

M.V. SILVA *, L.P. MAGALHÃES **

SUMÁRIO

O PROCESSADOR DE DIÁLOGO, uma ferramenta para definição, teste de protótipos e geração de estruturas de diálogos otimizadas é definido. Para apoio ao tratamento gráfico, é justificada a utilização de um Núcleo Gráfico Padrão e, em decorrência, a existência de um GERENCIADOR DE DIÁLOGO.

ABSTRACT

A DIALOGUE PROCESSOR, a tool for definition, prototyping and generation of optimized dialogue structures, is defined. The utilization of a Standards Graphics Package for Dialogue Processor support in graphics treatment is justified; as a consequence, the necessity of a DIALOGUE MANAGER is shown.

* Eng. Eletrônico e de Telecom., PUC/MG, 1981, atualmente inscrito no programa de mestrado do DEE-FEC-UNICAMP

** Dr. Ing., T.H. Darmstadt, Alemanha, 1981, atualmente Prof.-Ass.-Dr. DEE-FEC-UNICAMP

O presente trabalho apresenta o PROCESSADOR DE DIÁLOGO, já discutido anteriormente (/SILV 84/), fornecendo uma descrição funcional e topológica do mesmo. O Processador de Diálogo é uma ferramenta de apoio à geração de diálogos otimizados. Baseia-se na premissa de separar-se a atividade de diálogo da atividade a ser controlada pelo diálogo. Sendo a comunicação homem-máquina tratada isoladamente da aplicação, é possível utilizar um especialista em cada área para tratar da atividade correspondente.

São discutidas as principais características desta ferramenta de software para auxílio no projeto, análise e implementação ("prototyping") de diálogos homem-computador. Tal ferramenta deve ser independente da aplicação específica, da linguagem de programação utilizada pela aplicação e dos dispositivos a serem acionados para entrada e saída de dados. São características suas, a possibilidade de definir e alterar a definição de um diálogo, supervisionar a execução, coletar dados sobre o desempenho, analisar estes dados e promover uma adaptação do diálogo ao operador específico. Para a representação do diálogo, o Processador tem como recurso a Linguagem de Especificação de Diálogos (LED) que permite a definição da estrutura ou das estruturas de qualquer diálogo. Esta linguagem é baseada no método de representação de estruturas de diálogo através de Células de Diálogo (/SILV 85/).

O item 2 deste trabalho apresenta o Processador de Diálogo. O item 3 discute a utilização de um Núcleo Gráfico Padrão para suporte gráfico do Processador de Diálogo. O item 4 apresenta as conclusões.

2-O PROCESSADOR DE DIÁLOGO

A separação entre a aplicação e o diálogo que a controla nos sistemas interativos de computação, com a comunicação usuário-sistema sendo realizada por uma camada independente do programa de aplicação e do pacote gráfico de suporte, traz diversos benefícios, entre os quais (/GIIT 83/) :

- fomenta um trabalho interdisciplinar, com especialistas de aplicação, de núcleo gráfico, de interação, cada um realizando uma parte do projeto do software;
- permite ao programador de aplicação projetar suas rotinas considerando que rotinas de interação previamente implementadas estão ao seu dispor para realizar a troca de informações com o usuário;
- permite exploração de várias características de dispositivos físicos e de técnicas de interação para atender aos fatores humanos do operador;
- traz economia de custos, facilitando a construção da interface de

usuário, permitindo o seu aprimoramento para casos específicos e a sua reutilização por outras aplicações;

- permite consistência através de várias aplicações que podem fazer uso da mesma interface.

Entretanto, nenhum sistema de diálogo pode ser implementado, a priori, suficientemente otimizado para o tipo de aplicação em questão. Assim, surge uma necessidade de avaliação de desempenho da interação operador-sistema e de uma adaptação da estrutura do diálogo às características específicas de um determinado operador e de uma determinada aplicação.

Para suprir esta necessidade, deve-se obter recursos para a definição de estruturas alternativas para um mesmo diálogo, para a alteração de passos de uma determinada estrutura, para controle da execução do diálogo, para coleta de dados sobre o desempenho, para a avaliação destes dados e finalmente para a substituição de uma estrutura de diálogo por outra. Repetindo-se o processo de supervisão de execução e de coleta e avaliação de dados, pode-se determinar qual a estrutura mais eficiente, obtendo-se então uma otimização do trabalho a ser executado pelo operador quando utilizando o sistema.

