Computação

Interação Humano Computador

Francisco Carlos de Mattos Brito Oliveira
Fernando Antônio de Mattos Brito Oliveira


Geografia


História


Educação Física


Química


Ciências Biológicas


Artes Plásticas


Computação


Física


Matemática


Pedagogia

Computação

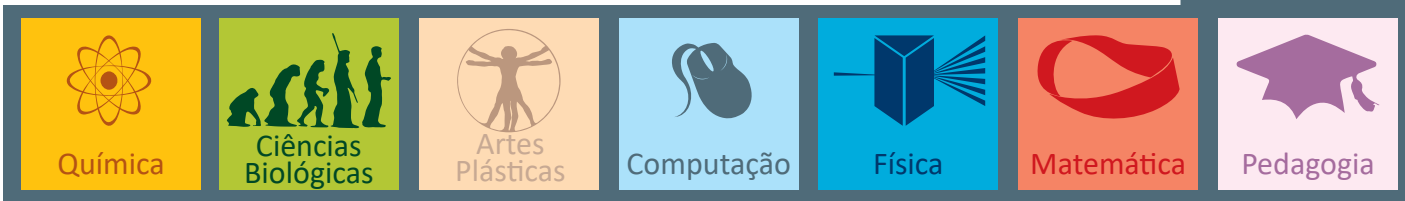
Interação Humano Computador

Francisco Carlos de Mattos Brito Oliveira
Fernando Antônio de Mattos Brito Oliveira

2ª edição
Fortaleza - Ceará

Ed UECE

2015



Editora Filiada à

ASSOCIAÇÃO BRASILEIRA DAS EDITORAS UNIVERSITÁRIAS

Editora da Universidade Estadual do Ceará – EdUECE
Av. Dr. Silas Munguba, 1700 – Campus do Itaperi – Reitoria – Fortaleza – Ceará
CEP: 60714-903 – Fone: (85) 3101-9893
Internet: www.uece.br – E-mail: eduece@uece.br

Secretaria de Apoio às Tecnologias Educacionais
Fone: (85) 3101-9962

Sumário

Apresentação

O presente livro tem como principais objetivos a discussão sobre a relevância do projeto de interface de sistemas interativos de computador e a apresentação de teorias e técnicas que auxiliam no processo de construção e avaliação dessas interfaces. Todo e qualquer contato que o usuário tem com o sistema faz-se através de sua interface. Uma interface projetada cuidadosamente pode levar à satisfação de seu usuário, aumento de produtividade, na qualidade do trabalho e à diminuição de erros. O raciocínio reverso é verdadeiro. Projetos que não levam em conta análise criteriosa de tarefas, de perfis dos usuários, do ambiente e das situações nas quais os sistemas serão utilizados podem se constituir em sistemas capazes levar seus usuários a cometerem erros com consequencias catastróficas. Sob o ponto de vista mercadológico, quando o usuário compara dois sistemas que oferecem a mesma funcionalidade, ele escolhe aquele com maior usabilidade.

Neste livro, discutimos a necessidade de se projetar sistemas com alto grau de usabilidade, confiáveis, que demandem menos horas de treinamento, maior performance e menor taxa de erros. Apresentamos várias das teorias vigentes, e discutimos métodos e aspectos práticos do projeto de interface de sistemas interativos.

Os autores

Capítulo 1

Projetando Sistemas que melhorem o desempenho humano

Objetivos

- Neste capítulo, apresentamos a área de interação humano computador como uma ciência multidisciplinar, nascida fundamentalmente da colaboração entre a psicologia e a engenharia de software. Em seguida discutimos quais objetivos de performance sistemas de computador devem buscar satisfazer. Apresentamos depois as qualidades que uma boa interface devem possuir, seguida de uma discussão sobre como se medir essas qualidades e porque elas são importantes. Discutimos também os grandes de desafios de se construir interfaces de qualidade. Fechamos essa unidade com a apresentação das linhas de pesquisa existentes para se fazer frente aos desafios apresentados.

1. Interação Humano Computador – Uma ciência multi--disciplina

> "A interface é o sistema".

A frase de Larry Tesler, cientista-chefe da Apple define bem a importância da experiência de uso que o sistema proporciona a seu usuário. A interface é a única maneira que o usuário tem de avaliar o sistema. A ele não interessa a linguagem de programação na qual o sistema foi desenvolvimento, o tipo de equipamento no qual a aplicação foi desenvolvida, nem a metodologia empregada na sua concepção e desenvolvimento. O conceito vai além da estética da tela (disposição de menus, cores, etc.). Ele se estende a questões como corretude, tempo de resposta, grau de dificuldade de uso, rapidez no desempenho de tarefas-chave, nível de erros cometidos por usuários durante o uso, facilidade de aprendizado, fadiga produzida pelo uso prolongado, acomodação de usuários com necessidades especiais, dentre outros.

Para desenvolver sistemas com tais características é necessário entender não somente como o computador funciona, mas também como o ser humano "funciona". Assim, cientistas da computação se juntaram a psicólogos dando início as atividades de pesquisa na área de interação humano--computador. A parceria, além de dar origem a uma nova área de pesquisa, é vantajosa para as duas "ciências-mães". A computação cria oportunidades

de experimentação para as teorias da psicologia. Programas desenvolvidos podem capturar dados relativos à sua execução, que em seguida, podem ser usados para confirmar ou invalidar teorias. Já as teorias validadas, orientam analistas de sistemas no projeto de interfaces.

Em praticamente todos os grandes avanços da computação observados recentemente vemos a forte influência da disciplina de interação humano-computador (IHC). O sucesso das redes sociais, por exemplo, vem da necessidade que as pessoas tem em se conectar (ou reconectar) com familiares e amigos a fim de compartilhar experiências. Também é fácil entender como as interfaces gestuais presentes em "tablets" e "smartphones" fazem sucesso: os gestos necessários para a interação nas aplicações quando não o mesmo, muito se assemelham com os que fazemos no mundo real, em situações semelhantes. Por exemplo, quando tocamos uma fotografia em um "tablet" com dois dedos e afastamos um do outro, estamos criando mais espaço entre os dois dedos, consequentemente aumentando a tamanho da fotografia. O gesto pela semelhança que teria com aquele que faríamos no mundo real, permite uma experiência sensorial mais rica e intuitiva para o usuário da aplicação. Essa experiência aumenta o grau de satisfação do usuário, diminui o esforço para o aprendizado e aumenta o nível de retenção (o usuário se lembrará com mais facilidade).

Igualmente fascinante em IHC é a oportunidade de se trabalhar a acessibilidade. Ao se projetar a interface tendo em vista as necessidades de pessoas especiais oferecemos a elas a oportunidade de se tornarem produtivas e úteis à sociedade. Interfaces acessíveis não envolvem apenas pessoas com alguma deficiência sensorial (cegos, surdos, etc.), elas também cobrem as limitações de interação trazidas pela idade (usuários idosos). A inclusão digital da terceira idade permite a essas pessoas que se mantenham intelectualmente mais ativas e produtivas por mais tempo.

Os exemplos apresentados acima mostram que uma nova geração de aplicativos surge com um grau de interatividade superior àquelas baseadas no modelo WIMP (Windows, ícones, Menus e Pointers (dispositivos de apontamento, como mouse)). O modelo WIMP predominou até os primeiros anos do século XXI e agora começa a ser suplantado. O modelo WIMP praticamente toda interação baseia-se em um único sentido, o da visão, sobrecarregando-o. Hoje falamos em interfaces multimodais, onde usuário usa mais de um sentido, como tato, audição, propriocepção, e até mesmo o olfato. O novo modelo também nos desprende do computador de mesa. Smartphones nos acompanham aonde quer que vamos, sabem onde estamos e aplicativos embarcados podem usar essa informação para criar uma vantagem competitiva para seu usuário. Todos esses desafios exigem mais conhecimento por parte do projetista de sistemas interativos.

Nesse primeiro capítulo discutimos como sistemas devidamente projetados podem melhorar a performance humana. No capítulo seguinte, apresentamos algumas teorias e métodos que apoiam o desenvolvimento de sistemas interativos. O capítulo três é dedicado ao debate acerca do enfoque gerencial do processo de construção desse tipo de sistemas. O quarto e último capítulo é dedicado ao debate sobre a técnica de construção de sistemas conhecida como *manipulação direta* seguida de uma breve apresentação de sistemas capazes de criar ambientes virtuais.

2. Maximizando o desempenho humano

Interface interativa de alta qualidade é produto de planejamento cuidadoso, sensibilidade às necessidades dos usuários e testes rigorosos. Quando a interface é bem projetada, ela praticamente desaparece deixando que o usuário se concentre em seu trabalho, diversão ou pesquisa. O *US Military Standard for Human Engineering Design* define objetivos que equipamentos (sistemas de computação, armamentos, etc.) devem atingir:

- Alcançar performance requisitada para usuário;
- Minimizar qualificação e tempo de treinamento para usuário;
- Alcançar confiabilidade homem-máquina adequada a tarefa;
- Facilitar padronização intra e entre sistemas.

3. Engenharia de Sistemas

3.1. Funcionalidade adequada

Antes de tudo é fundamental descobrirmos as funcionalidades e em que circunstancias elas são executadas. Essas funcionalidades normalmente remetem a tarefas que fazem parte das atribuições dos cargos que os usuários ocupam, por exemplo. A análise de tarefas é uma das principais atividades do projetista de sistemas interativos. Esse tipo de atividade será discutida mais adiante nesse texto. Nesse ponto é importante que o leitor saiba que essa é uma atividade "top down", ou seja, o projetista primeiro identifica as tarefas do jeito que os usuários definem, em seguida deve "quebra-la" em partes menores para entender os pormenores da atividade para em seguida, aplicar seus conhecimentos de tecnologia para implementa-la da maneira que pareça mais natural para aquele que vai utiliza-la no dia a dia.

Dentre essas tarefas, as mais complicadas de se modelar são aquelas em que os usuários tem realizar em situação de emergência na qual ele estará estressado, com pouco tempo para realiza-la e não poderá errar.

Outra coisa muito importante é projetar ações que possam ser reversíveis, ou seja, se o usuário errou não tem problema, basta que ele comande a ação de reversão e o sistema volta a posição anterior.

3.2. Confiabilidade, disponibilidade e segurança

O usuário precisa confiar no sistema. Sistema no qual o usuário não confia está fadado ao fracasso. A confiança do usuário no sistema é frágil, uma falha sequer e a confiança fica comprometida por muito tempo. Comandos devem funcionar como especificados e conforme descritos nos manuais do usuário. Os dados e valores exibidos em telas e relatórios devem corresponder à realidade, àquilo que está armazenado no banco de dados.

Outro fator importante é a disponibilidade. O sistema precisa sempre estar disponível. Nos dias de hoje não se espera nada menos do que sistemas que operem vinte e quatro horas por dia, sete dias por semana. Sistemas de computação não devem parar nem para fazer "backup". Estratégias de upgrades devem ser concebidas para deixar o sistema o mínimo de tempo indisponível.

O conceito de segurança está ligado ao de disponibilidade. Um sistema com pouca segurança tem muito mais probabilidade de se tornar indisponível. Entretanto a segurança na está somente ligado à disponibilidade. Muitas vezes a informação está correta mas não deve ser acessada por determinado usuário – Um funcionário não pertencente ao departamento de pessoal, por exemplo, não deve ter acesso aos salários dos demais funcionários da empresa.

Os três conceitos aparecem listados juntos nesse tópico pois suas ocorrências em um determinado nível podem tornar o sistema inviável, não interessando quão bom seja o projeto de sua interface.

3.3. Padronização, consistência, integração, portabilidade

A padronização da interface, dos objetos (normalmente visuais) com os quais os usuários interagem é fator decisivo para o sucesso de um sistema interativo. Pequenas diferenças entre aplicações irritam o usuário, aumentam o tempo de aprendizado e a incidência de erros. Assim a *padronização* refere-se a características comuns entre as interfaces tanto entre programas de uma mesma aplicação como entre diferentes aplicações.

O conceito de *Consistência* está muito próximo ao de padronização. *Consistência* refere-se a seqüências de ações comuns, termos, unidades, leiautes, cores, tipografias utilizadas em uma aplicação. A consistência é por si só um forte fator determinante de sucesso ou fracasso de um projeto. Interfaces inconsistentes causam basicamente os mesmos males discuti-

dos no parágrafo anterior. O usuário também espera que a consistência seja mantida entre as diversas versões da aplicação. Mudar o modo de se realizar uma tarefa de uma versão para outra é algo bastante perigoso de ser fazer. O conjunto de ações necessárias para o usuário atingir o mesmo objetivo utilizando a versão nova deve ser menor, as ações devem ser mais simples e intuitivas, caso contrário essa quebra de continuidade pode levar à frustração e perda de produtividade, além de pode trazer perda de mercado para a empresa que o construiu. A consistência também pode ser observada entre as ações do mundo real e o digital. Quanto mais consistente, em termos ação do usuário e reação do sistema, com as reações que ele esperaria no mundo real, melhor.

Já a *Portabilidade* diz respeito ao potencial de se converter dados e compartilhar interfaces entre vários sistemas e equipamentos de diferentes tipos. Hoje em dia essa característica está bem valorizada devido às tecnologias de *computação nas nuvens* – a informação está disponível aos usuários onde quer que eles estejam (desde tenham acesso à internet) e em vários dispositivos diferentes (desktops, tablets, smartphones, etc).

O conceito de *integração* está ligado à transferência automática de dados entre sistemas distintos. Por exemplo, o sistema de compras pode estar integrado ao de pagamento fazendo com que a pessoa que vai fazer o pagamento já saiba qual é a forma de pagamento negociada sem ter que ir ao seu colega de trabalho e perguntar.

3.4. Cronogramas e orçamentos

Cronogramas atrasados e orçamentos estourados têm uma alta probabilidade de impactar negativamente os projetos de sistemas interativos. O clima de animosidade que se estabelece quando o projeto fica fora do controle dificulta a colaboração entre os membros da equipe e entre a equipe e os diversos grupos de usuários engajados no desenvolvimento da aplicação. Em situações mais graves, gerentes podem ser substituídos, técnicos com conhecimento-chave podem deixar a equipe por insegurança ou simplesmente porque o ambiente de trabalho ficou intolerável.

Como forma de recuperar o tempo perdido, os gestores do processo podem decidir por eliminar certas fases de desenvolvimento, o que fatalmente levará a produtos de baixa qualidade e que podem comprometer a imagem da empresa. É importante salientar que testes planejados e executados cuidadosamente e de acordo com metodologias consagradas podem levar, ao contrário do que se imagina a princípio, a prazos mais curtos e aplicações de melhor qualidade.

4. Critérios para se medir a qualidade interfaces humano-computador

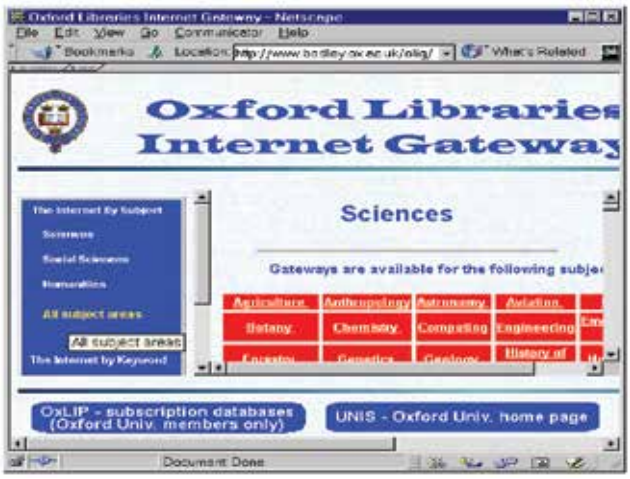
"*Não se pode administrar o que não se pode medir*", Peter Drucker. Para que tenhamos algum indicativo sobre a qualidade do projeto de interface de sistemas interativos que estamos propondo ou avaliando, precisamos ter critérios e métricas. Shneiderman (Shneiderman; Plaisant, 2005) propôs cinco fatores, que apresentamos a seguir.

4.1. Tempo para aprender

Medida simples e direta: Quanto tempo leva para *um usuário típico aprender* a usar o sistema para executar determinada tarefa. A definição levantada nos apresenta outra: *O usuário típico.* Usuário típico é aquele usuário representativo da comunidade de usuários daquela aplicação. Assim, se o projeto envolve desenvolvimento de interfaces para usuários com deficiência visual, o usuário típico é uma pessoa com deficiência visual.

Esse fator é muito valorizado tanto por empresas como por usuários em geral. Sistemas fácies de se aprender demandam menos treinamento e treinamento custa tempo e dinheiro. Além de tempo e dinheiro existem também questões de cunho administrativo. Se for complicado treinar uma pessoa para usar determinado sistema, a empresa pode pensar duas vezes em substituir um funcionário problemático. Tal situação não é apreciada por administradores.

O tempo para aprender também pode ser, por si só, fator determinante para um produto ou mesmo para a empresa que o concebeu. Um exemplo atual é a Google. Quanto tempo se leva para aprender a fazer uma pesquisa na internet usando o buscador deles? Basta alguém ver outra pessoa fazendo, não mais que poucos minutos. O que poucos sabem é que nem sempre foi assim. Os primeiros programas de busca na internet (figura abaixo) eram cheios de opções e isso complicava a vida de usuários novatos. Como saber em que categoria a informação que procuro se encontra? Quais foram os critérios utilizados na classificação? De quanto em quanto tempo ela é atualizada? O estilo *clean* da interface conquistou o mundo e ajudou a fazer daquela empresa uma das maiores do mundo. Digite algo que ele vai te ajudar a encontrar.



4.2. Performance

É medida em tempo gasto pelo usuário típico em realizar uma atividade bem representativa do domínio da aplicação. Um operador de telemarketing, por exemplo, passa a maior parte de seu dia de trabalho realizando um conjunto limitado de tarefas com ajuda com computador. A questão é que ele usa determinado programa, que o auxilia em determinada ação, centenas ou mesmo milhares de vezes ao dia. Portando o tempo que o usuário leva para realizar a tarefa é bastante relevante para a empresa. Com menos tempo, a empresa precisa de menos funcionários para contatar o mesmo número de clientes e isso representa ganhos de competitividade.

4.3. Taxas de erro

Erros podem causar grandes prejuízos. Perdas que podem envolver vidas, recursos financeiros, imagem, oportunidade, mercado. Outros erros podem ser facilmente corrigidos e sua ocorrência se deve ao fato de que o usuário busca naturalmente a realização de tarefas de forma acelerada. Obviamente se quer que o usuário tenha uma grande performance e não cometa erros. Existem várias técnicas para melhorar a relação performance/erro que apresentaremos no decorrer desse texto. Nessa altura é importante salientar o ponto central da questão: Se os erros não causarem grandes prejuízos e os prejuízos causados forem facilmente reversíveis, taxas de erros baixas são toleradas, para o bem da performance.

4.4. Tempo de retenção

Essa métrica diz respeito ao tempo que usuário mantém o conhecimento de como usar a aplicação. O tempo de retenção está relacionado com o tempo de aprendizado e à freqüência de uso. Obviamente quanto mais simples e

intuitivo for a forma com que os usuários interagem com o sistema, maior será o tempo de retenção. O tempo de retenção é importante para aplicações de não são utilizadas com freqüência, tais como edição de imagens para não profissionais que usam software desse tipo somente para ilustrar um texto ou outro, como alunos e professores. Assim, aqueles aplicativos que propiciarem um tempo maior de retenção levam vantagem.

