Interacção Humano-Computador
Exame

[Segundo ACM] - é uma disciplina que se concentra na concepção, avaliação e aplicação da computação interactiva, sistemas para o uso humano e com o estudo dos fenómenos importantes que os rodeiam.

Origens: Ergonomia - aspectos físicos
Factores Humanos - aspectos cognitivos
Interacções Humano-máquina

Principais Causas que levaram a "disciplina" Interacção Humano-Computador

- Acidentes Aéreos a seguir a II Guerra Mundial
  - ↳ aumento da tecnologia dos cockpits
  - ↳ pilotos tinham acidentes enquantos prestavam atenção aos sistemas.
- Em que entra a HCI?
  - ↳ os pilotos e utilizadores envolvidos no desenho do sistema
  - ↳ objectivos e tarefas analisados
  - ↳ protótipos construídos e usados para testar e descobrir problemas
  - ↳ teste aos prototipos
  - ↳ a aceitação depende dos resultados dos testes.

Dois Bons Exemplos da Importância da IHC:
- Desenho de dois botões num Avião
- Desenho de uma máquina de Radioterapia

Interacção Humano-Computador
- surgiu como disciplina autónoma das ciências da computação → anos 80
- Devido a:
  - Baixo preço da tecnologia
  - Migração da tecnologia
  - Necessidade de aumentar a produtividade dos utilizadores
- Disciplina muito desenvolvida
- É interdisciplinar - reúne vários áreas:
  - Psicologia
  - Ciências da Computação
  - Sociologia...

Projecto de Sistemas Interactivos

Sistema Interactivo - comunicação com o utilizador através da interface

2 Objectivos fundamentais ao Projectar

↳ funcionalidades que melhorem a eficiência, eficácia e desempenho do utilizador

↳ Uma interface que dê:

- as funcionalidades necessárias
- visibilidade do sistema e o efeito dos controlos (que se veja e compreenda a interface)

Estes sistemas envolvem uma parte que não controlamos: o utilizador

- é muito complexo
- não o conhecemos
- não o podemos controlar - (grande dificuldade)

Para o utilizador "o interface é o sistema"

[o desenvolvimento da interface de utilizador, envolve em geral metade ou mais do esforço]

Duas Abordagens ao projecto:

- Princípios de Usabilidade (independentes da técnica)
- Paradigmas de usabilidade (+ dependentes da tecnologia)

Devemos conhecer os exemplos de sucesso (paradigmas de usabilidade) e perceber porque funcionam (princípios de usabilidade)

Adaptar metodologias adequadas, projectar e testar, re-testar.... (até atingir os objectivos)

Desenvolvimento de um Sistema Interactivo (processo iterativo - 4 fases)

(iterativo)
- Análise de Tarefas
- Desenvolver vários desenhos usando tarefas
- Construir Protótipos
- Avaliar

É necessário conhecer: utilizador, paradigmas e princípios de usabilidade

Não projectar com cuidado tem um preço elevado:

- aumenta o tempo de desenvolvimento
- " custos e complexidade de manutenção
- " complexidade do código
- Prejudica a satisfação do utilizador

Usabilidade (Requeste não funcional)

↳ capacidade de um sistema iteractivo permitir que o utilizador alcance os seus objectivos com a sua ajuda

- Aspectos fundamentais para uma boa usabilidade:
  - ↳ fácil aprendizagem
  - ↳ utilização eficiente e eficaz
  - ↳ satisfação

A usabilidade é a capacidade de permitir aos utilizadores resolver determinadas tarefas de forma eficaz, eficiente e com satisfação

Eficaz: cumpre os objectivos pretendidos

Eficiente: rápido

Exemplos de Paradigmas de Usabilidade:

WIMP - anos 80

Manipulação directa (WYSIWYG) - anos 80

WWW - anos 90

...

Principios de Usabilidade (uma lista possível)

Compatibilidade com o utilizador

" com a tarefa

" com o fluxo de tarefas

" entre produtos

Feedback

Coerencia

Familiaridade

Simplicidade

Flexibilidade

Controlo

Invisibilidade da tecnologia

Robustez

Protecção

⇒ objectivos da usabilidade:

facilidade de Aprendizagem

facilidade de Utilização

Satisfação

—//—

Avaliação Heurística (Métodos de Avaliação de Usabilidade - Métodos Analíticos)

Objectivo: encontrar problemas de usabilidade. Pode ser usado nos vários fases do projecto

Como: conjunto de especialistas avaliam a interface de acordo com os heurísticas /principios de usabilidade.

Vantagens: Fácil, barato, rápido

Desvantagens: subjectivo (depende do avaliador)

Dificuldade em arranjar especialistas

Difícil Desempenhar o papel do utilizador

(Nº ideal de Avaliadores - 3 a 5)

Heurísticas de Usabilidade

1 - Visibilidade do Estado do Sistema

2 - Falar a linguagem do utilizador

3 - Utilizador tem controlo e liberdade

4 - Consistência e utilização de normas

5 - Evitar Erros

6 - Reconhecer em vez de lembrar

7 - Flexibilidade e Eficiência

8 - Desenho Simples e minimalista

9 - Ajudar a Reconhecer, diagnosticar e recuperar Erros

10 - Ajuda e Documentação

Grau de Severidade: (frequência, impacto, persistência)

0 - Mínimo

1 - Problemas Cosméticos

2 - Problemas Menores - Baixa Prioridade

3 - Problemas Maiores - Alta Prioridade

4 - Catastróficos - Tem de ser resolvido

Fases da Avaliação Heurística

Pré - Avaliação: informar sobre

Tipo de utilizador

Tipo de Tarefas

Tipo de Aplicação

Avaliação:

Individual para cada avaliador

2 Passagens: 1 - familiar

2 - Procurar erros mais específicos

Ser específico - especificar Heurística violada, sugerir soluções

Classificar a Gravidade

Produzir relatório

O Utilizador

Perfil do Utilizador

Sistemas de Processamento de informação (SPIH)

Experiência e Conhecimento

Trabalho e Tarefa => características muito variáveis entre utilizadores

Características físicas

Ambiente físico

Ferramentas

Sistema de Processamento de Informação Humano

Sub-sistema Perceptual - transporta sensações do mundo físico, detectadas pelos sistemas sensoriais

Sub-sistema Cognitivo - maioria das tarefas realizadas pelo humano envolve de forma complexa recuperação de factos e resolução de problemas.

Sub-sistema Motor - após o processamento perceptual e cognitivo, o pensamento é traduzido em acções

Sub-Sistema Perceptual

- memória - perceptual Buffer
- processo - reconhecimento de Padrões

Sub-Sistema Cognitivo

- memória - de curta duração (STM) (Pouca Limitação)
  - de longa duração (LTM) (Ponto forte)
- processo - atenção selectiva
  - resolução de problemas
  - aprendizagem

Sub-Sistema Motor

- O perceptual Buffer armazena a informação que entra através dos orgãos dos sentidos durante um curto intervalo de tempo (<0,1s)
- esta informação viaja pelo cerbero onde ocorrem niveis sucessivos de processamento
- Antes de chegar ao consciente ocorre o conhecimento de Padrões

[Sistema Perceptual - guarda a saída dos sistemas sensoriais
↳ Sensores associados. Memória de Imagem visual, Memória de Imagem Auditiva

Sistema Cognitivo - usa a informação da STM e da LTM para tomar uma decisão

Sistema motor - viabiliza a resposta do sistema cognitivo

Princípio Básico - ciclo Reconhecer - agir do processador Cognitivo]

Reconhecimento de Padrões

É um processo muito poderoso que ocorre abaixo do consciente

não usa apenas a informação presente.

