

CAPÍTULO 1

O QUE É INTERAÇÃO/INTERFACE HUMANO-COMPUTADOR

The ideal system so buries the technology that the user is not even aware of its presence. The goal is to let people get on with their activities, with the technology enhancing their productivity, their power, and their enjoyment, ever the more so because it is invisible, out of sight, out of mind. People should learn the task, not the technology. They should be able to take the tool to the task, not as today, where we must take the task to the tool. And these tools should follow three axioms of design: simplicity, versatility, and pleasurable

Norman, 1998, pg xii

INTRODUÇÃO

Novas tecnologias provêem poder às pessoas que as dominam. Sistemas computacionais e interfaces acessíveis são novas tecnologias em rápida disseminação. Explorar o poder do computador é tarefa para designers que entendem da tecnologia e são sensíveis às capacidades e necessidades humanas.

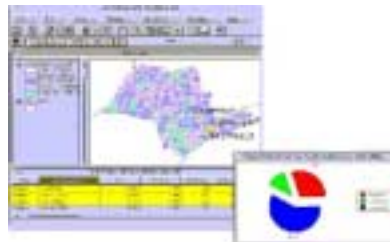
A performance humana no uso de computadores e de sistemas de informação tem sido uma área de pesquisa e desenvolvimento que muito se expandiu nas últimas décadas. Isso tem sido feito usando-se poderosas ferramentas computacionais na análise de dados coletados de acordo com métodos da Psicologia Experimental. Outras contribuições também advêm da Psicologia Educacional, do Design Instrucional e Gráfico, dos Fatores Humanos ou Ergonomia, e bem mais recentemente, da Antropologia e da Sociologia.

Interfaces de usuário têm produzido importantes histórias de sucesso tais como a da Netscape, America Online, Universal Online, ou Yahoo. Elas também tem produzido intensa competição, disputas por direitos autorais (por exemplo, Apple e Microsoft com relação à interface Windows), mega fusões (como a recente entre a America Online e TimeWarners), etc.

Individualmente, interfaces de usuário têm mudado a vida de muitas pessoas: médicos estão podendo fazer diagnósticos mais precisos; crianças estão expandindo os horizontes em ambientes de aprendizagem; artistas gráficos podem explorar mais possibilidades criativas; e pilotos têm mais segurança em seus vãos. Entretanto, algumas mudanças são perturbadoras e até desastrosas; freqüentemente usuários têm que lidar com frustração, medo e falha quando encontram design excessivamente complexos, com terminologia incompreensível e caóticos.

O crescente interesse no projeto de interfaces do usuário é bastante claro nos mais variados tipos de sistemas (Figuras 1.1, 1.2). Processadores de texto, ferramentas de edição, e softwares de manipulação de imagens são amplamente utilizados

FIGURA 1.1 – TELAS DO SIG ARCVIEW COM
MAPA MOSTRANDO A MALHA MUNICIPAL DO
ESTADO DE SÃO PAULO



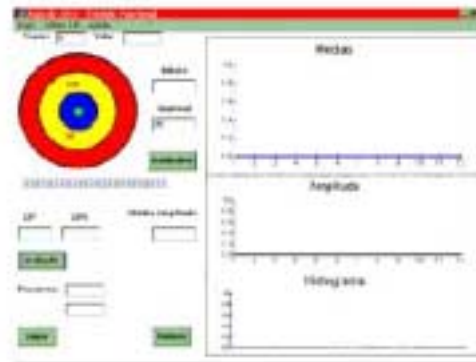


FIGURA 1.2 – TELA DO *JOGO DO ALVO* DESENVOLVIDO PELO NIED - UNICAMP NO PROJETO DE FORMAÇÃO DE RECURSOS HUMANOS PARA A EMPRESA ENXUTA

Correio eletrônico, vídeo conferência e a WWW têm oferecido novas mídias para comunicação. Bibliotecas digitais de imagens têm se expandido em aplicações que vão da medicina (Figura 1.3) até a exploração do espaço.



Figura 1.3 – IMAGENS OBTIDAS DO *SITE* VISIBLE MAN QUE POSSUI UMA VASTA BIBLIOTECA DE REFERÊNCIAS MÉDICAS

Visualização científica e simuladores remotos permitem experimentos seguros e treinamento a baixo custo. Acesso público e educacional a *sites* de museus (Figura 1.4), bibliotecas ou fontes de informação governamentais estão se ampliando (Figuras 1.5,1.6).



FIGURA 1.4 – SITE DO MUSEU DE ARTE DE SÃO PAULO – PÁGINA DE ENTRADA E PÁGINA PRINCIPAL DA EXPOSIÇÃO VIRTUAL DO MICHELANGELO



FIGURA 1.5 – SITES DA BIBLIOTECA NACIONAL E BIBLIOTECA DA ESCOLA DO FUTURO DA USP

FIGURA 1.6 – FONTES DE INFORMAÇÃO DO GOVERNO BRASILEIRO SOBRE DADOS ESTATÍSTICOS (IBGE) E AIDS



Ferramentas computacionais específicas e ambientes de programação permitem construir protótipos rápidos como as ferramentas de auxílio ao design de produtos industriais. Muitos de nós usam vários produtos eletrônicos, como os gravadores de vídeo cassete, fornos de microondas, telefones, etc. Arte, música, esportes e entretenimento são atualmente auxiliadas e suportadas por sistemas computacionais.

Profissionais das mais diferentes áreas têm contribuído significativamente para todo esse desenvolvimento, e dentre estes podemos destacar:

- **Designers de software** têm explorado maneiras melhores de organizar informação graficamente. Eles têm desenvolvido linguagens de consulta e facilidades visuais para entrada, busca e saída de informação. Têm usado sons (música e voz), representações tridimensionais, animação e vídeo para melhorar o conteúdo e a expressão das interfaces. Técnicas como manipulação direta, tele-presença, e realidade virtual mudam a maneira de interagir e de pensar sobre computadores.
- **Desenvolvedores de hardware** têm oferecido novos design de teclados e dispositivos de apontamento, além dos *displays* de alta resolução. Eles têm projetado sistemas com resposta rápida para complexas manipulações tridimensionais. Tecnologias que permitem entrada e saída por voz, entrada por gestos, telas de toque, em muito têm aumentado a facilidade de uso dos computadores.
- **Desenvolvedores na área de tecnologia educacional** estão criando tutoriais *online*, e materiais de treinamento e explorando novas abordagens de discussões em grupo, ensino a distância, apresentações de vídeo etc. **Designers gráficos** estão fortemente envolvidos com o *layout* visual, seleção de cores e animação. **Sociólogos, antropólogos, filósofos, administradores** estão tratando do

impacto organizacional, ansiedade computacional, treinamento, grupos de trabalho distribuídos, suporte computacional ao trabalho cooperativo, e mudanças sociais em geral.

Portanto, estamos vivendo um momento vital e estratégico para os desenvolvedores de interfaces. Pode-se dizer que a tecnologia está pronta. Temos portanto as pontes e túneis construídos e agora as estradas precisam ser pavimentadas e as sinalizações pintadas para tornar possível o pesado tráfego da grande leva de usuários (Schneiderman, 1998).

Neste capítulo estaremos apresentando a área de Interfaces Humano-Computador de modo a poder situar e apresentar ao leitor os problemas, a terminologia e conceitos envolvidos nessa área de estudo e aplicação. Vale lembrar que muitos dos aspectos aqui apresentados serão aprofundados em capítulos subsequentes.

INTERFACE HUMANO-COMPUTADOR

Quando o conceito de interface surgiu, ela era geralmente entendida como o hardware e o software com o qual homem e computador podiam se comunicar. A evolução do conceito levou à inclusão dos aspectos cognitivos e emocionais do usuário durante a comunicação.

Muito embora algumas pessoas ainda possam se lembrar dos antigos teletipos, é comum hoje em dia pensarem na interface como a tela e o que nela é mostrado. O nome interface é tomado como algo discreto e tangível, uma coisa que se pode desenhar, mapear, projetar e implementar, "encaixando-a" posteriormente a um conjunto já definido de funcionalidades. Um dos objetivos deste livro é acabar com essa idéia substituindo-a por outra que ajude os construtores de interfaces a irem na "direção correta".

De acordo com Brenda Laurel (1990) a "direção correta" é aquela que leva o usuário a ter mais poder. Por exemplo, uma nova versão de um editor de textos comumente oferece o dobro de opções que a versão anterior. E com isso se espera que o usuário possa customizar melhor seu uso e conseguir atingir objetivos mais complexos. Este objetivo nem sempre é conseguido, pois o enorme conjunto de funções e as convenções de interface que deverão ser aprendidas de modo a se poder usufruir as pretensas novas qualidades, na maioria dos casos, deixam o usuário atônito e cansado. Certamente as melhoras acrescentadas ao produto oferecem ao usuário mais poder e qualidade ao produto final, oferecendo mais graus de liberdade na sua concepção. Mas tudo isso se perde quando o custo para o usuário é muito alto. O que acontece é que a nova versão é adotada, muitas vezes por problemas de compatibilidade entre diferentes versões de um produto, mas toda melhoria é deixada de lado e o usuário continua usando o mesmo domínio de funções que ele já

conhecia. Concluindo, para que o usuário tenha mais poder, é preciso sim, que mais funcionalidade seja oferecida mas é fundamental a facilidade de uso.

UMA PRIMEIRA DEFINIÇÃO DE INTERFACES

Primariamente, como já dissemos, se visualiza uma interface como um lugar onde o contato entre duas entidades ocorre (por exemplo, a tela de um computador). O mundo está repleto de exemplos de interfaces: a maçaneta de uma porta, uma torneira, a direção de um carro, etc.

A forma das interfaces reflete as qualidades físicas das partes na interação. A maçaneta de uma porta é projetada para se adequar à natureza da mão que irá usá-la, o mesmo acontece com o câmbio de um carro (observe que a localização do câmbio dentro do carro sugere o uso por uma pessoa destra). Existem tesouras de dois tipos uma para pessoas destros e outra para pessoas canhotas.

O que muitas vezes é esquecido é que a forma da interface também reflete o que pode ser feito com ela. Tomando o exemplo da maçaneta, podemos ver que no mundo existem diversos formatos de maçaneta e de acordo com o formato sabemos como deve ser aberta uma porta: girando a maçaneta no sentido anti-horário, empurrando a porta, puxando a porta, etc. (Norman, 1988). O mesmo acontece com a forma das torneiras onde se deve girar ou empurrar ou levantar uma alavanca, etc.

Nos exemplos anteriores da porta e da torneira que foram feitas para serem abertas por um humano podemos dizer que o humano é o agente e a porta (ou torneira) são os pacientes dessa ação. Mas, temos também as portas, ou torneiras, que abrem automaticamente quando identificam através de um sensor ou uma câmera a presença de alguém (mesmo que esse alguém não queira abrir a porta). Nesse caso o sentimento que temos de quem está controlando a interação é bastante diferente. Em muitos banheiros públicos existem instalados aqueles secadores automáticos de ar quente para mãos e muitas vezes, mesmo não querendo usá-los eles se ligam porque nos encostamos próximos a eles ou sem querer passamos a mão perto do sensor. E as torneiras que sempre se fecham antes de acabarmos de lavar as mãos? Nesses casos, não é mais o humano que está no controle da interação.

Portanto, podemos ter como uma definição de base, que uma interface é uma superfície de contato que reflete as propriedades físicas das partes que interagem, as funções a serem executadas e o balanço entre poder e controle (Laurel, 1993).