4.5. Satisfação subjetiva

Essa métrica diz respeito à experiência que o usuário tem ao utilizar um aplicativo para realizar determinado tipo de tarefa. Usuários podem se sentir frustrados porque aplicação "não ajuda" na hora de realizar determinada ação, ou que não conseguiram achar a opção que os leva a certa ação, ou ainda que não conseguiram controlar o sistema. Normalmente a satisfação subjetiva é aferida por meio questionários aplicados com os usuários após o uso do sistema, em sessões de *debriefing.* Nessas sessões, um membro da equipe de desenvolvimento lê uma sentença para o usuário e pede que ele dê uma nota indicando o grau de concordância com a sentença. Uma sentença poderia ser: "Consegui fazer todas as tarefas rapidamente e sem muito esforço". Os graus de concordância possíveis seriam (-2 – discordo totalmente, -1 – Discordo, 0 – não concordo/não discordo/sem opinião, 1 - Concordo, 2 – Concordo plenamente). Essas são as escalas de *Likert*. Devido a atribuição de valores às sentenças é possível se fazer uma análise estatística acerca da satisfação subjetiva dos usuários que participam de estudos de usabilidade.

Vale salientar que muitas vezes o projetista terá de fazer escolhas entre quais qualidades ele vai querer que o sua aplicação possua, priorizando umas em detrimento de outras. É importante que essas escolhas sejam feitas de forma madura, que o projetista tenha entendimento do impacto que a não satisfação plena de determinado fator acarretará na aceitação do projeto pela comunidade de usuários. Aconselha-se envolver o usuário nesse processo de decisão.

5. Porque devemos construir sistemas interativos de qualidade

O interesse crescente na área de IHC advém da tomada de consciência dos prejuízos que sistemas mal projetados podem acarretar. Schneiderman identificou quatro tipos sistemas que mais se beneficiam das questões debatidas nesse texto: Sistemas de missão crítica, sistemas industriais e comerciais, aplicações de escritório e domésticas, e sistemas colaborativos. A seguir discutiremos cada um deles.

5.1. Sistemas de Missão Crítica

Exemplos desse tipo são: controle de tráfego aéreo, usinas nucleares (figura abaixo exibe sala de controle), suporte ao vôo, operações policiais e militares. Erros cometidos por usuários de desses sistemas podem levar à morte de pessoas. O tempo de treinamento para aplicações desse grupo é normalmente longo e objetiva a performance livre de erros. Os sistemas devem ser altamente confiáveis e eficientes. Os seus operadores devem ser manter eficientes mesmo em situação de alto estresse.

5.2. Sistemas Industriais e Comerciais

Nessa categoria temos sistemas bancários, de seguradoras, de controle de estoque, controle de reservas de hotel, de passagens aéreas e de telemarketing (figura abaixo). São características apreciadas nesse tipo de aplicação: baixa demanda por treinamento, velocidade de operação e baixas taxas de erro. O projetista deve também ficar atento a questões como fadiga e esgotamento do operador. Para esse tipo de sistemas é importante fazer análise dos tempos que as tarefas mais significativas leva e como fazer para melhora-los.

5.3. Aplicações de escritório, para uso doméstico e de entretenimento

Nesse grupo destacamos: processadores de texto, máquinas de auto-atendimento, video games, pacotes educacionais, correio eletrônico e teleconferência. Esse tipo de aplicação, facilidade de aprendizado e uso, baixas taxas de erro e satisfação subjetiva são as mais apreciadas. Dado a competição acirrada, usuário insatisfeitos podem facilmente migrar de solução.

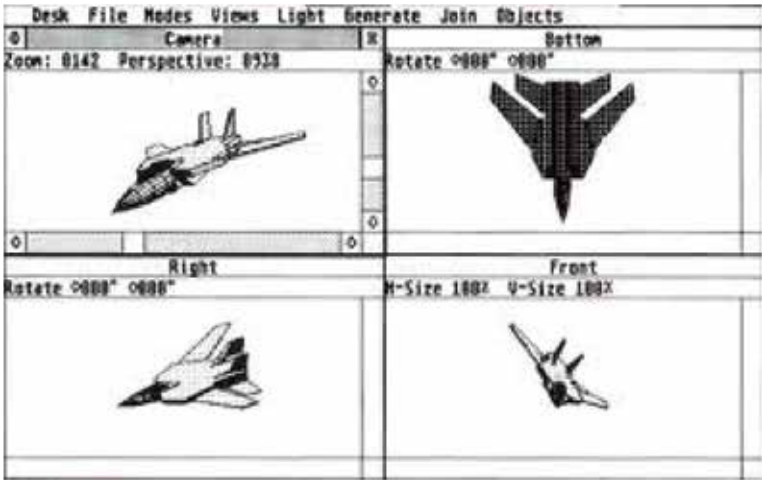
Um dos desafios para esse grupo de aplicações é a escolha das funcionalidades e a necessidade de se acomodar as demandas de usuários iniciantes, que procuram simplicidade de operação, com as dos usuários experimentados que desejam um conjunto maior e mais complexo de funcionalidades e rapidez na operação.

5.4. Sistemas exploratórios, de apoio à criação e sistemas colaborativos

Nos grupos de sistemas exploratórios temos navegadores de internet, processadores de texto colaborativos, uso de estatística para formação de hipóteses. Como exemplos de sistemas de apoio à criação, podemos citar aqueles que apóiam o desenho de arquitetos e engenheiros (Figura abaixo), os ambientes de desenvolvimento programas/sistemas integrados (IDEs) ou ainda sistemas de apoio à composição e edição musical. Já os sistemas colaborativos permitem a duas ou mais pessoas trabalharem no mesmo projeto mesmo que estejam separadas pelo tempo, ou pelo espaço.

A característica comum entre todos esses sistemas é o fato de que seus usuários possuem conhecimento específico nas suas respectivas áreas, mas não conhecem conceitos de computação, são altamente motivados e com expectativas igualmente elevadas. O uso de tais sistemas pode variar de ocasional a freqüente, e o projeto e avaliação desses sistemas é sempre complicado.

Para essa categoria de sistemas, é esperado que a interface "desapareça" durante o uso, permitindo que o operador foque na execução da sua tarefa. Tal objetivo é normalmente alcançado quando o computador fornece a possibilidade de manipulação direta de elementos do mundo real representados na aplicação. Através da manipulação direta, tarefas podem ser executadas, por exemplo, com o uso de gestos, o que propicia reação imediata do sistema, apresentando, por sua vez, novas possibilidades de interação.



6. Acomodando a Diversidade Humana

A diversidade das habilidades, formação, motivação, personalidade e estilos de trabalho entre humanos constitui um enorme desafio aos projetistas de sistemas interativos. É fundamental que este reconheça a diversidade humana, identifique-a entre usuários, entenda que impacto ela tem ou terá no(s) projeto(s) nos quais ele participa, além de conhecer técnicas para acomoda-las. Passaremos agora a listar e discutir algumas dessas diversidades.

6.1. Habilidades físicas e ambientes de trabalho

Pessoas tem diferentes habilidades perceptivas, cognitivas, e motoras. Assim um dos grandes desafios da área é o projetar interfaces que acomodem essas diferenças. Para ajudar, já existe muita literatura à disposição do projetista. A antropometria estuda as dimensões humanas (Dreyfuss, 1960). Essas dimensões são normalmente classificadas por faixa etária, sexo, grupo étnico. Esses dados satisfazem medidas de cerca de 95% da população de seus grupos. Esse é o percentual da população que a indústria acomoda quando cria seus produtos. A título de ilustração, os parâmetros utilizados para o projeto de teclados incluem a distância entre as teclas, o tamanho delas, a pressão que deve ser exercida para ser ativada, etc. Esses parâmetros satisfazem uma boa parte da população enquanto deixa uma fatia de fora. Quando possível é

importante considerar a possibilidades de ajustes nos dispositivos em fase de projeto. Exemplos de ajustes são: Controle de brilho/contraste para monitores, altura para teclado, velocidade de rastreio do mouse, etc.

Entretanto, não existem somente as medidas humanas estáticas. Em alguns casos precisamos de informações como a que distância podemos alcançar um objeto enquanto estamos sentados, da velocidade que o dedo humano consegue tocar um objeto, para citar algumas.

Existem também as metidas relacionadas à percepção como: Tempo de resposta a um estimulo visual, tempo de adaptação ao escuro/claro, tempo para identificar um objeto em um contexto, tempo para identificar a direção e velocidade de um objeto em movimento, diferenças de acuidade entre as visões periféricas e foveal. É importante também observar que podem existir pessoas daltônicas dentre os membros da comunidade de usuários. Existem ainda as questões ligadas à fadiga visual, principalmente aquela causada pelo uso contínuo de uma tela de computador. Isso tudo sem falar dos outros sentidos como audição, tato e até mesmo olfato.

A características físicas do ambiente de trabalho também desempenham papel importante na performance de usuários. O *American National Institute Standard for Human Factors Engineering of Visual Display Terminal Workstations* lista as seguintes preocupações que um projetista precisa observar:

- A superfície de trabalho;
- O espaço para pernas;
- A ajustabilidade de alturas e ângulos para cadeiras e superfícies de trabalho.
- O ajuste de postura e ângulo, altura do encosto, suporte lombar.

Existem ainda orientações para os níveis de iluminação, ruído, temperatura ambiente, umidade, diminuição de reflexo.

Projeto do ambiente de trabalho melhora a satisfação, performance, diminui a taxa de erro. Computadores no ambiente de trabalho também não podem impedir a colaboração entre as pessoas, limitar interação social ou ajuda na hora de resolver problemas. Projetos de ambientes de trabalho são normalmente discutidos sob tema da ergonomia, entretanto, a antropometria, sociologia, psicologia industrial, o comportamento organizacional e a antropometria são ciências que fornecem subsídios para esse tipo de trabalho.

6.2. Diferenças de personalidade

Uma das diferenças de estilo mais importantes entre usuários advém do seu gênero e são mais facilmente notadas em video games. Normalmente os jogos são desenvolvidos por homens, tendo em vista usuários homens. Mulheres normalmente se queixam da agressividade desses jogos e de sua trilha sonora. Termos como "Kill" ou "Abort", frequentemente utilizados em computação, são mal vistos pelo público feminino. É desafio corrente da área de IHC entender que características são mais atraentes para a mulher. Huff (Huff; Cooper, 1987) percebeu que mulheres se interessam mais por jogos que envolvem diálogo.

Entretanto, não é somente o gênero a fonte de diferentes preferências. A psicologia criou muitas escalas diferentes para definir tipos diferentes de personalidades. Existem aqueles que gostam de se arriscar e aqueles que tem aversão ao risco, aqueles que fazem questão de ter o controle absoluto da situação (locus de controle interno). Uns tem comportamento reflexivo e outros impulsivo, pensamento convergente e pensamento divergente, tolerância ao estresse, à ambigüidade, à ansiedade. Teorias como a *Myers-Briggs Type indicator (MBTI)* (Briggs, 1987) podem auxiliar no entendimento do relacionamento entre personalidades e profissões. O conhecimento do tipos diferentes de personalidades pode ser útil para projetistas de aplicações voltadas para educação, arte, música e entretenimento, dentre outras.

6.3. Diferenças culturais

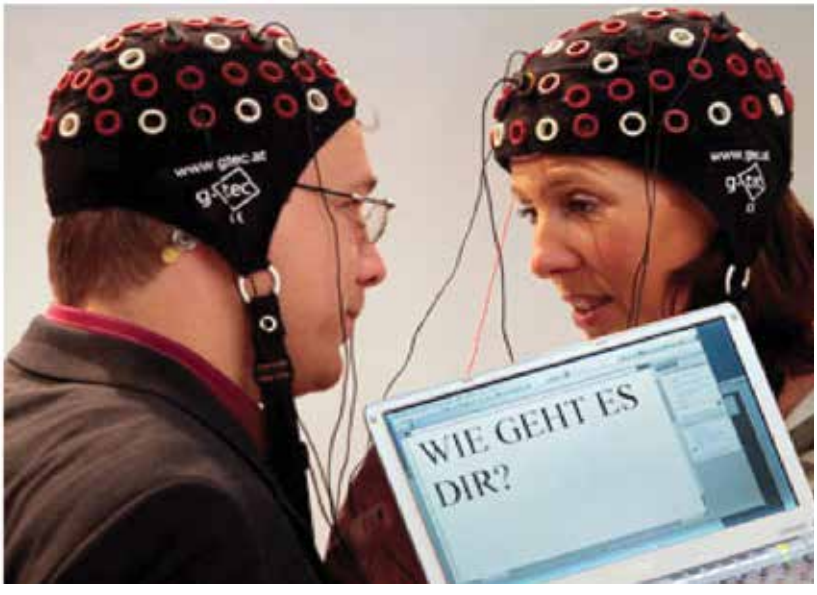
A medida em que o processo de globalização avança, mais se faz necessário entender como construir produtos que possam ser utilizados por pessoas de outros países e de outras culturas. Ao processo de adequação de um produto de software para o uso em diferentes culturas damos o nome de *localização.* Projetar o software tendo em mente uma eventual localização pode dar uma vantagem competitiva ao fornecedor.

Para facilitar o processo de localização, uma das práticas mais comuns é gravar todo o texto (instruções, ajuda, mensagens de erro e labels) em arquivos separados. Entretanto esse processo pode se tornar complicado. Chineses e japoneses, por exemplo, lêem a tela da direita para esquerda. Nesses lugares, o respeito à tradição pode leva-los a preferirem um tipo de interface mais convencional. Outros tipos de adequação envolvem caracteres, formatos de data e hora, de numerais, de moeda, de pesos e medidas, formatos de telefone e endereços, formas de tratamento, pontuação, documentos de identificação (Carteira de identidade, CPF, Passaporte). São também objetos de preocupação para o projetista: formas de ordenação, ícones, botões, cores, gramática, grafia, etiqueta, políticas, tom, formalidade e metáforas. Apesar de longa, a lista ainda pode estar incompleta. Para se assegurar que o sistema terá boa aceitação em diferentes culturas, recomenda-se estudos de usabilidade envolvendo pessoas dessas culturas.

6.4. Acessibilidade

As tecnologias da informação e comunicação estão cada vez mais presentes em atividades profissionais, nos lares, no entretenimento e na interação social. Por isso mesmo, elas se apresentam como um importante mecanismo de inclusão social para aqueles com algum tipo de deficiência. Podemos, por exemplo, trocar emails com uma pessoa cega e nunca saber que ela tem o problema. Livros digitais podem evitar a viagem de um cadeirante à biblioteca. O mecanismo de *closed-caption* permite ao surdo-mudo acompanhar o noticiário na televisão.

Muitas vezes, as soluções passam por itens de hardware. Por exemplo, existe o "refreshable braille display" que, acoplado ao computador permite ao usuário deficiente visual ler o conteúdo de arquivos de computador. Deficientes visuais também contam com a ajuda dos "ledores" de tela, que fazem uso de tecnologias *text-to-speech* (TTS) para terem acesso ao conteúdo de telas e arquivos de computador.



Dispositivos de telecomunicação para os deficientes auditivos, os TDDs (sigla em Inglês), permitem acesso à informação via telefone. Existe uma pletora de dispositivos e programas de computador disponíveis àqueles com algum tipo de deficiência física, como “mouse” ótico, (figura no alto) programas de reconhecimento de voz, e até mesmo interface cerebral (figura acima).

6.5. Usuários da terceira idade

O grupo de usuários da "melhor idade" tem muito a se beneficiar com o uso das tecnologias da informação e comunicação. Essas tecnologias permitem que essas pessoas façam coisas sem sair de casa, como pagar as contas do mês, fazer compras, etc. Além de poder dar mais segurança física a pessoas daquela faixa etária, essas tecnologias também oferecem a oportunidade deles saírem do isolamento social, aderindo a comunidades sociais na internet. Há sites de jogos, onde é possível se jogar com outra pessoa que esteja localizada remotamente, e a há as aplicações projetadas especificamente para comunicação, como Skype, MSN, além, claro, das rede sociais. Aplicativos projetados para desafiar as nossas capacidades cognitivas, como o Sudoku, são bastante recomendados por médicos como "exercício cerebral" para que pessoas nessa faixa etária continuem ativas.

O desafio para o projetista é que com o passar da idade nosso corpo sofre mudanças não uniformes em seu funcionamento fisiológico e psicológico. Há declínio de acuidade visual e auditiva. Existem também a diminuição nos tempos de execução de tarefas e de reação a estímulos. A idade também torna mais difícil a adaptação a mudanças de iluminação e o aprendizado. Uma ressalva importante nisso tudo é que a maioria das pessoas tem apenas um leve declínio nessas funções.

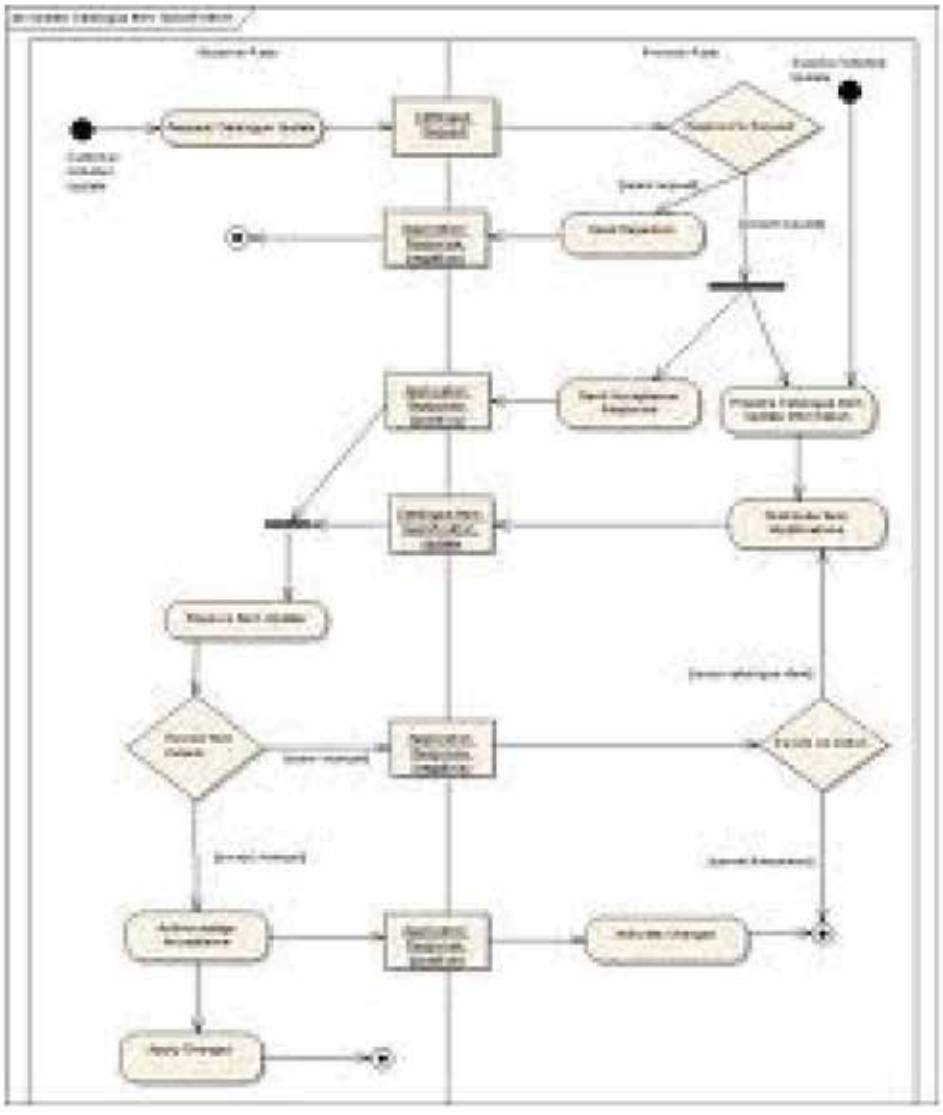
Vale a pena notar que esse é um mercado em franca acensão e que aplicações que entenderem melhor as necessidades especiais de interação desse grupo terão mais sucesso.