Resolve ambiguidades

Memória de Curta Duração (STM)

Tem uma capacidade muito reduzida (7 ± 2 unidades de informação) } limitações

Duração apenas de segundos

É aqui que são realizadas operações mentais conscientes

É muito lenta em relação a velocidade que recebe nova informação para processar

↳ problemas minorados com a organização da informação

↳ Características

letras e palavras que rimam são mais difíceis de distingir na STM

se houver actividades que distraiam pequenas quantidades de informação desaparecem

Taxa de esquecimento aumenta:

com a complexidade da actividade que distrai

depende da quantidade de informação a ser lembrada

quando foi armazena informação recentemente.

É mais fácil lembrar figuras do que palavras

Idade do utilizador influência a quantidade de informação a armazenar

Memória de Longa Duração (LTM)

Tem uma capacidade virtualmente ilimitada e permanente em duração } pontos fortes

Estrutura muito rica, complexa e dinâmica

Permite o acesso à informação através de muitos caminhos possíveis

O armazenamento e recuperação são lentos, difíceis e não fáceis.

Informação mais recentemente acedida é mais fácil de recuperar

Tulving sugeriu uma distinção útil entre os 3 componentes da LTM

- Semantic - armazena conceitos e ideias
- Episodic (autobiographical (narrativa) - contem a maioria dos eventos
- Procedural - relacionado com competências - mais robusto.

↳ algumas características

se a informação foi gerada como resposta ou pista → recuperação mais provavel

informação relacionada é mais fácil de recuperar do que informação isolada

recuperação mais fácil para imagens ou palavras visualizáveis

informação mais recentemente acedida é mais fácil de recuperar

Características da Memória Para o design -> Consequências

Não obrigar os utilizadores a reter informação (STM) para realizar operações complexas

Chunking: boa organização da informação no ecrã facilita aprendizagem

labels e nomes devem ser foneticamente distintos

Minimizar pedidos à STM quando à informação é armazenada

Recuperar nomes é melhor quando associados a uma imagem - ícones

Ter atenção aos pedidos à STM para utilizadores mais velhos

frequente utilização gera maior facilidade de recuperação - fornecer help...

Utilizar pistas redundantes para codificação da informação (nomes, formas, cores)

Os exemplos usados durante a aprendizagem devem ser análogos a condições reais.

Nos manuais devem fazer-se notas ao utilizador e erros usados na interface.

Aprendizagem

↳ envolve um grande game de fenómenos - memorização, aquisição de skills mentais ou motores

A aprendizagem é facilitada por:

analogia (semelhança de uma coisa com outra)

estrutura e Organização

Aprendizagem em unidades incrementais

Deve-se:

aproveitar tanto quanto possível os skills e conhecimentos actuais dos utilizadores

apresentar a informação de forma simples e organizada

Evitar nomes e conceitos novos desnecessários

aplicar uma metáfora

as interfaces devem ser flexíveis e permitir shortcuts aos utilizadores

as interfaces devem ter nos visíveis aos estratégias eficientes

Os utilizadores melhoram com a prática - acomodar vários níveis de experiência

Resolução de problemas

os utilizadores diferem entre si

são mais frequentemente heurísticos que algorítmicos

nem sempre escolhem estratégias óptimas

têm capacidade de aprender melhores estratégias com a prática.

Pontos fortes e fracos do STPH

Pontos fortes	Pontos fracos
Capacidade infinita da LTM	Capacidade limitada da STM
Duração e complexidade da LTM	Duração limitada da STM
Grande capacidade de aprender	Processamento afeito a erros
Reconhecimento de padrões	Acesso não fiável à LTM
Mecanismo de atenção poderoso	Processamento lento

Recomendação: distribuir tarefas entre o utilizador e o sistema computacional de acordo com as capacidades de cada um.

Exemplo: minimizar o carga da STM do utilizador

Perfil do utilizador → outros características

Experiência e conhecimento: nível de educação e cultura
capacidade de dactilografia
experiência com o sistema e a tarefa
língua materna
literacia informática

Trabalho e Tarefa: frequência de utilização
treino primário
tipo de utilização
utilização de outros sistemas
frequência de mudança

Características físicas: problema na perceção da cor
deficiências físicas
lateralidade
idade
sexo

Exemplos:

utilizadores com mt experiência do sistema, pouca com a tarefa → ajuda semântica
" " " " da tarefa, pouco do sistema → ajuda sintática
frequência de utilização elevada → facilidade de utilização
" " " baixa → facilidade em aprender e recordar
utilização obrigatória → facilidade de utilização
" opcional → " em aprender e recordar

Guideline Review - o especialista verifica se a interface está de acordo com uma lista de especificações fornecidas pelo cliente

Consistency Inspection - especialista avalia a consistência com uma família de especialistas (layout, cores...)

Vantagens	Desvantagens
avaliação detalhada	muito específico
lista de boas práticas	difícil aplicação

Cognitive walkthrough - particularmente usado para verificar a facilidade de aprendizagem de uma interface. O avaliador percorre todas as acções necessárias para realizar uma tarefa e tenta encontrar pontos em que podem ocorrer problemas

Vantagens	Desvantagens
Permite avaliação detalhada sem utilizadores	Bastante demorado
Pode ser aplicado em qualquer fase do desenvolvimento	Detecta em geral mais problemas do que os que existem
	Muito dirigido, podem faltar problemas

Pré-Requisitos:

descrição detalhada do protótipo

" do tipo de utilizadores

tarefas representativas

lista de acções para cada tarefa

Procedimento: o analista coloca-se no lugar do utilizador e para cada acção vai responder as seguintes questões:

1- A acção correcta está na interface

2- O utilizador vai associar a acção correcta ao que pretende fazer?

3- O sistema vai indicar claramente o resultado da acção escolhida

Modelos Mentais e Conceptuais

o nível mais elevado de uma interface de utilizador

o modelo conceptual é o enquadramento conceptual em que é fornecida a funcionalidade ao utilizador

para compreender como projectar um bom modelo conceptual é necessário compreender os modelos mentais

modelo mental - representação interna da conceptualização e compreensão

corrente do sistema por parte do utilizador
é uma tentativa por parte do projectista através de projectos de interface de
facilitar o desenvolvimento de bons modelos mentais

Como formamos modelos mentais
Usando o sistema
observando a utilização por parte de outros
lendo documentação

Os modelos mentais permitem
fazer previsões
determinar causas de eventos observados
determinar acções apropriadas que determinem as mudanças pretendidas
compreensão de dispositivos análogos

Estes modelos
são incompletos
instáveis
não são científicos
não têm limites definidos
Parcimoniosos

Principais directivas para obter um bom modelo
tornar visíveis processos e partes invisíveis
fornecer feedback
usar coerência (nomes, cores, localização no ecrã, ...)
usar uma metáfora

[Metáfora: explicação de modelos mentais do mundo Real]
Problemas no uso destes:
podem confundir o utilizador se forem incompletos
risco de sub-utilização das capacidades
utilizadores pouco experientes esperam que as metáforas sejam literais
utilizadores sofisticados raramente esperam que sejam "mágicas"]

—//—//—

Métodos Empíricos

Local	Laboratório	Ambiente Real
Vantagens	Equipamentos Especializado Ambiente controlado	Ambiente Natural, Contexto Iteração e Cooperação
Desvantagens	Falta de contexto Pouco Realista	Distracção, Ruido Maior Dificuldade de Observação