EVOLUÇÃO DE INTERFACES E SUA CONCEITUAÇÃO

Interface tornou-se uma tendência (ou moda, como nomeiam os mais incrédulos) como um importante conceito a ser explorado nos últimos anos, e isso é largamente atribuído a introdução dos computadores Macintosh da Apple. Certamente, quando se pensa hoje em dia em Interfaces Humano-Computador (IHC) imediatamente se visualiza ícones, menus, barras de rolagem ou talvez, linhas de comando e cursores piscando. Mas certamente interface não é só isso.

Podemos fazer um histórico analisando a geração de interfaces, da mesma forma com que analisamos gerações de computadores, ou seja, fazendo um forte paralelo com os componentes de hardware que as suportam (Tesler, 1991). Nielsen(1993) apresenta uma tabela onde ele faz esse relacionamento e também qualifica a categoria de usuários de computadores em cada geração, o que é de absoluta relevância para o desenvolvimento de interfaces (Tabela 1.1).

GERAÇÃO	TECNOLOGIA DE HARDWARE	MODO DE OPERAÇÃO	LINGUAGENS DE PROGRAMAÇÃO	TECNOLOGIA TERMINAL	TIPO DE USUÁRIOS	IMAGEM COMERCIAL	PARADIGMA DE INTERFACE DE USUÁRIO
-1945 pré-histórica	Mecânica e eletromecânica	Usado somente para cálculos	Movimento de cabos e chaves	Leitura de luzes que piscam e cartões perfurados	Os próprios inventores	Nenhuma (computadores não saíram dos laboratórios)	Nenhum
1945-1955 pioneira	Válvulas, máquinas enormes e com alta ocorrência de falha	Um usuário a cada tempo usa a máquina (por um tempo bastante limitado)	Linguagem de máquina 001100111101	TTY. Usados apenas nos centros de computação	Especialistas e pioneiros	Computador como uma máquina para cálculos	Programação, batch
1955-1965 histórica	Transistores, mais confiáveis. Computadores começam a ser usados fora de laboratórios	Batch (computador central não acessado diretamente)	Assembler ADD A,B	Terminais de linha glass TTY	Tecnocratas, profissionais de computação	Computador como um processador de Informação	Linguagens de Comando
1965-1980 tradicional	Circuito integrado. relação custo-benefício justifica a compra de computadores para muitas necessidades	Time-sharing	Linguagens de alto nível (Fortran, Pascal, C)	Terminais full screen, caracteres alfa-numéricos. Acesso remoto bastante comum	Grupos especializados sem conhecimento computacional (caixas automáticos, p.ex.)	Mecanização das atividades repetitivas e não criativas	Menus hierárquicos e preenchimento de formulários
1980-1995 moderna	VLSI. Pessoas podem comprar seu computador.	Computador pessoal para um único usuário	Linguagens orientadas a problemas/objetos (planilhas de cálculo)	Displays gráficos. estações de trabalho, portáteis	Profissionais de todo tipo e curiosos	Computador como uma ferramenta	WIMP (Window,Icons,Menu, e Point devices)
1995- futura	Integração de alta-escala. Pessoas podem comprar diversos computadores	Usuários conectados em rede e sistemas embutidos	Não imperativas, provavelmente gráficas	Dynabook, E/S multimídia, portabilidade simples, modem celular	Todas as pessoas	Computador como um aparelho eletrônico	Interfaces não baseadas em comando.

TABELA 1.1 - GERAÇÃO DE COMPUTADORES E DE INTERFACES DE USUÁRIOS (ADAPTADO DE NIELSEN,1993, P.50



De modo semelhante Walkers (1990) faz uma análise histórica da evolução de interfaces sob o aspecto do tipo de interação entre o usuário e o computador. No início havia um relacionamento um a um entre uma pessoa e o computador através de chaves e mostradores das primeiras máquinas como o ENIAC ou UNIVAC, EDVAC (foto ao lado)¹.

O advento dos cartões perfurados e do processamento em *batch* substituiu essa interação direta entre o homem e o computador por uma transação mediada pelo operador do computador. *Time sharing* e o uso dos teletipos trouxeram novamente o contato direto e conduziram o desenvolvimento das interfaces de linhas de comando e orientadas por menu (Figura 1.7). O estilo de diálogo é bastante simples, onde uma pessoa faz alguma coisa e o computador responde



FIGURA 1.7 – TELAS DE UM PROGRAMA COM ESTRUTURA DE DIÁLOGO POR MENU E LINEAR

Essa noção simplista de uma conversação levou ao desenvolvimento de um modelo de interação que trata o humano e o computador como duas entidades diferentes que conversam intermediadas por uma tela.

¹ Extraída de <http://ei.cs.vt.edu/~history/index.html> em fev./2000

Avanços da Lingüística têm demonstrado que diálogo não é linear, ou seja, quando dizemos alguma coisa, você pensa sobre o que dissemos e aí dá uma resposta, nós vamos pensar sobre a resposta e aí, e assim por diante. Portanto, para que o diálogo efetivamente ocorra é necessária a existência, ou a construção, de um meio comum de significados.

As atuais interfaces gráficas explicitamente representam o que vem a ser esse meio de significados comum, pela aparência e comportamento dos objetos na tela. Este conceito dá suporte a idéia de que uma interface é um contexto compartilhado de ação no qual tanto o computador como o humano são agentes (Laurel, 1993). Enganos, resultados inesperados e mensagens de erro são evidência típica de uma quebra na conversação, onde o pretenso meio de significados comum torna-se uma seara de desentendimentos.

A noção de **metáforas de interfaces** (Carroll *et al.*, 1988; Wozny, 1989) foi introduzida para prover às pessoas um esquema do funcionamento da interface que prevenisse tais desentendimentos, ou seja, facilitassem a criação desse contexto compartilhado. Por que metáforas?

METÁFORAS DE INTERFACE

Metáforas são parte integrante de nosso pensamento e linguagem. Elas aparecem não somente na poesia ou literatura, mas em nossa linguagem cotidiana. E geralmente as pessoas não se dão conta de que estão usando metáforas, elas são invisíveis. Alguns exemplos bastante comuns: *gastar* dinheiro, *atacar*, *defender* e *destruir* um argumento; *tratar* superficialmente um assunto; trânsito *engarrafado*; etc. As metáforas funcionam como modelos naturais, nos permitindo usar conhecimento familiar de objetos concretos e experiências para dar estrutura a conceitos mais abstratos. As características de metáforas em nossa linguagem são as mesmas que governam o funcionamento de metáforas de interfaces. Da mesma forma que metáforas invisíveis permeiam nossa linguagem cotidiana elas o fazem nas interfaces que usamos e projetamos (Erickson, 1990). Por exemplo, um usuário quando arrasta um documento de um diretório (ou *pasta*) para outro nos sistemas gerenciadores de arquivos de ambientes Windows, ele efetivamente acredita que está mudando o documento de lugar e o que efetivamente ocorre é que o *apontador* para o arquivo mudou (*apontador* também é uma metáfora).

Como as metáforas são usadas como modelos, uma metáfora de interface que sugira um modelo incorreto pode causar dificuldades para o usuário. Por exemplo, o clássico caso das funções *cortar* e *colar* dos editores atuais - quando se corta algum objeto ele fica guardado em um buffer (usuários principiantes acham que sumiu) e quando se cola em outra parte o objeto não cola como no real, ele "empurra" (para fazer a real função de colar é preciso marcar e depois colar).

E também, mesmo boas metáforas, não funcionam em sua totalidade. Por exemplo, considerando-se a metáfora da mesa de trabalho (*desktop metaphor*) dificilmente as pessoas conseguem explicar satisfatoriamente o funcionamento de uma função como a de busca, por exemplo, pois ela diverge significativamente da pretendida referência ao mundo real. Nesses casos, metáforas servem como auxiliares ao entendimento atuando como mediadores cognitivos cujos rótulos são menos técnicos que os do jargão computacional.

Mesmo não funcionando sempre, o seu uso crescente, especialmente em interfaces gráficas, favoreceu (ou forçou) a expansão do domínio da área de design de interfaces, com contribuições mais que relevantes de outras especialidades como design gráfico e industrial, lingüística, Psicologia e Educação dentre outras. Portanto, uma importante contribuição da abordagem metafórica foi ter tornado o design e estudo de interfaces uma preocupação inter(multi)(trans)disciplinar. Nas próximas seções deste capítulo, e no decorrer do livro como um todo, estaremos clarificando essa natureza inter(multi)(trans)disciplinar tanto no design como na avaliação de interfaces. No Capítulo 3 voltaremos a discutir sobre o entendimento e uso de metáforas no design de interfaces.

Concluindo, o que vimos nessas primeiras seções é que não se consegue ter um conceito simplista de interface como *os* aspectos do sistema com os quais o usuário tem contato, ou ainda a um pouco mais elaborada linguagem de entrada para o usuário, linguagem de saída para a máquina e um protocolo de interação (ACM CHI'85). Não se pode pensar em interfaces sem considerar o ser humano que vai usá-la, e portanto **interface e interação são conceitos que não podem ser estabelecidos ou analisados independentemente**. E no decorrer deste livro ao nos referirmos a interfaces estaremos focando a interação, o que para nós dá a amplitude desejada ao termo interface, pois pensar somente na "interface" é pensar muito pequeno. As preocupações usuais dos designers de interfaces - criar tipos mais legíveis, melhores barras de rolagem, integrar cor, som e voz - são todas importantes, mas são secundárias. A preocupação primeira deve ser a de melhorar o modo como as pessoas podem usar o computador para pensar e comunicar, observar e decidir, calcular e simular, discutir e projetar.

INTERAÇÃO HUMANO-COMPUTADOR (IHC)

Para que os computadores se tornem amplamente aceitos e efetivamente usados eles precisam ser bem projetados. Isso de maneira alguma quer dizer que o design deve ser adequado a todas as pessoas, mas os computadores devem ser projetados para as necessidades e capacidades de um grupo alvo. Certamente, usuários em geral não devem ser obrigados a pensar sobre como o computador funciona, da mesma forma que o funcionamento mecânico de um carro não é preocupação da maioria das pessoas. Entretanto, a posição dos pedais, direção e câmbio têm muito impacto sobre

o motorista, como também o design de sistemas computacionais têm efeito sobre seus usuários.

Empresas produtoras de software têm despertado para idéia de que a melhora no aspecto físico da interface do usuário proporciona maiores chances de sucesso de mercado. Para explorar essa nova dimensão do produto surgiu um termo amplamente usado - interface amigável ou sistema amigável (*user-friendly*). Na prática o significado do amigável está associado somente a uma interface, ou melhor, aos elementos na tela serem esteticamente mais agradáveis ou bonitos. Muito embora tenha implicado num avanço com relação às antigas interfaces, muitas empresas usaram o termo simplesmente como um atrativo de mercado. A maioria dos sistemas continua não atendendo às necessidades de seus usuários que tem que lidar com interfaces que mais parecem inimigas. E um outro aspecto, é o de quão pouco adequado é esse termo: primeiro, é desnecessariamente antropomórfico, usuários não precisam de máquinas para serem amigas, eles precisam de máquinas que lhes facilitem na execução de suas tarefas; segundo, significa que as necessidades dos usuários podem ser descritas em apenas uma dimensão, mais ou menos amigável - diferentes usuários têm diferentes necessidades e o que é amigável para um pode ser muito tedioso para outro.