Todas estas tarefas são referidas coletivamente como "prototyping", entendido como sendo a simulação da execução de um diálogo, podendo acarretar em monitoramento, análise e adaptação do mesmo a uma situação específica e na posterior geração de um produto final, constando da aplicação e seu diálogo otimizado.

Assim, é aqui proposta uma ferramenta, o PROCESSADOR DE DIÁLOGO, que pretende auxiliar o projetista na produção de diálogos, permitindo definição, "prototyping" e geração de um diálogo otimizado. A ferramenta possibilita ao projetista executar um projeto estruturado e, desde que este projetista tenha em mente as considerações de fatores humanos, ele está capacitado a traduzir em seu diálogo um sistema fácil e agradável ao usuário. Uma vez identificada a estrutura ótima para o diálogo entre um operador e uma aplicação específicos, o Processador de Diálogo está apto a gerar seu produto final, qual seja um sistema de aplicação envolvido com seu diálogo otimizado, onde todos aqueles componentes do Processador, utilizados no "prototyping" mas não mais necessários à execução do sistema interativo são subtraídos.

Pode-se visualizar três fases na utilização do Processador de Diálogo : definição, "prototyping" e geração, apresentadas a seguir.

2.1-Definição de Diálogos

Na fase de Definição, um diálogo específico é descrito através dos recursos oferecidos por uma Linguagem de Especificação de Diálogos (LED) (/SILV 85/), baseada no método de representação de estruturas de

diálogo por Células de Diálogo (/SILV 84/). Assim, cada passo em cada interação fica a cargo de uma Célula de Diálogo, que na prática, corresponde a uma rotina. A estrutura do diálogo estará então descrita através destas rotinas. Sendo a Linguagem de Especificação de Diálogos, uma extensão do FORTRAN-77, não são providos compiladores que codifiquem as descrições em forma objeto.

Assim, deve-se utilizar um pré-processor que leia as Células de Diálogo, interprete-as e gere rotinas em FORTRAN-77, através do mapeamento das primitivas extendidas da LED às primitivas básicas do FORTRAN-77. Após este pré-processamento, as rotinas podem ser submetidas ao compilador FORTRAN-77 que se encarrega de gerar rotinas em linguagem objeto, que são ligadas para obtenção do programa executável. Faz parte da função do pré-processor a análise do código que permita a detecção de erros de digitação quando da criação dos arquivos e de falhas na observação das regras da linguagem.

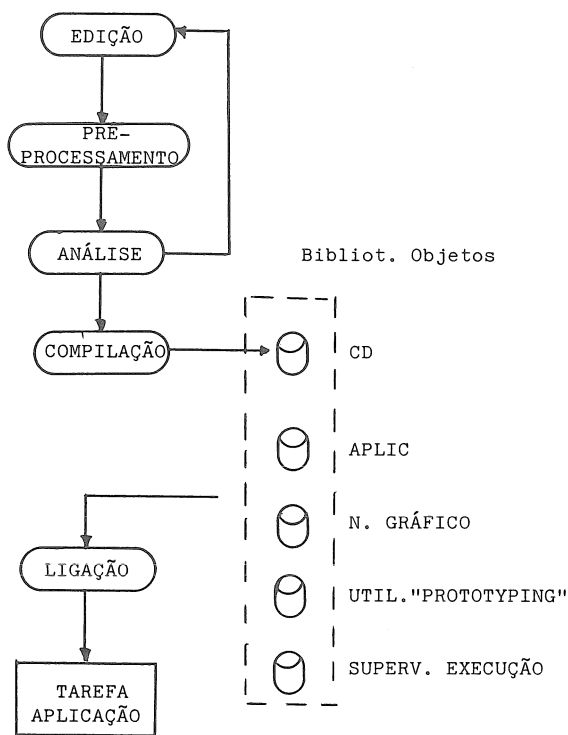


fig. 1 : Fase de definição.

Cada conjunto de células de diálogo capaz de realizar uma interação com o operador é reunido sob a forma de um subprograma (subprograma diálogo). Assim, pode-se ter diversos subprogramas diálogo capazes de atender a cada pedido de dado realizado pela aplicação. A otimização do diálogo está em se determinar, para cada interação, qual é o subprograma mais adequado.

