletrônicos ou flores, até sistemas que vendem produtos que os supermercados não vendem, como passagem aérea, por exemplo. Analisar esses outros sistemas pode ser útil para o designer identificar como os usuários

costumam atingir o objetivo de compra nesses outros casos e qual o suporte computacional costuma ser oferecido. Por analogia, as soluções de design encontradas podem apontar caminhos a serem explorados ou caminhos que não devem ser percorridos na elaboração da solução do caso atual.

Para recuperar casos com domínio e interações semelhantes, o designer poderia utilizar, por exemplo, as seguintes perguntas:

Quais **casos** possuem...

linguagem natural

usando descritores de casos de design

...manipulam algo semelhante a carrinho de compras?

- como outros recipientes com o mesmo tipo de conteúdo que carrinho de compras, ou
- como outras coisas manipuladas da mesma forma: buscar e escolher um conjunto, depois aplicar uma operação sobre ele

- **domínio.conceito** contendo *carrinho* ou
- **interação.passo** na forma $\langle _ , _ \rangle$ onde $(Conteúdo?, parte_de, carrinho)$ e $(Conteúdo?, parte_de, Recipiente?)$ ou
- **interação.sequência** na forma $\langle u, buscar, Coisas? \rangle, sequencial, [...], \langle u, Operação?, Coisas \rangle$

...tratamento de erros no objetivo buscar?

- **interação.passo** na forma $\langle d, informar, [F(erro na entrada)], _ , _ \rangle$ ou
- **interação.passo** na forma $\langle d, informar, [F(erro no processamento)], _ , _ \rangle$ ou
- **interação.passo** na forma $\langle d, informar, [F(erro na saída)], _ , _ \rangle$

...suporte tutorial?

- **interação.suporte** igual a *tutorial*

Ao buscar por casos com interação semelhantes, o designer tem oportunidade de perceber passos de interação que podem tornar a experiência de uso mais interessante ou prejudicá-la. Por exemplo, ele pode descobrir recursos de interação que facilitam a vida do usuário, tal como o uso de listas de compras definidas manualmente pelo usuário ou com base em um pedido específico.

Para recuperar casos com interfaces, sistemas e avaliações semelhantes, o designer poderia utilizar, por exemplo, as seguintes perguntas:

Quais **casos** possuem...

linguagem natural

usando descritores de casos de design

... widget botão de comando para plataforma web?

- **interface.widget** contendo *botão de comando* e
- **sistema.plataforma** contendo *web*

... interação por formulário para plataforma web?	· interface.estilo de interação contendo <i>formulário</i> e · sistema.plataforma contendo <i>web</i>
... uma boa organização visual?	· avaliação.princípio de design favorecido contendo <i>organização e estrutura visual</i>
... um contraste de cores ruim?	· avaliação.princípio de design prejudicado contendo <i>contraste</i>

Ao examinar interfaces semelhantes, o designer pode, por exemplo, identificar como botões de comando costumam ser nomeados, dimensionados e dispostos em sistemas web. Ele também pode aprender sobre layouts de formulários e o contraste das cores, formas e disposição dos elementos na interface.

A consulta à base de casos usando descritores tem o potencial de permitir ao designer controlar melhor quais casos devem ser recuperados. Se ele conseguir definir precisamente as características dos casos que lhe interessam, ele pode indicar um por um os valores dos descritores. Se, por outro lado, o designer tiver apenas uma noção do que ele deseja, ele pode indicar os descritores desejados por intervalos, atributos e relações. O conjunto de descritores definido como critério de busca possibilita ao designer certo controle sobre as similaridades e diferenças entre o caso atual e os casos recuperados. Os próprios descritores servem como dimensões de similaridades consideradas na recuperação de casos de design. Pesquisas mais gerais tendem a retornar casos com mais diferenças. Pesquisas específicas tendem a retornar casos com mais similaridades.

3.3 Reflexão em ação consultando casos de design de IHC

Como a consulta a casos existentes poderia ser utilizada durante o processo de design de IHC para apoiar a reflexão em ação? Schön (1986) sugere que o designer possa fazer analogias entre o caso atual e casos que já ocorreram no passado. Na construção dessas analogias, é fundamental que o designer busque identificar as semelhanças e, principalmente, as diferenças entre o caso atual e os casos recuperados. Assim, quando o designer conseguir construir analogias entre os problemas, ele também pode fazer analogias entre as formas de solucioná-los para encontrar caminhos a serem explorados na solução do problema atual único. Nesse processo de solução de problemas por analogias, algumas vezes a solução para o problema atual pode ser elaborada através de adaptações das soluções existentes. Como o problema atual possui alguma diferença em relação a todos os outros,

muito raramente as soluções propostas podem ser as mesmas por completo. Todas as soluções deveriam ter algumas diferenças entre si.

Nesse processo de reflexão em ação usando analogias, a comparação entre casos de design assume papel fundamental, pois ao compará-los o designer tem condições de identificar as diferenças e semelhanças entre eles. O designer precisa criticar as semelhanças e diferenças entre os casos para julgar se algo poderia ou não ser explorado na elaboração da solução do problema atual.

Esse tipo de análise e reflexão é muito comum durante avaliações formativas, principalmente quando o designer explicita várias soluções possíveis e decide o que deve continuar ou não sendo explorado. Se o designer desligar o seu espírito crítico ao analisar os casos recuperados, ele corre grandes riscos de tentar aproveitar algo que não deveria. O mesmo tipo de prejuízo ocorre quando o designer desliga seu espírito crítico durante uma avaliação de IHC, e deixa passar algo inadequado ao problema atual.