Por outro lado, pesquisadores estavam preocupados em como o uso de computadores pode efetivamente enriquecer o trabalho e a vida das pessoas. Em particular, eles estavam analisando as capacidades e limitações humanas, ou seja, estudando o lado humano da interação com sistemas computacionais. Isso implicava em procurar entender os processos psicológicos das pessoas quando interagem com computadores. Entretanto, com o desenvolvimento da área, em paralelo com avanços tecnológicos, tornou-se claro que outros aspectos ligados ao usuário e ao uso dos computadores precisavam ser incluídos: treinamento; práticas de trabalho; estrutura administrativa e organizacional; relações sociais; saúde; e todos os demais fatores importantes para o sucesso ou fracasso no uso de computadores.

O termo Interação Humano-Computador (IHC) foi adotado em meados dos anos 80 como um meio de descrever esse novo campo de estudo. E como já dissemos, o termo emerge da necessidade de mostrar que o foco de interesse é mais amplo que somente o design de interfaces e abrange todos os aspectos relacionados com a interação entre usuários e computadores.

Muito embora, ainda não exista uma definição estabelecida para IHC, acreditamos que a seguinte definição incorpora o espírito da área no momento: **IHC é a disciplina preocupada com o design, avaliação e implementação de sistemas computacionais interativos para uso humano e com o estudo dos principais fenômenos ao redor deles** A Figura 1.8 tenta expressar o conjunto de componentes contidos nessa definição.

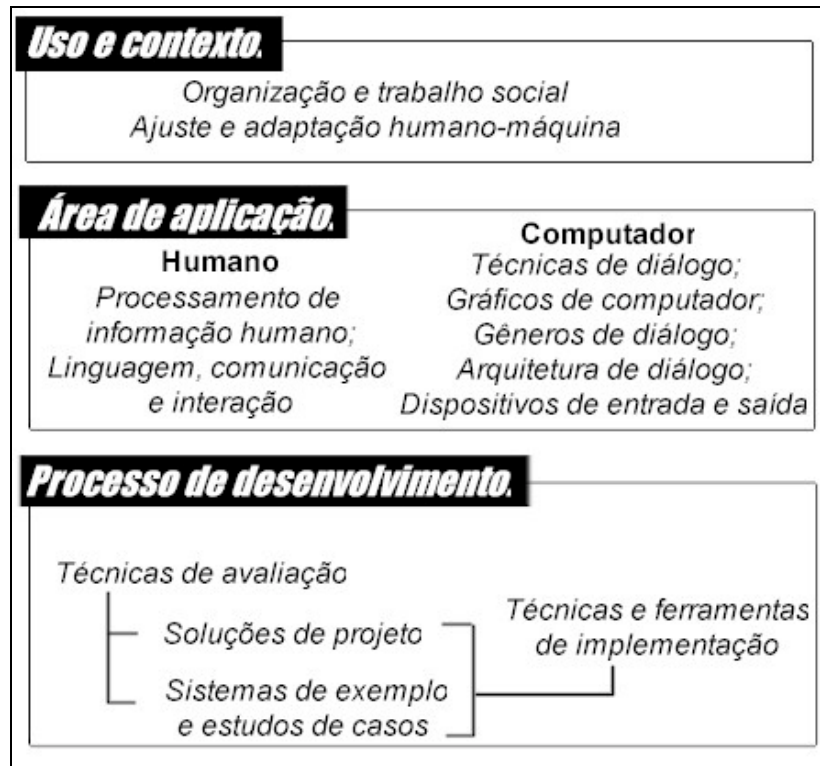


FIGURA 1.8 - INTERAÇÃO HUMANO-COMPUTADOR ADAPTADA DA DESCRIÇÃO DO COMITÊ SIGCHI 1992

Concluindo, IHC trata do design de sistemas computacionais que auxiliem as pessoas de forma a que possam executar suas atividades produtivamente e com segurança. IHC tem, portanto, papel no desenvolvimento de todo tipo de sistema, variando dos sistemas de controle de tráfego aéreo onde segurança é extremamente importante, até sistemas de escritório onde produtividade e satisfação são os parâmetros mais relevantes, até jogos, onde o envolvimento dos usuários é o requisito básico.

DESAFIOS DE IHC

Dado o rápido desenvolvimento da tecnologia, mais os conflitos e compromissos dos objetivos de um design e mais as diferentes componentes (e áreas de estudo) que caracterizam IHC, sem dúvida alguma ela é uma área com ricos desafios.

O desenvolvimento de máquinas mais rápidas e com maior poder de processamento, em conjunto com melhorias de tecnologias de hardware e software não pára, e abre inúmeras possibilidades para IHC. Dispositivos especiais possibilitam ao usuário "pegar" objetos dentro de um espaço virtual, e mesmo movimentar-se através de um espaço de realidade virtual. Aplicações multimídia, onde som, gráficos estáticos e dinâmicos, vídeo e texto são interligados são comuns hoje em dia. Desenvolvimentos recentes em telecomunicações têm possibilitado que grandes quantidades de diferentes tipos de informação possam ser enviadas através de redes. Imagens, vídeo, som e texto podem ser transmitidos com perda mínima de eficiência e qualidade. Informações de bancos de dados existentes em todo o mundo podem ser obtidas pelas pessoas de suas próprias casas. Essas mudanças trazem dois importantes desafios aos designers de IHC (Preece *et al*, 1994):

- Como dar conta da rápida evolução tecnológica?
- Como garantir que os design ofereçam uma boa IHC ao mesmo tempo que exploram o potencial e funcionalidade da nova tecnologia?

Um exemplo clássico desses problemas são os aparelhos de vídeo cassete. Enquanto a maioria das pessoas não tem problema algum em colocar uma fita, iniciar uma gravação ou dar um *play*, adiantar ou atrasar a fita, elas freqüentemente não acham assim tão fácil acertar o *timer* de forma a gravar um programa em um tempo futuro. Para a maioria dos gravadores de vídeo não é óbvio, a partir da interface entre a pessoa e a máquina, como a informação deve ser especificada para o sistema, e muito menos a resposta do sistema (quando existe) é clara. Certamente, se descobre que as coisas não funcionaram bem quando já é muito tarde. Tentando minorar essa dificuldade, muitos aparelhos de vídeo atualmente provêem um display das funções no televisor, mas mesmo assim as dificuldades permanecem (minoradas sem dúvida alguma).

O mesmo acontece com relação aos atuais aparelhos telefônicos. Enquanto as funcionalidades estavam restritas ao suporte de uma conversação tudo ia muito bem. Mas atualmente, a tecnologia permite conversas entre mais que duas pessoas; o sinal de que não está ocupado já não significa mais isso, pois os telefones tem *bip* que permite a interrupção, sem desligar, de uma conversa para atender outra; podemos transferir nossas ligações para outro número; etc. E aí, a mesma interface para dar conta de todas essas novas funcionalidades ficou complexa e não é mais óbvia. As pessoas têm problemas quando tentam operar essas funções e muitas desistem.

Não existe como negar que muitos sistemas computacionais foram projetados com interfaces extremamente pobres. O ponto que precisa ser entendido é que aumentar a funcionalidade não pode ser uma desculpa para um design pobre. Deve ser possível projetar boas interfaces cujos controles têm operações e efeitos relativamente óbvios e que também provêem um feedback imediato e útil.

Um bom exemplo é o dado por Norman (1988) com relação aos carros. Observando os controles dos painéis dos carros atuais podemos ver que eles têm cerca de 100 controles ou mais - dez ou mais para o equipamento de som, 5 ou mais para o sistema de ventilação, outros tantos para as janelas, limpadores de pára-brisa, luzes, para abrir e fechar portas, para dirigir o carro, etc. A maioria das pessoas, com pouca tentativa e erro (quase sempre enquanto dirige) ou após uma rápida olhada no manual, tem poucos problemas em lidar com todo o domínio de funções. Por que isso acontece, se não existe termo de comparação entre o número de funções e controles de um carro e de um gravador de vídeo? O que torna a interface do carro tão boa e a do vídeo tão pobre? Uma das razões é que o feedback nos carros é imediato e óbvio. Também, as pessoas que já dirigiram qualquer carro sabem o que esperar pois, muito embora, os carros sejam diferentes, a posição da maioria dos controles é a mesma ou similar, e símbolos similares são usados para indicar suas funções.

Portanto, os desafios de IHC são evidentes e a procura de soluções estabelece os objetivos da área que ao serem centrados no humano e não na tecnologia são sempre atuais.

OBJETIVOS DE IHC

Os objetivos de IHC são o de produzir sistemas usáveis, seguros e funcionais. Esses objetivos podem ser resumidos como desenvolver ou melhorar a segurança, utilidade, efetividade e usabilidade de sistemas que incluem computadores. Nesse contexto o termo sistemas se refere não somente ao hardware e o software mas a todo o ambiente que usa ou é afetado pelo uso da tecnologia computacional.

Nielsen (1993) engloba esses objetivos em um conceito mais amplo que ele denomina **aceitabilidade de um sistema** (Figura 1.9).

A **aceitabilidade geral** de um sistema é a combinação de sua aceitabilidade social e sua aceitabilidade prática. Como um exemplo de **aceitabilidade social**, podemos mencionar os sistemas atuais de controle das portas de entrada em bancos. Apesar de serem benéficos socialmente pois tentam impedir situações de assalto onde os usuários dos bancos ficam em sério risco, não são aceitos socialmente pois levam a que qualquer pessoa que queira entrar no banco tenha que esbarrar na porta trancada por inúmeras vezes até se desfazer de todo e qualquer objeto suspeito (o problema é que não se sabe quais os objetos que impedem a entrada).

A **aceitabilidade prática** trata dos tradicionais parâmetros de custo, confiabilidade, compatibilidade com sistemas existentes, etc., como também da categoria denominada "*usefulness*"



FIGURA 1.9 - ATRIBUTOS DE ACEITABILIDADE DE SISTEMAS (ADAPTADO DE NIELSEN, 1993)

"*Usefulness*" refere-se ao sistema poder ser usado para atingir um determinado objetivo. Novamente essa categoria é uma combinação de duas outras: utilidade e usabilidade. **Utilidade** deve verificar se a funcionalidade do sistema faz o que deve ser feito, ou seja, se um jogo efetivamente diverte e um software educacional auxilia o aprendizado. **Usabilidade** é a questão relacionada a quão bem os usuários podem usar a funcionalidade definida e este é um conceito chave em IHC, que trataremos mais detalhadamente ainda neste capítulo.

Portanto, a aceitabilidade de um sistema tem muitos componentes (daí a complexidade da tarefa), e IHC tem, de certa forma, que atender aos compromissos de todas essas categorias. Mas, como temos afirmado e reafirmado, a pesquisa de IHC é fundada na crença de que o centro e ponto básico de análise são as pessoas usando um sistema computacional. Suas necessidades, capacidades e preferências para executar diversas tarefas devem informar os meios como os sistemas devem ser projetados e implementados. As pessoas não devem ter que mudar radicalmente para se adequar ao sistema, o sistema sim deve ser projetado para se adequar aos seus requisitos.

A MULTI(INTER)(TRANS)DISCIPLINARIDADE EM IHC

Estabelecidos os objetivos de IHC tem-se a parte mais difícil que é a de que forma conseguir estes objetivos. Isso envolve uma perspectiva multidisciplinar, ou seja, resolver os problemas de IHC analisando diferentes perspectivas em seus multifacetados fatores: segurança, eficiência e produtividade, aspectos sociais e organizacionais, etc.

