UM ESTUDO SOBRE INTERFACES HOMEM-COMPUTADOR

Luciana Fernandes Schroeder
Centro Federal de Educação Tecnológica da Paraíba
Av. Primeiro de Maio, 720 – Jaguaribe
58015-430 – João Pessoa – Paraíba – Brasil
e-mail: luciana@cefetpb.br

Resumo

Este trabalho apresenta um estudo sobre como se deve projetar uma boa interface homem-computador. Para isso, o trabalho apresenta características desejáveis em uma interface além de algumas recomendações. A proposta deste estudo é o de auxiliar na melhoria da qualidade das interfaces de software.

1. Introdução

Até pouco tempo atrás, interface homem-computador não passava de algumas mensagens secas e campos por preencher em tela de fundo opaco. Pois bem, essa visão mudou, e muito. Hoje em dia, a interface de um sistema define seu sucesso ou fracasso em um mercado cada vez mais agressivo de software. Com o avanço tecnológico, o computador - antes de domínio de grandes empresas e universidades - hoje se encontra, na maioria dos setores, como escolas, escritórios, residências etc. O usuário leigo se tornou um foco importante do mercado e, para que ele sinta atração pela informática, foi necessário mudar. "Fácil de aprender" e "fácil de usar" são agora parâmetros utilizados como ponto de partida. Uma boa interface torna a interação usuário/sistema mais fácil e pode influir positivamente na produtividade final do trabalho. Na maioria dos casos é até mais importante do que um sistema com mais recursos ou eficiência do ponto de vista computacional, se ele não for "amigável" ao usuário.

1.1. O que é uma interface "amigável"?

O termo amigável (user-friendly) é comumente atribuído a interfaces. Sistemas que recebem este adjetivo reúnem as seguintes características:

- Facilidade de aprendizagem e uso: a interface é invisível (o usuário pode concentrar-se nas tarefas que necessita realizar); é previsível; é flexível e as pessoas gostam delas.
- Taxa de erro mínima: dificultar a ocorrência de erros cometidos pelos usuários é imprescindível .
- Recordação rápida: o usuário esporádico não deve ter que recorrer a manuais quando for usar o sistema.
- Atrativo: nem sempre o sistema mais poderoso é preferido pelo usuário. O usuário pode selecionar um sistema em que seu desempenho é inferior comparando com um outro, mas com o qual se sente mais "confortável".

1.2. Por que uma boa interface tem tanta importância?

O que observamos hoje é o aumento de preço do software e o contrário acontecendo com o hardware. Convém ressaltar que na aquisição de um software há investimento com o treinamento, outra área de alto custo. Uma boa interface reduz a necessidade de treinamento.

Softwares que apresentam dificuldades de aprendizado tendem a ser rejeitados pelos usuários. Comercialmente, o sucesso de vendas de software interativos está intimamente relacionado à facilidade de uso e aprendizado do produto.

2. Desenvolvimento de Interfaces

O desenvolvimento de interfaces de usuário é sempre considerado algo pessoal e intuitivo, ao gosto do desenvolvedor do sistema. No entanto, com o advento das interfaces gráficas, estudos têm sido realizados no sentido de orientar o desenvolvedor na observação de várias questões até então ignoradas.

2.1. Características Desejáveis

Existem algumas características desejáveis em uma interface. As básicas estão descritas abaixo:

a. Simplicidade

O usuário tem um trabalho a fazer. O sistema de informação supostamente deve facilitar este trabalho. Portanto projete seu sistema de informação de maneira clara e consistente. Uma interface pobremente organizada com várias funções avançadas distraem os usuários da realização das suas tarefas diárias. Uma interface bem organizada é aquela que suporta as tarefas diárias do usuário eficientemente. Funções básicas devem ser aparentes, enquanto que as avançadas podem ser menos óbvias.

b. Controlabilidade

A interface deve assumir um papel passivo, permitindo que o usuário detenha o controle da interação. Permita aos usuários personalizar sua interface e se estes se afastarem um ou dois dias da aplicação quando retornarem que eles encontrem a aplicação da maneira que a deixaram. Informe ao usuário onde ele está, o que ele pode fazer e como pode sair. Sempre ofereça a possibilidade de interromper ou continuar uma operação.