O espírito crítico do designer lhe motiva a questionar uma proposta de solução e a não se conformar com uma solução ruim ou que ainda pode ser melhorada. Esse espírito crítico é um dos motores do processo de reflexão em ação, que deve estar ativo durante todo o processo de design, seja com a consulta ou não de casos existentes.

Nesse sentido, podemos propor um conjunto de perguntas para estimular a comparação, a avaliação e o espírito crítico do designer durante a análise de casos de design existentes. As perguntas que o designer deve buscar responder são as seguintes:

- Quais são as semelhanças e diferenças entre a solução proposta e as existentes?
- Quais são os efeitos dessas semelhanças e diferenças sobre o uso?
- Considerando as soluções existentes, a solução proposta é satisfatória ou poderia ser melhorada? Por quê?
- Quais modificações poderiam ser exploradas na solução proposta? Por quê?

O designer pode começar a procurar semelhanças e diferenças entre as soluções de design usando os descritores como dimensões de comparação. Por exemplo, ao comparar os objetivos, o designer pode perceber que dois sites de super-

mercados permitem ao usuário comprar produto, apesar de somente um deles permitir ao usuário criar uma lista de compras para facilitar as futuras compras. Ao comparar as sequências de passos, o designer pode descobrir que a distância entre concluir um pedido e consultar os pedidos do cliente pode ser de um passo (<u: confirmar, [O(pedido)]>, sequencial, <u: consultar, [O(pedido)]>) ou de dois passos (<u: confirmar, [O(pedido)]>, sequencial, <u: consultar, [O(cliente)]>, sequencial, <u: consultar, [O(pedido)]>).

Depois de identificadas semelhanças e diferenças entre a solução proposta e outras soluções recuperadas da base de casos, o designer pode tentar prever possíveis consequências das características semelhantes sobre a experiência de uso. Um bom ponto de partida para tentar prever essas consequências é analisar os descritores de avaliação. Por exemplo, considerando-se os passos de interação, se uma sequência de passos possui média de tempo para ser concluída menor do que outra, então soluções semelhantes à primeira tendem a ser mais eficientes do que soluções semelhantes à segunda. Entretanto, como a experiência de uso é influenciada por mais de um fator da solução e do problema, é importante que o designer também considere os descritores de outros tipos de representação, além da avaliação. Por exemplo, uma sequência de interação pode ser eficiente quando o usuário é experiente no objetivo associado (descriptor experiência com as tarefas = conhece tarefas avançadas) e busca alcançar esse objetivo com frequência (descriptor objetivo.frequência = algumas vezes por mês), mas ser ineficiente para usuários inexperientes (descriptor experiência com as tarefas = sem experiência) que raramente buscam alcançar aquele objetivo (descriptor objetivo.frequência = poucas vezes por ano).

Uma vez identificadas semelhanças entre as soluções e previstos seus possíveis efeitos durante o uso, o designer tem insumos para julgar se a solução proposta é satisfatória ou não comparada às demais. Por exemplo, ele pode concluir que a solução proposta está muito aquém das soluções existentes e que deveria explorar algumas ideias empregadas em outras soluções; ou concluir que a solução proposta está próxima das soluções existentes com certos pontos de vantagens; ou ainda perceber que a solução proposta explora uma linha completamente diferente das soluções existentes e deve ser avaliada junto aos usuários.

Caso o designer perceba que a solução proposta não é satisfatória quando comparada a outras soluções ou que alguns aspectos podem ser melhorados, que modificações poderiam ser exploradas na solução proposta? O designer pode ex-

plorar pontos na solução proposta que poderiam ser semelhantes a outras soluções e não são, ou explorar pontos na solução proposta que poderiam ser diferentes de outras soluções e não são.

Nos mecanismos de adaptação (descriptor modificação da Tabela 3.9), encontramos dimensões em que as soluções de IHC podem variar (o que pode mudar durante o uso). Essas dimensões de variação apontam caminhos que o designer pode explorar para tornar a solução proposta mais semelhante ou mais diferente das soluções existentes. Essas dimensões de variação também servem para apontar caminhos que o designer pode explorar para propor diferentes alternativas de soluções. Se elas forem usadas para convergir soluções, elas podem contribuir com um raciocínio convergente. Se elas forem usadas para elaborar alternativas de solução, elas podem contribuir com um raciocínio divergente (nos termos de Lawson (1993): *parallel lines of thought*). Segundo Lawson (2006), a capacidade de mudar a direção dos pensamentos é uma habilidade de pessoas criativas.

A comparação entre os casos de design também pode ser apoiada por computador, caso os artefatos utilizados possuam um metamodelo processável. Por exemplo, Kim e Yoon (2005) apresentam um algoritmo que compara os passos de interação registrados em uma representação específica. Entretanto, nem todos os artefatos utilizados são facilmente comparáveis. Por exemplo, não é simples comparar computacionalmente cenários e personas escritos em linguagem natural. Além de comparar os artefatos, também pode ser útil comparar os descritores, dado que eles são uma abstração de características dos artefatos. Mesmo que a comparação feita por computador seja ótima, ainda assim caberá ao designer julgar as consequências das semelhanças e diferenças identificadas.

No próximo capítulo apresentamos uma pesquisa qualitativa que investiga a consulta a casos existentes durante atividades de design de IHC, conforme proposto neste capítulo.