Ético => estamos a avaliar o sistema e não o utilizador

Testes com utilizadores - aspectos especiais

confidencialidade

segurança (evitar ~~[illegible]~~ riscos)

liberdade (para parar quando quiser)

stress (minimizar)

Técnicas de Observação

observação -> directa: utilizador bingo notas

⊖ subjectivo

-> indirecta: gravação áudio e vídeo

⊖ complexa, demorada, afasta o utilizador

Pensar em voz alta: utilizador explica o que está a fazer

⊕ fornece mais informação

⊖ pode dificultar a tarefa

Recolher os dados do sistema (logging): monitorização, teclas

Técnicas de Interrogação

entrevista - observador conversa com o utilizador, utilizando perguntas preparadas

⊕ guiada, explorar outros assuntos

⊖ demorada e subjectiva

Questionário - perguntas sem ambiguidade, testado que podem ser abertas / fechadas

⊕ rápido

⊕ permite muito utilizadores

⊕ fácil analise e estatística

⊖ pouco flexível

Preparar teste com utilizadores

1 - Definir o teste (comparativo, experiência controlada / laboratório ou ambiente real)

2 - Definir tarefas (potenciais problemas; tarefas + importantes, ser específicas)

3- Medidas e métodos utilizados: (erros formulário de observador, loggings...)
4- Questionário para pesquisas de experiências: (info sobre utilizadores; avaliação geral (cores, layout))
5- Detalhes / planeamento testes: introdução / treino do utilizador, ordem de tarefas, estudo piloto

Estilos de Diálogos - Manipulação Directa

[menus, formulários, caixas de diálogo, teclas, etc...]
Por norma são usados mais do que um estilo no mesmo sistema.

Manipulação Directa - executam-se acções directamente nos objectos
Características:
representação contínua dos objectos
acções físicas ou selecções (em vez de linguagem de comandos)
operações rápidas e incrementais (reversíveis) com resultados imediatos
* A manipulação directa não implica a utilização de ícones
Exemplos: (WYSIWYG)
visualizar a página completa
ver como fica depois de imprimir
mostra acções com o cursor
ícones para tornar acções mais simples
feedback das acções
facilidade em reverter acções
Em Video Games: (wii, move, Kinect)
acções visuais e envolventes
acções físicas como comandos - imediatas no ecrã
não é necessário lembrar sintaxe

Não existe nenhuma interface de manipulação directa "pura"
É uma qualidade presente em vários graus:
Segundo Hutchins, Hollan e Norman uma interface tem os seguintes aspectos:
Distância - semântica articulatória = Envolvimento - conversação (linguagem de comandos) mundos modelos (manipulação directa)

Distância Semântica e Articulatória

Distância Semântica - distância subjectiva entre os objectivos do utilizador e a semântica da interface. (riqueza e minimalidade)
[é a distância entre o que o utilizador gostaria de dizer na linguagem da interface e o significado disponível pelos elementos da linguagem.

Distância articulatória - distância entre o significado das acções e a sua forma física.
[distância entre o significado e a forma dos elementos de linguagem da interface]

objectivo $\xleftrightarrow{\text{semântica}}$ significado da interface

$\updownarrow$ articulatória

forma física para executar

Exemplo: "Roder 90 graus" tem uma distância semântica baixa mas uma distância articulatória elevada

Resumo:

se os objectivos e acções não apoiam os objectivos do utilizador a distancia semântica é elevada.

se o objectivo do utilizador for desenhar rectangulos a aplicação que tenha rectangulos tem uma distância semântica menor.

quando a forma física das acções é mais próxima do seu significado a distância articulatória é menor.

[Escolher uma opção — distância articulatória maior
arrastar um ícone — " " menor]

Segundo Wolf e Rhyne existem 2 aspectos importantes em qualquer interacção com o utilizador:

Especificação dos Objectos	Especificação de Acções
Geração de Nomes	Correlação Visual (seleccionar)
Correlação Visual	Correlação gestual (desenhar um símbolo)
Geração Gestual	Geração de nome (escrever nome)
	acção análoga
	selecção codificada (escrever um comando)

A especificação dos objectos por correlação visual define a existência de manipulação directa.

O tipo de especificação das acções define o grau de manipulação directa.

Objectos		Acções	Interface - Directa
correlação visual	+	geração de nomes	$\downarrow$
		correlação visual	
		acção análoga	interface + directa

Manipulação Directa

Vantagens:

Fáceis de aprender e recordar
directos, WYSIWYG
Flexíveis, acções facilmente reversíveis
Feedback visual instantâneo e de contexto
menus propensos a erros

Desvantagens
não são auto-explicativos
podem tornar-se ineficientes (teclado + rápido)
é difícil desenhar ícones reconhecíveis
ícones ocupam mais espaço no ecrã

Perfil de Utilizador
Conhecimentos e experiência
baixa capacidade de dactilografia
experiência em sistemas modernos
experiência de janela moderna e alta
utilização frequente de outros sistemas
baixa literacia computacional
Trabalho e Janela
Baixa frequência de utilização
treino moderado
utilização ocasional
janelas pouco estruturadas

Alguns Princípios do Projecto
Minimizar as distâncias articulatórias e semânticas
Para uma boa interface de utilizador
coerência, bom modelo conceptual
feedback, boa organização da funcionalidade
boa disposição no ecrã, utilização eficaz de cor
boa manipulação de erros

Princípios Para ícones
representar a acção ou objecto de forma familiar
limitar o nº de ícones diferentes
realçar os ícones relativamente ao fundo
feedback visual da selecção
Ter atenção a animação do movimento
adicionar informação detalhada

evitar detalhes excessivos

contornos tornam os icones menos distintos

permitir definir icones

Estilos de Diálogo - Menus

Single Menu - selecção entre um ou mais icones

- binário (Messagebox Ok, Cancel)
- com vários opções (RadioButtons, CheckBox)

Pull Down Menu - sempre acessíveis

aceleradores (teclas de atalho Ex: Ctrl + P (Print)

Toolbox, Menus icónicos e Paletas - resposta num clique

possibilidade de adaptar

possível remover para optimizar espaço no ecrã

(conexão com botão direito do rato) Pop-Up Menus - responde a um clique

depende da posição do cursor

cobre uma área do ecrã (tentar reduzir o tamanho para não tapar contexto)

(barra office 2007) Ribbons - substitui menus e toolbars

agrupa comandos por tarefa

utilizadores avançados podem ter problemas em encontrar menus

ocupam mais espaço

(combobox's dos tipos de letras) Menus Listas Extensas - Scrolling Menus - parte do menu, possibilidade de ocultar as outras opções

aceleradores (letra B vai para as opções 3)

(Menu interface tipo TAC) Menu Listas Extensas - Fish Eye - interessante para 10 a 20 itens (todos ficam perceptíveis)

Estrutura dos Menus

Linear

Hierarquica

Rede

Vantagens dos Menus

AutoExplicativos

Requerem pouca memória (reconhecimento em vez de lembrança)

Requerem pouca capacidade de dactilografia

Manipulação de erros fácil

Feedback Imediato

Desvantagens

Pouco eficientes e flexíveis

Pouco práticos para muitas opções

Perfil do utilizador - Menus

Conhecimentos e Experiência

Pouca experiência de dactilografia

Pouca experiência com o sistema, a tarefa e aplicação

Utilização frequente de outros sistemas

Baixa literacia computacional

Trabalho e tarefa

Baixa frequência de utilização

Pouco ou nenhum treino

Utilização opcional

Tarefas muito estruturadas

Menus - Aspectos Relevantes no Projecto

Estrutura do Menu

Ordenação das opções

Selecção das opções

Invocação dos menus

Navegação

↳ Estrutura

adequar a estrutura do menu a estrutura da tarefa

minimizar a profundidade à custa da largura (dentro de certos limites)

Escolher uma ordenação adequada

assinar as escolhas não possíveis de momento

escolher nomes curtos e consistentes

agrupar itens por categoria

Organizar grupos em menus - ter em atenção:

criar grupos de itens similares

cobrir todas as possibilidades

0-9, 10-19, >20

Evitar sobreposições - níveis inferiores associados a um único menu superior

usar termos familiares bem definidos e distintos

↳ Ordenação

Ordenações mais típicas - tempo, numéricas (asc, desc), propriedades físicas (tamanho, área...)

