

Papai!

O vovô tem um
processador de texto
fantástico!
Revolucionário!



Você digita e ele já
imprime!!!!!!

SCC0560 - Interação Usuário- Computador (HCI)

Prof.: Dr. Rudinei Goularte
(rudinei@icmc.usp.br)

Aula 2 - Paradigmas e Princípios de Usabilidade

Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação
Sala 4-234

Paradigmas e Princípios de Usabilidade

Principal objetivo de um sistema
interativo?

Paradigmas e Princípios de Usabilidade

Projetistas de sistemas interativos
enfrentam duas questões:

- Como um sistema interativo pode ser projetado para assegurar usabilidade?
- Como a usabilidade de um sistema interativo pode ser demonstrada ou medida?

Paradigmas e Princípios de Usabilidade

- Projetar para maximizar a usabilidade
 - EXEMPLOS de estratégias para a construção efetiva de sistemas interativos provêm PARADIGMAS para o projeto de sistemas interativos com boa usabilidade. A evolução dos paradigmas de usabilidade também provê uma boa perspectiva da história da computação.
 - Através de exemplos!
 - PRINCÍPIOS abstratos oferecem um modo de compreender a usabilidade de modo mais geral.
 - Através de conceitos teóricos

Paradigmas de Interação

Construção de Bons Sistemas
Repetindo os casos de sucesso
→ conhecer a história

Mudanças de Paradigmas

Novos paradigmas são complementares
não necessariamente anulam os
anteriores

Associados à evolução tecnológica
demandam que os usuários repensem o
uso da tecnologia de modo inovador



<http://www.ualberta.ca/CNS/hyperdispatch/HyperDispatch19/1971-84.html>



<http://www.columbia.edu/acis/history/hollerith.html>



Redes de computadores e Tempo compartilhado

- Anos 1960-70s:
 - Processamento de programas em lote
 - Entrada: cartões perfurados, fitas de papel, etc
 - <http://www.columbia.edu/acis/history/hollerith.html>
 - http://www.swtpc.com/mholley/OAE80_Reader/OAE80_Index.htm
 - Saída: relatórios impressos
 - ...

Redes de computadores e Tempo compartilhado

- J.C.R. Licklider (~1960)
 - propôs o início das pesquisas em aplicações centradas no usuário
- Processamento em tempo compartilhado (*time-sharing*):
 - Permitiu que bons programadores passassem a "interagir" com computadores via terminais "remotos"

Video display units (VDUs)

- Tecnologia:
 - Surge o VDU como novo dispositivo físico
- (1962) Sketchpad:
 - Ivan Sutherland's (p/ Ph.D. no MIT)
 - Programa (MIT) para manipulação de imagens visuais em telas
 - Entrada via caneta ótica (*light pen*)
www.wikipedia.org/wiki/Sketchpad
<http://www.sun.com/960710/feature3/>

Toolkits de Programação

- (1960s) Douglas Engelbart
 - Pesquisador do *Stanford Research Institute*
 - Propôs uso do computador como mecanismo de complementar a capacidade humana de resolução de problemas
 - Defendeu a necessidade de *Toolkits* apropriados tanto para
 - produzir equipamento computacional
 - produzir software!
- NSL/Augment
 - Projetado a partir de 1963
 - Demonstrado em 1968
 - VIDEO: <http://sloan.stanford.edu/mousesite/1968Demo.html>
 - De quebra:
 - Inventou o mouse e o editor de texto
- Mais info:
 - <http://www.bootstrap.org/engelbart/index.jsp>

Computação pessoal: 1980s

- Os *toolkits* de Douglas Engelbart's
 - Usados por especialistas em computação
- O time de Seymour Papert (MIT)
 - Desenvolveu LOGO: uma linguagem de programação gráfica para crianças
 - Uma tartaruga desenhava uma trilha com sua cauda em um tanque de areia
 - A criança pode fazer de conta que está dentro da tartaruga e dirigir a tartaruga de modo a desenhar formas geométricas simples digitando frases simples tais como Go Forward e Turn Left

