

Interação Humano- Computador

Resumos
2014/2015

Bárbara Jael | 73241

IHC

2º Ano
2º semestre

Beatriz Sousa Santos
bss@ua.pt
gab. 1.17, IEETA

Paulo Dias
paulo.dias@ua.pt
gab. 0.05, IEETA

Avaliação

Exame (50%)

Trabalhos de grupo (50%)

- Apresentação de um PAPER - 15 min (10%)
- trabalho n.º 1 - heuristic evaluation (10%)
- trabalho n.º 2 - design and implementation (30%)

Apresentação do PAPER (mais de 8 páginas)

vantagens?

limitações? - podem ser ultrapassadas?

"Resumo"

apreciação crítica

$$\begin{aligned} \rightarrow 12 &= 2 \times 6 = \sqrt{2 \times 2} \times 3 \quad \text{fz} \sqrt{2} \times 3 \\ 6 &= 2 \times 3 = \sqrt{2 \times 2} \times 3 \quad \text{fz} \sqrt{2} \times 3 \\ \rightarrow 24 &= 2 \times 12 = \sqrt{2 \times 2} \times 2 \times 3 \quad \text{fz} \sqrt{2} \times 3 \\ \rightarrow 20 &= 5 \times \sqrt{2 \times 2} \end{aligned}$$

20	2
10	2
5	5
1	

DESIGN SIST. INTERATIVOS

Projeto participativo \rightarrow os utilizadores ~~estão~~ envolvidos como especialistas do domínio

- é orientado ao trabalho, e não orientado ao sistema
- é colaborativo — os utilizadores contribuem em todas as fases
- é iterativo — o design é avaliado e revisado em cada fase

Técnicas:

- brainstorming
- cenários
- fazer story boards
- workshops
- exercícios de papel e lápis

os modelos cognitivos ¹⁹⁸³ GOMS e ¹⁹⁸⁰ KLM modelam aspetos do conhecimento, intenções ou processamento do utilizador
(\hookrightarrow ambos propostos por Card, Moran e Newell)

GOMS — "Goals, Operators, Methods and Selections"

goals (objetivos): o que o utilizador pretende obter

operators (operadores): operações básicas que o utilizador tem de executar p/ usar o sistema; pode afetar ou não o sistema (ex) clicar numa tecla ou ler uma mensagem)

methods (métodos): possíveis decomposições de um objetivo em sub-objetivos (ex) seleccionar a opção "Guardar" ou clicar em "ctrl + S")

selections (selecções): Regras de selecção de possíveis métodos (levar em conta o tipo de utilizador e o estado do sistema)

problema de facto: o objetivo ser cumprido antes de todas as tarefas estarem cumpridas

(ex) • levantar dinheiro no multibanco: os multibancos antigos entregavam primeiro o dinheiro e só depois a cartão, visto que o objetivo do utilizador era o dinheiro, muitos cartões ficaram esquecidos na máquina

• tirar fotocópias: como, normalmente, as cópias ficam disponíveis ao utilizador (objetivo) antes de este remover o original, muitas vezes o utilizador pega nas cópias e vai embora sem o original

Ao analisar a estrutura de uma decomposição GOMS, podemos medir aproximadamente:

- a capacidade de memória a curto prazo (pela "profundidade" da estrutura do objetivo)
- o tempo necessário (p/ cada operador)

Apesar de poder descrever adequadamente como utilizadores experientes efetuam certas tarefas de rotina, tem limitações:

- não dá informação sobre o conhecimento do utilizador p/ estimar a formação (treino) e o tempo de transferência

	Análise de tarefas	GOMS
modelos:	aspectos do mundo real não faz parte do sistema	processos cognitivos do utilizador ao realizar a tarefa
ponto de vista:	externo	interno
ciclo de vida do SW:	fases iniciais	avaliação

KLM — "Keystroke-Level Model"

modela tarefas de interação unitária (sequências de comandos simples < 20s) na fase de execução

- (ex) • mudar a fonte de uma palavra
• usar "search and replace"

operadores:

B D H K M P R

- **K** — Keystroke (varia c/ a habilidade de escrita)
- **B** — pressionar botão no rato
- **P** — Apontar p/ um alvo (lei de Fitts)
- **H** — mudar entre teclado e rato (homming)
- **D** — desenhar usando o rato
- **M** — preparar mentalmente p/ uma ação física
- **R** — Resposta do sistema (pode ser ignorado)

é estimado um total do tempo que o utilizador leva p/ completar uma tarefa

- (ex)
- 1) Mover mão p/ o rato → H [rato]
 - 2) Posicionar cursor depois do erro → PB [esp]
 - 3) voltar c/ a mão p/ o teclado → H [teclado]
 - 4) eliminar caractere errado → MK [eliminar]
 - 5) adicionar u correto → K [char]
 - 6) reposicionar cursor → H [rato] MPB [esp]

$$T_{total} = 2T_K + 2T_P + 2T_B + 3T_H + 2T_M$$

A sua aplicação é limitada a micro-diálogos — apenas possibilita resultados aproximados
principal aplicação: é alternativo de comparação

Lei de Fitts (1954)

modelo empírico que explica características tradeoff da velocidade de precisão do movimento do músculo humano c/ alguma analogia ao teorema da capacidade de canal de Shannon

estima o tempo médio que um utilizador leva a seleccionar um alvo considerando a distância do cursor e a largura do alvo

$$T = \underbrace{a+b}_{\text{coeficientes}} \cdot \log_2 \left(2 \frac{D}{W} \right)$$

• quanto mais largo for o alvo, mais fácil será seleccioná-lo (não é necessário precisão)

• quanto mais longe o alvo estiver do cursor, mais vai demorar

Análise de tarefas: é a análise de como as pessoas realizam o seu trabalho → o que fazem

→ " " usam

→ " " precisam saber

esta análise é feita na parte da análise de requisitos

abordagens:

• decomposição de tarefas — divide as tarefas em sub-tarefas que devem ser realizadas numa ordem específica

• baseado em conhecimento — considera o que o utilizador precisa saber sobre os objetos e ações envolvidos na realização da tarefa e como o conhecimento é organizado

• baseado em relações — foca atores e objetos, as relações entre eles e as ações que realizam

HTA - "Hierarchical Task Analysis"

- produz:
- uma hierarquia de tarefas e sub-tarefas
 - planos e/ou as condições de sequência e execução

Regra de paragem da decomposição:

PARAR se $P \times C < \text{um dado valor}$

P → probabilidade de erro
 C → custo de erro

tarefas simples
não precisam de
decomposição a não
ser que sejam críticas

- fontes de informação p/ análise de tarefas:
- documentação
 - observação
 - entrevistas
- LA especialistas
- dispendioso

A qualidade dos resultados desta análise nunca será melhor que a informação original — "garbage in, garbage out"

São necessárias várias fases de captação de dados e análise, pois o processo de análise geralmente levanta novas questões

Aplicações da análise de tarefas:

- manuais e materiais de instrução ← os utilizadores foram observados enquanto realizam tarefas ao usar o sistema
 - desenho do sistema de alto-nível
 - design detalhado da interface do utilizador do sistema
- ← a análise de tarefas contribui p/ o design do novo sistema

notação de diálogo: nível sintático da IHC

está relacionada e/

- semântica do sistema (o que faz)
- apresentação do "u" (como o faz)

descrições formais de diálogo podem ser analisadas como

- ações incoerentes
- dificuldade em reverter erros
- falta de itens
- potenciais erros

vantagens:

- notação gráfica — mais fácil de compreender
- " textual — " adequada a análise formal

notação gráfica, vs. notação textual

- State Transition Networks
- " Charts
- Petri Nets
- Flow Charts

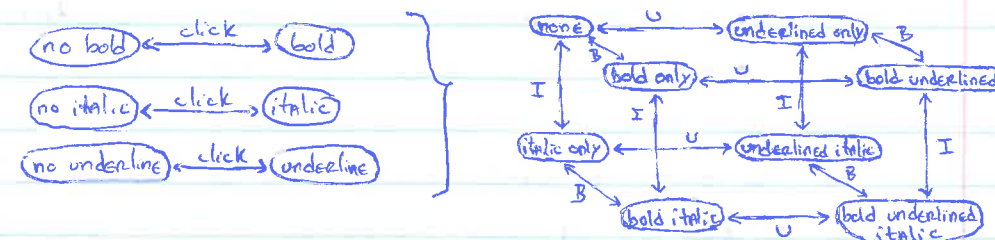
- gramáticas
- regras de produção

State Transition Networks (STN)

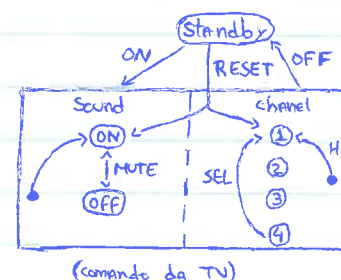
- adequado p/:
- diálogos sequenciais
 - " c/ escolha
 - " iterativos

falha em diálogos simultâneos

(ex) pôr um texto a negrito, itálico e sublinhado



State Charts



foram desenvolvidos p/ especificar graficamente sistemas reativos complexos e podem ser usados e/ou sistemas concorrentes

apesar de serem hierárquicos, mostram num único diagrama as alternativas de atividade simultânea

AVALIAÇÃO USABILIDADE

Avaliação heurística → método de avaliação analítico (pq é desenvolvida por peritos) proposto por Nielsen e Molich em 1990

debriefing (relatar resultado) → discutir características gerais da interface do utilizador e sugerir possíveis melhoramentos p/ resolver os principais problemas de usabilidade

deve/pode ser usada desde o início do desenvolvimento, pois a compreensão da interface irá afetar a usabilidade do produto final

os Analistas:

- devem trabalhar independentemente numa primeira fase, pois diferentes analistas detetam problemas diferentes (e depois consolidam essas descobertas num relatório)
- deverão ser 3 a 5, porém o nº exato depende de uma análise de custo-benefício (situações críticas precisam de muitos analistas)

limitações:

- método subjetivo
- envolve vários peritos em usabilidade
- não consegue encontrar todos os problemas de usabilidade

p/ minorar o facto de ser um método subjetivo, devem ser feitos muitos testes — quantos menos se tiverem, menos conclusiva é a avaliação

um problema de usabilidade é que, por vezes, os atributos entram em conflito. Por exemplo, a ^{learnability} e a eficiência podem afetar-se negativamente. Um sistema deve ser projetado cuidadosamente se tiver que satisfazer esses dois atributos — o uso de aceleradores normalmente resolve o conflito.

método de guerrilha → um método mais rápido e menos dispendioso p/ assegurar um ciclo de feedback que pode ser usado p/ avaliar e melhorar a formação (treino) de intervenientes

não pretende ser uma medida completa de eficiência

feedback qualitativo torna-se repetitivo após 5 ou 6 utilizadores e, portanto, trabalhar c/ uma amostra pequena pode trazer feedback de design significativo

heurísticas de Nielsen

- consistência e standards
- correspondência entre o sistema e o mundo real
- ajuda e documentação
- reconhecer em vez de relembrar
- estética e design minimalista
- controlo e liberdade do utilizador
- prevenção de erros
- ajuda a reconhecer, diagnosticar e recuperar de erros
- visibilidade do estado do sistema
- flexibilidade e eficiência de uso

c/ estas propostas, encontram-se problemas que talvez não seriam identificados em testes de utilização

ex) tipografia misturar letra maiúscula e minúscula — viola a heurística da consistência e acaba por atrapalhar os utilizadores _(confundir)

os graus de gravidade p/ cada problema

- permitem corrigir os erros mais urgentes
- combinam: — a frequência
— o impacto
— a persistência

a gravidade é calculada pela escala 0-4 (0K — gravíssimo)