Quando não à ordenação óbvia:

frequência de utilização

Ordem de utilização

Categorias Relacionais

ordem alfabética

↳ Selecção das Opções

usar apontadores ou mnemónicos

ter um tipo de selecção pré-definido

Distingir escolhas simples ou múltiplas (radio-button ou check)

Dar feedback das escolhas dos utilizadores

↳ Invocação dos Menus

usar menus permanentes em vez de pop-up

pop-up para utilizadores experientes

↳ Navegação

Ser coerente (design, nomes das opções)

Incluir descritores das opções se os nomes/icones forem pouco claros

aceleradores

permitir voltar para trás

Estudos Classicos - exemplificativos do que se pode fazer para comparar projectos alternativos

↳ Estrutura - Miller

24 estudantes - 6 por cada estrutura - 4 vezes cada

4 possiveis estruturas com 64 opções

2 opções por nivel, 6 niveis - 2^6

4 " " " 3 " - 4^3

8 " " " 2 " - 8^2

64 " " " 1 " 64^1

Foram medidos: tempos e erros

Princ. Resultado: melhor desempenho verifica-se para 8^2

↳ Ordenação - de Donald et. al.

100 utilizadores - 20 por tipo de ordenação

5 niveis de ordenação

- 2 niveis

categoria, categoria (CC)

categoria, alfabético (CA)

categoria, aleatório (CR)

- Alfabética (A)

- aleatória (R)

Definição de nomes explícitos (simular utilizadores "experientes" e "inexperientes")

Princ. Resultados: utilizadores experientes → melhor desempenho

para utilizadores experientes → ordenação alfabética

↳ Consistência - Teitelbaum and Granada

40 utilizadores com literacia computacional, - 20 vezes cada

Duas estruturas de menus, diferença na localização dos ícones (constante, aleatório)

Medtram-se: tempos médios de procura - blocos de 2 tentativas

Princ. Resultados: utilizadores sistema constante → melhor desempenho

melhoram com prática

Estilos de Diálogo - Outros Estilos

Formulários - adequados quando vários tipos de dados têm de ser preenchidos:

Mostra toda a informação ao utilizador

semelhança com formulário em papel

necessita de ajuda para a maior parte dos campos

Vantagens:

auto-explicativos

requerem pouca memória

acomodam parâmetros com muitos valores possíveis

fornecem contexto

Desvantagens:

assumem conhecimento dos entradas válidas

propensos a erros

pouco flexíveis

requerem capacidade de dactilografia

Perfil do utilizador - Formulários

Conhecimento e experiência

capacidade de dactilografia moderada a elevada

experiência com tarefa elevada

experiência com a aplicação baixa a moderada

literacia computacional moderada a elevada

melhoramentos visíveis

Características da tarefa:

frequência de utilização moderada a elevada

pouco ou nenhum treino

utilização de formulários em papel
importância da tarefa modelada
tarefa com elevada estrutura

Aspectos Relevantes do projecto - formulários

organização e disposição
títulos e campos
formato dos entrados
instruções
Navegação
Erros

↳ Organização e disposição

evitar apresentar a informação de forma não familiar
nas culturas ocidentais colocar o título à esquerda do campo
em colunas de campos, colocar o título em cima

↳ títulos e campos

negrito de alinhamento dos títulos (fluxograma da ordenação dos títulos)
distinguir os títulos dos campos; não mostrar o cursor sobre os títulos

↳ Instruções

indicar campos de preenchimento obrigatório
formatação clara
o utilizador deve poder escolher as unidades de entrada (prevenção de erros)
informação para preenchimento clara

↳ Erros

· mensagens de erro devem ser claras e bem visíveis

(caixa quando se imprime com as opções de impressão)

Dialog Boxes - combinação de menus e formulário

Directivas internas:

título representativo
sequência Top-left to botton-Rigth
layout consistente
margens, grelhas, espaços, linhas, caixas
terminologia, fontes, alinhamento
Botões normalizados

Directivas Externas

aparecimento/desaparecimento suave
fronteira pequena mas visível
tamanho pequeno - evitar oclusões

aparecimento perto dos itens relevantes mas sem oclusões
fácil de fechar

Teclas de funções

2 tipos:

Hard-keys - invocam directamente sempre determinada funcionalidade
[Exemplo: teclas dos pc's ou de uma máquina calcular]

Soft-keys - vão assumindo valores diferentes de acordo com o programa em utilização
[Exemplo: teclas F1 ... Fn ou o teclas genéricas do multibanco]

o seu valor pode programar-se
para serem mais fáceis de usar aprendem devem ser explicadas no ecrã
a limitação do número de teclas pode combater-se com o uso das teclas + shift
contudo a sua utilização torna-se mais difícil

Vantagens das teclas de funções

auto-explicativas
requerem pouca memória
fáceis de usar
flexíveis
requerem pouco (os genéricos) ou nenhum espaço no ecrã (específicos)
pouca capacidade de dactilografia

Desvantagens

número limitado de teclas disponíveis
expansões por hardware são caras
teclado + específico nas hardkeys

Perfil do Utilizador - teclas de funções

Conhecimento e experiência
capacidade de dactilografia baixa
experiência com a tarefa moderada a elevada
experiência com a aplicação moderada
literacia computacional moderada a elevada
Característica da tarefa
frequência de utilização baixa a elevada
pouco ou nenhum treino
utilização opcional
importância da tarefa baixa a elevada

Directivas do Projecto - teclas de funções

fornecer um número adequado de teclas para invocar a funcionalidade (ñ demasiada elevada)
usar:

- Espaço
  - tamanho, cores, formatos diferentes
  - organização em grupos lógicos
  - nomes claros e distintos
- teclas com consequências graves devem estar em locais menos acessíveis
- teclas mais usadas -> localizadas perto da "home row"
- minimizar utilização de teclas modificadoras (quando é necessário usar, fazê-lo correctamente)
- fornecer feedback quando as teclas são carregadas

Linguagem de Comandos - devem ser projectadas por forma a serem tão usadas quanto possível

Vantagens
- poderosas
- flexíveis
- eficientes
- requerem pouco espaço no ecrã

Desvantagens
- difíceis de aprender e lembrar
- não são auto-explicativas
- propensas a erros
- requerem dactilografia
- melhoramentos não são visíveis

Perfil do Utilizador - Linguagem de Comandos

Conhecimento e experiência
- experiência com tarefa elevada
- experiência com aplicação elevada
- literacia computacional elevada
- capacidade de dactilografia elevada

Características da tarefa
- frequência de utilização elevada
- treino formal
- utilização obrigatória

