

UNIVERSIDADE FEDERAL DA PARAIBA
CENTRO DE CIENCIAS E TECNOLOGIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

DESENVOLVIMENTO DO GERENCIADOR DE DIALOGOS E DE FERRAMENTAS DE
PROTOTIPAGEM DO SISTEMA AGILE

CLAUDIA DANTAS PROCOPIO

CAMPINA GRANDE - PB

ABRIL - 1992

P. 962d

CLAUDIA DANTAS PROCOPIO

DESENVOLVIMENTO DO GERENCIADOR DE DIALOGOS E DE FERRAMENTAS DE
PROTOTIPAGEM DO SISTEMA AGILE

Dissertação apresentada ao Curso de MESTRADO
EM SISTEMAS E COMPUTAÇÃO da Universidade
Federal da Paraíba, em cumprimento às
exigências para obtenção do Grau de Mestre.

AREA DE CONCENTRAÇÃO: CIENCIA DA COMPUTAÇÃO

MARIA DE FATIMA QUEIROZ VIEIRA TURNELL

Orientadora

DIS
04.5125/04
P923d



P963d Procopio, Claudia Dantas
Desenvolvimento do gerenciador de dialogos e de
ferramentas de prototipagem do sistema agile / Claudia
Dantas Procopio. - Campina Grande, 1992.
44 f. : il.

Dissertacao (Mestrado em Sistemas e Computacao) -
Universidade Federal da Paraiba, Centro de Ciencias e
Tecnologia.

1. Gerenciador de Dialogos 2. Ferramentas de
Prototipagem 3. Dissertacao I. Turnell, Maria de Fatima
Queiroz Vieira, Dra. II. Universidade Federal da Paraiba -
Campina Grande (PB) III. Titulo

CDU 004.512.5(043)

DESENVOLVIMENTO DO GERENCIADOR DE DIÁLOGOS E DE FERRAMENTAS DE
PROTOTIPAGEM DO SISTEMA AGILE

CLAUDIA DANTAS PROCOPIO

DISSERTAÇÃO APROVADA EM 30.04.1992

Maria de Fatima Q. V. Turnell
MARIA DE FATIMA Q. V. TURNELL, Ph.D
Orientador

Ana Regina Cavalcanti da Rocha
ANA REGINA CAVALCANTI DA ROCHA, D.Sc
Componente da Banca

Pedro Sergio Nicolletti
PEDRO SERGIO NICOLLETTI, M.Sc
Componente da Banca

Campina Grande, 30 de abril de 1992

SUMÁRIO

Página

1.0 - INTRODUÇÃO.....	01
1.1 Justificativa.....	01
1.2 Decisões de implementação.....	03
1.3 Escopo do trabalho.....	04
2.0 - CONCEITUAÇÃO.....	06
2.1 Sistemas Gerenciadores de Interface do Usuário.....	06
2.2 Sistemas de Prototipagem Rápida de Interfaces.....	11
2.3 Gerenciadores de Diálogos.....	12
2.4 Terminologia usada no trabalho.....	12
3.0 - O AMBIENTE AGILE.....	16
3.1 Arquitetura do Sistema.....	16
3.2 Filosofia de Funcionamento.....	18
3.2.1 Paradigma de interface.....	18
3.2.2 Eventos/Ações.....	22
3.3 Ambiente de instalação.....	23
3.4 Ambientes de interação.....	23
3.4.1 Prototipagem.....	24
3.4.2 Simulação.....	24
4.0 - GERENCIADOR DE DIALOGOS.....	25
4.1 Algoritmo de gerencia.....	26
4.2 Interpretador de diálogos.....	27
4.2.1 Geração do evento.....	27
4.2.2 Análise de eventos.....	31

4.3 Sequenciador de diálogos.....	46
4.3.1 Pilha de diálogos.....	46
4.3.2 Descrição das ações.....	48
4.3.3 Ativação das ações.....	53
5.0 - FERRAMENTAS DE PROTOTIPAGEM.....	57
5.1 Filosofia de desenvolvimento.....	57
5.2 Editor de interfaces.....	58
5.2.1 Árvore de sequenciamento.....	58
5.2.2 Exemplo do sequenciamento de um menu.....	62
5.2.3 Exemplo do sequenciamento de um formulário.....	62
5.3 Gerador de documentação	64
6.0 - CONCLUSÕES.....	70
6.1 Considerações.....	70
6.2 Testes.....	73
6.3 Dificuldades.....	74
6.4 Sugestões para trabalhos futuros.....	75
BIBLIOGRAFIA.....	77