Um resumo dos principais fatores que devem ser levados em conta pode ser visto na Tabela 1.2 (Preece, 1994). Primeiramente, tem-se os fatores relacionados com o usuário como o conforto, saúde, ambiente de trabalho ou ergonomia do equipamento

a ser utilizado. Analisar esses fatores é tarefa bastante complexa pois eles não são independentes, interagem fortemente uns com os outros.

Outro ponto que em muito aumenta a complexidade da análise dos fatores ligados ao usuário, é que eles não são homogêneos em termos de requisitos e características pessoais. Humanos compartilham muitas características físicas e psicológicas, mas são bastante heterogêneos em termos de qualidades como habilidades cognitivas e motivação. Essas diferenças individuais têm importância fundamental no design da interface de um sistema computacional.

FATORES ORGANIZACIONAIS TREINAMENTO, POLÍTICAS, ORGANIZAÇÃO DO TRABALHO, ETC.		FATORES AMBIENTAIS BARULHO, AQUECIMENTO, VENTILAÇÃO, LUMINOSIDADE, ETC.	
SAÚDE E SEGURANÇA estresse, dores de cabeça, perturbações musculares, etc.	capacidades e processos cognitivos O USUÁRIO motivação, satisfação, personalidade, experiência, etc.		CONFORTO posição física, layout do equipamento, etc.
INTERFACE DO USUÁRIO dispositivos de entrada e saída, estrutura do diálogo, uso de cores, ícones, comandos, gráficos, linguagem natural, 3-D, materiais de suporte ao usuário, multimídia, etc.			
TAREFA fácil, complexa, nova, alocação de tarefas, repetitiva, monitoramento, habilidades, componentes, etc.			
RESTRIÇÕES custos, orçamentos, equipe, equipamento, estrutura do local de trabalho, etc.			
FUNCIONALIDADE DO SISTEMA hardware, software, aplicação			
PRODUTIVIDADE aumento da qualidade, diminuição de custos, diminuição de erros, diminuição de trabalho, diminuição do tempo de produção, aumento da criatividade , oportunidades para idéias criativas em direção a novos produtos, etc.			

TABELA 1.2 - FATORES EM IHC (ADAPTADO DE PREECE, 1994, p.31)

Voltemos ao exemplo dos carros, considerando agora os bancos. Se todos tivessem o mesmo formato, nenhuma dificuldade haveria para o designer projetar sempre o banco ideal. Comparado com diferenças psicológicas, as diferenças físicas podem ser

consideradas triviais de lidar. Um modo de tratar essa diversidade é projetar sistemas flexíveis que possam ser "customizados" de forma a se adequar às necessidades individuais. Isto, de certa forma, está sendo feito no design dos bancos nos carros atuais, onde um bom número de modos para ajuste são disponíveis. Também, sistemas computacionais, como editores de texto por exemplo, oferecem atualmente uma série de opções para se adequar à experiência e preferência de usuários.

Portanto, na análise dos fatores humanos envolvidos em IHC diversas disciplinas são necessárias (Figura 1.10).



FIGURA 1.10 - DISCIPLINAS QUE CONTRIBUEM EM IHC (ADAPTADO DE PREECE, 1994, P. 38)

Temos as principais: Psicologia Cognitiva, Psicologia Social e Organizacional, Ergonomia (termo europeu) ou Fatores Humanos (termo americano) e a Ciência da Computação. Outras áreas de estudo que tem tido uma crescente influência em IHC incluem: Inteligência Artificial, Linguística, Psicologia, Filosofia, Sociologia, Antropologia, Engenharia e Design.

CIÊNCIA DA COMPUTAÇÃO contribui provendo conhecimento sobre as possibilidades da tecnologia e oferecendo idéias sobre como explorar todo o seu potencial. Também os profissionais de computação têm se preocupado em desenvolver ferramentas de software auxiliares ao design, implementação e manutenção de sistemas: linguagens de programação, ferramentas de prototipação, sistemas de gerenciamento de interfaces de usuário (UIMS), ambientes de design de interfaces de usuário (UIDE), ferramentas de *debugging* e teste, etc. Alguns esforços têm sido feitos no sentido de prover métodos rigorosos de analisar a forma como IHC é projetada e incorporada em sistemas, que incluem arquiteturas de sistemas, abstrações e notações. Conceitos de reuso e de engenharia reversa também são utilizados em IHC. Em particular, tem havido a preocupação de prover meios para que designers iniciantes possam reusar trabalhos de colegas mais experientes, como bibliotecas de código, por exemplo. Os sofisticados sistemas gráficos usados em visualização e em realidade virtual também são resultados da ciência da computação.

PSICOLOGIA COGNITIVA. A preocupação principal da Psicologia é entender o comportamento humano e os processos mentais subjacentes. A Psicologia Cognitiva adotou a noção de processamento de informação como modelo para o comportamento humano e tenta colocar tudo que vemos, sentimos, tocamos, cheiramos, etc. , em termos desse modelo. Como poderá ser visto no Capítulo 2, importantes tópicos de IHC são o estudo da percepção, atenção, memória, aprendizagem, solução de problemas, etc. O objetivo da Psicologia Cognitiva tem sido o de caracterizar esses processos em termos de suas capacidades e limitações. Por exemplo, uma das principais preocupações da área nos anos 60 e 70 era identificar a quantidade de informação que podia ser processada e lembrada de uma só vez. Recentemente, existe a preocupação em caracterizar o modo como as pessoas trabalham entre si e com vários artefatos, entre eles o computador. Um dos principais resultados desses estudos é a cognição distribuída. Psicólogos cognitivistas têm se preocupado em aplicar princípios psicológicos em IHC usando uma variedade de métodos: desenvolvimento de *guidelines*, uso de modelos para prever o desempenho humano no uso de computadores, métodos empíricos para testar sistemas computacionais, etc.

PSICOLOGIA SOCIAL tem como preocupação estudar a natureza e causas do comportamento humano no contexto social. Pode-se resumir as preocupações básicas da Psicologia Social em quatro pontos (Vaske e Grantam, 1990):

- a influência de um indivíduo nas atitudes e comportamentos de outra pessoa
- impacto de um grupo sobre o comportamento e as atitudes de seus membros
- impacto de um membro nas atividades e estrutura de um grupo
- relacionamento entre estrutura e atividades de diferentes grupos

E a tecnologia desempenha um papel importante em todos esses aspectos.

A PSICOLOGIA ORGANIZACIONAL dá aos designers o conhecimento sobre estruturas organizacionais e sociais e sobre como a introdução de computadores influencia práticas de trabalho. Em grandes organizações, por exemplo, o computador serve tanto como meio de comunicação, quanto para fazer a folha de pagamento e contabilidade em geral, para controlar entrada e saída de pessoas, etc. Isso envolve entender a estrutura e funcionamento de organizações em termos de autoridade e poder, tamanho e complexidade, eficiência, fluxo de informação, tecnologia, práticas de trabalho, ambiente de trabalho e contexto social. Modelos de mudanças organizacionais com a inclusão da tecnologia são bastante úteis a esse entendimento.

FATORES HUMANOS, OU ERGONOMIA, teve um grande desenvolvimento a partir da segunda grande guerra, atendendo a demanda de diversas disciplinas. Seu objetivo é conceber e fazer o design de diversas ferramentas e artefatos para diferentes ambientes de trabalho, domésticos e de diversão, adequados às capacidades e necessidades de usuários. O objetivo é maximizar a segurança, eficiência e confiabilidade da performance do usuário, tornando as tarefas mais fáceis e aumentando os sentimentos de conforto e satisfação. As primeiras contribuições dos

especialistas em fatores humanos para IHC foram no design do hardware (teclados mais ergonômicos, posições do vídeo, etc.) e nos aspectos de software que poderiam resultar em efeitos fisiológicos adversos nos humanos, como a forma da apresentação de informação na tela do vídeo.

LINGÜÍSTICA é o estudo científico da linguagem (Lyons, 1970). Muita atenção tem sido dada atualmente aos resultados da lingüística como fontes de conhecimento importantes para IHC. O uso imediato e mais tradicional é o de explorar a estrutura da linguagem natural na concepção de interfaces, principalmente para facilitar o acesso e consulta a bases de dados. Também na concepção de linguagens de programação mais fáceis de serem aprendidas resultados da lingüística estão presentes (por exemplo, na linguagem de programação Logo, voltada para a Educação). Estudos derivados, que consideram o estudo da linguagem enquanto forma de comunicação, não apenas textual, têm tido muita relevância hoje em dia em IHC (Semiótica e Engenharia Semiótica, que serão tratadas no Capítulo 3, são um exemplo). Também na internacionalização de interfaces e localização de software a lingüística tem tido um papel cada vez mais importante. Internacionalização é a preocupação em isolar os fatores culturais de um produto (por exemplo, textos, ícones, datas etc.) de outros que podem ser considerados genéricos culturalmente. Localização é exatamente o processo de colocar os aspectos culturais em um produto previamente internacionalizado (Russo e Boor, 1993).

INTELIGÊNCIA ARTIFICIAL (IA) é um ramo da ciência da computação cujo objetivo é desenvolver sistemas computacionais que exibam características que nós associamos com inteligência no comportamento humano. A preocupação central é com o desenvolvimento de estruturas de representação do conhecimento que são utilizadas pelo ser humano no processo de solução de problemas. Métodos e técnicas de IA, tais como o uso de regras de produção, têm sido usados por IHC no desenvolvimento de sistemas especialistas e tutores com interfaces inteligentes. IA também se relaciona com IHC no processo de interação dos usuários com interfaces inteligentes no sentido do uso de linguagem natural (textual e falada), na necessidade do sistema ter que justificar uma recomendação, nos sistemas de ajuda contextualizados e que efetivamente atendam às necessidades dos usuários, etc. Atualmente, grande ênfase tem sido dada no desenvolvimento de agentes de interfaces inteligentes, que auxiliam os usuários na navegação, busca de informação, organização da informação, etc. O objetivo no uso desses agentes é o de reduzir a sobrecarga cognitiva que muitos usuários têm atualmente ao lidar com a quantidade de informação apresentada, na maioria das vezes, de forma hipertextual.

FILOSOFIA, SOCIOLOGIA E ANTROPOLOGIA são, das disciplinas que contribuem com IHC, as tradicionalmente denominadas *soft sciences* (Preece, 1994). Com isso, se está querendo dizer que elas não estão diretamente envolvidas com o design real de um sistema computacional do mesmo modo que as *hard sciences* que oferecem métodos, técnicas e implementações. Elas estão mais diretamente envolvidas com os

desenvolvimentos da tecnologia de informação e com a transferência de tecnologia. Isso na verdade vem mudando, pois atualmente métodos da Sociologia e Antropologia têm sido aplicados no design e avaliação de sistemas. Uma dessas técnicas é a etno-metodologia onde a premissa básica é não assumir um modelo *a priori* do que vai acontecer quando as pessoas usam o computador, ao invés disso, analisar o comportamento na observação do que acontece durante o uso em seu contexto real de uso. Portanto, a ênfase é em entender o que acontece quando as pessoas se comunicam entre si ou com as máquinas, enquanto e depois que isso acontece, e não modelar e prever de antemão como o faz a Psicologia Cognitiva. A razão da aplicação desses métodos na análise de IHC é a de que uma descrição mais precisa da interação entre usuários, seu trabalho, a tecnologia em uso e no ambiente real de uso precisa ser obtida. Trabalho cooperativo auxiliado pelo computador (CSCW) o qual objetiva prover ferramentas de software que possibilitem a execução cooperativa (compartilhando software e hardware) de tarefas é uma área de aplicação e desenvolvimento que depende diretamente do resultado dessa descrição.