7. Linhas de pesquisa em IHC

Como discutido brevemente, a computação oferece ao psicólogos a oportunidade de entender processos cognitivos e estruturas de memória em humanos. Por outro lado, a psicologia pode afetar enormemente o processo de construção de novas tecnologias. Pesquisadores de diversas áreas do conhecimento vêm contribuindo e usufruindo dos avanços da área de IHC. Entretanto, não são somente os psicólogos que exercem e sofrem influência nessa área multi-disciplinar. Identificamos também a colaboração de cientistas da informação, de negócios, educadores, antropólogos e sociólogos. Várias linhas de pesquisa nasceram a partir dessa colaboração. A seguir discutimos uma série de linhas de pesquisa na área IHC. Em tempo, ressaltamos a lista não é completa nem exaustiva.

7.1. Especificação e implementação da interação

As ferramentas modernas de construção de interface são de extrema utilidade e aumentam sobremaneira a produtividade. A questão é que a utilidade delas é limitada a seqüências de ações bem triviais, como montagem de menus, telas, e outros widgets. Acontece que várias técnicas de elicitação de requisitos em voga nos dias de hoje apregoam uma participação maior dos usuários na definição de como deve ser sua interação com o sistema. Além disso, novos dispositivos e formas de interação se tornam disponíveis, o que corrobora com a necessidade de atualização constante nos métodos de especificação e implementação da interação.



7.2. Manipulação Direta

São interfaces visuais nas quais os usuários operam sobre uma representação de objetos de interesse. Estudos empíricos podem ajudar no nosso entendimento sobre qual a representação analógica ou metafórica a ser implementada.

Novas formas de manipulação direta compreendem: linguagens visuais, visualização espacial, controle remoto, telepresença, realidade virtual e realidade aumentada.

7.3. Dispositivos de entrada e apresentação de dados

Esses incluem telas de alta resolução sensíveis ao toque, "stylus", canetas, voz, gestos, mouse, luvas e joysticks.

A escolha do dispositivo de entrada depende muito da tarefa a ser executada e critérios de avaliação envolvem velocidade, acurácia, fadiga, facilidade de correção de erros e satisfação subjetiva. Esses critérios devem ser mensurados a partir de experimentação extensiva envolvendo usuários representativos da comunidade de usuários-alvo realizando tarefas semelhantes ou, se possível, iguais àquelas que serão realizadas no dia-a-dia de uso do sistema. Métodos quantitativos devem ser empregados para a certificação de que o emprego de determinado dispositivo propicia ganhos de performance aos usuários.

7.4. Assistência online

Um dos grandes desafios para a área de IHC é a construção de sistemas interativos que propiciem o treinamento "just-in-time", ou seja, que permita aos usuários o treinamento durante o uso. Em outras palavras, permitir que os usuários aprendam a utilizar o sistema a medida em que vai se utilizando dele. A grande dificuldade desse tipo de treinamento está no fato de que os usuários vão adquirindo habilidades à medida que usam o sistema, vão deixando de ser novatos e passando a ser especialistas. Essa mudança faz com que os requisitos para interação se modifiquem, pois os usuários vão conseguindo realizar tarefas mais rapidamente, ficando assim menos dispostos a navegar por menus, preferindo, por exemplo, atalhos. Passada a fase inicial de aprendizado, esses usuários prestam mais atenção em detalhes de operação do sistema e adotam uma postura mais inquisitiva. O sistema deve, portanto, auxilia-lo nessa exploração. Tal exploração não está limitada à busca de mais performance na execução de tarefas realizadas diariamente, ela envolve também a descoberta de novas funcionalidades. Em um treinamento formal, os usuários são "bombardeados" com muitas informações acerca do funcionamento do sistema, e o nível de assimilação é relativamente baixo, pois eles ainda não estão prontos para se apropriar daquele conhecimento operacional mais sofisticado. Eles normalmente se atém às rotinas mais básicas que serão utilizadas no apoio à tarefas diárias.

7.5. Exploração da informação

Com a disponibilização crescente de conteúdo multimedia quer seja na internet, que seja em bases de dados corporativas ou científicas, aumenta a demanda por ferramentas e estratégias que permitam aos usuários filtrar, se-

lecionar, re-estruturar suas informações rapidamente sem medo de desorientação ou de ficarem perdidos. A "information visualization" ou simplesmente "infovis" é o nome da sub-área de IHC que estuda essas ferramentas e técnicas. Entre as técnicas preferidas estão: Grandes telas, apontamento direto, grafos, caledogramas, dendogramas e mapas de temperatura.

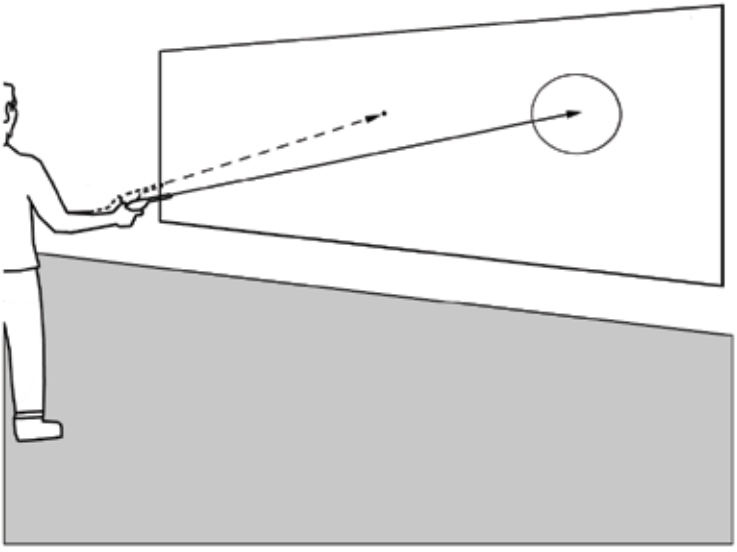
7.5.1. Grandes telas

Nós humanos temos grande capacidade de indexar informações espacialmente. Essa indexação nos é benéfica, pois é uma forma de aumentarmos nossa memória. Podemos não saber o detalhe da informação, mas sabemos onde encontra-lo. Assim, grandes telas aumentam a nossa área de trabalho, podendo diminuir as demandas cognitivas das tarefas já que podemos "espalhar" as informações necessárias às tarefas na área disponível na tela.

7.5.2. Apontamento direto

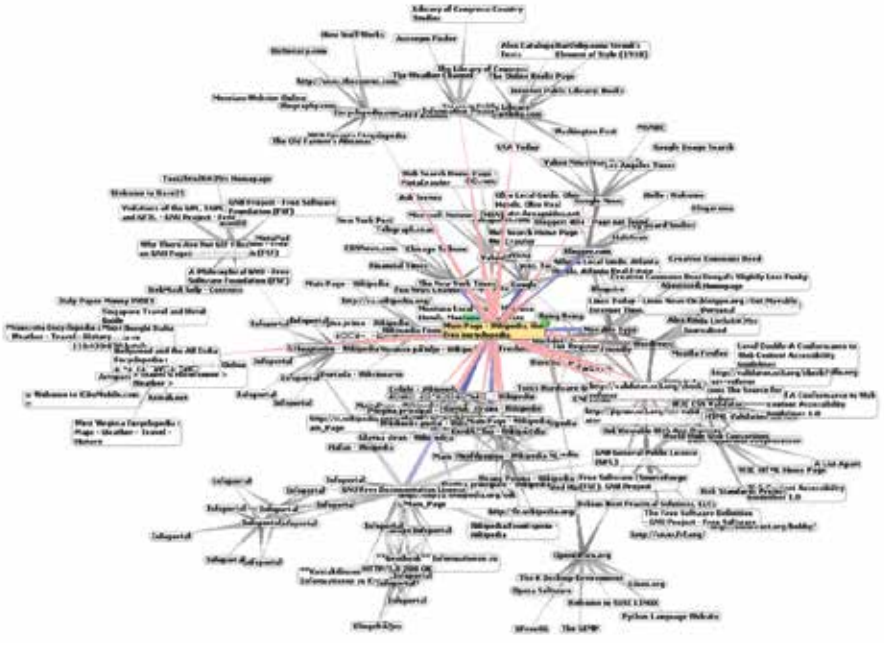
Desenhos e linguagem são matérias-primas do pensamento. Muitas vezes utilizamos o quadro para desenhar e com isso ganhamos tempo e agilidade na resolução de problemas. Isso deve-se ao fato de que nós utilizamos meio ambiente para "descarregar" conteúdo cognitivo – o que temos em mente. Um exemplo simples é o fato de ser muito mais fácil fazermos contas aritméticas com o auxilio do lápis e papel do que somente de cabeça. Voltando ao processo de resolução de problemas com o uso de desenho, faz sentido apontar para os desenhos que fazemos, raciocinar com eles com se fossem peças de um quebra-cabeças e construir frases que somente nós entendemos, como : "Se eu fizer primeiro isso <<aponta para *isso*>>, e depois aquilo outro <<apon-

ta para *aquilo>>,* acho que pode dar certo". O ato de apontar ajuda na fusão dos componentes visuais e lingüísticos de nosso raciocínio facilitando o processo de sua consolidação.



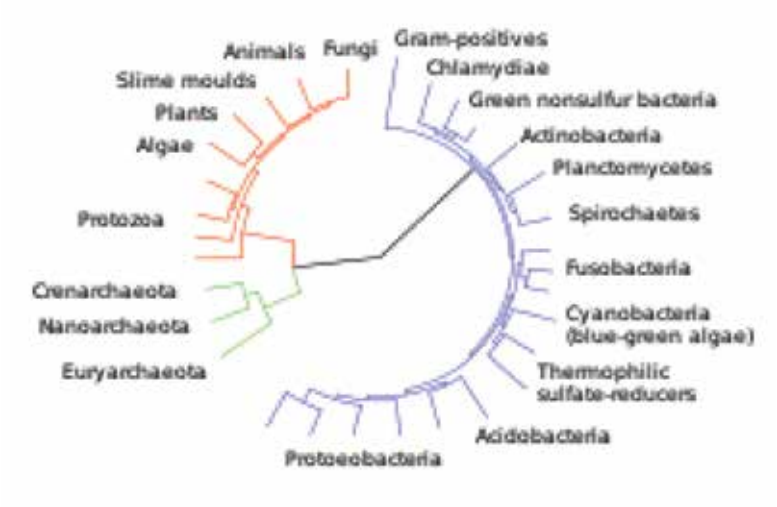
7.5.3. Grafos

Grafos (figura abaixo) são uma técnica de visualização de dados muito utilizada na análise de redes socais, em cartografia e na área de bioinformática



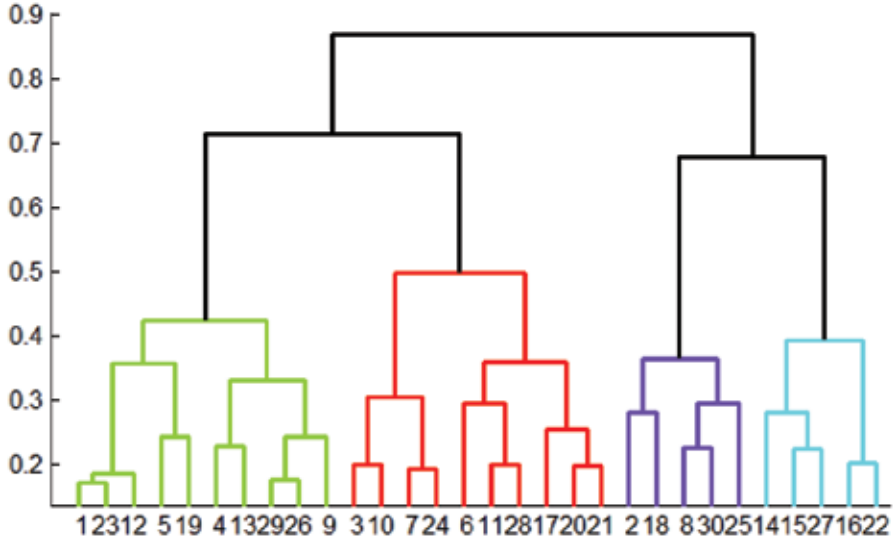
7.5.4. Caledogramas

Caledogramas (figura abaixo) são bastante utilizados como método de classificação de espécies de organismos, o qual consiste em agrupar organismo ancestral, todos os seus descendentes e nada mais. Obviamente a técnica pode ser utilizada para outros tipos de agrupamentos.



7.5.5. Dendogramas

Dendogramas são diagramas em formato de árvore frequentemente usados para ilustrar agrupamentos hierárquicos.



7.5.6. Mapas de temperatura

Os mapas de temperatura (figura abaixo) são representações de dados exibidas em formato de tabela onde uma categoria (um tipo de dado) é exibido como cor. Quanto mais alto o valor daquela variável, mais forte a cor. Mapas de temperatura na *web* podem mostrar quais são os sites mais *quentes*, mais freqüentados, com o maior número de acessos.

Atividades de avaliação

1. Você concorda com a frase de Larry Tesler ("A interface é o sistema")? Rediga uma pequena redação de 10 linhas a respeito do tema.

2. Como a área de IHC influencia a psicologia?

3. O que a usabilidade de equipamentos de guerra tem a ensinar para a usabilidade de computadores e sistemas que utilizamos em escritórios?

4. Debata o tema confiabilidade. Por que a confiabilidade é tão importante para área de IHC?

5. Por que a "computação nas nuvens" atende a um anseio da área de IHC?

6. Em sua opinião, o IPad traz uma satisfação subjetiva maior que um outro Tablet? Justifique sua resposta.

Capítulo 2

Teorias e Métodos

Objetivos

- Neste capítulo apresentamos teorias de alto nível que fornecem subsídio geral para o projetista de sistemas interativos. Em seguida, introduzimos o modelo Objeto-Ação proposto do Shneiderman (Shneiderman 2005) que preconiza o entendimento, em detalhes, da tarefa e o reconhecimento do papel desempenhado por todos os objetos utilizados na sua execução. Em seguida discutimos o impacto de fatores como a frequencia de uso, perfis de tarefas e estilos de interação sobre o projeto de interface de sistemas interativos. Depois apresentamos dicas práticas para a construção de interfaces de qualidade. Encerramos a unidade com um breve debate acerca de quais tipos de tarefas sistemas automatizados levam vantagens sobre o operador humano (trabalho manual), e em quais o ser humano tem desempenho superior a sistemas automatizados.

1. Introdução

Um bom projetista não pode se confiar apenas em julgamentos intuitivos. Nesse capítulo apresentaremos técnicas que nos dão direcionamento tanto de alto nível na forma de teorias e modelos, como princípios de nível médio e dicas práticas.

2. Teorias de alto nível

2.1. Introdução

Diversas teorias são empregadas no projeto de sistemas interativos. Existem as teorias exploratórias, que nos ajudam a observar o comportamento, descrever atividade e a comparar conceitos de alto nível entre dois projetos e treinamento. Já as teorias preditivas ajudam os projetistas a comparar projetos no que diz respeito ao tempo de execução de determinada tarefa, taxas de erro, etc. Algumas teorias focam nas atividades cognitivas ou perceptivas como tempo para encontrar um determinado item em uma tela, ou tempo para converter um caractere de itálico para negrito. Teorias relacionadas à predição de tarefas motoras são as mais bem estabelecidas e podem prever com um bom grau de exatidão o tempo em que o usuário navegará até determinado

item da tela. Teorias preceptivas tem sido usadas com sucesso na predição dos tempos gastos por usuário na leitura de textos livres, textos em listas, e em telas. Atividades cognitivas complexas normalmente envolvem várias subtarefas o que torna a tarefa de predição bastante complicada. O tempo de uso do sistema também pode acarretar em grandes diferenças de performance. Um usuário novato pode levar até 100 vezes mais tempo para completar uma tarefa do que alguém experimentado.

Atualmente existem centenas de teorias na área de IHC. Muitas delas ainda estão passando por um processo de amadurecimento tanto por aqueles que as propuseram como por aqueles que acham que podem melhorar o que já está posto. Isso significa que o campo ainda não está maduro e que devemos esperar novas e talvez radicais mudanças pela frente. A seguir discutiremos algumas dessas teorias.

2.2. O Modelo de Foley

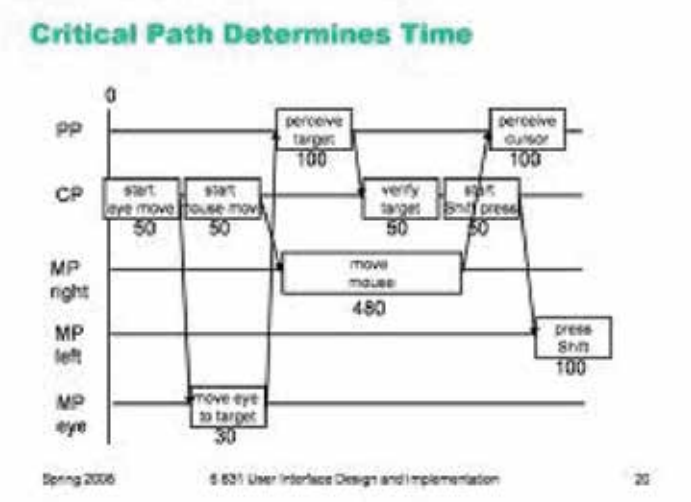
Proposto por Foley(Foley et al., 1987), o modelo prevê uma abordagem "topdown" para o desenvolvimento de sistemas interativos em quatro níveis: Conceitual, semântico, sintático e léxico.