Computação pessoal

- Alan Kay (influenciado por Papert e Engelbart)
um dos fundadores do Xerox PARC (Palo Alto Research Center)
 - Trabalhou para incorporar Smalltalk nos computadores pessoais que se tornavam acessíveis
- Meados de 70: concebeu o Dynabook
 - Não existe ainda: o mais próximo de hoje seria um Tablet PC mas o custo teria que ser MUITO baixo de modo que qualquer um pudesse possuir um Dynabook
 - <http://www.honco.net/os/kay.html>
- "The best way to predict the future is to invent it."

Windows

- ANTES: uma tarefa por computador/terminal
- Computação pessoal "de fato"
- Windows: permite uma tarefa por janela
 - Mapeia o fato de que os seres humanos trabalham, na verdade, com várias tarefas ao mesmo tempo e sob condições de interrupção constante
- WIMP interface: Windows, Icons, Menus and Pointers
<http://cne.gmu.edu/itcore/userinterface/GUIHistory3.html>

Metáforas

- Exemplos
 - tartaruga
 - escritório
 - carrinho de compras
 - máquina de escrever
- Em cada caso
 - Como funciona?
 - O que ajuda?
 - Onde falha?
- Conclusão
 - Definição?
 - Em que ajuda?
 - Qual o problema?

Metáforas

Conclusão:

- Definição
 - metáfora é o uso de conceitos no mundo real para facilitar o APRENDIZADO de um novo sistema
- Em que ajuda
 - o aprendizado, por permitir que conceitos utilizados no mundo real sejam reutilizados no projeto
- Qual o problema?
 - A metáfora não se aplica 100% no projeto, e assim em algum momento o seu uso falha
 - Ex: aprendendo a utilizar um editor de texto, há problema pois barra espaço no teclado apaga (!) se a inserção não estiver ativada, e isso não acontece nunca na máquina de escrever tradicional!

Manipulação Direta

- Ben Shneiderman, 1982
- *Windows, menus, buttons e metaphors*
- WIMP
 - *Windows*
 - *Icons*
 - *Menu*
 - *Pointing device*

Manipulação Direta

- Características
 - Visibilidade dos objetos de interesse
 - Ação incremental na interface com *feedback* rápido em todas as ações
 - Reversibilidade de todas as ações
 - Correção sintática de todas as ações
 - Ações atuam diretamente sobre os objetos visíveis

Manipulação Direta

- *Model-world metaphor*

Interface é um 'mundo' no qual o usuário pode agir, e cujo estado muda em consequência dessas ações

 - deixa de ser uma 'mediadora' do diálogo entre usuário e sistema, e passa a ser o próprio sistema
- Paradigma WYSIWYG
 - Diferença entre a representação do objeto e o objeto de interesse final é mínima

Linguagem vs. Ação de agentes

- Linha de comando
 - `dir -wt *.java`
 - `ps -u mpp`
 - `lp cv.pdf`
 - `mv ../../src/*.java ../../backup/.`
- Manipulação direta (WIMP)
- Agentes
 - Aprendem as ações do usuário e as executam

Hypertext/WWW

- Vanevar Bush: 1945
 - Memex
 - <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>
 - http://www.kerryr.net/pioneers/memex_pic.htm
- Ted Nelson: 1965
 - Xanadu
 - <http://www.xanadu.com/>
- Berners-Lee: 1989...
 - *World Wide Web*
- *Web 2.0*

CSCW

- *human-computer interaction*
 - Apoio às atividades de um indivíduo
 - Indivíduo interage com Computador
- *human-human interaction*
 - Apoio às atividades colaborativas realizadas por grupos de indivíduos
 - Indivíduo interage com outros indivíduos!
 - Com suporte de dispositivos (hardware e software) computacionais