ENGENHARIA é uma ciência aplicada direcionada à construção e testes empíricos de modelos. Basicamente, a Engenharia usa os resultados da ciência em geral na produção de artefatos. Na maioria dos aspectos, a grande influência da engenharia em IHC tem sido via Engenharia de Software.

DESIGN tem oferecido a IHC conhecimento mais que evidente, como por exemplo na área de design gráfico. Muitos autores, afirmam que o envolvimento e o crescente interesse de designers gráficos no projeto de telas de sistemas computacionais consolidou IHC como uma área de estudo. O processo de design de IHC também foi influenciado pela prática de design gráfico. Por exemplo, a prática de gerar diversas alternativas para serem avaliadas em sessões de *brainstorming* entre colegas logo no início de um projeto tem sido adotada atualmente por um grande número de designers de interfaces. Com o advento da WEB, como uma nova mídia de comunicação, dificilmente se tem uma equipe de desenvolvimento que não tenha um designer gráfico.

A contribuição dessas disciplinas em IHC certamente é uma via de duas mãos, ou seja, certamente IHC alterou também a prática em cada uma dessas disciplinas. Por exemplo, hoje em dia não existe engenheiro ou designer que não faça uso de ferramentas de design tanto para produzir projetos arquitetônicos ou mecânicos quanto para fazer o projeto de uma nova cadeira. E de modo mais geral, em todas essas áreas, ferramentas de visualização, busca, compilação, análise de informação têm sido geradas e amplamente utilizadas.

PRINCÍPIOS DE DESIGN

Se pensamos na complexidade da maioria dos sistemas computacionais, vemos que o potencial de se ter uma precária IHC é bastante alto. Daí alguns autores definirem princípios básicos que ajudem a garantir uma boa IHC.

PARTINDO DOS OBJETOS QUE NOS CERCAM

Norman (1988), partindo da experiência de observar e vivenciar as frustrações que as pessoas experimentam com objetos do cotidiano que não conseguem saber como usar, com embalagens que parecem impossíveis de serem abertas, com portas que mais parecem uma armadilha, com máquinas de lavar e secadoras que têm se tornado cada vez mais poderosas e confusas, identifica alguns princípios básicos de um bom design, que segundo ele, constituem uma forma de Psicologia - a Psicologia de como as pessoas interagem com objetos. Os quatro princípios (visibilidade e *affordances*; bom modelo conceitual; bons mapeamentos e feedback), como veremos a seguir, são altamente interrelacionados e difíceis de serem tratados e estudados isoladamente.

- **Visibilidade e *Affordances***

O usuário necessita ajuda. Apenas as coisas necessárias têm que estar visíveis: para indicar quais as partes podem ser operadas e como, para indicar como o usuário interage com um dispositivo. Visibilidade indica o mapeamento entre ações pretendidas e as ações reais. Indica também distinções importantes - por exemplo, diferenciar a vasilha do sal da do açúcar.

A visibilidade do efeito das operações indica se a operação foi feita como pretendida, como por exemplo, se as luzes foram acesas corretamente, se a temperatura de um forno foi ajustada corretamente, etc. A falta de visibilidade é que torna muitos dispositivos controlados por computadores tão difíceis de serem operados.

A título de ilustração, vamos considerar um exemplo muito simples já mencionado anteriormente, o das portas. Quantos de nós já experimentamos frustrações no uso de portas cuja funcionalidade é extremamente simples - abrir e fechar, e nada mais. Muitas vezes empurramos portas que deveríamos puxar, abrimos para a direita quando deveria ser para a esquerda, empurramos portas que deveriam ser deslizadas em alguma direção, que também muitas vezes não descobrimos qual é. As partes corretas deveriam estar visíveis. Designers deveriam prover sinais que claramente indicassem que uma porta deveria ser empurrada, simplesmente colocando a barra de empurrar em um dos lados da porta e nada no outro. Os pilares de suporte deveriam estar visíveis e tudo isso

sem nenhum prejuízo da estética tão largamente procurada. A barra horizontal para empurrar e os pilares são sinais naturais e portanto naturalmente interpretados, conduzindo ao que se denomina de design natural (Norman, 1988).

Outro exemplo clássico de falta de visibilidade, que também já mencionamos, é o dos modernos telefones, com múltiplas funções e com uma interface de uso que não as deixa visíveis. E um exemplo favorável é o dos carros, que oferecem uma boa visibilidade na maioria de suas inúmeras funções.

A mente humana é extraordinária no processo de dar sentido ao mundo. Considere os objetos - livros, rádios, eletrodomésticos, máquinas de escritório, etc. - que fazem parte de nossa vida. Objetos com um bom design são fáceis de interpretar e entender. Eles contêm "dicas" visíveis de sua operação. Objetos com design pobre são difíceis e frustrantes de usar. Eles não provêm indicações ou o que é muito pior, provêm, muitas vezes, falsas "dicas".

Affordance é o termo definido para se referir às propriedades percebidas e propriedades reais de um objeto, que deveriam determinar como ele pode ser usado. Uma cadeira é para sentar e também pode ser carregada. Vidro é para dar transparência, e aparenta fragilidade. Madeira dá solidez, opacidade, suporte, e possibilidade de escavar. Botões são para girar, teclas para pressionar, tesouras para cortar, etc. Quando se tem a predominância da *affordance* o usuário sabe o que fazer somente olhando, não sendo preciso figuras, rótulos ou instruções. Objetos complexos podem requerer explicações, mas objetos simples não. Quando estes necessitam rótulos ou instruções é porque o design não está bom.

- **Bom modelo conceitual**

Um bom modelo conceitual permite prever o efeito de ações. Sem um bom modelo conceitual opera-se sob comando, cegamente. Efetua-se as operações receitadas, sem saber que efeitos esperar ou, o que fazer se as coisas não derem certo. Conforme as coisas vão dando certo, aprende-se a operar. Agora, quando as coisas dão errado ou quando se depara com situações novas necessita-se de um maior entendimento, de um bom modelo.

Consideremos o exemplo de uma tesoura. Mesmo que nunca tenhamos visto uma anteriormente, é claro o seu limitado número de funções possíveis. Os buracos deixam claro que algo deve ser colocado neles, e a única coisa lógica de se colocar e que pode encaixar são os dedos. Os buracos têm *affordances*, que possibilitam os dedos serem inseridos. O tamanho dos buracos provêm restrições que limitam quais dedos podem ser usados: o buraco maior sugere diversos dedos e o menor apenas um. O mapeamento entre os buracos e os dedos é então sugerido e restringido pelos buracos. Entretanto, a operação não é limitada à colocação dos dedos corretos. A tesoura irá funcionar com qualquer

dedo. Consegue-se entender a tesoura e seu funcionamento porque suas partes são visíveis e as implicações claras. O **modelo conceitual** é portanto claro, e até óbvio, e existe um efetivo uso de *affordances*.

Um contra exemplo, é o de um relógio digital simples, com dois ou até quatro botões no mostrador. Para que servem esses botões? Como descobrir se são de puxar e empurrar e não girar? Como acertar a hora, a data? Não existe um relacionamento evidente entre os controles e suas funções, nenhum mapeamento aparente.

Para objetos do dia a dia, modelos conceituais podem ser bastante simples, mas quando consideramos, a complexidade de sistemas computacionais a relevância de um bom modelo é mais que óbvia.

- **Bons mapeamentos**

Mapeamento é o termo técnico para denotar o relacionamento entre duas entidades. No caso de interfaces, indica o relacionamento entre os controles e seus movimentos e os resultados no mundo. Vamos novamente recorrer ao exemplo dos carros e os mapeamentos envolvidos em dirigir um carro. Quando queremos ir para a direita, devemos virar o volante também para a direita (sentido horário). O usuário identifica dois mapeamentos: o controle que afeta a direção e que o volante precisa ser virado em uma de duas direções. Ambos são arbitrários, mas a roda e o sentido horário são escolhas naturais: visíveis, muito relacionada ao resultado esperado, e provêem um feedback imediato. O mapeamento é facilmente aprendido e sempre lembrado.

Mapeamentos naturais, aqueles que aproveitam analogias físicas e padrões culturais, levam ao entendimento imediato. Por exemplo, é comum designers utilizarem analogias especiais: para mover um objeto para cima, move-se o controle também para cima (Norman, 1988).

Problemas de mapeamento são muitos e uma das principais causas das dificuldades que os usuários encontram no uso de objetos. Também retomando, consideremos os telefones. Suponha que se deseja redirecionar as ligações de um número para outro. As instruções são geralmente do seguinte tipo: tecle #, em seguida o número 9 e finalmente o número para o qual deseja desviar suas chamadas. Uma descrição incompleta de um procedimento arbitrário. O que acontece se eu errar no meio do caminho o que faço? Por que #? Por que 9? E a ausência de feedback é outro aspecto. Como sei se deu certo ou não?

Um objeto é fácil de ser usado quando existe um conjunto visível de ações possíveis, e os controles exploram mapeamentos naturais. O princípio é simples mas raramente incorporado aos design.

- **Feedback**

Retornar ao usuário informação sobre as ações que foram feitas, quais os resultados obtidos, é um conceito conhecido da teoria da informação e controle. Imagine falar com uma pessoa sem ouvir sua própria voz (a famosa ausência de "retorno" que os músicos tanto reclamam nos palcos), ou desenhar com um lápis que não risca, ou seja, sem nenhum feedback. Acreditamos não ser necessário retomar os exemplos clássicos já amplamente discutidos: carros (bom feedback) e telefones (nenhum feedback para algumas funções mais avançadas). Exemplificando com interfaces computacionais, quantos de nós mandamos imprimir documentos em impressoras de rede e sempre nos deparamos com problemas clássicos: qual foi mesmo a impressora? a impressão já terminou ou não? o documento foi mesmo impresso ou houve algum problema? acabou o papel da impressora (que está em outra sala muitas vezes bem distante) antes do término da impressão?

Um aspecto interessante e que pode ser depreendido desses exemplos que mencionamos é o que Norman (1988) chama de **paradoxo da tecnologia**. A tecnologia oferece potencial para tornar nossa vida mais simples e agradável, e cada nova tecnologia traz mais benefícios. E ao mesmo tempo adiciona tamanha complexidade que faz aumentar nossa dificuldade e frustração. O relógio é um bom exemplo disso. Ninguém tinha problemas com os clássicos relógios cuja única função era marcar as horas e com um único botão conseguimos fazer todos os ajustes necessários. Os relógios digitais atuais ampliaram em muito as funcionalidades básicas: eles marcam data, dão alarme, são cronômetros, mostram hora no mundo todo e os mais modernos mostram inclusive o horário mundial da WEB. Mas adicionar todas essas funções causa problemas. Como fazer o design de um relógio com tantas funções e ao mesmo tempo limitar seu tamanho, custo e complexidade de uso? Quantos botões deveriam ter os relógios para torná-los fáceis de aprender e operar? Não existem respostas simples. Sempre que o número de funções excede o número de controles, o design torna-se arbitrário e não natural, e complicado. A mesma tecnologia que simplifica a vida provendo um maior número de funcionalidades em um objeto, também a complica tornando muito mais difícil aprender, e usar. Esse é o paradoxo da tecnologia e o grande desafio dos designers é minimizar esses efeitos.