O nível conceitual reflete o modelo mental do usuário do sistema interativo. Como exemplo de modelo mental, citamos o da transação bancária, onde tanto o funcionário do banco como o cliente sabem que uma transação envolve ações que devem ser feitas em conjunto. Um saque deve compreender consulta a saldo, entrega do numerário (se saldo for suficiente) e atualização do saldo. É no nível semântico que definimos os significados dos termos saldo, numerário, etc. No nível sintático definimos como utilizamos os termos definidos no nível acima para realizar determinada tarefa definida no domínio de aplicação. Em nosso exemplo como consulta saldo, libera numerário e atualiza saldo são utilizados em conjunto para realizar a tarefa efetuar saque. Finalmente, no nível léxico definimos como mecanismos intrínsecos aos dispositivos de entrada e apresentação de dados serão utilizados. Por exemplo, para a operação saque realizada em um caixa eletrônico, o cliente usará uma maquina de auto-atendimento que dispõe de botões, normalmente um para cada opção, mas não tem mouse. Já a mesma aplicação sendo implementada em um micro computador, o projetista poderá contar com o mouse, com a tecla <<tab>> para facilitar a navegação do usuário entre um campo e outro.

A conveniência do método está em sua característica "topdown" que propicia modularidade ao projeto. Projetistas devem começar trabalhando na camada conceitual e ir mapeando todas as transições entre os níveis.

2.3. Os Modelos GOMS e *Keystroke*

Os dois modelos foram propostos por Card, Moran e Newell (Card et al., 1983). O acrônimo GOMS vem das palavras inglesas (Goals, objetivos; Operators, operadores; Methods, Métodos e Selection Rules, regras de seleção). Para os autores, usuários tem objetivos (editar um documento) e sub-objetivos (inserir uma palavra). Os objetivos são atingidos por meio da utilização de métodos, como mover o cursor para a posição desejada. Já os operadores são "*atos cognitivos, motores, ou perceptivos elementares cuja execução é necessária para mudar em qualquer aspecto o estado mental do usuário ou afetar o ambiente de tarefa*" (Card et al., 1983). São exemplos de operadores: Pressionar uma tecla, mover a mão em direção ao mouse, lembrar-se do nome de um arquivo, verificar a posição do cursor. As regras de seleção são estruturas de controle utilizadas na escolha entre os diversos métodos disponíveis para se realizar a mesma tarefa. A figura abaixo mostra os tempos gastos por um usuário durante a execução de cada operador. Como o usuário pode escolher diferentes caminhos para alcançar seu objetivo, o pior caso, aquele em que ele demoraria mais tempo, determina o tempo que é levado em consideração.



Já o modelo de nível de keystroke tenta predizer tempos de execução sem erros de tarefas por parte de usuários experientes somando os tempos que eles levam em subtarefas como: digitação, clique do mouse, achar onde o cursor do mouse se encontra, desenhar, raciocinar e esperar pelo tempo de resposta do sistema. Esse modelos estão focados na execução de tarefas realizadas por usuários experientes e não se preocupam com o processo de aprendizado, resolução de problemas, recuperação de erros, satisfação subjetiva ou retenção.

Uma alternativa aos dois métodos descritos acima é o emprego de diagramas de transição (figura abaixo). Esses diagramas são velhos conhecidos

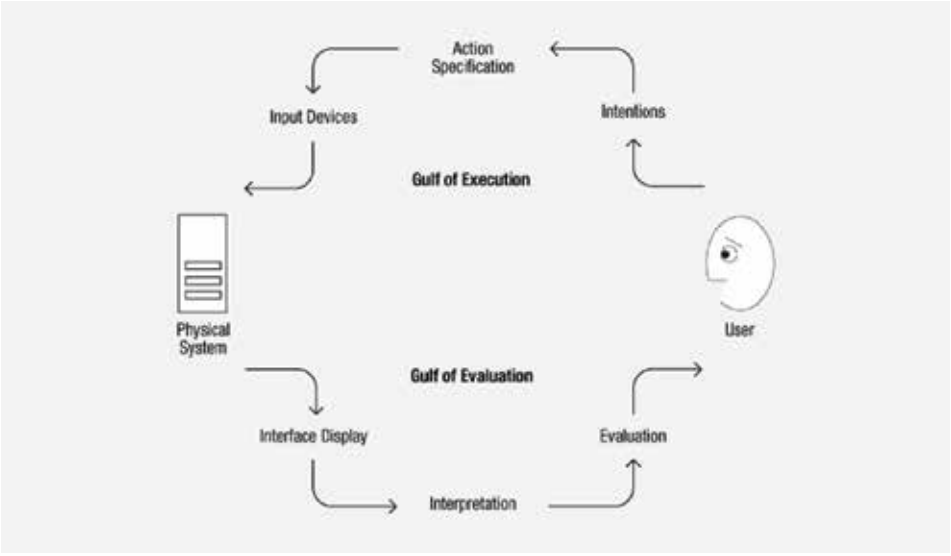
engenharia de software e estão presentes na UML (Unified Modeling Language). Eles são úteis tanto na fase de projeto, como na de instrução, além de preditores de tempo de aprendizado, execução e de erros.

2.4. Modelos de estágios de ação

Norman descreve *sete estágios de ação* (Card et al., 1983) como modelo de interação humano-computador. São eles:

1. Formular um objetivo;
2. Formular uma intenção;
3. Especificar uma ação;
4. Executar uma ação;
5. Perceber o estado do sistema;
6. Interpretar o estado do sistema;
7. Avaliar o resultado.

Quando colocamos em seqüência temos: o usuário formula uma intenção conceitual, reformula-a em termos das semânticas de diversos comandos, constrói a sintaxe requerida, finalmente produz ação de mover o mouse para selecionar um ponto na tela. Esses estágios de ação constituem o que Norman chamou de *ciclos de ação* e *avaliação*. Foi também a partir da proposição desses estágios que se identificou o *golfo da execução,* o descompasso entre as intenções do usuário e as ações permitidas pelo sistema. Em contrapartida, o *golfo de avaliação* corresponde ao descompasso entre a representação do sistema e as expectativas do usuário. A figura abaixo ilustra os *golfos de execução e de avaliação.*



A partir desse modelo, Norman sugere quatro princípios para um bom projeto. Primeiro, o estado do sistema e as alternativas de ação devem estar sempre visíveis. Segundo, deve haver um bom modelo conceitual consistente com a imagem do sistema. Terceiro, a interface deve incluir bons mapeamentos que revelem as relações entre os estágios. Quarto, o usuário deve receber *feedback* contínuo. Norman enfatiza muito o estudo dos erros que, segundo ele, ocorrem nas transições entre objetivos e intenções, intenções e ações, ações e execuções.

O modelo ajuda descrever o processo de exploração de uma interface realizado pelo usuário. A medida em que os usuários tentam atingir seus objetivos, eles estão sujeitos a quatro pontos críticos: 1) Eles podem formular objetivos inadequados, 2) Podem não encontrar o objeto de interface correto devido a um *label* ou ícone que não pôde ser compreendido, 3) podem não saber como executar determinada ação e 4) podem receber *feedback* inapropriado ou de difícil entendimento.

2.5. Teorias voltadas para o uso de *Widgets*

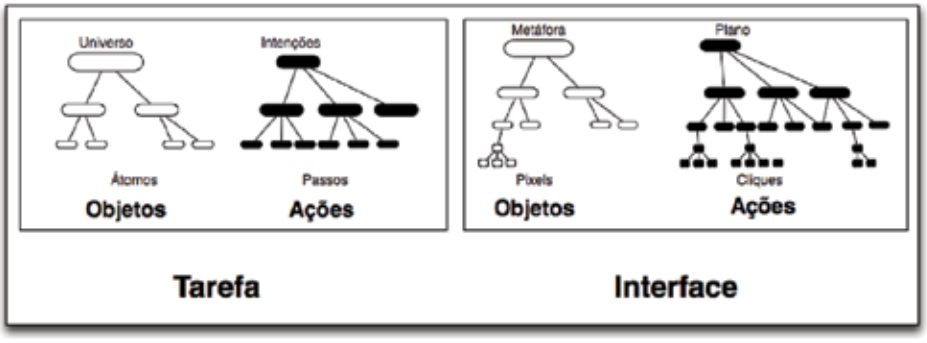
A maioria dos programas de computador são baseados em *Widgets:* Labels, campos, caixa de escolha, *radio bottoms, check boxes,* etc (figura ao lado). A maioria dos ambientes de programação também possuem o que chamamos de *pintores de tela*, ou seja, os programadores constroem as telas de seus programas simplesmente abrindo sua "caixa de ferramentas", onde ficam guardados todos modelos de *widgets,* arrasta um que queira acrescentar à sua tela, criada originalmente em branco, e solta-o.

Vê-se, portanto, que o uso dos widgets oferece uma oportunidade de medir a qualidade da interface. Cada um tipo (ou classe) de widget tem sua própria complexidade de uso. É mais fácil clicar em um *checkbox,* do que

percorrer uma lista de valores e escolher aquele que nos interessa. Assim, uma interface que tenha mais widgets mais "caros" de se operar, será forçosamente uma interface que levará a uma pior performance do usuário, produzirá fadiga mais cedo, e inclusive pode aumentar a taxa de erros durante a sua operação. Também deve-se considerar que a seqüência na qual esses objetos aparecem na tela tem bastante influência nas questões discutidas acima. A seqüência precisa ser lógica e ter coerência com outros objetos, principalmente objetos reais. Por exemplo, tela formulário deve apresentar a mesma seqüência de campos que a correspondente em papel.

3. O Modelo de Objeto-Ação

O método, proposto do Shneiderman (Shneiderman; Plaisant, 2005), começa pelo entendimento da tarefa. A tarefa inclui o universo de objetos do mundo real com os quais o usuário trabalha para alcançar seu objetivos e as ações que ele realiza sobre esses objetos. Objetos de alto nível de uma tarefa podem ser uma biblioteca de fotos, estatísticas do mercado de ações, ou mesmo contatos de uma aplicação de rede social (ver figura abaixo). Esses objetos podem ser decompostos em informações a respeito de uma única ação do mercado de ações, em um nível menor, da cotação do momento daquela ação. As ações da tarefa começam a partir das intenções de alto nível que são decompostas em objetivos individuais e em seguida, em passos individuais.



Depois da identificação dos objetos de tarefa, suas ações e como elas podem ser decompostas, o projetista pode criar representações metafóricas dos objetos de interface e suas ações. Objetos de interface são conjuntos de pixels que podem ser copiados ou modificados de tal forma que venham a representar objetos do mundo real da tarefa e que respondam às ações dos usuários como forma de guia-los na execução das tarefas. Em seguida, o projetista deve fazer com que as ações dos objetos de interface se tornem visíveis aos usuários de tal forma que eles seja capazes de decompor seu plano em uma série de ações intermediárias como abrir uma caixa de diálogo, clicar determinado botão, etc.

Trata-se de um modelo exploratório que foca tanto nos objetos da tarefa e suas ações como nos objetos de interface e suas ações. Como os detalhes de sintaxe são mínimos, usuários que conhecem os objetos de domínio da tarefa podem aprender a utilizar a interface com relativa facilidade. Tarefas incluem hierarquias de objetos e ações de alto e baixo nível. Hierarquias não são perfeitas, mas são compreensíveis, úteis e largamente aceitas pelos usuários.

4. Freqüência de uso, perfis de tarefas e estilos de interação

"Conhecei vosso usuário", Hansen (Hansen, 1971). A idéia simples, mas de difícil implementação, além de ser subvalorizada. Muitos projetistas acham que conhecem seus usuários. Projetistas de sucesso entendem que pessoas pensam, aprendem e resolvem problemas de formas diferentes. Todo projeto deveria começar com o conhecimento da comunidade de usuários, incluindo perfis populacionais que incluem idade, sexo, habilidades físicas, educação, cultura, etnia, treinamento, motivação, objetivos e personalidade. Além da diversidade humana, temos as diversidades de situação, de tarefas, e de freqüência de uso, e impacto do erro, o que torna ainda maior o desafio para o projetista de sistemas interativos.

A seguir discutiremos os diferentes perfis de usuários conforme a freqüência de uso, perfis de tarefa e estilos de interação.

4.1. Usuários segundo sua freqüência de uso

4.1.1. Usuário novato

Na classe de usuários novatos podemos identificar dois tipos de usuários: os verdadeiramente novatos e aqueles que estão usando o sistema pela primeira vez. A diferença é que o segundo grupo é formado por especialistas no negócio e estão apenas começando a ter contato com o sistema. O objetivo do projetista é diminuir o nível de ansiedade desses usuários o que pode acarretar dificuldades no aprendizado. Uma das primeiras providencias é utilizar vocabulário do domínio da aplicação. Reduzir o número de ações para a realização de tarefas para que o usuário as execute e com isso reduza ansiedade, ganhe confiança, e obtenha reforço positivo. Feedback informativo acerca da realização das tarefas ("arquivo salvo com sucesso") e mensagens de erro construtivas devem ser emitidas sempre que um erro for cometido. Tutoriais online que descrevem passo a passo como determinada tarefa é alcançada também são importantes.

4.1.2. Usuário intermitente

Já os usuários intermitentes com algum conhecimento são aqueles que tem contato com diversos sistemas, mas de forma intermitente. Conhecem a tarefa que querem desempenhar, mas podem não lembram exatamente onde estão as opções do menus. Menus bem estruturados, uso consistente de terminologia, seqüência de ações consistente, mensagens de fácil entendimento e guias para padrões frequentes de uso podem ajudar esse tipo de usuário na redescoberta da seqüência de ações necessárias para a realização da tarefa intencionada.

4.1.3. Usuário experiente

Os usuários experientes já são familiarizados tanto com o domínio da aplicação como com a interface. Eles buscam performance, querem realizar suas atividades o mais rápido possível. Eles esperam curtos tempos de resposta, feedbacks breves e que não distraiam, e a capacidade de realizar o que desejam teclando e/ou selecionando (clicando) pouco. Esses usuários estão sempre à procura de construção de macros ou outros mecanismos que diminuam o número de passos para realizar uma tarefa.

Construir sistemas que acomodem essas três categorias não é simples. Muitas vezes é preciso fazer refinamentos sucessivos da interface, sem perder de vista os elementos da tarefa. Uma das estratégias é a de propiciar uma estratégia de aprendizado estruturada em níveis. Os novatos podem aprender

um subconjunto de ações mínimo necessário para que desempenhem suas tarefas mais elementares. Com poucas opções, a possibilidade de erro diminui. Depois de adquirirem confiança advinda do uso prático do sistema, esses usuários podem ser apresentados a mais opções. Outra estratégia é permitir que usuário defina o nível de detalhes de feedback do sistema.

4.2. Perfis de tarefa

Todo projetista deve concordar que o conjunto de tarefas deve ser identificado antes que o projeto possa prosseguir. O problema é que a análise de tarefa, ou é feita informalmente, ou é não é feita. Ações de tarefa de alto nível podem ser decompostas em múltiplas ações de tarefa de nível intermediário, que por sua vez, podem ser refinadas em ações atômicas que o usuário executa com um único comando ou seleção de menu. Definir o conjunto apropriado de ações atômicas é uma tarefa complicada. Se as ações atômicas forem muito pequenas, os usuários podem se frustrar com o número de ações necessárias para realizar uma ação de alto nível.

A freqüência com que cada ação é executada pode ajudar o projetista a estruturar as tarefas dentro do sistema. Tarefas realizadas com mais freqüência devem ser executadas de forma simples e rápida mesmo que isso vem acarretar o atraso na execução de tarefas não tão frequentes. Tarefas frequentes podem ser disparadas a partir de uma única tecla ou combinações de teclas. Abaixo um exemplo de tabela freqüência de atividades por tipo de cargo em um sistema de controle de pacientes internados.

Cargo	Atividade		
	Pesquisa paciente	Atualiza prontuário	Avalia sistema
Enfermeiro	45%	55%	0%
Médico	60%	40%	0%
Supervisor	20%	0%	0%
Chefe de enfermagem	52%	40%	0%
Contas médicas	65%	0%	0%
Analista de Sistemas	0%	0%	50%

4.3. Estilos de Interação

Quando a análise de tarefas estiver completa e os objetos e ações tiverem sido identificados, o projetista pode escolher entre os estilos primários de interação: Manipulação direta, menus, preenchimento do formulários, linguagem de comandos ou ainda, linguagem natural.

A *manipulação direta* ocorre quando o projetista consegue criar uma representação visual do mundo onde a ação ocorre. Quando isso é possí-

vel, as tarefas dos usuários ficam bastante simplificadas. Exemplos de tais sistemas são vídeo games e sistemas de controle de tráfego aéreo, onde os usuários interagem diretamente com os elementos visuais que representam, nos exemplos, uma personagem ou uma aeronave.

Em sistemas de *menus*, os usuários lêem uma lista de itens, seleciona aquele que julga mais apropriado para a sua tarefa e observa o efeito de sua ação. Se a terminologia e significado dos itens forem facilmente inteligíveis e distintos, os usuários podem alcançar seus objetivos com menos esforço e terá menos o que memorizar. O maior benefício da abordagem é que existe uma clara estrutura que serve para ajudar no processo de escolha do usuário.

Para entrada de dados, o ideal é o uso de *formulários*. Nesse tipo de interação o usuário deve entender os *labels,* que tipos de valores podem ser inseridos, o método de entrada da dados, e saber responder às eventuais mensagens de erro.

Para usuários frequentes, as *linguagens de comando* oferecem a forte sensação de locus de controle e iniciativa. Uma vez que a linguagem fora dominada, os usuários podem expressar possibilidades complexas rapidamente, sem terem de ler menus. Entretanto, nessa modalidade de interação, a freqüência de erros é maior, treinamento se faz indispensável, retenção mais difícil, e assistência on-line mais difícil.

Muitos pesquisadores buscam o desenvolvimento de tecnologias que propiciem aos computadores a interação com humanos através de linguagem natural. Interfaces de linguagem natural normalmente fornecem pouco contexto para o comando seguinte e frequentemente requer diálogos de clarificação.

A seguir, veja a tabela comparativa entre os métodos primários de interação.

Estilo	Vantagens	Desvantagens
Manipulação Direta	Apresenta visualmente os conceitos da tarefa	Difícil de se definir a metáfora correta
	Aprendizado fácil	Difícil de se programar
	Fácil retenção (fácil do usuário lembrar)	
	Evita erros	
	Encoraja exploração	
	Promove satisfação subjetiva	
Menus	Diminui o tempo de aprendizado	Muitos menus pode ser prejudicial
	Reduz o numero de teclas a serem digitadas	Diminuem a velocidade de operação dos usuários experientes

Estilo	Vantagens	Desvantagens
	Estruturam o processo de decisão	Consumem espaço de tela
	Permitem suporte ao erro	
Preenchimento de formulários	Simplificam a entrada de dados	Consumem espaço de tela
	Requerem pouco treinamento	
	São convenientes para oferecer ajuda sensível ao contexto aos usuários	
Linguagem de comandos	São flexíveis	Manipulação de erros fraca
	Usuários experientes gostam, pois dessa forma normalmente é possível interagir com o sistema em alta velocidade	Requer mais treinamento e memorização do comandos
	Apóia a iniciativa do usuário	
	Permite a criação de macros	
Linguagem natural	Alivia o usuário da necessidade de aprendizado de sintaxe linguagem específica do sistema em questão	Requer construção de diálogos de clarificação.
		É de difícil contextualização

5. As Oito Regras de Ouro do Projeto de Interface

Apresentamos a seguir as oito regras de outro do projeto de interface propostas por Shneiderman (Shneiderman; Plaisant, 2005).

