

Linguagem de comandos

- As linguagens de comandos tiveram origem nos sistemas operacionais e se caracterizam pelo seu impacto direto nos dispositivos físicos ou na informação.
- Linguagens de comandos podem consistir de comandos simples ou ter sintaxes complexas.
- Elas podem incluir comandos tão expressivos quanto crípticos, como o comando a seguir, do UNIX, que elimina as linhas e branco de um arquivo: `grep -v ^$ filea > fileb`
- Podem oferecer *prompts* enxutos ou se aproximar de sistemas de seleção por menus.

```

C:\WINDOWS>cd ..

C:\>dir c:\users\laura

O volume da unidade C nao possui nome
O número de série do volume é 3561-13F7
Diretório de C:\USERS\LAURA

.          <DIR>      12/03/96  13:40
..         <DIR>      12/03/96  13:40
NELSON  TXT          3.422 12/03/96  13:53
NELSON  WRI          4.224 12/03/96  13:45
PLNING  TXT          2.399 01/04/96  12:21
INFEDU3 DOC         997.888 09/04/96  9:13
        6 arquivo(s)  1.007.933 bytes
                        432.504.832 bytes livres

C:\>copy a:infedu.doc c:

Nao preparada - Erro lendo unidade A
Anular, Repetir, Falhar?a

```

Figura 3.3: Exemplo de linguagem de comandos (DOS)

Vantagens

- Flexibilidade;
- Permite iniciativa do usuário;
- Dá maior poder a usuários experientes;
- Potencialmente rápido para execução de tarefas complexas;

Desvantagens

- Requer treino substancial;
- Requer memorização de comandos e parâmetros;
- Dificulta retenção de informação;
- Tratamento de erros pobre (característica não intrínseca, embora tradicionalmente associada).

Diretrizes de projeto

- Crie modelo explícito de objetos e ações;
- Escolha nomes significativos e distintivos;
- Especifique os comandos de forma consistente (hierarquia, ordem dos argumentos, ação-objeto);
- Forneça regras de abreviação consistentes;
- Forneça a possibilidade da criação de macros para usuários experientes.

Design de linguagens de comandos

O primeiro passo para o projeto de uma linguagem de comandos é a **especificação da funcionalidade** completa do sistema, concretizada em uma relação de ações e objeto da interface.

Esta relação deve levar à determinação da **sintaxe da linguagem**,

assim como à determinação das escolhas **lexicais**, que devem ser realizadas de forma sistemática.

Projeto de linguagens de comandos: parâmetros

comando: função

argumentos: objetos a serem manipulados

qualificadores da função

delimitadores: marcadores sintáticos (brancos, vírgulas e outros)

a) Consistência da especificação

Siga a estrutura sentencial (lingüística) na descrição da linha de comando:

Exemplo: “Salve (a mensagem) identificador-mensagem (como) nome”

salve mensagem nome

Em última análise, escolha uma ordem e seja consistente com ela!

b) Uso de palavras significativas X símbolos

Palavras significativas são de fácil interpretação.

Exemplo: Change “OK” to “Ok”

Símbolos dão eficiência e rapidez para usuários experientes.

Exemplo: RS;/OK/,/Ok/*

c) Uso de estrutura hierárquica e termos congruentes

estrutura hierárquica: **verbo + objeto + qualificador**

congruência: uso de pares de opostos e partições

Exemplo: avançar/voltar; direita/esquerda/acima/abaixo

d) Estratégias de abreviação

- truncamento (n 1^{os});
- eliminação de vogais;
- primeira e última letra;
- outros padrões de abreviação;
- Deve ser dada atenção ao resultado e ao som da abreviação (aspectos culturais).

Às vezes é necessário combinar duas regras. Nesses casos, é importante deixá-las aparentes e aplicá-las sempre na mesma ordem!

Linguagem natural

- O estilo de interação por Linguagem Natural pode ser considerado uma variação das linguagens de comandos.