USABILIDADE DE SISTEMAS COMPUTACIONAIS

Nielsen (1993) já mais direcionado para sistemas computacionais explora o design e propõe princípios que levem a um aumento da usabilidade, que como vimos é um dos critérios que definem a aceitabilidade de um sistema (Figura 1.9). Nielsen explicita seus princípios de design a partir de alguns *slogans*, que ele define como *slogans de usabilidade*. A seguir apresentamos alguns deles:

- **Sua melhor tentativa não é boa o suficiente**

É impossível fazer o design de uma interface ótima simplesmente baseado em nossas melhores idéias. Usuários tem um potencial infinito para mal interpretar elementos de interface e para fazer suas tarefas de modo diferente do que imaginamos. Portanto, o design é sempre melhor se trabalhamos baseados no entendimento do usuário e de

suas tarefas. Sempre temos que nos preocupar em validar ou avaliar nossos design usando as diferentes formas de avaliação existentes (o Capítulo 4 trata especificamente de avaliação de interfaces) e estar abertos para efetuar um redesign a partir dos resultados dessas avaliações.

- **Usuário está sempre certo**

A atitude do designer quando verifica que o usuário tem problemas de interação com um determinado aspecto da interface, não deve ser a de julgar que o usuário é ignorante ou então, que ele não tentou o suficiente ou ainda, deixar passar que um dia o usuário aprende. Por exemplo, quando escrevemos um determinado procedimento de operação e verificamos que o usuário sempre erra em uma determinada parte do procedimento, certamente a solução não é colocar um aviso em destaque do tipo LEIA CUIDADOSAMENTE ESSE TRECHO. O que deve ser feito é aceitar que o texto está mal escrito e que precisa ser reformulado. Portanto, o designer de interfaces deve adquirir uma certa humildade e aceitar a necessidade de modificar uma "grande idéia" de forma a resolver problemas dos usuários.

- **Usuário não está sempre certo**

Também não se deve ir ao extremo de construir uma interface somente a partir do que os usuários gostariam. Usuários freqüentemente não sabem o que é bom para eles. Qualquer um de nós teria dificuldades em prever como gostaríamos de interagir com um sistema em potencial com o qual não temos nenhuma experiência. Temos a tendência de rejeitar *a priori* qualquer grande inovação em objetos com os quais estamos familiarizados e que atendem satisfatoriamente nossas necessidades.

- **Usuários não são designers**

Uma solução simples para atender a diversidade de usuários seria a de prover interfaces flexíveis que pudessem ser amplamente customizadas e aí cada usuário teria exatamente a interface que melhor lhe satisfizesse (análogo aos bancos de carros modernos mencionados anteriormente). Estudos demonstram que usuários novatos não customizam suas interfaces, mesmo quando essas facilidades estão disponíveis (Jorgensen e Sauer, 1990). Mas existem alguns outros bons motivos para não se dar à customização uma importância indevida: primeiro, customização é fácil somente se puder produzir um design coerente a partir do conjunto de opções disponíveis; segundo, o processo de customização

também vai exigir uma interface e portanto adiciona complexidade; terceiro, muita customização leva a que cada usuário tenha uma interface muito diferente de outro usuário (isso dificulta, por exemplo, o pedido a ajuda entre colegas que é um dos principais métodos de *help* usado tanto por novatos como por especialistas (Sellen e Nicol, 1990)); e quarto, usuários nem sempre adotam as decisões de design mais apropriadas.

- **Designers não são usuários**

Designers são humanos e certamente usam computadores, mas são diferentes de usuários em diversos aspectos básicos: a experiência computacional e o conhecimento dos fundamentos conceituais do design do sistema. Conseqüentemente o designer olha uma determinada tela ou uma determinada mensagem e acredita que são perfeitamente claras e adequadas, mesmo que sejam incompreensíveis para quem não conhece o sistema. Conhecer sobre um sistema é uma via de mão única, impossível voltar e fazer o papel de um novato.

- **Menos é mais (*less is more*)**

Uma das frequentes soluções de design que têm sido adotadas é colocar no sistema todas as opções e características imagináveis, pois se tudo está disponível então todos ficarão satisfeitos. Essa tendência é verificada nos softwares, como editores de texto por exemplo, que a cada nova versão tem dobrado de tamanho gerando o fenômeno denominado '*fatware*' (Perratore *et al*, 1993). Cada elemento em uma interface acarreta uma sobrecarga ao usuário que tem que considerar se o usa ou não. Ter poucas opções, as necessárias à tarefa, geralmente significa uma melhor usabilidade, pois o usuário pode se concentrar em entender essas poucas opções.

- **Help não ajuda (*help doesn't*)**

Muitas vezes, senão na maioria delas, vemos usuários perdidos tentando encontrar informação na enorme quantidade de material de *help* que acompanha um sistema, e quando a encontra não consegue entendê-la. Também, a existência de *helps* acrescenta mais complexidade à interface e na maioria das vezes sem grande efetividade. Em qualquer situação, deve-se ter claro que a existência de um *help* não pode ser usada como desculpa para um design ruim. Sempre é melhor o usuário poder operar um sistema sem ter que usar um *help* e o design deve usar isso como um requisito básico.

Usabilidade é definida em função de múltiplos componentes e é tradicionalmente associada com cinco atributos de usabilidade (Nielsen, 1993):

- **Facilidade de aprendizagem (*learnability*)**

O sistema precisa ser fácil de aprender de forma que o usuário possa rapidamente começar a interagir. Segundo Nielsen, é o mais importante atributo de usabilidade, por ser a primeira experiência que qualquer usuário tem com um

sistema. Certamente existem sistemas para aplicações altamente especializadas e complexas onde se prevê um extenso trabalho de treinamento em seu uso, mas na maioria dos casos um sistema deve ser fácil de aprender.

Quando se analisa a facilidade de aprendizagem, é preciso ter em mente que geralmente o usuário não aprende toda uma interface antes de começar a usá-la. Pelo contrário, o aprendizado ocorre do uso. Portanto, esse fator é avaliado em função do tempo que o usuário demora para atingir um suficiente grau de proficiência na execução de suas tarefas.

- **Eficiência**

O sistema precisa ser eficiente no uso, de forma que uma vez aprendido o usuário tenha um elevado nível de produtividade. Portanto, eficiência refere-se a usuários experientes, após um certo tempo de uso. Um modo típico de avaliar esse atributo é definir de alguma forma o que significa um usuário experiente e avaliar um grupo desses executando tarefas típicas de um sistema.

- **Facilidade de relembrar (*memorability*)**

O sistema precisa ser facilmente relembrado, de forma que o usuário ao voltar a usá-lo depois de um certo tempo não tenha novamente que aprendê-lo. Esse atributo tanto se refere a usuários casuais (que é uma categoria com um número grande de usuários na maioria dos sistemas) como para aqueles sistemas utilitários que são inerentemente usados em períodos específicos como os sistemas para confecção de relatórios de atividades trienais, de imposto de renda, etc. Certamente, aumentar a facilidade de aprendizagem também torna a interface mais fácil de ser relembrada, mas tipicamente usuários que retornam a um sistema são diferentes dos usuários principiantes.

Raramente se avalia esse item, mas é notável a preocupação com ele nas modernas interfaces gráficas onde tudo que for possível é visível. Usuários desses sistemas não precisam lembrar o que está disponível, pois o sistema sempre o relembra quando necessário.

- **Erros**

Neste contexto, erro é definido como uma ação que não leva ao resultado esperado, um "engano" portanto. O sistema precisa ter uma pequena taxa de erros, ou seja, o usuário não pode cometer muitos erros durante o seu uso e, em errando, deve ser fácil a recuperação, sem perda de trabalho. Erros catastróficos (o usuário perder seu trabalho, não perceber que errou, etc.) não podem ocorrer.

- **Satisfação subjetiva**

Os usuários devem gostar do sistema, ou seja, deve ser agradável de forma que o usuário fique satisfeito ao usá-lo. Esse atributo é muito relevante quando se

considera sistemas usados fora do ambiente de trabalho, tais como jogos, sistemas domésticos em geral, etc. Para esses sistemas o entretenimento e envolvimento são muitas vezes, valores mais importantes que velocidade de processamento (claro, quando esta não compromete o resultado).

Nielsen (1993) ressalta que satisfação subjetiva como atributo de usabilidade é diferente dos estudos que avaliam atitudes gerais das pessoas com relação a computadores. Esses estudos são efetuados no contexto da aceitabilidade social de computadores (LaLomia e Sidowski, 1991) e não como atributo de usabilidade. Entretanto, certamente os sentimentos que as pessoas têm com relação a computadores em muito afeta sua interação com um determinado sistema. Pouco se conhece sobre a relação entre atributos de um determinado sistema e a atitude geral de uma pessoa com relação a computadores. Um dado importante é que usuários que percebem que têm um alto grau de controle sobre os computadores têm atitudes positivas com relação a ele (Kay, 1989). E dar controle ao usuário não é das soluções atuais de design mais frequentes, muito pelo contrário.

Satisfação subjetiva pode ser medida simplesmente perguntando ao usuário sobre suas opiniões subjetivas. Para um único usuário o resultado desse questionamento é subjetivo, mas quando se considera muitos usuários, a média das respostas passa a ser uma medida objetiva. Isso é o que é feito na maioria dos estudos de usabilidade. Tipo de questionários usados e formas de avaliar satisfação subjetiva são tratados no Capítulo 4.

O que se pode depreender dos princípios de usabilidade é que eles tratam basicamente de dois aspectos: a tarefa e as características individuais dos usuários. Portanto, mais uma vez, conhecer o usuário é fundamental para se fazer o design de um sistema usável. Entender os principais modos de classificar usuários ajuda a fazer um bom design que atenda a maior diversidade desses.

Na análise do usuário sua experiência é um fator relevante, e essa experiência deve ser analisada em três dimensões: com relação ao uso do sistema, com relação ao uso de computadores em geral e com relação ao domínio da aplicação. Nielsen (1993) apresenta essa diferenciação em um gráfico que ele denomina de *cube do usuário* (Figura 1.11).

O importante é ressaltar que o uso do sistema altera, e tem como um de seus objetivos, a categoria do usuário (um novato não é um eterno novato) e isso tem importantes implicações no design. Algumas interfaces são e devem ser projetadas com ênfase apenas em novatos - sistemas de informação sobre museus ou quiosques em parques e exposições, sistemas que necessariamente são alterados anualmente, etc. - que são categorias de sistemas onde a facilidade de aprendizagem é o requisito mestre. Mas a maioria das interfaces é projetada tendo em vista tanto os expertos como os novatos e portanto precisa acomodar ambos os estilos (não esquecendo que

novatos são futuros especialistas). Frequentemente, interfaces que são boas para novatos também o são para especialistas, mas sempre é possível prover múltiplos estilos de interação, de tal forma que os usuários iniciem aprendendo um estilo mais fácil e depois migrem para outro mais eficiente.