Ubiquitous computing Computação Ubíqua

- 1991, Mark Weiser
 - Computação embutida no ambiente de modo transparente
 - Apoio ao usuário sem este "utilizar o computador"
 - Afastamento do paradigma de computador pessoal
 - Calm computing, Natural interfaces, Augmented Reality, Context-aware computing
- Escala
 - Jarda (*yard*, ~1m)
 - Dispositivos públicos, 1 para vários indivíduos
 - Pé (*foot*, ~30cm)
 - Dispositivos pessoais, 1 para cada indivíduo
 - Polegada (*inch*, ~2.5cm)
 - Dispositivos pessoais (vários para um indivíduo)
 - Mais recente:
 - Cada vez menores
 - Públicos (vários embutidos no ambiente...)
 - Sempre provendo serviços ao usuário de modo individual ou coletivo

Alguns dos responsáveis pela evolução

- Vannevar Bush
- Douglas Engelbart
- Licklider
- Ivan Sutherland
- Simon Papert
- Ted Nelson
- Ben Shneiderman
- ...
- Tim Berners-Lee

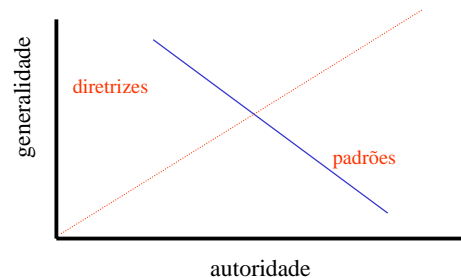
Princípios de Interação

Como construir sistemas seguindo heurísticas abstratas

Regras de usabilidade

- Autoridade vs Generalidade
 - Padrões (*standards*)
 - Heurísticas e Diretrizes (*Guidelines*)

Regras de projeto



Padrões (Standards)

- Definidos por organizações para assegurar conformidade a um conjunto de regras por uma ampla comunidade
 - ISO: International Organization for Standardization
 - BSI: British Standards Institution

Padrões: lembram deste?

- ISO standard 9241
 - **Usability:** the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments
 - **Effectiveness:** the accuracy and completeness with which specified users can achieve specified goals in particular environments
 - **Efficiency:** The resources expended in relation to the accuracy and completeness of goals achieved
 - **Satisfaction:** The comfort and acceptability of the work system to its users and other people affected by its use

Padrões

- ISO standard 9241 (em inglês!)
 - Ergonomic requirements for office work with visual display terminals (VDTs)
 - <http://www.system-concepts.com/stds/status.html>
 - 17 partes, algumas quase completas, outras só planejadas...
 - 7 partes relativas a aspectos de hardware
 - 7 partes relativas a aspectos software
 - A maioria em estágio bem preliminar...
 - Um dos componentes diz respeito à especificação de usabilidade, e se aplica tanto a projetos de hardware como de software

Guidelines (diretrizes)

- MITRE Corporation
 - Guidelines for Designing User Interface Software (1986)
 - <http://www.hcibib.org/sam/>
- Questões
 - Estilos de diálogo, uso de cores,

Guidelines (diretrizes)

- Diretrizes abstratas: utilizadas para especificação de requisitos
 - Ex: Consistência
 - Apple guidelines
 - "effective applications are both consistent within themselves and consistent with one another"
 - "the user, not the computer, initiates and controls all actions."
- Diretrizes específicas: utilizadas no projeto detalhado
 - Ex:
 - toda página do projeto tem que ter o logo em tamanho X no canto superior direito...