c. Familiaridade

Onde for possível utilize representações do mundo real na interface. Representações do mundo real torna a interface familiar e intuitiva facilitando o aprendizado. Os controles do sistema devem ser claramente visíveis e suas funções identificadas. Representações visuais oferecem ligações e lembranças que ajudam os usuários a relembrar relações, e reconhecer o que o computador está fazendo. Alguns formatos e definições já consagrados em outros sistemas devem ser mantidos, assim como as inovações devem ser bem elaboradas para não confundir o usuário. Faça uso da cultura (ex.: vermelho = pare/perigo).

d. Consistência

Os mesmos comandos devem fazer sempre as mesmas coisas, os recursos que o usuário aprende a usar em uma parte do sistema devem estar disponíveis, e da mesma forma se possível, em outras partes do mesmo. A consistência visual do sistema é imperativa.
A consistência é importante, pois além de manter a interface dentro de um padrão oferece também ao usuário mais segurança, uma vez que as estruturas utilizadas seguem uma metodologia similar de formação, seguindo os mesmos critérios de apresentação. Isso reduz o esforço de aprendizado, pois permite que o usuário desenvolva um modelo conceitual da interface. Se tais cuidados forem tomados em relação a sistemas, obtém-se padronização e consistência entre sistemas, o que permite a portabilidade de experiência,

ou seja, que o usuário use em um novo sistema a experiência obtida na interação com outros sistemas.

e. Complacência

A interface deve permitir que o usuário recupere-se de situações de erro, bem como considerar que ele pode esquecer informações já apresentadas.

f. Eficiência

A interface deve minimizar o esforço gasto para executar uma tarefa.

g. Conveniência

A interface deve permitir fácil acesso a todas operações.

h. Flexibilidade

A interface deve prover mais de uma maneira do usuário efetuar uma dada operação. Um exemplo de flexibilidade desejável em uma interface é a de oferecer um atalho a uma determinada janela, pode ser este na forma de um botão ou teclas de atalho.

i. Prestimosidade

A interface deve ser prestativa, fornecendo ajuda quando requisitada ou quando perceber que o usuário se encontra em dificuldades. A ajuda, na forma de mensagens de erro, conselhos, etc., deve ser clara e precisa, não conduzindo o usuário a situações embaraçosas, nem exigindo dele conhecimentos que ele não tenha ou não possa obter pelo próprio sistema. O usuário não deve necessitar de outros recursos, fora os oferecidos pela interface, para efetuar sua tarefa. Problemas devem ser notificados ao usuário tão logo sejam detectados e, se possível, antes que ocorram.

j. Naturalidade

A interface deve se comunicar com o usuário de maneira natural, não exigindo deste conhecimento técnico para decifrar a mensagem .

k. Satisfação

A interface deve satisfazer o usuário, não frustando-o. Ela não deve demorar na resposta e deve permitir que o usuário obtenha ajuda em qualquer ponto da interação.

Quando ocorrerem conflitos na tentativa de atender a todas as características citadas, compromissos devem ser assumidos, levando-se em conta aspectos como os tipos de usuário do sistema e as tarefas a serem desenvolvidas por eles.
Uma característica muito exigida da interface é que esta seja natural e intuitiva, mas não podemos esquecer que a interação entre o computador e o ser humano não é natural, exemplo disto são os incontáveis cursos de informática ofertados no mercado. Logo, natural e intuitiva para o usuário é aquela interface que se comporta de maneira semelhante às interfaces já conhecidas por ele. Os conceitos de natural e intuitiva estão, então, ligados a características de consistência.

2.2. Recomendações

A seguir são apresentados alguns exemplos de recomendações ergonômicas para construção de uma interface homem-computador.

a. Presteza

Projete um sistema que informe e conduza o usuário na interação.