FIGURA 1.11 - TRÊS DIMENSÕES DAS DIFERENTES EXPERIÊNCIAS DE USUÁRIOS (ADAPTADO DE NIELSEN, 1993, P.44)

Um modo típico de permitir essa evolução natural do aprendizado inicial para um uso mais eficiente é o uso de aceleradores na interface. Aceleradores são elementos de interface que permitem que usuários realizem tarefas frequentes de forma mais rápida. Exemplos são as teclas de função, abreviação de nome de comandos, uso de duplo-clique para ativar objetos, etc. Muitos sistemas também provêem dois conjuntos de menu, os menus curtos para novatos e os longos para usuários mais experientes. Isto permite ao sistema oferecer um maior conjunto de opções para os usuários mais avançados sem confundir os principiantes. Importante é estar ciente de que ter ambos os estilos de interação aumenta a complexidade da interface e pode vir a ser um problema. Portanto é importante, fazer o design da interface de tal modo que o principiante não esteja exposto ao modo especialista. Por exemplo, sistemas que possibilitam o uso de comandos em sua forma abreviada devem ser cuidadosos no sentido de prover também a forma extensa em mensagens de erro e *helps*, por exemplo.

A **experiência com computadores** também tem um impacto no design da interface. Usuários experientes em um amplo conjunto de aplicações têm mais idéia de que características procurar e de como o computador normalmente trata várias situações. Especificamente, usuários com experiência em programação são aptos a usar as macro-linguagens e outros meios mais complexos de combinar comandos em aplicações simples, não orientadas a programação, como editores de texto, por exemplo.

Finalmente, a terceira dimensão que é **a experiência no domínio da tarefa** tem importância fundamental. Interfaces projetadas para especialistas podem fazer uso de terminologia e jargão específico de uma área de especialidade. Usuários com pouca experiência terão que ter mais explicação sobre o que o sistema faz e sobre o que as diferentes opções significam, e a terminologia usada não pode ser tão abreviada e densa quanto a dirigida para especialistas. Por exemplo, consideremos um sistema auxiliar a escolha de investimentos financeiros. Caso o sistema seja dirigido a não especialistas muito cuidado deve ser tomado dada a complexidade do domínio e a especificidade de sua terminologia. Poucos de nós sabe diferenciar, por exemplo, uma ação ordinária de uma preferencial, no mercado de aplicação em ações.

Diferenças entre usuários tem outras dimensões além da experiência. Alguns fatores são fáceis de serem verificados, como a idade (Czaja, 1988) e sexo (Fowler e Murray, 1987). Outros são menos óbvios como diferenças em habilidades de raciocínio (Gomez *et al*, 1986), estilos de aprendizagem (Sein e Bostron, 1989), etc. Diferenças culturais também são extremamente relevantes e serão tratadas quando discutirmos interfaces internacionais.

Parece complexo, e realmente o é, obter ótimos graus de usabilidade em todos os atributos simultaneamente. Compromissos são inerentes ao processo de design. Por exemplo, o desejo de evitar erros catastróficos pode levar a se ter uma interface menos eficiente de usar, no estilo das interfaces que a cada ação solicita ao usuário a confirmação e a re-confirmação antes da ação ser executada. Então, é importante estabelecer os objetivos de usabilidade a serem atingidos, quais os atributos a serem priorizados e isso é definido pelo contexto específico ao qual é dirigido um projeto.

Em **aplicações de escritório, domésticas e de entretenimento** - processadores de texto, jogos, softwares educacionais, etc. - facilidade de aprendizagem, baixa taxa de erros e satisfação subjetiva são fundamentais e devem ser maximizados. A escolha das funcionalidades adequadas nesse tipo de sistemas é muito difícil, pois toda gama de usuários deve ser alvo e certamente é mais que desejável uma evolução agradável do nível principiante para o especialista.

Sistemas críticos - controle de tráfego aéreo, reatores nucleares, operações militares, etc. - são sistemas de alto custo onde espera-se alta confiabilidade e efetividade. Também são sistemas altamente complexos onde um treinamento é aceitável de forma a obter rapidamente ausência de erro, mesmo sob estresse. Satisfação subjetiva é menos valorizada pois os usuários são bem motivados. E como são sistemas de uso freqüente a memorização é naturalmente obtida. Eficiência portanto é o atributo base do design de sistemas dessa categoria.

Sistemas de uso comercial e industrial - banco, seguros, reserva aérea, aluguel de carros, gerenciamento de cartão de crédito, etc. - que são de uso amplo, exigem um elevado custo de treinamento. Então facilidade de aprendizagem é fundamental de

forma a se ter performance rápida para muitos a custos razoáveis. Satisfação subjetiva tem importância modesta pois os usuários são motivados ao uso e a memorização é obtida a partir do uso freqüente.

Os sistemas exploratórios, cooperativos e criativos - enciclopédias eletrônicas, escrita cooperativa, tomada de decisão, simulação científica, sistemas para composição musical, diagnóstico médico, auxiliares de projetos de arquitetura, etc. - são sistemas geralmente direcionados a usuários peritos no domínio da tarefa, mas não experientes com computadores. São usuários altamente motivados e a preocupação central do design é a de deixar o computador transparente de forma a que o usuário somente se preocupe com a tarefa. Então, baixa taxa de erros e facilidade de aprendizagem devem ser os atributos mais relevantes.

Certamente, não podemos nos esquecer que existem outras considerações que não a usabilidade que levam a design que viola princípios de usabilidade. Por exemplo, considerações quanto a segurança freqüentemente requerem acesso a controles que não "passam" em nenhuma avaliação de usabilidade. Certamente, mensagens de erro construtivas são difíceis em resposta a senhas erradas. Outros sistemas, têm funções escondidas do usuário comum, como por exemplo, funções especiais de *boot* ou de utilização apenas pela administração do sistema, por exemplo.

USABILIDADE NA WEB

Dados disponíveis apontam que em 1998 cerca de três bilhões de dólares deixaram de ser ganhos na WEB norte-americana por causa de design mal feito de páginas, que dificultava a compra em vez de facilitar. Essa estimativa dá conta de um debate que ganha cada vez mais espaço: como equilibrar o uso de recursos visuais capazes de atrair a atenção do usuário e ao mesmo tempo tornar os *sites* fáceis de entender e usar? A questão pode ser parafraseada: Como fazer uso da tecnologia disponível e ao mesmo tempo aumentar a usabilidade de *sites* da Web?

Com cerca de 10 milhões de *sites* na Web em Janeiro de 2000 (com a previsão de cerca de 25 milhões até o final do ano e 100 milhões em 2002), usuários têm mais escolhas que nunca. Por que então eles iriam gastar seu tempo em algo que é confuso, lento, e que não satisfaz suas necessidades?

Como dissemos no início deste capítulo, o número de pessoas que usa a Internet está crescendo sem parar. O crescimento trouxe mudanças no perfil do usuário. No começo predominavam os especialistas e agora predominam os novatos, que mal sabem ligar o computador e que algumas vezes tem rejeição a ele. Assim, deslumbrar-se com a tecnologia não tem mais razão de ser. Com a enorme oferta de alternativas, usuários da Web tem uma notável impaciência e insistência em

gratificação imediata. Se não conseguem entender como usar um *Website* em poucos minutos, eles concluem que não vale a pena perder seu tempo. E então o abandonam.

Usabilidade assumiu uma importância na economia da Internet como nunca teve antes (Nielsen, 1999). No desenvolvimento tradicional de produtos, usuários não experimentam a usabilidade do produto até que o tenham comprado. Por exemplo, somente quando se compra um VCR é que se descobre, o que é bastante comum, o quanto é difícil programá-lo. Mas isso não importa mais para o fabricante, em um primeiro momento, pois a compra já foi faturada. A indústria de software já tem um pouco mais de preocupação com a usabilidade de seus produtos, dado o suporte que é preciso ser dado ao usuário e que tem um custo altamente significativo no produto.

A Web reverteu esse cenário. Agora o usuário experimenta a usabilidade de um *site* antes de se comprometer a usá-lo e antes de ter gasto qualquer dinheiro com potenciais compras e a equação é bastante simples:

- no design de produtos e de softwares tradicionais, usuários pagam antes e experimentam a usabilidade depois
- na Web usuários experimentam a usabilidade antes e pagam depois.

Portanto, é clara a extrema importância da usabilidade no design para a Web.

Um exemplo muito citado é o da IBM dos Estados Unidos (Figura 1.12). A empresa constatou que o recurso mais popular em seu *site* era a função de busca, porque as pessoas não conseguiam descobrir como navegar, e o segundo mais popular era o botão de ajuda. A solução foi um amplo processo de redesign, envolvendo centenas de pessoas e milhões de dólares. Resultado: na primeira semana depois do redesign, em fevereiro de 1999, o uso do botão de ajuda caiu 84% enquanto as vendas aumentaram 400%.



FIGURA 1.12 – PÁGINAS DO SITE DA IBM AMERICANA

Outro exemplo, também bastante conhecido, é o *site* da Amazon Books (Figura 1.13). No início ele era exuberante, mas ultimamente ele tem mudado muito, perdendo os "fogos de artifício" em favor da funcionalidade e rapidez e, principalmente, do relacionamento com o usuário. Certamente o que impulsionou a Web não foi a multimídia, mas a capacidade de relacionamento que nenhuma outra mídia ofereceu até então, e hoje os melhores *sites* são os que estão aprendendo a se relacionar com o usuário.



FIGURA 1.13 – PÁGINA DE ENTRADA DO SITE DA AMAZON BOOKS EM FEVEREIRO DE 2000

Um paralelo esclarecedor pode ser feito com uma agência bancária - *imagine se você fosse a uma agência bancária para fazer um simples depósito e lhe obrigassem a sentar na recepção, ouvir uma propaganda sobre o elenco de serviços que aquele*

banco oferece, passar para outra sala, ouvir mais coisas, esperar, e só então ser transferido para a fila do caixa. É assim que muitos sites são construídos (Pedro Mozart, Gazeta Mercantil, 20 de outubro de 1999, p. C-8)

No design para a Web existem basicamente duas abordagens: uma artística onde o designer se expressa e outra dirigida a resolver o problema do usuário. Certamente existe a necessidade da arte, da diversão e do prazer na Web, mas acreditamos que o principal objetivo dos projetos para a Web deva ser o de tornar fácil para os usuários executarem tarefas úteis. Nada diferente do que advogamos em todo este livro.

Para garantir usabilidade em design para a Web podemos estabelecer alguns princípios básicos (Nielsen, 1999):

- **Clareza na arquitetura da informação**

É essencial que o usuário consiga discernir o que é prioritário e o que é secundário no *site*. Ou seja, antes de mais nada é preciso chegar a um bom arranjo da informação. Os usuários sempre terão dificuldades em encontrar o que procuram, então devem ser ajudados provendo-se um senso de como a informação está estruturada e localizada. Para se conseguir isso, uma das alternativas adotadas em alguns *sites*, é prover um mapa do *site*, de forma que os usuários saibam onde estão e para onde podem ir.

- **Facilidade de navegação**

Uma máxima é que o usuário deveria conseguir acessar a informação desejada no máximo em três cliques. E conseguir organizar a informação dentro disso já é um bom princípio.

- **Simplicidade**

Quem navega quer encontrar o mais rapidamente possível o objetivo da busca. Portanto, a pirotecnia deve ser evitada, dando ao usuário paz e tranquilidade para que possa analisar a informação. Cuidados devem ser tomados para que a simplicidade não signifique ausência de informação. Por exemplo, ao se entrar em uma homepage do *site* de uma instituição ou projeto o usuário precisa que duas perguntas básicas sejam respondidas: Onde eu estou? O que posso obter nesse *site*?