Além das Células de Diálogo, são necessários à execução do diálogo, o programa de aplicação que fará uso do diálogo (ou um simulador dele, caso se pretenda testar o diálogo separadamente ou previamente em relação ao desenvolvimento do programa de aplicação), as rotinas de um núcleo gráfico, que dão apoio à geração e manipulação de figuras para a comunicação com o operador, o Supervisor de Execução que controla a execução do "prototyping" e rotinas (na forma de FUNCTION's) de apoio ao "prototyping" (para coleta de dados, análise, etc, algumas providas pelo implementador do Processador de Diálogo e outras pelo próprio autor de diálogos.

A tarefa aplicação, definida como sendo o programa de aplicação ligado ao supervisor de execução, aos subprogramas de diálogo, às rotinas do núcleo gráfico e aos utilitários de "prototyping", através do uso do ligador do sistema operacional do computador hospedeiro.

A figura 1 apresenta a sequência de atividades desde a criação das células de diálogo até a obtenção da tarefa aplicação.

2.2-"Prototyping"

Uma vez obtida a tarefa aplicação, deve ser realizado o "prototyping". A tarefa aplicação é executada e o supervisor de execução é acionado toda vez que a aplicação desejar se comunicar com o usuário. O supervisor se configura então como a ligação entre a aplicação e o diálogo, incumbindo-se de acionar um dos subprogramas diálogo providos para aquela interação e passar-lhe o controle de execução. O subprograma é executado e, nos locais definidos pelo autor de diálogos, são ativadas funções de "prototyping" (providas na forma de FUNCTION's) que realizarão coleta e análise de dados (e outras atividades previstas). Quando o controle é devolvido ao supervisor de execução é acionado outro subprograma diálogo para a mesma interação e repete-se o processo que permite então a avaliação dos subprogramas diálogo projetados para aquela interação. A seguir, o controle é devolvido ao programa de aplicação que retoma sua execução. A cada nova interação, ou seja, a cada nova necessidade de comunicação entre o programa de aplicação e o operador, um novo conjunto de subprogramas diálogo é executado, um subprograma por vez, permitindo avaliar os diversos subprogramas providos para cada interação. Ao final da execução da tarefa aplicação, pode-se determinar, pelos dados coletados, qual subprograma é apropriado para cada interação com a aplicação em questão e o conjunto destes subprogramas constitui o diálogo otimizado para aquela aplicação. A figura 2 apresenta o diagrama

da tarefa aplicação sendo executada, onde são exemplificadas m interações com n possibilidades de sequências de diálogo diferentes para cada uma das interações.

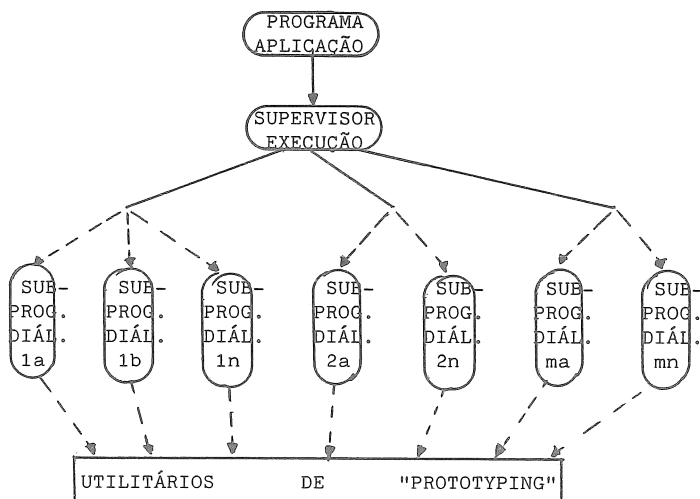


fig. 2 : Fase de "prototyping"

2.3-Geração do Diálogo Otimizado

A geração é realizada após o elemento LIMPADOR processar os subprogramas diálogo selecionados na fase anterior e eliminar das células de diálogo correspondentes todas as primitivas específicas de "prototyping", desnecessárias à operação normal do diálogo.

As células de diálogo resultantes são novamente pré-processadas, compiladas e ligadas, agora ao programa de aplicação, ao supervisor de execução e às rotinas do núcleo gráfico. Deve-se notar que o supervisor de execução agora utilizado é uma versão simplificada daquele utilizado para o "prototyping", já que não são mais necessários os testes e controles para acionar diversas sequências de diálogo para atender a cada solicitação de interação efetuada pela aplicação. Já foi identificada a sequência ótima para cada interação e o supervisor simplesmente faz a interface entre a aplicação e as sequências ótimas. A operação de obtenção da tarefa aplicação ótima é ilustrada na figura 3.