5.1. Regra 1: Mantenha a consistência

Essa é uma regra simples, mas uma das mais comumente violadas. Tarefas semelhantes devem ser realizadas a partir de ações semelhantes, encontradas na mesma região do sistema. Terminologia idêntica deve ser utilizada em menus, listas, telas de ajuda. É preciso o uso consistente de cores, de leiaute, de letras maiúsculas e fontes.

5.2. Regra 2: Permita que usuários frequentes se utilizem de atalhos

A medida que vão se tornando confiantes com o uso do sistema, usuários procuram formas de realizar as mesmas ações com um numero menor de interações. Abreviações, teclas especiais (teclas de função) e macros são apreciadas por esses tipos de usuários. Baixo tempo de resposta e atualização rápida da tela também são valorizadas.

Tables	
Action	**Keystroke**
Go to next cell	Tab
Go to previous cell	SHIFT+Tab
Go to beginning of column	ALT+PageUp
Highlight to beginning of column	ALT+SHIFT+PageUp
Go to end of column	ALT+PageDown
Highlight to end of column	ALT+SHIFT+PageDown
Go to beginning of row	ALT+Home
Highlight to beginning of row	ALT+SHIFT+Home
Go to end of row	ALT+End
Highlight to end of row	ALT+SHIFT+End
Column break	CTRL+SHIFT+Enter

5.3. Regra 3: Ofereça feedback informativo

O sistema deve responder a toda e qualquer ação do usuário. Respostas curtas são indicadas para ações simples e corriqueiras, enquanto ações mais relevantes demandas respostas mais elaboradas.

Representações visuais dos objetos de interesse fornecem ambientes convenientes para se mostrar as mudanças de forma mais explícita.

5.4. Regra 4: Projete diálogos auto-contidos

Seqüências de ações devem ser organizadas em grupos com começo, meio e fim. Feedback informativo deve ser fornecido ao usuário cada vez que uma fase for completada, o que deve propiciar ao usuário uma sensação de realização, de alívio e indicação de que ele está no caminho certo.

5.5. Regra 5: Elabore estratégias para a prevenir erros e facilitar sua recuperação

Usuários de sistemas interativos cometem muito mais erros do que se imagina. Usuários experientes de sistemas operacionais e editores de texto cometem erros ou usam estratégias ineficientes em 31% do tempo. A taxa de erros de usuários experientes de planilha eletrônica chegam a 50%. Isso acarreta grande perda de produtividade.

Melhorar a qualidade das mensagens de erro pode ser uma estratégia. Mensagens devem ser mais especificas, em tom positivo e construtivas – dizer o que o usuário deve fazer em vez de somente reportar o erro. Mas o importante mesmo é evitar que os erros ocorram.

Devemos entender a natureza dos erros. Segundo Norman (Norman, 1983), projetistas podem ajudar a evitar os deslizes dos usuários organizando telas e menus de forma que as opções sejam bem distintas, o que torna mais

difícil o usuário escolher a opção errada. Aqui também se advoga a necessidade de se projetar sistemas que permitam a reversão das ações.

Uma das formas de se prevenir erros é o projeto seqüências de ações completas. Por exemplo, em sistemas que exigem a conexão com outros sistemas, todo o processo de conexão pode ser automatizado. Para os usuários de processadores de texto, podemos ter *templates* que garantam que o texto fique formatado adequadamente. Prefira uma seleção de itens de menu a um preenchimento que exija digitação, não permita a digitação de caracteres alfanuméricos em campos numéricos. Ações erradas devem, se possível, não afetar o estado do sistema.

5.6. Regra 6: Permita a fácil reversão das ações

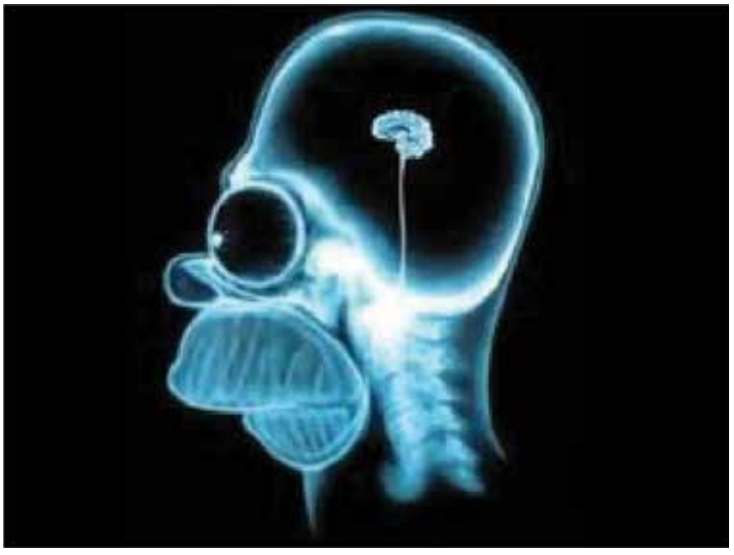
Sempre que possível, ações devem ser reversíveis (observar figura ao lado). Essa característica alivia a ansiedade uma vez que o usuário sabe que os erros podem ser fácil e rapidamente corrigidos. Isso o incentiva a explorar opções do sistema que ainda não lhes são familiares.

5.7. Regra 7: Apóie o locus interno de controle

Usuários experientes gostam da sensação de que estão no comando e que o sistema responde às suas ações. Respostas inesperadas do sistema, sequências tediosas de entrada de dados, impossibilidade ou dificuldade em se obter uma determinada informação podem causar ansiedade e descontentamento.

5.8. Regra 8: Diminua a carga de memória a curto prazo

A limitação do processamento humano da informação em memória de curto prazo – a regra de que humanos se lembram de "sete mais ou menos dois" blocos de informação – requer que telas sejam mantidas simples, que seja dado tempo suficiente para que o usuário se acostume com telas e eventuais códigos, e seqüências de ações. Em relatórios com múltiplas páginas, um breve resumo da informação exibida deve ser oferecido ao usuário ao final de cada página. Sempre que apropriado, devemos oferecer ajuda ao operador a respeito da sintaxe correta de comandos, significado de códigos e campos.



6. Entrada e apresentação de dados

Atividades relativas a entrada e leitura de dados costumam a tomar grandes quantidades de tempo dos usuários de computador, principalmente nas empresas. Smith e Mosier (Smith et al., 1986) nos oferecem cinco objetivos de alto nível que devem ser observados em interfaces para entrada de dados. A lista é apresentada a seguir.

1. **Consistência nas transações de entrada de dados.** Esse preceito defende a consistência de seqüências de ações em todas as circunstâncias. De uma forma mais prática, podemos pensar, por exemplo, que todos os formulários de entrada devem possuir botões que propiciem ações como *limpar formulário, salvar formulário,* etc. Esses botões devem ter a mesma funcionalidade para qualquer formulário.
2. **Minimizar as ações do usuário.** Quanto menos ações o sistema exigir de seu usuário, menos chances esses terão de errar. Utilizar objetos como *listas, radio buttons* dentre outros é, a princípio, mais vantajoso do que permitir longas seqüências de digitação que têm o potencial de gerar erros. Entretanto, o projetista não pode perder de vista o efeito *homing* que é o tempo gasto pelo usuário em tirar sua mão do teclado, procurar o mouse e navegar com ele até o ponto na tela onde se encontra o objeto desejado. Usuários experientes preferem digitar de seis a oito caracteres em vez de ter de fazer o *homing.* Outra observação é a de que devemos evitar a entrada de dados redundante, o que é altamente irritante para o usuário.
3. **Minimizar a carga de memória.** Nesse ponto, a sugestão é de que não devemos exigir do usuário a memorização listas de códigos nem de sintaxe de comando complexas.

4. **Compatibilidade entre a entrada e a apresentação de dados.** A disposição dos campos em uma tela de entrada de dados deve ser a mais semelhante possível àquela de apresentação de dados. Alguns sistemas implementam exatamente a mesma tela tanto para a entrada como para a apresentação de dados, a diferença está no *estado* do formulário, ora ele está no modo de entrada de dados, ora no estado de apresentação.
5. **Flexibilidade para o controle do usuário sobre a entrada de dados.** É importante que o projetista entenda que os dados a serem inseridos no sistema fazem parte de um contexto de trabalho que já existia e vigia antes da existência do sistema. Isso posto, é provável que os usuários já possuam uma seqüência em eles agrupam os dados mentalmente. Por exemplo, controladores de tráfego aéreo pensam sempre primeiro no horário de chegada dos vôos, já em outras ocasiões é a altitude que é mais valorizada por aqueles profissionais. Esses dados ditos "mais valorizados" devem aparecer primeiro nas telas de entradas de dados.

7. Obtendo a atenção do usuário

O usuário é normalmente "bombardeado" com muitas informações disponibilizadas por sistemas de computação. Torna-se portanto um desafio ao projetista desenhar sistemas que obtenham atenção imediata do operador quando este deva tomar ações em espaço de tempo muito curto. Nesse tipo de situação, o sistema deve ser projetado para atrair a atenção do usuário para aquela situação emergencial. A seguir listamos, baseado em Wickens (Hollands; Wickens, 1999), algumas técnicas existentes para se conseguir a atenção do operador:

- **Intensidade.** Utilize apenas dois níveis, reserve a alta intensidade para situações em que queria atrair a atenção do usuário.
- **Marcação.** Sublinhe, coloque a informação dentre de uma caixa, aponte para ela com uma seta, ou use um indicador como um asterisco, um bullet, um traço, um sinal de + ou mesmo um X.
- **Tamanho.** Use no máximo quatro tamanhos de fonte. Os tamanhos maiores atraem mais atenção.
- **Cores.** Use no máximo quatro padrões de cores, deixe outras cores reservadas para ocasiões onde se queira atrair a atenção
- **Cores piscando.** Troque de cores (objeto piscando, mudando de uma cor para outra) com muito cuidado e em áreas limitadas.
- **Audio.** Tons suaves devem se utilizador para feedback positivo enquanto que tons mais agudos servem para alertar os usuários em situações raras de emergência.

É importante ressaltar que há sempre o perigo de se construir telas poluídas com muita informação e, para contornar a situação, não se deve abusar das técnicas descritas acima. Usuários novatos precisam de telas simples, organizadas logicamente, com labels bem escolhidos para guiá-los em suas ações. Usuários experientes não precisam de muitos labels nos campos – aumento sutil de luminosidade ou representação posicional são suficientes.

8. Entre a automação e o controle humano

Apesar dos crescentes níveis de automação, graças principalmente à padronização de rotinas e tarefas nas empresas, o que facilita sobremaneira o projeto de sistemas de computação em alguns tipos de atividades, os humanos ainda levam vantagem. Abaixo, tabela comparativa, compilada a partir de Brown (Brown, 1998), que delineia melhor em que situações os humanos são melhores que as máquinas. É importante ressaltar, entretanto, que essa "fronteira" não é definitiva e que à medida que as tecnologias avançam, mais as máquinas irão se tornando "capazes" de exercer funções até antes restrita a humanos.

Humanos são geralmente melhores em	Máquinas são geralmente melhores em
Perceber estímulos de baixa intensidade	Perceber estímulos fora alcance humano
Detectar estímulo em ambiente de com muito ruído	Contar ou medir quantidades físicas
Perceber eventos inesperados	Armazenar quantidades de informação codificada
Lembrar de princípios e estratégias	Monitorar eventos pré-definidos, especialmente quando esses eventos ocorrem de forma esporádica
Relembrar de detalhes pertinentes aparentemente sem relacionamento com a situação atual	Responder rápida e consistentemente aos sinais de entrada
Recorrer à experiência durante o processo de tomada de decisão	Recuperar consistentemente informação detalhada
Recorrer a alternativas caso a primeira estratégia não se funcione	Processar dados quantitativos de maneira pré-definida
Raciocinar indutivamente: generalizar a partir de observações	Agir dedutivamente: Inferir a partir de um princípio geral
Agir diante de emergências não previstas ou diante de situação inusitada	Realizar ações repetitivas pré-programadas de forma confiável
Utilizar-se de princípios para resolver diversos tipos de problemas	Empregar grandes quantidades de força física de forma altamente controlada
Fazer avaliações subjetivas	Realizar várias tarefas simultaneamente
Propor novas soluções	Manter-se em atividade mesmo sob pesada carga informacional

Humanos são geralmente melhores em	Máquinas são geralmente melhores em
Concentrar-se nas tarefas importantes quando estiver sobrecarregado de trabalho	Manter-se em um bom nível de performance durante longos períodos
Adaptar resposta física quando exigido pela situação	

Atividades de avaliação

1. Elabore uma seqüência de ações, nos níveis conceitual, semântico, sintático e léxico (modelo de Foley) para a operação "Alugar filme na locadora"
2. Como o modelo GOMS pode predizer performance do usuário durante o uso de um programa de computador que não existe?
3. Como o modelo de Widgets pode pode predizer performance do usuário durante o uso de um programa de computador que não existe?
4. Explique porque é difícil construir uma aplicação que satisfaça tanto a usuários novatos como os experientes?
5. Explique porque é importante construirmos os perfis de tarefa
6. A tabela de Brown (entre a automação e o controle humano) pode já estar desatualizada. Cite pelo menos três exemplos de possíveis desatualizarão da tabela (O que já saiu da coluna "humanos são melhores…" para a coluna "maquinas são melhores")

Capítulo 3

Gerenciando os processos do Projeto

Objetivos

- Neste capítulo discutimos a importância do apoio organizacional ao desenvolvimento de sistemas baseados na engenharia da usabilidade. Em seguida, apresentamos métodos que descrevem como o processo de construção deve ser conduzido dentro das organizações. Discutimos técnicas consagradas na área como a observação etnográfica, o projeto participativo e o desenvolvimento baseado em cenários. Finalmente, abordamos os cuidados que o gerente desse tipo de projeto devem ter acerca de como os funcionários da organização que irá receber o sistema vêem a iniciativa.

1. Introdução

Atualmente a formação dos usuários está mais ligada ao fluxo de trabalho do que à tecnologia propriamente dita. Assim, projetos de sistemas interativos devem estar baseados em cuidadosas observações dos usuários no desempenho de suas tarefas, refinados por detalhadas análises de freqüência e seqüência de tarefas e validado por protótipos construídos desde os primeiros estágios e em seguida por testes de usabilidade e aceitação. Os projetos da atualidade devem acomodar as habilidades, objetivos e preferências dos usuários. Projetistas procuram interação direta com os usuários durante as fases de projeto, desenvolvimento e durante todo o ciclo de vida do sistema.

Em tudo mundo, a Engenharia da Usabilidade tem se firmado como uma disciplina com práticas e padrões bem estabelecidos. Gerentes de projetos são convidados a adaptar as práticas apresentadas nesse capítulo a fim de adequa-las a seus orçamentos, cronogramas e estrutura organizacional. Entretanto, é importante que as empresas dêem apoio organizacional à usabilidade. Esse ponto é importante uma vez que as empresas dão muito mais importância aos processos de engenharia de software que aos de usabilidade.

Também discutiremos os três pilares sobre os quais o processo de desenvolvimento de sistemas interativos se assenta: 1) Um documento de guidelines para o processo propriamente dito; 2) Ferramentas de software para o desenvolvimento de interface; 3) A revisão de especialistas e os testes de usabilidade.

2. Apoio organizacional à usabilidade

Quando os dois produtos possuem funcionalidades semelhantes, a da usabilidade passa a ser o principal critério de comparação. Assim aquele que tiver usabilidade superior tem vantagem. Cientes disso, muitas empresas desenvolvedoras de software criaram laboratórios de usabilidade que são utilizados para revisão de especialistas e na condução de testes de usabilidade.

Especialistas de fora do projeto podem fornecer observações importantes e enriquecedoras, enquanto os testes de usabilidade realizados nas principais tarefas do domínio de aplicação implementadas em um sistema de computação são muito importantes para o entendimento de performance dos futuros usuários.

Mesmo em empresas de pequeno porte é aconselhável a existência de um pequeno grupo especializado em técnicas de projeto e teste tanto de interface como de usabilidade.

Em trabalho na IBM, Karat (Karat, 1990) aponta ganhos de até cem dólares a cada dólar investido em usabilidade. Karat também identificou ganhos na redução do tempo de desenvolvimento do software, redução nos custos de manutenção, aumento de lucratividade devido a satisfação do cliente, além do aumento da produtividade e eficiência do usuário.

Para Shneiderman a atividade de projetar sistemas interativos é inerentemente criativa e imprevisível. Projetistas de sistemas interativos misturam conhecimento extenso sobre o que é possível ser desenvolvido com um "senso estético místico sobre o que atrai o usuário".

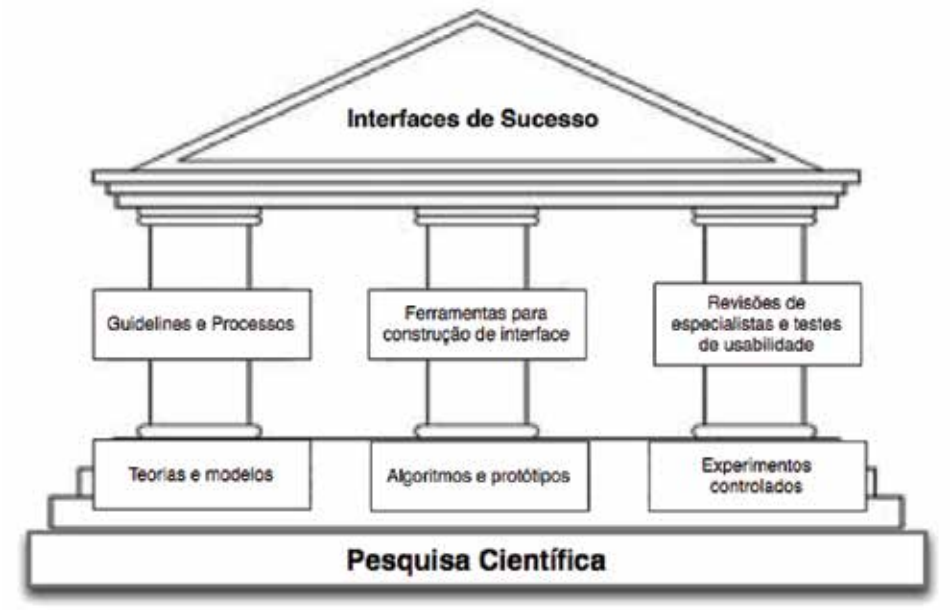
Carroll e Rosson (Carroll; Rosson, 1985) caracterizam o projeto como:

- um processo, ele não é um estado nem pode ser adequadamente representado estaticamente;
- "não-hierárquico", não é nem "top-down" nem "bottom-up";
- um processo radicalmente transformacional, envolve o desenvolvimento de soluções parciais e provisórias que podem, ao final de contas, não ter nenhum papel no projeto final;
- intrinsecamente envolve a descoberta de novos objetivos.

Essas características tornam o processo de desenvolvimento de sistemas interativos complicado em termos de gerenciamento. Orçamentos e prazos podem facilmente serem comprometidos. Portanto, a alta administração das empresas, precisam primeiro se tornar cientes dos benefícios dos investimentos em usabilidade para em seguida apoiar as iniciativas a respeito.