Princípios de usabilidade: três categorias

- 1. Learnability
 - (facilidade de aprendizado)
- 2. Flexibility
 - (flexibilidade)
- 3. Robustness
 - (robustez)

1. Learnability (facilidade de aprendizado)

- 1a. Predictability - Previsibilidade
 - (facilidade de **prever** o resultado da interação)
- 1b. Synthesizability -
 - (facilidade de **inferir** como a interação funciona depois de utilizar um pouco)
- 1c. Familiarity
 - (**familiaridade**)
- 1d. Generalizability
 - (facilidade de **generalizar** o mecanismo geral de interação depois de utilizar um pouco)
- 1e. Consistency
 - (consistência)

1. Learnability (facilidade de aprendizado)

- 1a. Predictability (facilidade de prever o resultado da interação com base no passado)
 - antes de interagir, apenas observando, o usuário já sabe o que vai acontecer como resultado de uma interação
 - Pode inferir o que é possível fazer
- 1b. Synthesizability (facilidade de avaliar o efeito das ações passadas no estado atual)
 - O usuário consegue formar um modelo mental do comportamento do sistema, e consegue concluir como a interação ocorre depois de utilizar um pouco o sistema e perceber os resultados de ações passadas
 - Forma um modelo mental da operação que permite avaliar o efeito de ações passadas no estado atual do sistema

1. Learnability (facilidade de aprendizado)

- 1c. Familiarity (familiaridade)
 - O usuário entende a interação porque ela é parecida com outras às quais ele está acostumado a usar em outros sistemas ou no mundo real
- 1d. Generalizability (facilidade de generalizar o resultado da interação)
 - O usuário consegue aplicar soluções semelhantes em várias situações ou sistemas que são semelhantes de alguma forma
- 1e. Consistency (consistência)
 - O quanto o comportamento é similar em situação similares
 - O mais importante dos princípios da categoria de facilidade de aprendizado; os demais dependem deste!

2. Flexibility (Flexibilidade)

- 2a. Dialog Initiative
 - (iniciativa do diálogo)
- 2b. Multi-threading
 - (suporte a múltiplas tarefas/diálogos)
- 2c. Task Migratability
 - (transferência de controle entre sistema e usuário para execução de tarefas)
- 2d. Substitutivity
 - (formas alternativas de entrar/exibir informação)
- 2e. Customizability
 - (capacidade de adaptação da interface)

2. Flexibility (Flexibilidade)

- 2a. Dialog Initiative (iniciativa do diálogo)
 - Dependendo da situação, o usuário ou o sistema inicia a interação
 - Preferência do usuário deve ser maximizada, do sistema minimizada
- 2b. Multi-threading
 - O usuário deve poder fazer várias coisas ao mesmo tempo; múltiplos diálogos em andamento
 - ex: multi-modalidade: usar vários canais de comunicação
- 2c. Task migratability (migração do controle de tarefas)
 - Possibilidade de transferir o controle de uma tarefa entre sistema e usuário e vice-versa
 - ex: correção ortográfica (ser humano pode fazer, mas software pode ajudar)

2. Flexibility (Flexibilidade)

- 2d. Substitutivity
 - uma ação (entrada ou saída) pode ser realizada de mais de um modo
 - usar cm ou *inch*
 - mostrar resultado de vários modos diferentes
- 2e. Customizability (personalização)
 - o usuário deve poder personalizar a interação (e a interface) de acordo com suas necessidades ou preferências
 - Adaptabilidade x adaptatividade
 - gerenciada pelo usuário x pelo sistema (automática)

3. Robustness (robustez)

- 3a. Observability
 - Capacidade que o usuário tem de avaliar o estado interno do sistema a partir da representação perceptível da interface
- 3b. Recoverability
 - Habilidade do usuário realizar uma ação corretiva uma vez que tenha percebido que um erro aconteceu
- 3c. Responsiveness
 - Como o usuário percebe o taxa de comunicação com o sistema, tempo necessário para perceber mudanças de estado no sistema em resposta a ações
- 3d. Task conformance
 - O quanto os serviços do sistema suportam todas as tarefas que o usuário precisa realizar, da maneira que o usuário espera

- Projeto: parte 0!
- Não esqueçam - no máximo até próxima quarta-feira!

- That's it!