Exemplos:

- Dirigir a entrada de dados indicando o formato adequado e os valores aceitáveis (ex.:__/__/__).
- Exibir as unidades de medidas dos dados a digitar (cm , mm, m)
- Para cada campo de dados, fornecer um rótulo. Quando necessário, fornecer no rótulo informações suplementares.
- Indicar o tamanho do campo, quando ele é limitado (ex: Código: _____).
- Dar um título a cada janela.
- Fornecer ajuda on-line e orientação.

b. Concisão

A capacidade da memória de curto termo é limitada. Consequentemente, quanto menos entradas, menor a probabilidade de cometer erros.

Exemplos:

- Para dados numéricos, a entrada de zeros à esquerda não deve ser necessária.
- Se os códigos forem mais longos que 4 ou 5 caracteres, use mnemônicos ou abreviaturas.
- Permitir ao usuário entradas de dados sucintas.
- Quando uma unidade de medida está associada a um campo, inclua a unidade como parte do campo de dados, em vez de fazer o usuário digitá-la.

c. Ações Mínimas

A maioria dos erros ocorrem por culpa do desenvolvedor, muitos deles podem ser evitados tomando algumas medidas consideradas mínimas que reduzem a possibilidade dos erros ocorrerem.

Exemplos:

- Minimize o número de passos necessários para se fazer uma seleção em menu.
- Não faça o usuário entrar com dados que poderiam ser gerados pelo computador.
- Trate a entrada do usuário. Ex. Salário – somente valores numéricos.
- Ofereça uma caixa de seleção ao invés do usuário digitar nomes.
- Faça uso de valores default, mostrando-os automaticamente nos seus lugares apropriados. Esta ação não só agilizará o trabalho do usuário, mas também lhe mostrará o tipo esperado.

d. Mensagens de Erro

A qualidade das mensagens favorece o aprendizado do sistema, indicando ao usuário a razão ou a natureza do erro cometido, o que ele fez de errado, o que ele deveria ter feito e o que ele deve fazer.

Exemplos:

- Fornecer mensagens de erro orientadas a tarefas.
- Utilizar termos tão específicos quanto possível para as mensagens de erros.
- Utilizar mensagens de erro tão breves quanto possível.
- Adotar um vocabulário neutro, não personalizado, não repreensivo nas mensagens de erro; evitar o humor.
- Nunca grite com o usuário, ou seja, não use "!" nas mensagens.
- Não use a palavra "Erro", este termo só é relevante ao desenvolvedor.

3. Conclusão

A construção de interfaces ainda não possui um padrão de desenvolvimento sistemático que possa ser aplicado com sucesso garantido. Porém, não desenvolver um projeto de interface que contemple a necessidade por interatividade dos atuais sistemas de informação poderá definir entre o sucesso ou fracasso de um novo software. Aconselhamos que esse processo de desenvolvimento sofra refinamento através de avaliações até a obtenção de um projeto satisfatório. Versões refinadas do projeto melhoram substancialmente o produto final.

4.Bibliografia

[1] IBM. Principles For Good User Interfaces, 1997, http://www.ibm.com.

[2] Carlow International Incorporated. Human-Computer Interface Guidelines, Aug.1992, http://groucho.gsfc.nasa.gov/Code_520/Code_522/Documents/HCI_Guidelines/.

[3] Lewis C. & Rieman. Task-Centered User Interface Design - A Practical Introduction, 1995.

[4] Lucena, F.N. & Liesenberg. H. K. E., Interfaces Homem-Computador : Uma primeira Introdução, Unicamp, 1994.

[5] SoftPólis, núcleo Softex-2000 de Florianópolis, LabIUtil, Laboratório de Utilizabilidade UFSC/SENAI-SC/CTAI. Ergolist, 1996,http://www.ctai.rct-sc.br.

[6] Vavassori, F. B.. Método Heurístico para Avaliação e Projeto de Interface Homem-Software (Monografia), Universidade Católica de Pelotas, 1995.

[7] Watanabe, Y.S.C. & Oliveira, L.A.M., Projeto PATI : Recomendações para desenvolvimento de Interfaces Homem-Máquina, Telebrás, 1994.