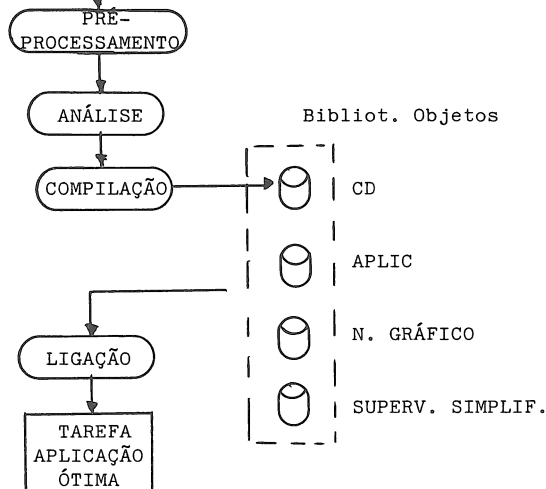
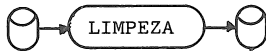


fig. 3 : Fase de geração.

3-USO DE UM NÚCLEO GRÁFICO PADRÃO NO APOIO AO PROCESSADOR DE DIÁLOGO

O Supervisor de Execução (figura 2) será enfocado mais detalhadamente. Para tal, ele é analisado como sendo constituído por três elementos : um Supervisor de "Prototyping", um Selecionador de Sequências de Diálogo e um Gerenciador de Recursos Gráficos.

O Selecionador de Sequências é responsável por, mediante uma solicitação de interação da aplicação com o operador, determinar quais são (no "prototyping") ou qual é (na execução normal da tarefa aplicação ótima) o(s) subprograma(s) diálogo previsto(s) para aquela interação específica.

Em termos de implementação, o diálogo apresenta-se (conforme pode ser visto na figura 2) como uma árvore, onde o Selecionador de Sequências é a raiz. Cada ramo imediatamente abaixo do Selecionador corresponde a uma necessidade de comunicação do programa de aplicação com o operador. Mas, podem ser oferecidas diversas possibilidades de realização de cada uma destas comunicações. Assim, a cada ramo deixando a raiz da árvore pode corresponder uma subárvore, cujos ramos representam

as possibilidades de atendimento daquela necessidade de comunicação. A decisão por um ramo ou outro depende de valores de parâmetros passados pelo programa de aplicação ao Seleccionador que lhe permitem determinar qual o tipo de necessidade de comunicação e então qual ramo deve ser escolhido.

O Supervisor de "Prototyping" tem sua atividade restrita à fase de "prototyping", já que, em operação normal do diálogo, ele não é necessário. Sua função é a de selecionar ordenadamente, para cada solicitação de interação da aplicação em relação ao operador, cada sequência de diálogo, na forma de subprograma diálogo, provida para atender àquela interação, permitindo a avaliação de todas elas.

Antevendo a implementação, o Supervisor de "Prototyping" agiria como uma estrutura do tipo CASE, segundo a qual, condições seriam manipuladas e testadas implicando no acionamento de uma rotina (subprograma diálogo) para realizar a interação com o operador. Após o retorno do controle de execução ao Supervisor de "Prototyping", as condições são alteradas permitindo o acionamento de outra rotina, e o processo se repete até que todas as rotinas providas para a execução do diálogo correspondente a um dado pedido de interação do programa de aplicação com o operador sejam executadas.

Associando à descrição do Seleccionador, o Supervisor de "Prototyping" estaria logo abaixo na árvore e se incumbiria de executar cada ramo da subárvore correspondente ao ramo inicial escolhido pelo Seleccionador.

O Gerenciador de Recursos Gráficos surge em consequência :

- da utilização de recursos de núcleos gráficos padrões pelo diálogo; tem então o objetivo de adequar os recursos do núcleo gráfico às necessidades do diálogo;
- da necessidade de supervisão dos elementos de comunicação definidos nas células de diálogo;
- do desejo de proporcionar à aplicação acesso aos recursos do núcleo gráfico.