Questões Relevantes no Projecto
- semântica, sintaxe, léxico e interacção

Directivas - Linguagem de comando
- Equilibrar riqueza e minimalismo conceitos análogos a distância semântica (MLD)
- usar síntaxe coerente
- gramática acção - objecto -> natural e fácil de lembrar

abreviaturas dos comandos -> simples e coerentes, obtidas de forma a serem mais fáceis de lembrar (truncatura e não omitindo vogais)

Permitir as seguintes facilidades de interacção

defaults

edição de comandos

interpretação inteligente

type-ahead

feedback

ajuda e documentação

tornar a linguagem: "user tailorable" [Ex: delete -> did you mean "delete"? Y OR N]

Linguagem Natural - a comunicação entre um computador e um utilizador através de linguagem natural envolve: reconhecimento e geração

As linguagens naturais como estilo de diálogo não são verdadeiramente naturais, são linguagens naturais restritas

Dizem em "habilidade" - facilidade e naturalidade com que os utilizadores podem usar a linguagem

↳ "habilidade" -> relacionada com os domínios da linguagem

conceptual - conjunto total de objectos e acções que fornecem

funcional - o que pode ser expresso directamente através da linguagem

sintáctico - formas sintácticas que podem ser compreendidas

lexical - a variedade de palavras que podem ser compreendidas

limitações no domínio conceptual não são perturbadoras, mas as limitações aos outros domínios tornam a linguagem menos "habitável"

[Exemplo: o SI de uma universidade -> base de dados dos funcionários, pode ser consultada com linguagem natural

Domínio Conceptual - informação sobre os funcionários

1. pergunta: "Quanto ganha o director do centro" -> pode estar fora do domínio funcional

2 questões ↳ "Quem é o director do centro?": R: Sr. XX

↳ "Quanto ganha o Sr. XX?"

Questões como: "Qual o vencimento do Sr. XX" ou "Qual é o salário do Sr. XX"? podem não ser reconhecidas por causa do léxico.]

Vantagens

Fáceis de aprender e recordar

Poderosas

Flexíveis

Eficientes

Desvantagens

Assumem conhecimento do domínio do problema

Requerem extensos diálogos de clarificação

Requerem capacidade de dactilografia (se forem escritos)

melhoramentos não são visíveis

Podem criar expectativas irrealistas, gerar reações negativas, encorajar comportamentos

Implementação cara

Perfil do utilizador - linguagem natural

Conhecimento e experiência

capacidade de dactilografia elevada

experiência com tarefa elevada

experiência com aplicação baixa

literacia computacional baixa

Características da tarefa

frequência de utilização baixa

pouco ou nenhum treino

utilização opcional

importância da tarefa baixa

Directivas - linguagem natural

fornecer uma linguagem (restrita) que seja habitável aos quatro domínios

gerar saídas usando formas válidas quanto aos quatro domínios da linguagem

(Exemplo: usar sempre palavras que o sistema reconheça)

Interfaces 3D - usadas cada vez mais em:

Realidade virtual aumentada

3D workspaces

visualização de informação

Problemas:

desorientação do utilizador (no mundo real usamos mais informação)

Exemplos:

- Realidade virtual e Aumentada

  Entretenimento, treino, visualização de dados, revisão de projectos

- Dispositivos de entrada/saída com haptic feedback

  ambiente de pintura com um pincel 3D, inclui ferramentas essenciais. este dispositivo determina a orientação do pincel e fornece força de contacto entre a tela e o pincel

- Prespective wall

visualização de files (tipo e data da ultima modificação)

usa-se 3D para melhor aproveitar o espaço do ecrã

a animação alivia a carga cognitiva do utilizador para o sistema perceptual

Disposição da Informação e Design do ecrã

↳ um mau design do ecrã pode degradar o desempenho do utilizador

necessário ter cuidado

Diretrizes → layout de ecrãs

Disposição Geral

Texto - mensagens, instruções

Número

Técnicas de codificação

Cor

↳ **Disposição da Informação**

incluir toda apenas informação necessária

começar no canto superior esquerdo

agrupar itens logicamente

usar espaços

evitar usar apenas maiusculas

distinguir campos dos títulos

↳ **Texto** - mensagens:

breves e concisas

nivel de detalhe ajustado ao conhecimento do utilizador

afirmativas e construtivas

especificas e compreensivas

↳ **Mensagens de Erro**

esclarecedoras

simples

escritas na linguagem do utilizador

↳ **Numeros**

inteiros devem ser alinhados à direita

reais devem ser alinhados pela vírgula

evitar os zeros à esquerda sempre que desnecessários

dividir os números longos agrupados em grupos de 3 ou 4 digitos

↳ **Técnicas de codificação**

Blinding, Bold, Italic, size, font, underling, shape, sound, color, borders...

usar parcimoniosamente técnicas de codificação

Cor

Utilização da Cor (traz benefícios)

pode facilitar a discriminação em displays complexos

pode aumentar a satisfação

pode enfatizar a organização lógica da informação

chama a atenção para avisos

pode desencadear reações emocionais

⚠ É necessário cuidado - pode degradar o desempenho do utilizador

Directivas - Utilização da cor

usar cor com parcimónia, limitar o número de cores e quantidade

projectar primeiro sem cor

evitar a utilização simultânea de cores saturadas

garantir contraste entre o fundo e o texto

evitar cores saturadas no fundo

não veicular informação apenas através da cor

assegurar que a codificação da cor apoia a tarefa

tornar código de cor o mais evidente possível

permitir ao utilizador controlar o código de cor

ter em atenção o significado cultural das cores

Utilização da cor

codificação através da cor é potencialmente sensível a interacções com o contexto

cores fortes (saturadas) complementares produzem maus resultados

pequenas manchas de cor num fundo neutro sublinham informação

representação de informação quantitativa

Visão da cor

visão humana -> sensível a radiação electromagnética numa peq. gama de comprimento de onda

a cor é percebida em função do comprimento de onda pode variar com as pessoas

2 tipos de visão:

Escotópica - funciona a níveis baixos de luminância, não é sensível ao comprimento de onda

Fotópica - funciona a níveis mais elevados de luminosidade, é sensível ao comprimento de onda da luz.

Daltonismo - deficiência na compreensão das cores

Processo de Percepção da cor

olhos -> sensores ; cérebro -> processador

no olho a luz passa pela íris, é focada pelo cristalino e projecta-se na retina

na retina existem 2 tipos de células sensoriais:

só existe 1 tipo ← bastonetes - visão escotópica } existem em quantidades diferentes e distribuidos de forma diferente na retina

3 tipos ← Cones - visão fotópica

↳ vermelhos - 64%

verdes - 32%

azuis - 2%

objectos percebidos com uma cor que se relaciona com o espectro da luz reflectida

↳ diferentes espectros podem provocar sensações de cores iguais

Existem 2 tipos de produção de cor

↳ aditivos (televisores, monitores)

↳ subtractivos (impressoras)

A cor pode ser descrita como 4 grandezas psicofísicas (relacionam-se de alguma forma com grandezas físicas)

↳ Hue - relacionada com o que geralmente chamamos cor (posição no espectro)

↳ saturação - intensidade da cor

↳ luminosidade - reflector de luz

↳ brilho - objectos emissores de luz

Colorimetria

Qualquer cor pode ser representada pela sobreposição de 3 cores básicas

$$C = a_1R + a_2G + a_3B$$

Modelos de Cor

Existem vários modelos de cor, que incluem um sistema de coordenadas tridimensionais

RGB → modelo standard dos monitores

↳ cubo num espaço tridimensional que inclui todas as cores disponíveis no monitor

pode variar de monitor para monitor

subconjunto de cores visíveis pelo olho humano

não é um espaço perceptualmente uniforme

Outros modelos de Cor

Modelo CMY - Cyan, Magenta, Yellow → sistemas subtractivos

CMY são complementos do RGB

Cores primárias em sistemas subtractivos

Modelo HSV - Hue, saturation value → especificação pela utilizador

plano v=1, contém o modelo RGB

cores complementares com 180° entre si

Modelo HLS - Hue, lightness, saturation

cores saturadas para L=0,5

Design Centrado no Utilizador - Interactive Software Lifecycle

Abordagens complementares ao projecto

- princípios de usabilidade (independentes da tecnologia)
- paradigmas de usabilidade (dependentes da tecnologia)
- directivas e normas (mais específicas)

metodologia de desenvolvimento SW

avaliação

É necessário: conhecer exemplos de sucesso, perceber porque funcionam (princípios e adaptar metodologias adequadas, e testar, re-testar....