3. Os três pilares do projeto

Os três pilares (figura abaixo) descritos abaixo foram propostos por Shneiderman (Shneiderman; Plaisant, 2005) com o intuito de ajudar o projetista a "tornar boas idéias em sistemas de sucesso". Para o autor, o uso da técnica sozinha não garante o sucesso do projeto, mas, como veremos a seguir, ela está baseada em projetos e pesquisas quem obtiveram sucesso no passado.



3.1. Guidelines e processos

Antes de mais nada, o grupo de desenvolvimento deve elaborar um documento contendo *guidelines,* ou seja, orientações que devem guiar todos os projetistas da organização. Um dos motivos pelos quais a Apple tem sucesso é a existência de um conjunto de princípios nos quais todos os desenvolveres devem seguir, o que assegurou a harmonia no *design* de todas a aplicações do sistema. Os *guidelines* da Microsoft tem sido refinados por anos e servem também como ferramenta educacional para os programadores que queiram desenvolver na plataforma.

Cada projeto tem necessidades distintas, mas *guidelines* devem considerar:

Palavras e ícones

- Terminologias (objetos e ações), abreviações, e em que circunstancias usar letras maiúsculas;
- Tipo e tamanho de fonte e estilos (negrito, itálico e sublinhado).
- Ícones, gráficos e espessura das linhas;

- Uso de cores, inclusive cores de fundo, uso de recursos como realces e trocas de cores (piscando).

Leiautes de telas

- Seleção de menus, preenchimento de formulários e formatos de caixas de diálogos;
- Como as mensagens de erro e feedback serão construídas;
- Margens, alinhamento e espaços em branco;
- Entrada e apresentação de dados para itens e listas;
- Formatação de cabeçalhos e rodapés.

Dispositivos de entrada e saída

- Teclados, tela, controle de cursos e dispositivos de apontamento;
- Sons, feedback de voz, toque, gestos e outros modos de entrada especiais;
- Tempos de resposta para diversos tipos de tarefa.

Seqüências de ação

- Clicar, arrastar e soltar ou gesticular,
- Sintaxe de comando, semântica e seqüências;
- Teclas de função programáveis;
- Procedimentos de recuperação de erros

Treinamento

- Ajuda on-line e tutoriais
- Materiais de treinamento e de referência

Shneiderman (Shneiderman; Plaisant, 2005) advoga que o processo de criação dos *guidelines* deve ganhar visibilidade e apoio dentro da organização. *Guidelines* controversas, como quando se utilizar da alertas de voz, deve ser revisto por outros colegas e testado empiricamente.

Esses documentos devem estar em constante estado de adequação, devem ser refinados para se adaptar às novas necessidades e às experiências acumuladas através do uso.

3.2. Ferramentas de software de apoio ao desenvolvimento da interface

Uma das dificuldades no projeto de sistemas interativos encontra-se no fato de que clientes e usuários normalmente não tem idéia de como o sistema ficará até que ele esteja pronto. Alguns sistemas apresentam avanços em diversas áreas e os usuários podem não entender os impactos de determinadas

decisões de projeto. Em geral, as mudanças são muito mais complicadas e caras quando implementadas em estágios mais avançados do projeto.

A questão ainda não está totalmente resolvida, mas alguns problemas podem ser evitados se usuários e clientes tiverem uma visão realística do sistema ainda em seus estágios iniciais. Felizmente, nos dias de hoje existem uma pletora de ferramentas de prototipagem onde podemos construir telas, menus, formulários, simular cliques de mouse, etc. Projetistas são aconselhados a se utilizarem desses recursos e envolverem os clientes no processo de projeto, obtendo deles o "aceite" dos protótipos antes que esses sejam encaminhados para a programação.

Na figura acima temos um protótipo de um programa feito no aplicativo Pencil (gratuito). Os "balões" em azul mostram como a interação acontecerá. Obviamente existem outros aplicativos que já permitem alguma interação do usuário com o sistema. Entretanto, vale a pena ressaltar que todos projetos tem orçamentos e que devemos encontrar equilíbrio entre o tamanho do esforço empregado na construção dos protótipos. Assim, utilizando uma ferramenta como o Pencil podemos construir protótipos rapidamente e a baixo custo. Isso além de aumentar a produtividade, fazendo com que o projetista possa construir mais protótipos diferentes ou mais versões de um mesmo protótipo.

3.3. Revisão de especialistas e testes de usabilidade

Produtores de espetáculos de teatro sabem que antes da estréia devem fazer ensaios onde todos as personagens vestem com as suas fantasias e o palco está montado exatamente da forma em que estará no primeiro dia de espetáculo. Projetistas de aeronaves testam seus protótipos em túneis de vento e em programas de computador que simulam com o máximo de precisão as mais diversas condições que o aparelho vai encontrar no mundo real. Da mesma forma, projetistas de sistemas interativos devem realizar muitos pequenos testes piloto e além de alguns testes completos. Esses testes devem envolver usuários representativos e pessoas especialistas nas tarefas contidas no escopo do sistema.

Existe uma variedade de métodos de revisão a serem empregados com os usuários especialistas, dentre os quais podemos citar:

- **Avaliação Heurística.** Os revisores fazem uma crítica da interface a fim de determinar sua conformidade com uma lista de heurísticas de projeto semelhantes às oito regras de ouro. Faz uma enorme diferença se os revisores são familiarizados com as regras e se são capazes de interpretá-las e aplica-las.
- **Revisão das guidelines.** Nesse tipo de revisão, a interface é checada com relação à sua conformidade com o documento de guidelines.
- **Inspeção de consistência.** Aqui os especialistas verificam a consistência da interface com relação às demais aplicações da organização, checando terminologia, cores, leiaute, formatos de entrada e apresentação de dados, etc. Essa inspeção envolve não somente o sistema, mas também os materiais de treinamento e ajuda online.
- **Passo a passo cognitivo** (Cognitive Walkthrough). Esse tipo de inspeção, os especialistas testam a interface passo a passo enquanto realizam as

tarefas mais importantes do sistema, principalmente aquelas de maior freqüência. Simulações de erros e das rotinas de recuperação deles também são realizadas.

- **Inspeção formal de usabilidade.** Nessa técnica, os participantes se reúnem como se fossem julgar a interface. Existe a figura do moderador ou "juiz" que apresenta a interface e discute seus méritos e fraquezas. A equipe do projeto pode contestar a apresentação do julgador. Esse tipo de inspeção tem um apelo educacional.

Após resistência inicial, gerentes começam a entender os benefícios dos testes de usabilidade e já colocam esses testes em nos cronogramas de seus projetos. Grandes empresas de computação possuem laboratórios de usabilidade. A IBM, por exemplo, possui uma elaborada instalação na Flórida com 16 laboratórios de usabilidade dispostos em formato circular e com um prédio central que abriga um banco de dados que armazena dados relativos ao uso e performance de usuários em experimentos relacionados à usabilidade.

Normalmente testes de usabilidade compreendem uma lista de tarefas, um questionário que serve para medir o nível de satisfação do usuário acrescido de perguntas que ajudam o projetista/pesquisador a entender a experiência vivida pelo usuário durante a execução das atividades previstas no teste. Deve-se pedir permissão por escrito dos usuários para filma-los durante a execução das tarefas. Nessa filmagem devemos utilizar várias câmeras posicionadas de maneira a capturar aspectos de denunciem situações como frustração, dificuldade, satisfação, etc. Para isso câmeras devem focar rosto, braços e campo de trabalho. Para ajudar no trabalho de análise dos testes, é válido pedir ao usuário que pense em voz alta, o chamado método think aloud.

É igualmente importante que se tenha dados quantitativos de performance dos usuários com relação à execução das tarefas listadas nos testes. Esses dados podem ser comparados aos qualitativos (descritos no parágrafo acima). Quando os dados qualitativos são coerentes com os quantitativos, as conclusões da avaliação ficam mais consolidadas, pois se sabe que o usuário estava sendo sincero em durante a entrevista. Os dados qualitativos ajudam o projetista a entender como o sistema impactou a forma de realizar cada uma das tarefas e o que pode levar a ajustes na interface na busca por melhores números. Dentre os dados quantitativos podemos listar os seguintes:

- Tempo que os usuários levam para aprender determinadas funções;
- Tempo que os usuários levam para executar cada uma das tarefas elencadas no teste;
- Taxas de erros na execução das tarefas do teste;
- Grau de satisfação do usuário com o uso do sistema.

4. Metodologias de desenvolvimento

4.1. Introdução

Shneiderman (Shneiderman; Plaisant, 2005) estima que cerca de sessenta porcento dos projetos fracassam dos quais vinte e cinco porcento deles nunca terminam enquanto os outros trinta e cinco alcançam apenas sucesso parcial. Uma abordagem de desenvolvimento que leve considere as questões de usabilidade desde os estágios iniciais do projeto resultam em diminuição dramática de custos de desenvolvimento e manutenção. Tais abordagens produzem sistemas mais fáceis de se aprender e de se utilizar, aumentam a produtividade e diminuem as taxas de erro.

Metodologias de engenharia de software como a UML tem-se se mostrado úteis para que projetistas e gerentes mantenham seus projetos dentro dos prazos e orçamentos estimados. Entretanto, nem sempre fornecem *guidelines* úteis à construção de interfaces de qualidade.

Várias metodologias de projetos focadas no projeto de interfaces foram propostas, como Hix e Hartson (Hix; Hartson, 1993) e Nielsen (Nielson, 1994). A título de ilustração detalharemos a metodologia *Logical User-Center design* proposta por Kreitzberg (Kreitzberg, 1996). Ela identifica seis estágios no processo de desenvolvimento de sistemas interativos. Tais estágios estão abaixo listados.

Estágio 1: Desenvolvimento do conceito do produto

- Propor conceito de alto nível;
- Estabelecer objetivos de negócio;
- Montar equipe de usabilidade;
- Identificar população de usuários;
- Levantar questões de âmbito técnico e ambiental;
- Propor plano envolvendo recursos humanos, cronograma e orçamento.

Estágio 2: Pesquisa e análise das necessidades

- Dividir o grupo usuários em sub-grupos homogêneos;
- Dividir as atividades de trabalho em unidades de tarefa;
- Conduzir análise de necessidades através da construção de cenários e do projeto participativo;
- Propor o primeiro rascunho do fluxo de trabalho e seqüência de tarefas;
- Identificar os objetivos principais e estruturas que serão utilizadas da construção da interface;

- Pesquisar e resolver questões técnicas e outras limitações relativas ao projeto.

Estágio 3: Conceitos de projeto e protótipos de telas-chave

- Propor critérios de usabilidade específicos baseados nas necessidades dos usuários;
- Iniciar a construção do documento de *guidelines* e de estilo;
- Identificar o conjunto de "telas-chave", como: login, tela de entrada e telas dos principais processos;
- Desenvolver protótipos das telas-chave utilizando-se de ferramenta de prototipação rápida;
- Conduzir revisões iniciais e testes de usabilidade.

Estágio 4: Refinamento do projeto através de iterações

- Expanda os protótipos das telas-chave e programe-as;
- Conduza revisões heurísticas e revisões de especialistas;
- Conduza testes de usabilidade completos;
- Entregue protótipos e especificação para programação.

Estágio 5: Implemente o sistema

- Utilize-se de padrões;
- Administre mudanças tardias nos requerimentos;
- Desenvolva *help* online, manuais e tutoriais;

Estágio 6: Apóie o processo de implantação

- Forneça suporte e treinamento;
- Realize manutenção, mantenha histórico de utilização.

Em seguida, discutiremos três técnicas bastante utilizadas no processo de desenvolvimento de sistemas interativos: Observação etnográfica, projeto participativo e desenvolvimento baseado em cenários.

4.2. Observação etnográfica

Praticamente todas metodologias incluem a observação de usuários em seu ambiente de trabalho como um dos passos iniciais no processo de construção de sistemas interativos. Para esses usuários, trabalhar em uma organização e em um lugar e em uma época acabam constituindo uma cultura única. Por esse motivo métodos etnográficos de observação tendem a se tornar cada vez mais importantes. "*Um etnógrafo participa, de modo explícito ou não, no cotidiano das pessoas por um período extenso de tempo, observando o que acontece, ouvindo o que é falado e fazendo perguntas*", Hammersley e Atkinson (Hammersley; Atkinson, 2007). Como etnográficos, projetistas de interface ganham entendimento sobre o comportamento individual e o contexto organizacional. O trabalho desses projetistas se diferencia do etnógrafo tradicional porque eles observam as interfaces em uso com o propósito de melhora-la ou de construir algo inteiramente novo. Enquanto etnógrafos tradicionais entram em um processo de imersão na cultura, o que normalmente leva semanas ou meses, projetistas de interface limitam esse tempo para dias ou mesmo horas e mesmo assim ainda conseguem obter informações necessárias ao seu ofício. Métodos etnográficos já foram empregados em observações realizadas em escritório, controle de tráfego aéreo e em vários outros domínios.

Essa atividade também precisa de *guidelines* para diminuir as possibilidades de serem feitas observações irrelevantes ou que se deixe de perceber detalhes que devam ser considerados. Rose (Rose et al., 1995) elaborou uma série de *guidelines* com objetivo de tornar o processo de observação etnográfica mais produtivo, as quais transcrevemos a seguir:

Preparação

- Entenda as políticas e cultura da organização;
- Familiarize-se com o sistema de informática e sua história;
- Proponha objetivos iniciais da observação e prepare questionário da pesquisa;
- Obtenha permissão para observar e entrevistar.

Estudo de Campo

- Estabeleça bom relacionamento com gerentes e usuários;
- Observe e/ou entreviste usuários em seu ambiente de trabalho e colete dados qualitativos e quantitativos;
- Investigue com mais profundidade *pistas* que tenha emergido das entrevistas;
- Grave todas as visitas e entrevistas.

Análise

- Compile os dados obtidos em bancos de dados numéricos, textuais e de multimedia;
- Quantifique os dados e realize testes estatísticos;
- Interprete os resultados dos testes estatísticos;
- Refine seus objetivos à luz dos resultados dos resultados de sua análise.

Divulgação dos resultados

- Considere relatar os achados de sua pesquisa para várias audiências;
- Prepare relatórios e apresente resultados.

À primeira vista esses guidelines parecem óbvios, mas segui-los requer dos projetistas atenção ao interpretar cada situação. A título de exemplo, um gerente pode reclamar que seus funcionários não preenchem o formulário de controle de atividades. Já os funcionários se queixam que o tal preenchimento é muito trabalhoso e leva tempo preenche-lo.

4.3. Projeto participativo

É fácil entender porque devemos envolver os usuários nas várias fases dos projetos de sistemas interativos. Um maior envolvimento resulta em informações mais precisas acerca das tarefas pertencentes ao escopo do sistema, mais oportunidades para que eles influenciem nas decisões de projeto, facilita o processo de aceitação do sistema e aumenta a sensação de que o projeto também é deles.

A questão que se levanta é até onde esse envolvimento deve ir. Se for muito extenso, encarece o projeto e aumenta o tempo de execução. Outro potencial problema é que as pessoas que não estão envolvidas no processo podem oferecer resistência ao sistema. A participação de usuários incompetentes pode gerar interfaces de baixa qualidade uma vez que o projetista fica tentado a satisfazê-los.

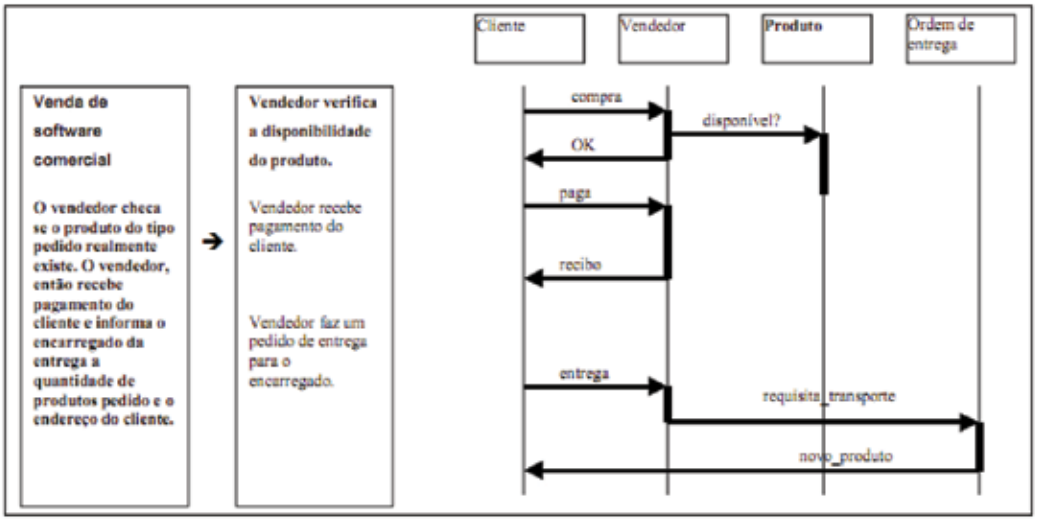
Uma das metodologias mais tradicionais de projeto de interfaces participativo é o chamado *Plastic Interface for Collaborative Technology Initiative through Video Exploration* (PICTIVE) de Muller (Muller, 1992). Nessa abordagem, usuários usam cartolina, etiquetas ou fitas adesivas para construir um protótipo de baixa fidelidade. A cartolina é a tela em branco sobre a qual os próprios usuários vão criando os objetos que nela aparecerão. Com etiquetas e/ou fitas pode-se construir botões e outros objetos de interação. Os usuários, após concordarem em como a interação será feita, fazem um vídeo deles mesmos interagindo com o protótipo. Segundo Muller, sob liderança adequada, a metodologia promove o surgimento de novas e interessantes formas de interação e os usuários "se divertem" com esse tipo de trabalho.

É importante lembrar que os usuários devem se comprometer com o projeto, com uma agenda de reuniões sistemáticas e, principalmente com a necessidade de se honrar prazos. Como normalmente são profissionais de outras áreas, provavelmente estão lotados em outros departamentos e possuem outras obrigações.

4.4. Desenvolvimento baseado em cenários

Nessa prática, bastante utilizada nos dias de hoje durante o processo de elicitação de requisitos. Na construção dos cenários descrevemos situações em que os usuários se utilizarão recursos fornecidos pelo sistema para resolver situações encontradas no dia-a-dia de suas atividades. Quando possível e especialmente quando no cenário envolve mais de um ator e um processo de colaboração entre eles, é aconselhável criar um pequeno teatro e "executar" a atividade conjunta. Além dos usuários-chave, esses "teatrinhos" devem ser encenados nos locais de trabalho para que os atores possam utilizar o espaço e os objetos normalmente empregados na execução das tarefas encenadas, por exemplo: Recepção de hotéis, laboratórios de análises clínicas, etc. Os cenários podem representar situações normais ou emergenciais. Pode-se construir cenários descrevendo a utilização do sistema por usuários novatos e outro cenário onde a mesma atividade é executada por um usuário experiente.

A figura abaixo exemplifica um exemplo de cenário.



(Breitman, 2000) pg. 24.

5. Impacto social

A introdução de sistemas interativos normalmente causam enorme impacto não somente nas organizações, mas também nas vidas das pessoas que as fazem. A fim de minimizar os riscos associados ao seu desenvolvimento e implantação, é interessante que se produza um documento bastante bem fundamentado antecipando os impactos que o sistema deve causar para que seja apresentado aos tomadores de decisão das empresas. A circulação desse documento, além de ser uma medida preventiva, pode elicitar sugestões produtivas por parte desse grupo. Essas sugestões podem levar a correções de rumo ainda em uma fase inicial do projeto, o que é bem mais barato do que em fases mais adiantadas.