Cognitive Walkthrough (1992, Wharton)

É um método analítico, pois não envolve utilizadores

baseado no facto de que os utilizadores normalmente preferem aprender a mexer num sistema utilizando-o, ao invés de ler manuais de instruções

É um método que produz resultados rapidamente e c/ pouco custo e deve ser aplicado numa fase inicial (antes de começar a escrever código)

focado em avaliar a ^{learnability} ~~aprendizabilidade~~ (o quão fácil é p/ novos utilizadores realizarem tarefas c/ o sistema)

problemas comuns:

- o avaliador não sabe como executar a tarefa
- envolver uma extensa análise e documentação e muitas vezes são detetados muitos potenciais problemas, o que consome muito tempo

Ética em métodos empíricos

- pedir consentimento explícito
- confidencialidade
- segurança (evitar qualquer risco)
- liberdade (os utilizadores podem desistir quando quiserem)
- limite de stress

num teste de usabilidade (método empírico) recorre-se à observação e a perguntas

CIF — "Common Industry Format"

especifica o formato de apresentação dos resultados duma avaliação sumativa

o tipo mais comum de avaliação de usabilidade é formativa, ou seja, concebida p/ identificar os problemas que podem ser corrigidos

a avaliação sumativa produz métricas de usabilidade que descrevem o quão utilizável é um produto quando utilizado num contexto particular

protótipo em papel (de baixa fidelidade)

- vantagens:
- rápido e sem custo
 - extrai o feedback dos utilizadores sobre aspectos gerais
 - fácil de modificar mesmo durante testes
 - de valor p/ testar o modelo conceptual da interface de utilizador

Adequado especificamente p/ obter feedback sobre

- conceitos e terminologia
- navegação
- conteúdos
- funcionalidades

não precisa ter muito detalhe nem ser muito realista

think-aloud

a avaliação formativa pretende saber que detalhes da interação estão bem e como se pode melhorar o design da interação

este método de observação ajuda a fazer uma avaliação formativa nos testes de usabilidade: é pedido ao participante que vá pensando em voz alta ("think-aloud") enquanto usa o sistema durante o teste de usabilidade, verbalizando as suas ações p/ serem tomadas notas

vantagem: ao contrário do que acontece numa avaliação sumativa, o resultado ajuda a moldar o produto

desvantagem: pensar em voz alta abrandar o processo de pensamento e, consequentemente, incrementa a atenção, o que pode impedir a determinação de erros que aconteceriam numa situação real

experiências controladas

① definir:

- a hipótese (ex) o método X é melhor que o Y
- o input (ex) método de navegação [variável independente]
- o output (ex) performance e opinião [variável dependente]
- design da experiência (ex) dentro-de-grupos
- outras variáveis

② seleccionar os participantes

③ preparar a documentação

- lista de tarefas e dificuldade } p/ os utilizadores
- questionário final
- lista de tarefas p/ o observador tirar notas

④ fazer um teste piloto

⑤ tratar da logística

entre-grupos vs. dentro-de-grupos

dentro-de-grupos: todos os participantes usam ambos os métodos (entre-grupos é tipo o que fizemos c/ a nokia)

protocolo de uma experiência controlada

- ① breve explicação dos métodos
- ② treinar antes de usar cada método
- ③ medidas de desempenho registadas pelo sistema
- ④ satisfação e opinião dos utilizadores obtidas através do questionário

métodos de avaliação de usabilidade baseados em modelos

métodos que procuram aforar o grau de usabilidade de um sistema a partir de modelos da sua interface e/ou dos seus utilizadores, ou seja, de representações que capturam e modelam a forma como os utilizadores interagem c/ o sistema

(ex) ~~um destes~~ modelos: GOMS e KLM

avaliações de campo vs. de laboratório

avaliações de laboratório são mais apropriadas p/ prevenir situações de perigo

avaliações de campo são mais apropriadas p/ quando o contexto em que ocorre a experiência é essencial p/ os resultados