O ciclo de vida da engenharia de usabilidade envolve 3 níveis diferentes, em que estes níveis são compostos por tarefas, muitas delas iterativas, depois de cumprir todo este ciclo de vida, então sim consegue-se um produto com excelente usabilidade.

1 nível - Análise de Requisitos

- Perfil dos Utilizadores - estabelece as principais características do utilizador para desenhar o UI
- Análise de tarefas - obter um modelo conhecido no utilizador, extrair os requisitos de usab. do produto
- Est. de Metas de usabilidade - estabelecer objectivos qualitativos e quantitativos de usabilidade
- Plataformas Compatíveis e Restrições - estabelece restrições e compatibilidade da plataforma que limitam as alternativas do desenho.
- Princípios Gerais do desenho - identifica as linhas gerais que podem ser relevantes

2 nível - Desenho, testes e Desenvolvimento [- Nível 1]

- Trabalho Re-engenharia - restruturar o modelo de trabalho actual do utilizador para realizar o potencial de automação, apoiar mais fácilmente os objectivos
- Modelo de Desenho Conceptual - estabelecer regras de base coerentes, elevado nível de estrutura de desenho da interface
- Modelo Conceptual Mock-ups - apoiar a avaliação, refinamento e validação do modelo conceptual
- Modelo interactivo de Avaliação conceptual - avaliar, aperfeiçoar e validar o modelo conceptual

2 nível - Desenho, testes e desenvolvimento [- Nível 2]

- Padrões de Projecto de Screen - estabelecer um conjunto de padrões para definir o estágio de projecto detalhado da interface
- Padrões de Projecto de Screen de Prototipagem - apoiar a avaliação, aperfeiçoamento e validação do projecto
- Padrões de avaliação de desenho de screen interactivo - avaliar, refinar e validar o desenho
- Guia de estilo de desenvolvimento - documento do modelo conceptual de desenho, normas e saída de análise de requisitos

2 nível - Desenho, testes e Desenvolvimento [- Nível 3]

- Detalhes do Desenho de interfaces - desenho de interface de forma completa e detalhada
- Avaliação de interface de Utilizador - avaliar, refinar e validar subconjuntos chave do desenho

3 Nível - Instalação

Feedback do utilizador - obten dados de usabilidade depois do produto ser instalado e utilizado. Informa do desenho para versões mais recentes ou produtos relacionados.

Filosofia Geral da Eng. Usabilidade

Desenho da interface de utilizador é fundamental

a integração da eng. usabilidade com o s/w devem ser adaptados

a análise de requisitos compensa

projecto abordado de cima para baixo → processo estruturado

Design, teste de desenvolvimento deve ser iterativo

ciclo de vida inteiro pode ser "mergulhado" em sub-conjuntos de funcionalidades

existem vários técnicos para realizar cada tarefa

técnicas alternativas têm de tornar o ciclo de vida mais flexível

aplicações optimas do ciclo → requer equipamentos multifuncionais

Design the User Experience - ilustra o caminho para produtos mais usáveis

4 fases:

Análise - cumprir com os principais partes interessados para definir a visão

incluir tarefas de usabilidade no plano

formar uma equipa multidisciplinar → especialização completa

estabelecer metas dos objectivos de usabilidade

realizar estudos

olhar para a concorrência

criar perfis de utilizador

análise de tarefas

cenários de documentos do utilizador

Desenho - debaten conceitos de design e metáforas

desenvolver modelo de fluxo de tabela de navegação

fazer orientações do conceito de design

criar protótipos

realizar testes de usabilidade

criar projecto detalhado de alta fidelidade

testes de usabilidade

padrões de documentos e orientações

Implementação - fazer avaliação heurística

trabalhar em estreita colaboração com equipa → como o projecto é implementado

Realizar testes de usabilidade

Desenvolvimento - usar pesquisas para obter feedback do utilizador

Realizar estudos de campo para obter informações

verificar objectivos através de testes de usabilidade

Modelo para o projecto de Interface de Utilizador

modelos:

Avaliadores - projecto tem características desejadas?

Geradores - contribuem directamente para o projecto

Também no projecto de interface são necessários modelos

vários tipos de modelos podem ser utilizados.

Modelo de utilizador

Análise de tarefas

Interacção de Diálogo

Modelo de sistema

↳ Modelo de Utilizador

* modelos para obtenção dos requisitos do utilizador - contexto social e organizacional

** aspectos relacionados com os processos mentais, perceptuais e motores dos utilizadores

* 1- Obtenção dos Requisitos

parte muito importante na eng. software

foca-se frequentemente nos requisitos funcionais: o que o sistema deve ser capaz de fazer

requisitos não funcionais: é usável, é aceitável

modelos socio-técnicos - reconhecem que a tecnologia surge como parte de um ambiente organizacional.

USTM - User Skills and Task Match

OSTA - Open System Task Analysis

ETHICS - Effective Technical Human Implementation of Computer System

Metodologias Soft System - ênfase na compreensão da situação e não obter uma solução

Projecto Participado - abrange o ciclo de desenvolvimento e incorpora o utilizador como membro activo da equipa

↓

Projecto Participado

os utilizadores são envolvidos como membros da equipa

características:

orientado para o trabalho não para o sistema

colaborativo - o utilizador contribui para as fases

iterativo - sujeito a avaliação e revisão em cada fase

utiliza um conjunto de métodos para transferir a informação entre utilizadores e projectistas

workshops, exercícios a lápis, cenários, story boarding, brainstorming

** Modelos Cognitivos -> representam o utilizador numa interacção com o sistema, i.e., modelam aspectos do conhecimento, intenções ou processamento do utilizador

[objectivos de alto nível -> actividade motora]

Vários modelos cognitivos:

- Hierarquia de objectivos e tarefas (GOMS)
- Modelos Linguísticos
- Modelos físicos e de dispositivo (KLM - Level Model)

GOMS - goals, operators, methods and selections

Goals - o que o utilizador pretende atingir

operators - operações básicas que o utilizador tem de executar para utilizar o sistema

methods - vários possíveis decomposições do objectivo em subobjectivos

selections - regras de selecção dos métodos possíveis (depende do utilizador e do sistema)

[exemplo: salvar um ficheiro]

[exemplo: nas máquinas antigas de ATM deve. 1º o dinheiro e dps o cartão, muitos utilizadores esqueciam-se do cartão]

as regras de selecção devem ser ajustadas ao utilizador dependendo do seu perfil

a análise da decomposição GOMS pode produzir medidas de desempenho

- requisitos da STM (profundidade da estrutura de objectivos)
- ideia de tempo necessário (um tempo por cada operador)

capacidades:

- descreve como os utilizadores experientes desempenham tarefas de rotina
- permite estimar os tempos de um modelo de dispositivos

Limitações:

- não permite estimar tempos de treino de transferência

KLM - Keystroke Level Model

permite fazer previsões do desempenho do utilizador

modela tarefas unitárias de interacção (sequências de comandos simples < 20 segundos)

tarefas têm 2 fases:

- aquisição - construção da representação mental
- execução - recorrendo aos sistemas

A KLM apenas modela na execução -> decomposto em 7 operadores

- K - Keystroke -> varia com a capacidade de dactilografia
- B - Button -> botão pressionado do rato
- P - Pointing - ponto entre o rato e o teclado
- D - Drawing - desenho do uso do rato
- M - Mentally - preparação mental da acção física

R -> System Response - por vezes pode ser ignorado

[existem tempos computados]

os tempos estão relacionados com o chunking e o tipo de utilizador

aplicabilidade limitada ao monodiálogo

apenas pretende fornecer resultados

principal aplicação -> comparação de alternativas

II - Análise de Tarefas

consiste na análise da forma como as pessoas executam os seus trabalhos

o que fazem, o que utilizam, o que precisam de saber

[Exemplo: apanhar o: casa

ir buscar o apanhador

acertar a terminação adequada

apanhar comportamentos

saco bem cheio, esvaziar → é preciso saber de apanhadores

arrumar o apanhador]

Vários abordagens

decomposição da tarefa

técnicas baseadas no conhecimento

análise baseada em relações entre entidades

Principais ferramentas: observação

Principais Objectivos: produção de materiais de treino e documentação no caso da observação dos sistemas existentes

na projecção de um novo sistema -> ajuda na obtenção de requisitos do sistema

Diferença entre Análise de Tarefas e GOMS

a análise de tarefas descreve as tarefas que o utilizador executa no ponto de vista exterior

Modela também aspectos que não fazem parte do sistema computacional

O GOMS pretende compreender os processos cognitivos do utilizador ao executar a tarefa

análise de tarefas -> mais usado no início do ciclo de vida do s/w

GOMS -> mais usado nas tarefas de utilização

Decomposição de Tarefas

A HTA (Hierarchical Task Analysis) é uma das técnicas de análises de tarefas mais usadas. Resulta:

hierarquias de tarefas e sub-tarefas

planos com a ordem e condições de execução

Quando se deve parar a decomposição?

Parar se $P \times C$ < dado valor

probabilidade de cometer um erro ↑ ↳ custo do erro

Ou seja: tarefas simples não necessitam de ser decompostas a não ser que sejam críticas.
[exemplo: preparar uma chávena de chá]

Tipos de Planos do HTA

Sequência fixa

Tarefas opcionais

Esperar por acontecimentos

Ciclos

partilha do tempo

arbitrários

Mistura de vários tipos

A forma e qualidade do resultado final → depende muito da experiência do analista

Diferentes analistas produzem diferentes análises (sobretudo nos detalhes) → variam conforme o objectivo do analista

Fontes de Informação Para Análise de Tarefas

os resultados das tarefas não podem ser melhores que os dados originais (garbage in garbage out)

o processo de análise provoca novas questões → deve haver vários períodos de recolha e análise de dados

Fontes de informação: documentação, observação, entrevistas, ...

↳ Documentação

a melhor fonte são os manuais, livros de instruções, manuais de treino, etc.

as fontes descrevem o que as pessoas supostamente fazem e não o que efectivamente fazem

descrevem as funções do equipamento, não o modo de utilizar

esta informação deve ser usada com base para a observação directa e entrevistas aos utilizadores

atenção na forma como as entrevistas são feitas

↳ Observação

é sempre necessário observar (formal ou informal) para compreender as tarefas

um bom ponto de partida → observação dos utilizadores e leitura de documentação

deve seguir-se uma observação mais intensa: laboratório, campo, passiva, activa (perguntas, part-walk...)

↳ entrevistas

forma directa e rápida de obter info. sobre as tarefas → especialistas especialista no domínio

perguntas mais gerais

perguntas mais específicas

descrição de uma tarefa na forma ~HTA

Utilização da Análise de tarefas - vários utilizações possíveis:

manuais e materiais de ensino → observar os utilizadores a usarem o sistema

projecto de alto nível de um sistema

projecto detalhado de interfaces de utilizador de um sistema

(→ análise de tarefas contribui para o projecto de um novo sistema)

III. Notação de Diálogo

diálogo - nível sintáctico da IHC

↳ relacionado com:

- semântica do sistema (o que faz)
- apresentação do sistema (como parece)

descrições formais podem ser analisadas quanto:

- ações inconsistentes
- dificuldades em inventar situações
- itens em falta
- erros potenciais.

as notações usadas podem ser:

- gráficas - mais fáceis de compreender
- textuais - mais adaptadas à análise formal)

Mais comuns:

- Gráficas: State Transition Networks
  - Petri Nets
  - State Charts ...
- textuais - gramars, production Rules

State transitions Networks

têm sido muito usados para descrever diálogos

(exemplo: ferramenta simples para desenhar um dispositivo, tem:

- superfície de desenho
- menu com opções

A STN representa bem diálogos:

- sequencias
- com escolhas
- iterativos

falham na descrição de diálogos com partes concorrentes - podem ser activadas em simultâneo

STN para caixa de Diálogo com 3 toggles

Se considerarmos cada toggle individualmente temos STN's simples - 2 estados

não explica o que acontece quando o utilizador actua em 2 botões

Se o utilizador actua em 2 botões temos 4 estados

se o utilizador actua em 3 botões tem 8 estados

N botões → 2^N estados

Grande limitação da STN para representar interfaces de utilizadores

Petri Nets

um dos formalismos mais antigos das Ciências de computação

usados para:

Síntese de circuitos assíncronos

programação concorrente

usados para especificar aspectos dos interfaces de utilizador

numa "petri net" o sistema pode estar em vários "estados" em simultâneo

definem-se:

Places (círculos)

Counters (marcas pretas)

Places for user input (elipses)

Quando todos os places com arcos para uma transacção tiverem counters, esta transição pode ocorrer, sendo os counters removidos e colocados na saída de transacção

State Charts

podem ser vistos como uma forma de STN

especificam graficamente sistemas reactivos complexos e tratam problemas de concorrência

hierárquicos, num único diagrama mostram as alternativas e representam acb. concorrente

Dispositivos de Entrada e Saída

Teclados

Disposição das teclas

Características operacionais:

Tamanho, ângulo, área de descanso das mãos, espaçamento entre teclas, força de actuação nas teclas, superfície de topo das teclas, gama de deslocamento das teclas, feedback de activação, arabamentos de topo, indicadores de home row

Este podem ser: alfanuméricos, numéricos, funções, cursores

Os teclados ergonómicos evitam o RSI, WRULD, KRP

Keyless Keyboard pretende:

eliminar o movimento dos dedos

minimizar o movimento do pulso

ser fácil de aprender

Dispositivos Apontadores

São usados para: apontar e seleccionar um alvo, desenhar, posicionar objectos, orientar e rodar objectos, definir percursos entre pontos, manipular texto, etc...

Variam na eficiência com que executam estas tarefas e dividem-se em: (segundo Shneiderman, 98)

Controlo directo

touch screen

light pen

Controlo Indirecto

rato

track ball

mesa digitalizadora

joystick (track point)

touch pad

↳ Ratos - podem ser:

mecânicos

ópticos

variam em precisão, forma, nº de botões...