Shneiderman e Rose (Shneiderman; Rose, 1997) sugerem um roteiro para a construção do documento de impacto social, transcrito abaixo:

Descreva o novo sistema e seus benefícios

- Transmita os objetivos de alto nível do novo sistema;
- Identifique os *stakeholders;*
- Identifique os benefícios específicos.

Aborde preocupações e potenciais barreiras

- Antecipe mudanças em funções de trabalho e potenciais demissões;
- Aborde questões relacionadas a segurança e privacidade;
- Discuta responsabilidades por mal uso e falha do sistema;

- Evite julgamentos direcionados;
- Discuta direitos individuais e benefícios sociais;
- Avalie questões acerca de até que ponto sistemas devem ser centralizados ou descentralizados;
- Preserve princípios democráticos;
- Assegure acessibilidade;
- Mantenha a simplicidade e preserve o que funciona

Descreva o processo de desenvolvimento

- Apresente uma estimativa de cronograma para o projeto;
- Proponha um processo de tomada de decisão;
- Discuta como será o envolvimento dos *stakeholders;*
- Revorganize as necessidades de aumento de pessoal, treinamento e equipamento;
- Proponha plano para *backups* de dados e equipamentos;
- Proponha um plano de migração para o novo sistema;
- Proponha um plano para medir o sucesso do novo sistema.

Atividades de avaliação

1. O que você entende sobre Engenharia da Usabilidade?

2. Por que é tão importante o apoio da organização para a usabilidade?

3. O que você entende sobre Revisões de especialistas?

4. Qual o papel das guidelines no desenvolvimento de sistemas?

5. O que é observação etnográfica e quais benefícios ela traz para o projeto de sistemas interativos?

6. Quais são os principais problemas da metodologia Projeto Participativo?

Capítulo 4

Manipulação direta e ambientes virtuais

Objetivos

- Este capítulo começa com uma discussão de como projetos de sucesso baseados em manipulação direta revolucionaram o uso do computador. Em seguida debatemos as razões do sucesso de tal técnica e a apresentação de situações onde seu emprego não é aconselhável. Depois apresentamos ambientes de desenvolvimento baseados em manipulação direta, seguida da apresentação de sistemas de automação de lares, de manipulação direta remota e de ambientes virtuais.

1. Introdução

Para Shneiderman (Shneiderman; Plaisant, 2005) as características de sistemas interativos que realmente fazem a diferença são a visibilidade de objetos e ações e a possibilidade de se tomar ações rápidas, reversíveis e incrementais. Interfaces devem possibilitar a manipulação direta de objetos e ações relacionados ao domínio da tarefa. Ao interagirem com interfaces com essas características, usuários reportam sentimentos positivos como:

- Domínio da interface;
- Competência em realizar as tarefas;
- Facilidade de aprendizado tanto de funções básicas como avançadas;
- Confiança de que continuarão a dominar a interface mesmo se deixarem de usa-la por algum tempo;
- Satisfação ao usar o sistema;
- Vontade de ensinar aos outros;
- Desejo de explorar aspectos mais avançados do sistema.

Nesse capitulo discutiremos sistemas que incorporam algumas dessas características.

2. Sistemas baseados em manipulação direta

2.1. Editores de texto

Processadores de texto e planilhas eletrônicas estão entre os primeiros aplicativos que se utilizaram do conceito de manipulação. A estratégia ajudou a revolucionar os escritórios de tudo o mundo. Um dos pilares para dessa abordagem é o conceito *WYSISYG*, do Inglês *What you see is what you get,* o que em tradu-

ção livre seria *o que você vê é o que você leva,* ou seja o que você vê na tela, é o que sairá na impressão. Hoje em dia isso até parece trivial, mas no passado não muito distante, existia a necessidade de se elaborar uma série de comandos antes para se imprimir uma página de texto no formato desejado. Portanto, não se tinha segurança se a página sairia como se queria. Outra melhoria muito importante e revolucionária para época foi o fato da tela ter assumido um formato de "página", exibindo o texto completo. Negritos, sublinhados e itálicos também passaram a aparecer na tela de forma natural. O uso do cursor e mouse permitiram que os usuários dessas ferramentas selecionassem diretamente a parte do texto a ser afetada por um comando de formatação, por exemplo. Nos modernos processadores de texto, as ações são facilmente revertidas, o que da muito mais conforto e segurança para o usuário.

Pode-se afirmar que uma pletora de ferramentas surgiram a partir do desenvolvimento de processadores de texto, tais como softwares de apresentação de slides (por exemplo, Powerpoint), planilhas eletrônicas e softwares para publicidade (*publishing*).

A figura abaixo mostra uma antiga tela do "não saudoso" software de edição de textos, *Wordstar*.

2.2. Planilhas eletrônicas

Dan Bricklin e Bob Frankston criaram a primeira planilha eletrônica, o VisiCalc. Dan estava frustrado em ter de fazer cálculos repetitivos durante as tarefas do curso de negócios de Harvard. A ferramenta, conforme descrita pelos seus criadores era uma "planilha eletrônica de cálculo automático" que permitia computação e exibição imediata de resultados em células espalhadas entre as 254 linhas e 63 colunas.

A simulação da planilha de trabalho de um contador facilitava o entendimento de um usuário acerca dos objetos e de quais ações poderiam ser realizadas sobre eles. Usuários de planilhas eletrônicas podem, por exemplo, tentar planos alternativos e observar seus efeitos em vendas e lucro. No primeiro contato com o programa, usuários ficavam maravilhados ao verem as células da planilha se recalcularem automaticamente quando determinado valor em outra célula era alterado. Era como o computador estivesse fazendo o trabalho deles bem debaixo de seus olhos.

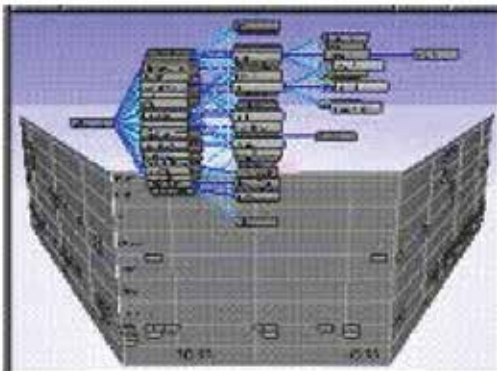
VisiCale e seus criadores

Obviamente muitos outros concorrentes apareceram e incluíram outras funcionalidades como a capacidade de se gerar gráficos e até banco de dados e o VisiCalc saiu do mercado. Mas a contribuição dos dois alunos não mudou somente os escritórios de todo mundo, ela ensinou uma importante lição aos projetistas de sistemas interativos: O valor da manipulação direta.

2.3. Gerenciamento espacial de dados

São diversos os usos potenciais de aplicações cuja abordagem é o gerenciamento espacial de dados. A título de ilustração, uma aplicação militar pode oferecer a um oficial da Marinha um sistema de rastreamento de navios baseado no mapas e em zoom contínuo. Assim o oficial pode identificar comboios de navios como pequenos pontos no meio do oceano. Após selecionar os pontos, o oficial pode fazer um zoom, que pode ser implementado a partir de um movimento com o mouse (Clica-se o botão direito do mouse e arrasta-o para direita para aumentar o nível de zoom, ou para esquerda para diminuir). A medida que os objetos vão se tornando mais visíveis, mais detalhes a respeito deles vão se ficando disponíveis (Prefixo do navio, foto, nome e matrícula de seu comandante, etc.).

O sistema da Xerox chamado de Information Visualizer (figura abaixo) é um conjunto de ferramentas que permite explorações animadas e em três dimensões de edifícios, diretórios (pastas) e diagramas organizacionais.



O sucesso dos sistemas de gerenciamento espacial de dados depende da habilidade dos projetistas na escolha de ícones, representações gráficas, e leiautes de dados com apelo natural e compreensível aos usuários. A satisfação de pairar sobre um objeto e explorá-lo com o zoom atrai e seduz até o mais ansioso dos usuários, que demanda poder e informações adicionais.

2.4. Video games

Para muitos pesquisadores é na área de "games", que muitos dos conceitos que estamos apresentando são aplicados de forma mais efetiva. Para ilustrar essa assertiva, vejamos o que acontece com uma pessoa jogando um antigo e elementar video game, o *pong* (figura ao lado).

Tudo que o usuário tem de fazer é mexer um botão da direita para esquerda e verá um elemento gráfico se mexendo na tela que representa sua raquete. Tudo que um usuário novato precisa fazer para aprender a jogar o jogo é observar outro jogador em ação por não mais que uns trinta segundos.

Jogos mais modernos promovem competição entre jogadores que podem inclusive morar em países diferentes, possuem gráficos tridimensionais e em alta resolução, além de interfaces gestuais. Esses jogos propiciam diversão estimulante, desafios para usuários novatos e experientes e muitas lições que podem ser empregadas em outros projetos.

Esses jogos criam um campo de ação visual convincente. Os comandos são ações físicas cujo resultados são exibidos imediatamente na tela. Não existe nenhuma sintaxe que deva ser memorizada e, portanto não há mensagens de erro. Se usuário acha que a sua espaçonave está muito para esquerda, ele simplesmente move o joystick para direita. Se a brincadeira é de luta de boxe, ele pode interagir utilizando nada mais que seu próprio corpo, dando socos no ar. As mensagens de erro são dispensáveis, pois as ações são óbvias e facilmente reversíveis. Esses princípios podem ser aplicados em aplicações voltadas para escritório, computação pessoal ou outros domínios de sistemas interativos.

Entretanto, é importante salientar que existem diferenças importantes entre o ambiente de diversão e o de trabalho que devem ser consideradas pelo projetista. A pletora de recursos multimodais oferecidos ao usuário em ambiente de jogo pode ser vista como distração para um funcionário em um escritório. No jogo há muitos eventos aleatórios projetados para desafiar o jogador, usuários em outro ambiente de interação preferem que o sistema tenha comportamento previsível. Jogadores estão normalmente envolvidos em competições com outros jogadores ou mesmo com o sistema. Em um ambiente de escritório, por exemplo, a colaboração é mais valorizada do que a competição e o *locus* de controle interno (usuário no comando) é mais importante do que o externo.

2.5. Projeto auxiliados por computador

Os sistemas de CAD (*Computer Aided Design,* figuras abaixo), concebidos para auxiliar engenheiros e arquitetos enquanto projetam automóveis, circuitos integrados, espaços interiores, etc., também se valem dos conceitos de manipulação direta. O engenheiro eletrônico, por exemplo, pode ver um esquema de circuito na sua tela e, com uns cliques de mouse, mover resistências e capacitores para dentro do circuito ou ainda trocar por outros de diferentes especificações. Quando o projeto é concluído, o computador pode fornecer informação sobre corrente, voltagem, custos de fabricação, além da fazer uma checagem técnica do projeto à busca de eventuais imperfeições no projeto.

O que satisfaz os usuários desse tipo de aplicação é a capacidade de manipulação direta do objeto de interesse e as múltiplas possibilidades criadas a partir dessa manipulação. Isso é bastante diferente da antiga interação feita a partir da emissão de comandos via teclado.

3. Explicações acerca da manipulação direta

De acordo com o *Princípio da Transparência* de Rutkowski (Rutkowski, 1982), os usuários sentem satisfação interagir com sistemas que possibilitam a manipulação direta porque são "*capazes de aplicar o intelecto diretamente na execução da tarefa em questão; a ferramenta em uso tende a desaparecer*". Heckel (Heckel, 1991) lamenta que "*Nossos instintos e treinamento de engenheiros encorajam-nos a pensar logicamente em vez de visualmente e isso é contraproducente em na construção de sistemas amigáveis.*"

Hutchins e colegas (Hutchins et al., 1985) revêem os conceitos de manipulação direta e oferecem uma inteligente decomposição de conceitos. Eles

descrevem um "*sentimento de envolvimento direto num mundo de objetos, o que é bem diferente de se comunicar com ele através de um intermediário*".

A psicologia, em particular o estudo de como resolvemos problemas e como aprendemos, pode nos ajudar a entender a satisfação do usuário em interagir com sistemas que permitem a manipulação direta. Décadas atrás, Montessori (Montessori, 1973) mostrou que crianças aprendem conceitos de matemática mais facilmente se, em vez de símbolos aritméticos, por exemplo, o professor utilizar objetos físicos para apresentar conceitos como a divisão, soma, subtração ou multiplicação. Bruner (Bruner et al., 1966) levou esse mesmo princípio para o ensino de fatoração polinomial. Finalmente, Arnheim (Arnheim, 1969) mostrou que representações físicas, espaciais, ou visuais são mais simples de se memorizar do que representações numéricas ou textuais.

Chega-se a conclusão, portanto, que nós humanos raciocinamos melhor em cima do concreto do que do abstrato. E isso é fácil de se entender. Ao raciocinarmos sobre coisas abstratas, temos que manter em mente, o seu significado. Quando temos evidências perceptivas dos objetos da ação, não precisamos fazer uma operação de transformação mental a mais. Assim nossas mentes estão mais livres para focar nos objetivos das nossas tarefas.

4. Manipulação direta pode não ser sempre a melhor saída

Representações espaciais ou visuais não se constituem necessariamente uma interface superior quando comparadas a interfaces mais tradicionais. No desenvolvimento de sistemas, por exemplo, os diversos diagramas são úteis até certo ponto. Diagramas complexos facilmente se tornam confusos e difíceis de ler. Estudos mostram que abordagens gráficas são mais eficientes quando a tarefa exige reconhecimento de padrões, mas não quando o campo visual fica muito cheio de objetos e a tarefa exige informações detalhadas. Para usuários experientes, uma tabela com cinqüenta nomes de documentos pode ser mais apropriada do que apenas dez ícones de documentos abreviados de tal forma a caber no tamanho de ícone.

Um segundo problema é que os usuários devem aprender o significado dos componentes das representações visuais. Um ícone, por exemplo, pode ter um significado para o projetista, mas pode requerer um tempo de aprendizado igual ou maior de aprendizado que uma palavra. Em aeroportos, por exemplo, devido ao fato de servirem a uma comunidade multilingual, boa parte da comunicação é feita por meio de ícones cujo significado nem sempre é óbvio.

Outra questão é a que em vários casos, usuários experientes, por motivo de performance, preferem usar teclado a ter que parar a digitação, pro-

curar o mouse e apontar para o objeto na tela. O teclado continua sendo o dispositivo de manipulação direta mais efetivo para diversos tipos de tarefas.

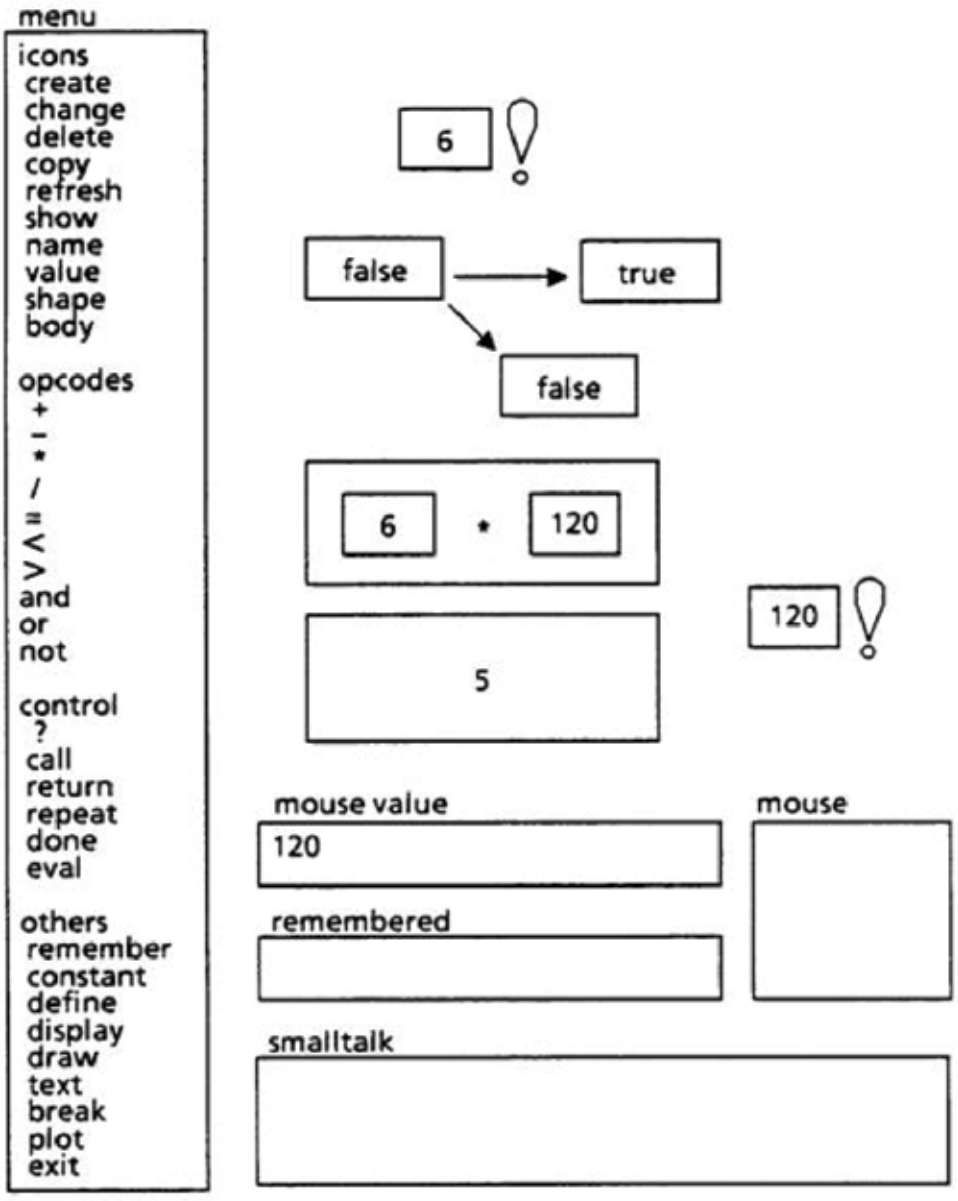
A escolha dos objetos e ações adequados para cada tipo de tarefa não é necessariamente algo simples. Metáforas, analogias e modelos com um mínimo conjunto de conceitos são um bom ponto de partida. Problemas podem surgir quando o projetista escolhe uma metáfora não apropriada ou de difícil entendimento para a comunidade de usuários. Em geral, o projetista tem formação diferente daquelas pessoas que fazem parte da comunidade de usuários e não convive com elas. Assim é importante que o projetista apresente e teste seu modelo conceitual com membros da comunidade o mais cedo possível no processo. É também recomendável a confecção de um documento que apresente o modelo com suas suposições e limitações.

5. Ambientes de desenvolvimento de sistemas baseados em manipulação direta

Não basta produzir programas de computador que implementem a manipulação direta. Um ambiente de desenvolvimento de sistemas que implemente tais conceitos, certamente aumenta a produtividade do programador. Na programação de robôs, por exemplo, os profissionais primeiro testam os movimentos necessários para uma determinada tarefa, um a um, antes de executar a seqüência toda em alta velocidade. Esse conceito de programação de manipulação direta também se aplica a brocas de precisão, câmeras de televisão com movimento previamente definido, etc.