*Qual é o salário de Jorge García?
O salário de Jorge García é de R\$ 1800,00.
Em que departamento ele trabalha?
Ele trabalha no departamento de Compras.*

Figura 3.4: Exemplo de interface me linguagem natural

Vantagens

- Elimina necessidade de aprender sintaxe;
- Diminui distância homem-máquina através do uso de código familiar.

Desvantagens

- Linguagem natural pode ser ambígua;
- Disparidade entre competência lingüística do usuário e do sistema é intransponível;
- Contexto não fica naturalmente aparente.

Manipulação direta

O estilo de interação por Manipulação Direta tem origem no lançamento do paradigma do Modelo de Mundo na interface do Macintosh.

Ele se caracteriza pela **representação, na tela, dos objetos do mundo modelado**, que são **manipulados pelo usuário como se estivessem no mundo da aplicação**.

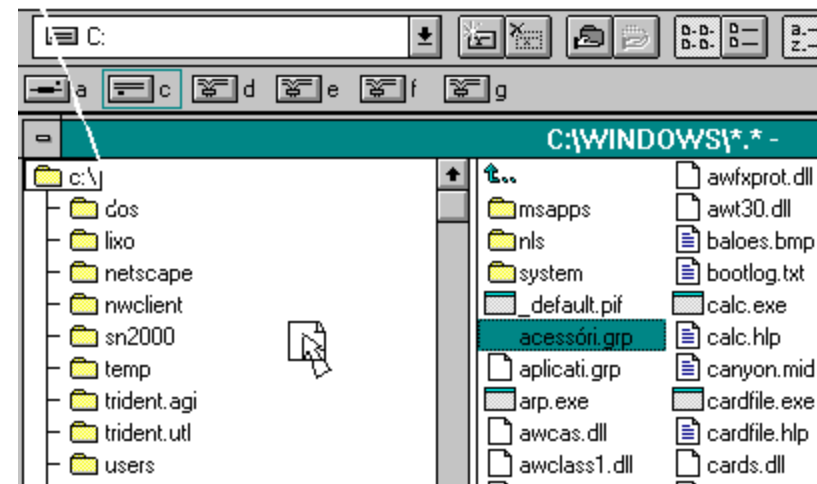


Figura 3.5: Exemplo de manipulação direta (interface do WINDOWS)

Vantagens

- Objeto de interesse e tarefa representados visualmente;
- Facilidade de aprendizado;
- Facilidade de retenção;
- Prevenção de erros;
- Encoraja exploração do ambiente;
- Dá a sensação subjetiva de envolvimento e satisfação.

Desvantagens

- É de uso difícil em ambientes que envolvem conceitos abstratos e significados finos;
- não acomoda certos tipos de aspectos do objeto. Modificações feitas no conteúdo, propriedades internas (tais como permissões de acesso), não são representadas.
- É dependente de cultura;
- Requer facilidades gráficas.

Interfaces Gráficas e de tipo WIMP

Há dois conceitos normalmente identificados com o de interfaces por manipulação direta: o de interfaces gráficas e o de interfaces de tipo WIMP.

A rigor, as interfaces gráficas são aquelas que têm capacidades gráficas, isto é, alta precisão de representação visual.

As interfaces de tipo **WIMP** (**W**indows, **I**cons, **M**enus and **P**ointing devices – janelas, ícones, menus e dispositivos de apontamento), que imperam nas aplicações para computadores pessoais, são interfaces mistas que integram as capacidades inerentes aos dispositivos em questão.

A interface de manipulação direta ou gráfica como a melhor opção

Embora as interfaces gráficas tenham aspectos que as tornam em geral melhores do que as outras, não se pode afirmar que, por ser gráfica, uma interface seja necessariamente boa.

Mas há aspectos nos quais elas são neutras. Assim como qualquer tipo de interface, as interfaces gráficas necessitam verificar os requisitos de uma boa interface, sem os quais elas podem chegar a ser tão ruins como qualquer outra.