Enviam coordenadas relativas

Vantagens

Relação directa entre o movimento da mão e do cursor (distância, velocidade, direcção)

Permitem controlar a velocidade

Movimento contínuo em todas as direcções

Desvantagens

- requerem movimento da mão entre teclado e rato

espaço adicional

aprendizagem de coordenação mão-olho

↳ Mesas Digitalizadoras

Electromagnéticas

Sónicas

Resistivas

Fornecem coordenadas absolutas

Variam em: resolução, tamanho, precisão, preço

Vantagens

Relação directa entre o movimento da mão e do cursor (distância, velocidade, direcção

controlar a velocidade

movimento contínuo em todas as direcções

resoluções elevadas

Desvantagens

requerem movimento da mão entre o teclado e a mesa

espaço adicional

↳ TouchScreens

resistivos - mais antigos, mais baratos, mais usados

ópticos

sónicos - padrões de onda estacionários (+ caros, + recentes)

capacitivos

Vantagens

relação directa entre o movimento da mão e do cursor (distância, velocidade, direcção)

não requerem espaço adicional

permitem movimento contínuo em todas as direcções

Desvantagens

o dedo pode ser muito grande para a precisão pretendida

cansativos em períodos longos

o dedo pode ocultar parte do ecrã

o ecrã pode sujar-se

outros problemas: paralaxe (ópticos), temperatura e humidade (capacitivos), desalinhamento (resistivos)

Directivas - dispositivos de entrada

Escolher o dispositivo com base em análise e teste cuidado da tarefa

Minimizar movimentos entre mãos e olhos

usar teclas de cursor para tarefas que envolvam

muito manipulação de texto

movimento através de arrays de objectos discretos

usar touch screen quando

não há treino

os alvos são grandes, discretos e espalhados

a frequência de utilizam é pequena

o espaço é importante

não há entrada de texto

Reconhecimento de voz

Primeiro sistema de voz desenvolvido em 1972

Está a ser mais utilizado

Tem desafios de 2 tipos: tecnológicos, factores humanos

Contudo tem melhorado bastante, existem ainda problemas tecnológicos

no processamento de discurso contínuo
na extensão do vocabulário
dependência em relação ao utilizador
na taxa de erro aumenta com cansaço, stress, constipações...
ambientes ruidosos

} a investigação contínua por forma a ultrapassar estes problemas

Vantagens - quando o utilizador:

tem deficiência física

tem de se deslocar

tem os olhos ocupados

está em ambientes apertados ou com pouca visibilidade

Desvantagens

a voz é transiente (transitória)

não existe mecanismo natural de feedback

perturbam os outros pessoas

falta de privacidade

podem ser mais lentos e cansativos (sobreconhecem a STT)

Objectivos - Reconhecimento por voz

fornecer estrutura ao diálogo de saída para guiar a entrada

usar vocabulário distinto e familiar

considerar a entrada de voz quando as limitações da tecnologia forem aceitáveis:

treinar os utilizadores

ruído ambiente

extensão do vocabulário

custo dos erros

ter em atenção quando se deve utilizar ou não o reconhecimento de erros

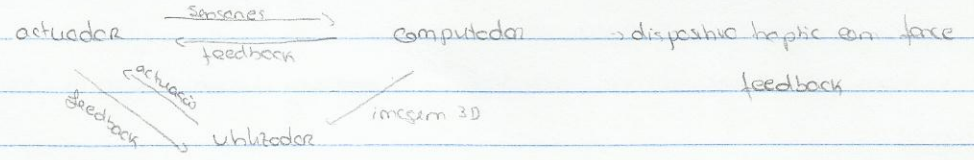
Eye Trackers

fornece como saída:

saída de vídeo com movimento de cursor

coordenadas x,y

Raio da pupila



O futuro?

- Parece provável que venham a ser mais utilizados

reconhecimento por voz, entrada gestual, entradas com as emoções, apontadores 3D, interacção por voz, wearable devices, ambientes wide-body, feedback tátil

Displays

Existem ecrãs com base em vários tecnologias: CRTs, plasmas, LCD's, electroluminescentes, electroforéticos, 3D, etc...

São geralmente usados em sistemas nosten scan

	CRT	LCD	Plasma
Vantagens	Brilho Preço	volume, peso consumo	espessura, tamanho brilho
Desvantagens	volume, peso consumo	preço, brilho e ângulo de visão	preço consumo

[Ecrãs Electroluminescentes - usam um material que emite luz quando sujeito a um campo electrico alto

LCD's - um pixel num LCD colestérico tem 3 estados (reflexivo, transparente, intermédio

Dispositivos Electroforéticos

Têm propriedades do papel:

permanece uma imagem visível por reflexão de visão elevada

flexível e barato

Não pode apagar-se e escrever-se através de um sinal elétrico

Organic LEDs (OLEDs)

permitem obter displays: flexíveis, mais eficientes, mais brilhantes que os LCD's

Estereoscopia: disparidade binocular

a disparidade binocular devido à diferente posição dos dois olhos fornece uma pista visual de profundidade: estereopsis ou visão estereo

Displays Autoestereoscópicos

permitem ver imagens 3D sem necessidade de óculos

permitem a visualização em vários pontos do espaço

são ainda dispendiosos

Dextroscope

permite ao operador ver objectos 3D enquanto coloca as mãos no ambiente virtual, para os manipular.

HMDs - Head Mounted Displays

podem ser totalmente imersivos ou see-through

podem incluir ou não head-tracker

podem incluir ou não head-phones stereo

factores determinantes da qualidade e preço (resolução, campo de visão, tamanho e peso)

Boom

Usa tubos de raios catódicos

Confortável de utilizar, não é necessário vestir-lo

tracker embutido

Resolução elevada; CORO

Síntese de voz

Existem diferentes tipos

Discurso digitalizado - gravado e digitalizado

Discurso sintetizado - concatenando sons básicos

Existem dificuldades técnicas na síntese de voz devido à riqueza da voz humana

diferentes sinais de pronúncia

sentido pode ser alterado pela entoação

variações de entoação podem reflectir estados emocionais

A qualidade de um sintetizador de voz implica muito mais do que simples inteligibilidade

Vantagens e Desvantagens: mesmas do que o reconhecimento por voz

Directivas - Síntese de voz

considerar como alternativa quando o utilizador se move, mãos e olhos ocupados

evitar em ambientes abertos, quando a privacidade e segurança são importantes

evitar em frequência de utilização elevada

começar por palavras não críticas que forneçam contexto

usar taxas de saída de aprox. 180 palavras por minuto

colocar 1º objecto, depois solução

possibilidade de repetir mensagens.

Conclusões

não usar a tecnologia apenas porque é nova

é necessário compreender a sua usabilidade para diferentes tipos de utilizadores e tarefas

Avaliação de interfaces de utilizador

Estilos - laboratório (complementares, sempre possível usar os 2)

campo

Métodos - Analíticos (sem utilizadores)
- avaliação heurística
- cognitive walkthrough
- M. baseados em revisão
- M. baseados em modelos

guideline review
consistency inspection
(complementares - sempre que possível usar 2 ou +)

Empíricos (com utilizadores)
- experiências controladas
- observação
- inquérito (entrevistas, questionários)

Na escolha dos métodos de avaliação a usar devemos ter em conta aspectos:

etapa, estilo, ~~se é~~ objectivo, medida, informação, imediato, ~~se é~~ intrusivo, tempo, equipamento, perícia

(ver quadros -> último slide)