A primeira consequência vem da necessidade de execução das seguintes funções :

- gerenciamento das transformações de coordenadas das informações gráficas definidas pelo diálogo e mapeadas ao sistema gráfico;
- gerenciamento de segmentos; mapeamento dos elementos de comunicação aos segmentos gráficos do sistema gráfico, com o necessário provimento da identificação de cada segmento a ser criado;

- gerenciamento do efeito das funções de controle do sistema gráfico para adequá-las às necessidades do diálogo;
- mapeamento dos parâmetros necessários à ativação de primitivas do núcleo gráfico a partir das primitivas gráficas da LED.

A segunda consequência atende às seguintes funções :

- atribuição de valores para a identificação dos elementos de comunicação criados;
- controle do tempo de vida de cada elemento. Finda a duração definida, o elemento deve ser explicitamente deletado;
- vinculação das figuras aos elementos correspondentes, conforme definido na célula. Se um elemento é alterado, a figura correspondente também deve ser alterada.
- manutenção de informações sobre o desenrolar do diálogo, tais como:
 - estado corrente do diálogo : inativo, ativo (existe célula em execução);
 - identificação da célula em execução;
 - lista das células correntemente ativas (aquela em execução é a mais interna na hierarquia);
 - estado de erro (se e onde ocorreu);
 - tabela de correlação entre a identificação de um dispositivo gráfico pelo núcleo gráfico e a identificação daquele dispositivo pelo gerenciador de diálogo;
 - lista dos elementos definidos, contendo o identificador, o tipo de elemento, os atributos correntes e a células em que cada elemento está definido.

A terceira consequência tem a ver com o oferecimento de facilidades gráficas à aplicação. Pode ser desejado permitir à aplicação acesso às facilidades gráficas do núcleo que dá suporte ao Processador de Diálogo. Ocorre, entretanto, que restrições devem ser feitas a este acesso, para que não haja conflitos de acesso entre a aplicação e a camada de diálogo, o que prejudicaria a representação final. A solução apresenta-se como sendo a utilização uniforme do núcleo gráfico tanto por parte do diálogo quanto da aplicação, através da definição de uma primitiva do Gerenciador para cada recurso do núcleo gráfico, a ser mapeada a estes recursos após tratamento das restrições pelo Gerenciador.

As funções deste Gerenciador são discutidos em maior grau de

4-CONCLUSÃO

As vantagens de uso do Processador de Diálogo podem ser resumidas como :

- formalizar, ou padronizar, a arte de projetar diálogos. Um tratamento padronizado possibilita o rápido aprendizado da técnica;
- oferecer uma Linguagem de Especificação de Diálogos clara, legível e objetiva, mapeada sobre uma linguagem padrão;
- permitir testar diversos diálogos, compará-los e selecionar o de melhor efetividade, adaptando então o diálogo a um operador e a uma aplicação específicos;
- permitir projeto modular, onde cada camada do sistema interativo fica a cargo do especialista correspondente;
- possibilitar a utilização de informações gráficas no diálogo entre operador e aplicação, o que traz aumento de efetividade;
- permitir economia de custos, pois o Processador pode ser utilizado por diversas aplicações;
- permitir explorar características de dispositivos e técnicas de interação com o objetivo de tornar a interface "amigável" ao usuário.

A decisão de utilização do núcleo gráfico padrão como apoio ao Processador de Diálogo veio do desejo de se ter uma interface com o mundo exterior portátil e consistente. Esta decisão trouxe a necessidade de se definir um elemento Gerenciador de Recursos Gráficos, que se responsabilize por supervisionar a utilização dos recursos do núcleo gráfico adequando-os às necessidades da camada de diálogo, e para permitir às aplicações que se utilizam do Processador de Diálogo, o acesso às mesmas facilidades gráficas oferecidas ao diálogo.

5-REFERÊNCIAS BIBLIOGRÁFICAS

- /GIIT 83/ - GRAPHICAL INPUT INTERACTION TECHNIQUE WORKSHOP SUMMARY - Computer Graphics - 17 (1) - JAN 1983
- /SILV 84/ - Silva, M.V.; Magalhães, L.P.; "Processadores Universais de Diálogo : Formas de Representação do Diálogo" - Anais do IV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - JUL 1984
- /SILV 85/ - Silva, M.V.; "PROCESSADOR DE DIÁLOGO : Uma Ferramenta para Geração de Sistemas Interativos" - Tese de Mestrado - UNICAMP - Campinas - SP - ABRIL 1985