FIGURA 1.14 – PÁGINA DE ENTRADA DO NÚCLEO DE INFORMÁTICA APLICADA À EDUCAÇÃO – NIED – DA UNICAMP – SIMPLES MAS SEM INFORMAÇÃO

Muitos exemplos podem ser encontrados na rede de páginas que de tão simples (Figura 1.14) omitem informações básicas como essas. Sempre temos que ter em mente que o que importa na Web é a informação e ela não pode ser omitida em função de uma pretensa simplicidade.

- **A relevância do conteúdo**

Se nas revistas ou na televisão, por exemplo, a sedução passa muito pela beleza das imagens, na Web o conteúdo é o que mais importa para atrair e prender a atenção do usuário. Sempre que questionados sobre *sites*, usuários se referem a qualidade e relevância do conteúdo. Um bom texto para essa mídia tem que ser o mais conciso e objetivo possível, não promocional ou publicitário, como impera hoje, com perda de credibilidade. É preciso alterar o estilo de escrita, de forma a ser otimizado para leitores *online* que freqüentemente imprimem textos e que necessitam páginas bem curtas com a informação secundária deixada para páginas de suporte.

- **Manter a consistência**

Assim, como para qualquer outro tipo de software, a consistência é um poderoso princípio de usabilidade na Web. Quando as coisas acontecem sempre do mesmo jeito, os usuários não precisam se preocupar a respeito do que irá acontecer. Ao contrário, eles sabem o que vai acontecer baseados numa experiência anterior. Isso leva a adoção de procedimentos padrões, como por exemplo, o uso de cores. *Layouts* ambiciosos devem ser abandonados. As fontes a serem usadas devem ser as mais comuns, pois o designer não sabe as fontes que o usuário tem instaladas.

Outro aspecto bastante verificado e que transparece no design, é o de se gerenciar um projeto para a Web da mesma forma que qualquer outro projeto corporativo tradicional. Isso conduz a um design com uma interface inconsistente. Ao invés disso, um *Website* deve ser gerenciado como um projeto único de interface com o usuário.

- **Tempo suportável**

O tempo de carga das páginas deve ser necessariamente curto. Estudos indicam que 10 segundos é o máximo de tempo antes que as pessoas percam o interesse. Mas na Web os usuários já têm uma baixa expectativa, então esse limite pode aumentar para 15 segundos e mesmo assim ser aceitável

- **Foco nos usuários**

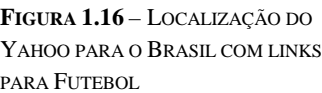
Novamente, todos os princípios podem ser sumarizados em um só: o foco deve estar nas atividades dos usuários. Deixar-se embevecido pelas últimas tecnologias

Resumindo, pessoas são extremamente dirigidas a um objetivo quando usam a Web. Elas têm alguma coisa específica que querem fazer e não toleram nada que dificulte atingir esse objetivo. Portanto, o princípio mestre do design para a Web é "sair do caminho" de forma a que o usuário possa fazer o que quer da maneira mais rápida possível.

Exportar software é vital para qualquer indústria produtora de software. Interfaces internacionais são aquelas projetadas para serem usadas em mais de um país. Fato mais que definitivo é que não basta um produto traduzido em muitas outras línguas. Fazer o design de uma interface internacional pode ou não envolver tradução de linguagem, mas certamente deve envolver conhecimento sobre as necessidades e cultura de outros países. Usuários desejam um produto que seja adequado às suas características culturais e práticas de trabalho e, algumas vezes, isso não implica necessariamente em uma tradução. Equipes de design de software têm então pela frente o desafio de garantir a usabilidade de seus produtos para todo o mercado global.

[illegible]

FIGURA 1.15 – PÁGINA DE ENTRADA DO SITE DO YAHOO AMERICANO EM FEVEREIRO DE 2000



Mesmo não sendo um fenômeno novo, garantir a usabilidade internacional ganha absoluta relevância nos dias de hoje graças à fantástica expansão da WWW (literalmente rede MUNDIAL). No contexto da Web, inicialmente faz mais sentido o processo de internacionalização ao invés de localização, pois em muitos países o número de usuários não justifica uma localização.

À primeira vista, pode parecer que a tendência atual do uso de interfaces gráficas e o uso de elementos gráficos ao invés de palavras resolve grande parte dos problemas. Mas isso não é bem verdade, pois símbolos gráficos e padrões de cores não são necessariamente universais. Podemos classificar os símbolos gráficos em três categorias distintas (Rogers, 1989):

- **Símbolos de semelhança ou ícones:** retratam o objeto que representam. Por exemplo, a figura de uma impressora para indicar a função imprimir, a de uma tesoura para a cortar, etc.
- **Símbolos de referência:** retratam algum objeto que por referência ou analogia pode representar o conceito que o símbolo está querendo representar. Por exemplo, o uso de uma lupa indicando que se pode ver o conteúdo de um arquivo (ou um *preview* de um documento), o uso de um semáforo indicando as funções de parar, prosseguir e esperar de um depurador, etc.
- **Símbolos arbitrários:** formas arbitrárias que somente tem significado por convenção. Por exemplo, todos os sinais de trânsito (que têm sido fonte de inspiração para muitos designers dada a sua característica de padrão internacional). Certamente, essa é a categoria de símbolos mais difícil de aprender para o usuário, a menos que sejam tão largamente utilizados que a convenção deixa de ser menos importante. Por exemplo, poucos usuários se dão conta que o uso do símbolo "?" para indicar dúvida é absolutamente arbitrário (Nielsen, 1993)

Símbolos por semelhança geralmente são comuns a muitos países, desde que não sejam de natureza muito específica. Por exemplo, símbolos de esportes onde em alguns países o esporte não é nem conhecido - o caso do *squash* na Hungria (Brugger, 1990). Portanto o uso de símbolos por semelhança, é preferível para a internacionalização (e para interfaces sem esse objetivo também).

Outro exemplo, de problema interessante em interfaces gráficas é o uso dos marcadores em *check boxes*. Em muitas interfaces o uso do sinal X teve que ser substituído pelo V (sinal de visto) pois em testes no Japão, o X era entendido como exclusão e não seleção da opção.

Conceitualmente os sinais gráficos presentes em interfaces têm sua origem e entendimento na Semiótica, que será tratada no Capítulo 3.

Além desses problemas gráficos, tem-se o problema da absoluta falta de tradição no processo de tradução da linguagem. Cada software e cada designer tem seus próprios termos. O pior é que nem em produtos de um mesmo fabricante a consistência é garantida.

Portanto, interfaces gráficas não possuem uma garantia inerente de serem internacionais. O design precisa efetivamente ser orientado à internacionalização desde o início do processo. E os princípios de IHC tratados ao longo deste livro são os mesmos, embora tratados em algumas partes com um maior grau de profundidade. Em primeiro lugar os usuários e suas tarefas, localizados em um contexto.

REFERÊNCIAS:

Brigger, C. (1990) Advances in the international standardization of public information symbols. *Information Design Journal* 6, 1, 79-88

Carroll, J. M. e Thomas, J. C. (1988) Fun. ACM SIGCHI Bulletin 19,3 (January), 21-24

Czaja, S. J., Hammond, K., Blascovitch, J. J., e Swede, H. (1989) Age related differences in learning to use text-editing system. *Behaviour & Information Technology* 8, 4, 309-319.

Erickson, T.D. (1990) Working with Interface Metaphors. Em B. Laurel (ed.) (1990) *The Art of Human-Computer Interface Design*. Reading, Mass.: Addison-Wesley

Fernandes, T.(1995) *Global Interfaces Design: A Guide to Design International User Interfaces*. Academic Press Professional, Boston, MA

Fowler, C. J. H. e Murray, D. (1987) Gender and cognitive style differences at the human-computer interface. *Proc. IFIP INTERACT'87 Second Intl. Conf. Human-Computer Interaction* (Stuttgart, Germany, 1-4 September), 709-714

Gomez, L. M., Egan, D. E., e Bowers, C. (1986) Learning to use a text-editor : Some learner characteristics that predict success. *Human-Computer Interaction*, 2, 1, 1-23

Jorgensen, A. H. (1990) The personal touch: A study of users customization practice. *Proc. IFIP INTERACT'90 Third Intl. Conf. Human Computer Interaction* (Cambridge, U.K., 27-31 August), 549-554

Kay, R.H. (1989) A practical and theoretical approach to assessing computer attitudes: The computer attitude measure (CAM). *Journal on Research on Computing in Education*, 456-463

LaLomia, M. J. e Sidowski, J. B. (1991) Measurements of computer attitudes: A review. *Intl. J. Human-Computer Interaction*, 3,2,171-197

Laurel, B. ed. (1990) *The Art of Human-Computer Interface Design*. Reading, Mass.: Addison-Wesley

Laurel, B. (1993) *Computer as a Theatre*. Reading, Mass.: Addison-Wesley

Lyons, J. (1970) *New Horizons in Linguistics*. Hamondsworth:Penguin

Nielsen, J.(1990) *Design User Interfaces for International Use*. Elsevier Science Pub., Amsterdam, The Hetherlands

Nielsen, J. (1993) *Usability Engineering*. Academic Press, Cambridge, MA

Nielsen, J. (1999) *Design Web Usability*. New Riders Publish., Indianapolis, Indiana USA

Norman, D. A. (1988) *The Psychology of Everyday Things*. Basic Books, New York

Perratore, E., Thompson, T., Udell, J., e Malloy, R. (1993) Fighting fatware. *Byte* 18, 4(April), 98-108

Preece, J. (1993) *A Guide to Usability: Human Factors in Computing*. Reading, Mass.: Addison-Wesley

Preece, J. *et al.* (1994), *Human-Computer Interaction*. Reading, Mass.: Addison-Wesley

Proceedings ACM CHI'85 Conference: Human Factors in Computing Systems. Lorraine Borman and Bill Curtis (eds.), San Francisco (April 14-18, 1985)

Proceedings ACM CHI'92 Conference: Human Factors in Computing Systems. Penny Bauersfeld, John Bennett and Gene Lynch(ed.), Monterey, CA(May 03-07,1992)

Rogers, Y. (1989) Icons at the Interface: Their usefulness. *Interacting with Computers*, 1,1(April), 105-117

Russo, P. e Boor, S. (1993) How fluent is your interface? Design for international users, in *Bridges Between Worlds. INTERCHI'93 Conferencing Proceedings*, Reading, Mass.: Addison-Wesley

Sein, M. K. e Bostron, R. P. (1989) Individual differences and consceptual models in training novice users. *Human-Computer Interaction*, 4, 3, 197-229

Sellen, A. e Nicol, A. (1990) Building user-centered On-line Help. Em B. Laurel (ed.) *The Art of Human-Computer Interface Design*. Reading, Mass.: Addison-Wesley

Schneiderman, B. (1980) *Software Psychology: Human factors in Computer and Information Systems*. Little, Brown, Boston (1980)

Schneiderman, B. (1998) *Design the User Interface - 3rd Edition*. Reading, Mass.: Addison-Wesley

Tesler, L.G. (1991) Networked Computing in the 1990s. *Scientific American* 265, 3(September), 86-93

Vaske, J. J. e Grantan, C.E. (1990) *Socializing the Human-Computer Environment*. Norwood, NJ:Ablex

Walkers, J. (1990) Through the Looking Glass. Em B. Laurel (ed.) *The Art of Human-Computer Interface Design*. Reading, Mass.: Addison-Wesley

Wozny, L. A. (1989) The application of metaphor, analogy, and conceptual models in computer systems. *Interacting with Computers* 1, 3, 273-283