Exemplos deste tipo de erro são a falta de consistência no uso de rótulos funcionais, nas ações das funções, e nas mensagens de erro.

Exemplo de falha que podem comprometer “o melhor estilo”

Uso do termo “excluir” em um ambiente e “remover” em outro, ambos para rotular a função de eliminar um objeto;

Uso do termo “excluir” para a função de eliminar um tipo de objeto e “remover” para a mesma função agir sobre outro.

*“Tem certeza de que deseja **excluir** a pasta trabalho?” “Tem certeza de que deseja **remover** o arquivo ata?”*

Ao mesmo tempo, as interfaces de manipulação direta têm problemas sérios no tratamento de conceitos abstratos, de nuances de significado e do caráter dinâmico (ações).

Além disso, contrariamente à crença, elas são, sim, dependentes de cultura.

Fatores a influenciarem a seleção do melhor estilo de interação

Aspectos ligados à tarefa

Se houver entrada de dados pesada,

então **formulário** ou **linguagem de comandos**;

Se existir versão da informação em papel a ser mantida,

então **formulário**;

Se existir notação formal familiar (ex.: matemática, física etc.),

então **linguagem de comandos**;

Se existir representação visual bem-definida e um número modesto de objetos e operações conseguirem representar o domínio de ação,

então **manipulação direta**;

Se forem requeridas muitas decisões e a seleção precisar ser feita a partir de um espaço não-familiar e grande,

então **seleção por menus** (2 a 12 itens) ou **linguagem de comandos**;

Se o usuário não tiver habilidade para digitação,

então **seleção por menus**, **manipulação direta** ou **linguagem natural** (com processamento de voz);

Se usuário tiver nível de instrução pobre,

então **manipulação direta** (bem guiada) ou **linguagem natural** (com proc. de voz);

Se a exploração do sistema e a intuição forem metas importantes,
então **manipulação direta**.

Aspectos ligados ao nível de conhecimento do usuário - estilo de interação

Se usuário for novato,

então **seleção por menus** ou **manipulação direta**;

Se usuário tiver conhecimento modesto sobre o domínio da tarefa e certa capacidade computacional ,

então **seleção por menus, manipulação direta** ou **formulário**;

Se for perseguido o uso intermitente,

então **seleção por menus, manipulação direta, formulário, linguagem de comandos com auxílio *on-line* ou material de bolso, ou linguagem natural**.

Se usuário utilizar o sistema com frequência (usuário experiente),

então **linguagem de comandos com atalhos, seleção por menus com digitação (*type-ahead*), manipulação direta com atalhos adequados** ou **formulários com *display* conciso**.

Combinação de estilos de interação

A apresentação de regras para a seleção de estilo de interação é meramente didática. Na prática, os projetistas combinam estilos diversos sempre que necessário.

Exemplos

- Dados que necessitam ser entrados dentro de interfaces por menus são fornecidos pelo usuário por meio de entrada direta (formulário para preenchimento de campo);
- Estilos alternativos podem ser fornecidos quando diferentes perfis de usuários o justificarem (Exemplo: linguagens de comandos com formulários ou menus que aparecem após alguns segundos sem entrada de dados, para usuários menos experientes);

- Os chamados “*command menus*”, onde menus hierárquicos em vários níveis de profundidade são exibidos com a alternativa de seleções sucessivas por meio da digitação da primeira letra da opção desejada (comando). Assim, o usuário novato lança mão do recurso dos menus, enquanto o experiente digita diretamente a seqüência de iniciais das opções selecionadas.
- Formulários com opção de digitação de comandos;
- Interfaces de manipulação direta com menus de opções para objetos e/ou funções que não tenham visualização fácil (conceitos abstratos);
- Interfaces de linguagem natural orientadas por menus.

Conclusões

Não há um único estilo melhor do que os outros em termos absolutos.

As diretrizes, que devem ser vistas como tais, devem ser utilizadas em conjunto com os parâmetros resultantes da modelagem do **usuário** e das **tarefas**.

Em geral, combinam-se vários estilos de interação.