Esse conceito foi levado para a automação de escritório sob a forma de *macros*. *Macros* nada mais são do que um conjunto de comandos previamente gravados que, quando acionadas, as executam trazendo produtividade e diminuindo os erros do operador. Durante a execução de uma *macro* outra pode ser chamada, o que aumenta e muito as possibilidades de construção de encadeamento complexo de ações. Hoje programas de automação de escritório como o *Excel* possuem sua própria linguagem de programação que permite aos seus usuários a criação de partes de programas que são capazes de executar operações em planilhas eletrônicas.

Um trabalho que inspirou muitos foi sistema *Pygmalion* de Smith (Smith, 1977) (figura abaixo). O sistema permitia que programas aritméticos fossem especificados visualmente através de ícones. Muitos outros projetos de pesquisa tentaram criar sistemas de programação de manipulação direta.



Malsby e Witten (Maulsby; Witten, 1989)desenvolveram sistemas capazes de inferir um programa a partir de exemplos, questionando os usuários para resolver ambigüidades. Em certos domínios de aplicação, as interfaces se tornaram previsíveis e úteis, o problema acontecia quando o sistema inferia errado. Os usuários então perdiam a confiança rapidamente.

Para se criar uma ferramenta confiável que funcione em muitas situações sem resultados imprevisíveis, projetistas precisam vencer os *cinco desafios da programação* da interface do usuário, de acordo com Potter (Potter, 1993):

- Generalização computacional suficiente (estruturas de controle como seqüência, decisão e repetição);
- Acesso às estruturas de dados apropriadas e seus operadores;
- Facilidade de programação (por especificação, por exemplo ou por demonstração, com modularização, passagem de argumentos, etc) e facilidade na edição de programas;
- Simplicidade na invocação e assinalamento de parâmetros (manipulação direta, biblioteca de ícones com comportamentos pré-definidos, ferramenta de depuração integrada);
- Baixo risco (alta probabilidade de programas livres de erros e operações do tipo *undo* para facilitar a recuperação de erros)

Green e Petre (Green; Petre, 1996) propuseram o *framework de dimensões cognitivas* para auxiliar o projetista na análise de questões ligadas aos ambientes de programação visual. O *framework* sugere um vocabulário para facilitar as discussões questões de alto nível relacionadas à interface. O termo *viscosidade,* por exemplo, é usado para descrever o grau de dificuldade em se realizar mudanças em um programa. Já *avaliação progressiva* descreve a capacidade de execução parcial de programas. São também dimensões: *Consistência, difusão, dependências escondidas, comprometimento prematuro* e *visibilidade*.

6. Construindo sistemas baseados em manipulação direta

Para Shneiderman (Shneiderman; Plaisant, 2005), o truque para a criação de sistemas de manipulação direta está na proposição de uma representação ou modelo apropriado da realidade. Muitos projetistas podem ter dificuldades em pensar sobre problemas de informação de forma visual. Mas com a prática, eles podem acabar achando o método mais natural. Vários projetistas aplicam os conceitos de manipulação direta a partir de metáforas. Por exemplo, se você está interessado em construir um programa de agenda de endereços, pode começar com a imagem de uma agenda real. Com essa imagem diante de você, imagine as ações que seus usuários podem fazer e como eles fariam essas mesmas ações com uma agenda de verdade. A partir daí veja como essas ações podem ser implementadas na sua aplicação de tal forma que se pareçam ao máximo com aquelas realizadas com o objeto do mundo real. Obviamente, nem todas as ações do mundo real poderão ser modeladas em sua aplicação e como já discutido anteriormente, nem todas as tarefas são realizadas de forma mais eficiente através da manipulação direta.

Manipulação direta tem o poder de atrair usuários porque as ações são facilmente compreensíveis, simples e até agradáveis de se executar. Quando

as ações se tornam simples e reversibilidade é assegurada, a retenção fica facilitada, há uma diminuição da ansiedade, os usuários se sentem em controle e o nível da satisfação aumenta.

7. Automação de lares

Existem muitas oportunidades para o desenvolvimento de sistemas interativos baseados em manipulação direta quando nos referimos à automação de lares. Essas oportunidades se estendem desde o controle do temperatura do ar-condicionado, passando pela hora em que o sistema de aspersão do jardim e chegando até a programação de eletrodomésticos.

Essas possibilidades incluem o uso de comandos de voz, algo que somente agora começa a se concretizar dado à complexidade de algoritmos e dos modelos matemáticos por trás deles. Um exemplo bem recente de sucesso no emprego de comandos de voz vem do aplicativo *Siri* disponibilizado pela Apple em seus *smartphones* a partir do modelo iPhone 4S (figura abaixo).

Dispositivos como o controle remoto universal (figura abaixo), apesar de não puder se dizer que são populares, devido ao preço proibitivo, estão no mercado a algum tempo. Eles foram concebidos, primeiro para reunir em um só dispositivo as funcionalidades dos diversos tipos de controle remoto: o da televisão, do DVD, do equipamento de som, etc. Os mais modernos controlam também computadores, ar-condicionados e iluminação.

O sucesso desses sistemas depende muito da facilidade de aprendizado tanto na hora de instalar e configurar o sistema como na momento de utilização (de controle) dos sistemas instalados. Esses sistemas são normalmente adquiridos por pessoas de alto poder aquisitivo e, portanto, exigentes.

8. Manipulação direta remota

A medida em que os problemas de redes de comunicação de dados são superados e conexões rápidas e confiáveis são disponibilizadas, grandes oportunidades para sistemas de manipulação direta se abrem. Dentre os domínios de aplicação que podem se beneficiar de tais tecnologias temos: automação de escritório, toda a área de trabalho colaborativos apoiado por computador ou *CSCW* do Inglês (Computer Supported Collaborative Work) – incluindo telemedicina, e educação – principalmente educação à distância.

Manipulação direta remota também pode permitir que o desenvolvimento de aplicações no espaço, subaquáticas e em ambientes hostis seja realizado de forma segura e viável economicamente.

Ambientes remotamente controlados em medicina podem permitir acesso de moradores de centros menos desenvolvidos a especialistas residentes em grandes centros. Em outro cenário, um patologista pode examinar amostra de tecido ou de fluídos a partir do uso de um microscópio que faz a captura remota das imagens e as transmite para uma tela de alta resolução. Nesse ultimo cenário, o sistema de manipulação remota poderia oferecer ao patologista controles que permitissem ajustes:

- Da magnificação da imagem;
- De foco;
- De iluminação (ajuste bidirecional contínuo ou passo a passo);
- De posição (ajuste do posicionamento das amostras sob as lentes do microscópio).

A figura abaixo apresenta um microscópio controlado remotamente.

Para Shneiderman (Shneiderman; Plaisant, 2005) a arquitetura de ambientes remotos introduz vários fatores complicadores, a saber:

- **Retardos.** Normalmente são causados por baixa qualidade nos canais de comunicação que ligam os dois pontos onde as atividades estão ocorrendo. Existem os retardos de transmissão, que no exemplo acima seria o tempo que leva para um comando alcançar o microscópio e os retardos de operação, o tempo gasto para que o microscópio responda.
- **Feedback incompleto.** Um exemplo de feedback incompleto pode ser o do microscópio respondendo tão lentamente aos comandos que inviabiliza o seu uso de forma mais natural pelo patologista.
- **Feedback de fontes múltiplas.** Além da imagem o patologista pode também ter acesso ao audio para facilitar a colaboração com um auxiliar que se encontra junto ao microscópio. Os retardos de comunicação entre o patologista e o microscópio podem ser diferentes daqueles do canal de audio entre ele e seu auxiliar. Retardos distintos em canais diferentes podem complicar ainda mais a tarefa.
- **Interferências imprevistas.** Não é somente a comunicação que pode ser fonte de desafios, diversas outras coisas podem dar errado. No nosso exemplo do microscópio operado remotamente, os próprios componentes mecânicos podem falhar e comprometer o funcionamento do sistema.

Como vários desses fatores estão fora do alcance do projetista, resta-lhe propor sistemas que possam suportar esses diversos tipos de adversidades. Uma estratégia poder ser a de minimizar, até onde possível, o volume de dados trafegando pelos canais de comunicação. Outra seria a de ter canais de comunicação redundantes, onde o "reserva" poderia

ser acionado automaticamente no momento em que o "principal" estivesse indisponível. Uma terceira seria a de optar por comunicação assíncrona, onde o operador captura o dado, remete-o ao operador remoto que o analisa e retorna ao com o seu devido parecer. Esse tipo de estratégia é chamada de *store and forward*.

9. Ambientes virtuais

Quando se pensa em simuladores de vôo imagina-se logo quanto não custa para criar um sistema desses. Os valores são tão altos no Brasil até hoje se conta nos dedos de uma mão o número deles. A maioria de nossos pilotos precisa se deslocar ao estrangeiro para ter acesso a um deles.

Quando se pensa na viabilidade econômica desse tipo de projeto fica fácil entender a sua importância. Um simulador de vôo que possa ser de fato utilizado para treinamento de pilotos profissionais custa algo em torno de cem milhões de dólares. Já a aeronave cujo vôo ele simula pode custar quatro vezes mais. Além da questão puramente econômica, existem também vários outros argumentos a favor da construção desses sistemas. Um deles é o humano. Não somente pelas vidas de pilotos-alunos que o uso do sistema poupa, mas também pela melhoria da qualidade do treinamento que proporciona. Determinadas situações de pane são muito perigosas para serem feitas em aeronaves de verdade. Já no simulador, a única coisa que pode sair ferida é o ego do aluno.

Agora, o que nos interessa, para esse texto, é entender como os componentes desse tipo de sistema são colocados juntos para se criar um *ambiente virtual*. As janelas são substituídas por telas de computador de alta resolução. Os sons são produzidos por sistemas estéreos distribuídos em alto falantes de grande fidelidade posicionados estrategicamente com o intuito de aumentar a sensação de *imersão* – termo bastante comum na área de realidade virtual que significa o grau de realismo proporcionado pelo sistema. Ainda com essa mesma intenção, o nosso piloto-aluno senta-se em uma poltrona equipada com motores hidráulicos que reproduzem turbulências, deslocamentos do centro de gravidade nos seus corpos que lhes propicia experiências perceptivas muito semelhantes àquelas experimentadas situações em vôo real.

Vários outros sistema de realidade virtual estão surgindo e todos eles com um enorme potencial de benefícios tanto para as empresas que os criarem, pois um mercado promissor se abre para elas, como também para sociedade como um todo. Não só podemos ter barateados os custos de formação de pilotos com os teremos mais bem treinados. Se levarmos a discussão para a educação médica, por exemplo, podemos ter sistemas de educação que simulem cirurgias e procedimentos complexos. Simulações essas que podem ser repetidas várias vezes para melhor assimilação, podem também ser interrompidas para que o instrutor corrija os alunos no momento apropriado, etc. Agora imagine que, com o avanço das redes de computadores, com conexões mais rápidas e confiáveis, esse treinamento pode ser dado à distância. A possibilidade de treinamento de profissionais de saúde à distancia abre caminho uma melhoria na qualidade de vida de milhões que moram em comunidades remotas.

Os princípios de manipulação direta discutidos nesse texto podem ser úteis àqueles que estão projetando novos sistemas de realidade virtual ou estão refinando os existentes. Usuários devem ser capazes de selecionar ações rapidamente tanto pelo ato de apontar como através de gestos. Seu controle deve incremental e suas ações, reversíveis. O sistema deve mudar de estado imediatamente após cada ação de seu usuário, quer seja atualizado telas, emitindo sons ou mesmo ativando outros sistemas como movimentando a poltrona do piloto. As respostas do sistema, quando bem projetadas e implementadas, proporcionam e aumentam o senso de causalidade reforçam o aprendizado, aumentam performance e satisfação subjetiva. Objetos de interface e as ações que eles implementam devem ser simples e pertencerem ao domínio da aplicação.

Atividades de avaliação

1. Por que geralmente sistemas baseados em manipulação direta geram mais altos níveos de satisfação subjetiva?
2. Por que se diz que a área de jogos é onde se consegue ver mais claramente os conceitos da manipulação direta em ação?
3. Porque nem sempre o uso da manipulação direta é adequado?
4. Desenvolver sistemas de manipulação direta é mais caro. Em que situações devemos usar essa abordagem?
5. Quais são as dificuldades de se aplicar os conceitos de manipulação direta em ambientes de desenvolvimento de sistemas?
6. O Brasil tem hoje grande riqueza mineral com o pré-sal (Petróleo). O problema é que o mineral encontra-se a grandes profundezas no oceano Atlântico. Como a manipulação direta remota pode ajudar a resolver esse problema?

Referências

ARNHEIM, R. Visual thinking. . Univ of California Pr, 1969.

BREITMAN, K. K. Evolução de Cenários. v. , 2000.

BRIGGS, K. C. Myers-Briggs type indicator. . Consulting Psychologists Press Palo Alto, 1987.

BROWN, C. M. Human-computer interface design guidelines. . Intellect Books, 1998.

BRUNER, J. S.; PRESIDENT; OF HARVARD COLLEGE, F. Toward a theory of instruction. . Taylor \& Francis, 1966.

CARD, S.; MORAN, T.; NEWELL, A. The GOMS model of manuscript editing. The psychology of human-computer interaction, v. 139–147, 1983.

CARROLL, J. M.; ROSSON, M. B. Usability specifications as a tool in iterative development. Advances in human-computer interaction, v. 1, n. , 1–28, 1985.

DREYFUSS, H. The measure of man: human factors in design. . Whitney Library of Design New York, 1960.

FOLEY, J.; KIM, W. C.; GIBBS, C. A.Algorithms to transform the formal specification of a user-computer interface. In: PROCEEDINGS INTERACT 87, 2ND

IFIP CONFERENCE ON HUMAN-COMPUTER INTERACTION, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, , 1987, .

GREEN, T. R. G.; PETRE, M. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. Journal of visual languages and computing, v. 7, n. 2, 131–174, 1996.

HAMMERSLEY, M.; ATKINSON, P. Ethnography: Principles in practice. . Taylor \& Francis, 2007.

HANSEN, W. J.User engineering principles for interactive systems. In: PROCEEDINGS OF THE NOVEMBER 16-18, 1971, FALL JOINT COMPUTER CONFERENCE, , 1971, .

HECKEL, P. The elements of friendly software design. . Sybex Inc., 1991.

HIX, D.; HARTSON, H. R. Developing user interfaces: ensuring usability through product \& process. . John Wiley \& Sons, Inc., 1993.

HOLLANDS, J. G.; WICKENS, C. D. Engineering psychology and human performance. . Prentice Hall New Jersey, 1999.

HUFF, C.; COOPER, J. Sex Bias in Educational Software: The Effect of Designers' Stereotypes on the Software They Design1. Journal of Applied Social Psychology, v. 17, n. 6, 519–532, 1987.

HUTCHINS, E. L.; HOLLAN, J. D.; NORMAN, D. A. Direct manipulation interfaces. Human-computer interaction, v. 1, n. 4, 311–338, 1985.

KARAT, C. M.Cost-benefit analysis of usability engineering techniques. In: PROCEEDINGS OF THE HUMAN FACTORS AND ERGONOMICS SOCIETY ANNUAL MEETING, 34, n. 12839–843 , 1990, .

KREITZBERG, C. Managing for usability. Multimedia: A management perspective, v. 65–88, 1996.

MAULSBY, D. L.; WITTEN, I. H.Inducing programs in a direct-manipulation environment. In: ACM SIGCHI BULLETIN, 20, n. SI57–62 , 1989, .

MONTESSORI, M. From Childhood to Adolescence: Including Erdkinder and the Function of the University. . Schocken Books, 1973.

MULLER, M. J.Retrospective on a year of participatory design using the PICTIVE technique. In: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, , 1992, .

NIELSON, J. Special issue: usability laboratories. Behaviour \& Information Technology, v. 13, n. 1-2, , 1994.

NORMAN, D. A. Design rules based on analyses of human error. Communications of the ACM, v. 26, n. 4, 254–258, 1983.

POTTER, R. Just-in-time programming. Watch What I Do: Programming by Demonstration, MIT Press, Cambridge, MA, v. 513–526, 1993.

ROSE, A.; PLAISANT, C.; SHNEIDERMAN, B.Using ethnographic methods in user interface re-engineering. In: PROC. DIS 95: SYMPOSIUM ON DESIGNING INTERACTIVE SYSTEMS, , 1995, .

RUTKOWSKI, C. An introduction to the human applications standard computer interface. Byte, v. 7, n. , 291–310, 1982.

SHNEIDERMAN, B.; ROSE, A. Social impact statements: engaging public participation in information technology design, Human values and the design of computer technology. Center for the Study of Language and Information, Stanford, CA, v. , 1997.

SHNEIDERMAN, S. B.; PLAISANT, C. Designing the user interface 4 th edition. . Pearson Addison Wesley, USA, 2005.

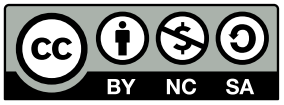
SMITH, D. C. Pygmalion: A computer program to model and stimulate creative thought. . Birkhäuser, 1977.

SMITH, S. L.; MOSIER, J. N.; CORPORATION, M.; DIVISION, U. S. A. F. S. C. E. S. Guidelines for designing user interface software. . Citeseer, 1986

Sobre os autores

Francisco Carlos de Mattos Brito Oliveira: Possui graduação em Ciências da Computação pela Universidade Estadual do Ceará (1990), mestrado em Informática Aplicada pela Universidade de Fortaleza (2002) e doutorado em Computer Science and Applications pela Universidade Virginia Tech, EUA (2010). Têm interesses de pesquisa em: Interação Multi-modal, Comunicação entre humanos eletronicamente mediada, Tecnologias assistivas e Visão computacional. Possui larga experiência em gestão de projetos de grande porte em computação.

Fernando Antônio de Mattos Brito Oliveira: Mestre em Engenharia Agrícola pela Universidade Federal de Visçosa (1999). Atual na área de tecnologia desde 2001, possui diversos cursos de gerência de projetos e atuação relevante na gestão de projetos de tecnologia da informação.



Computação

Fiel à sua missão de interiorizar o ensino superior no estado do Ceará, a UECE, como uma instituição que participa do Sistema Universidade Aberta do Brasil, vem ampliando a oferta de cursos de graduação e de pós-graduação na modalidade de educação a distância e gerando experiências e possibilidades inovadoras com uso das novas plataformas tecnológicas decorrentes da popularização da internet, do funcionamento do cinturão digital e da massificação dos computadores pessoais.

Comprometida com a formação de professores em todos os níveis e a qualificação dos servidores públicos para bem servir ao Estado, os cursos da UAB/UECE atendem aos padrões de qualidade estabelecidos pelos normativos legais do Governo Federal e se articulam com as demandas de desenvolvimento das regiões do Ceará.

ISBN 978-85-78265-65-6

UNIVERSIDADE ESTADUAL DO CEARÁ

UAB UNIVERSIDADE ABERTA DO BRASIL

CAPES