APENDICE - Guia do usuário

LISTA DE FIGURAS

FIGURA 3.1	- Modelo de Seeheim.....	16
FIGURA 3.2	- Arquitetura de Agile.....	17
FIGURA 3.3	- Informações referentes a um Diálogo.....	20
FIGURA 4.1	- Estrutura Evento.....	29
FIGURA 4.2	- Vetor dos Menus.....	32
FIGURA 4.3	- Estrutura INFO_SEQ_MENU.....	33
FIGURA 4.4	- Estrutura Itens.....	33
FIGURA 4.5	- Vetor dos Comandos.....	34
FIGURA 4.6	- Estrutura INFO_SEQ_COMANDO.....	35
FIGURA 4.7	- Estrutura Comandos.....	36
FIGURA 4.8	- Vetor das Perguntas e Respostas.....	37
FIGURA 4.9	- Estrutura INFO_SEQ_PERGUNTA.....	38
FIGURA 4.10	- Estrutura Respostas.....	39
FIGURA 4.11	- Vetor dos Formulários.....	39
FIGURA 4.12	- Estrutura INFO_SEQ_FORMULARIO.....	40
FIGURA 4.13	- Estrutura Campos.....	41
FIGURA 4.14	- Estrutura da Pilha de Diálogos.....	47
FIGURA 4.15	- Estrutura de uma Ação.....	48
FIGURA 4.16	- Estrutura de Retornos.....	52
FIGURA 5.1	- Árvore de Sequenciamento dos Menus.....	60

LISTA DE TABELAS

TABELA 3.1 - Texto do Diálogo e Eventos a serem Reconhecidos.....	21
TABELA 4.1 - Eventos Repassados pelo Gerenciador de Apresentação.	30
TABELA 6.1 - Tabela de Critérios.....	73

ABSTRACT

This dissertation describes the development of a **Dialog Manager** module which when combined with the **Presentation Manager Module** described in [SANTOS 92] forms a system which offers resources for rapid interface prototyping - the AGILE system.

As well as the Dialog Manager, this dissertation describes the development of two software tools; the **Interface Editor** and the **Documentation Generator**, both of which were integrated into the system's architecture with the purpose of aiding the interface designers in the specification and documentation of their prototypes.

RESUMO

Esta dissertação descreve o desenvolvimento de um módulo **Gerenciador de Diálogos**, que compõem juntamente com o módulo **Gerenciador de Apresentação** descrito em [SANTOS 92], um sistema que oferece recursos para prototipagem rápida de interfaces - o sistema **AGILE**.

Além do **Gerenciador de Diálogos**, esta dissertação descreve o desenvolvimento de duas ferramentas, o **Editor de Interfaces** e o **Gerador de Documentação**, que foram integradas à arquitetura do sistema, com o objetivo de auxiliar os projetistas de interface na especificação de seus protótipos.

1.0 - INTRODUÇÃO

1.1 Justificativa

A preocupação com a forma de apresentar as informações para o usuário de computadores, tornando-a mais natural, surgiu com o aparecimento dos sistemas interativos e o uso de terminais de vídeo, e se intensificou à medida que estes se tornaram mais populares [HUCKLE 81].

Em 1962, há 30 anos atrás, Ivan E. Sutherland defendendo sua tese de doutorado pelo Massachusetts Institute of Technology, apresentou o Sketchpad, considerada a pioneira das interfaces usuário-computador. No entanto, só em 1984 com o lançamento do MacIntosh, a Apple Computer Inc. tornou conhecida a expressão interface amigável a milhares de pessoas em todo o mundo [PERRY 89].

O crescimento e diversificação do universo de usuários de computador, ocasionado principalmente pelo barateamento do custo do hardware, e a evolução dos conceitos na área de interface combinada com computadores cada vez mais rápidos, e com maior capacidade de armazenamento, fez surgir uma nova sub-divisão dentro da tecnologia CASE (Computer Aided Software Engineering): os Sistemas de gerenciamento de Interface do Usuário - SGIU [BRANCO 90].

No entanto, os métodos de interação com o usuário, nem

sempre são facilmente definidos. Surgiram então os sistemas de prototipagem rápida de interfaces usuário-computador [LUQUI 89], que têm como objetivo minorar esta dificuldade, permitindo uma maior participação do usuário no processo de definição da interface antes mesmo de sua implementação.

Nosso trabalho consistiu no projeto e implementação de componentes básicos de Agile - um Sistema de Prototipagem de Interfaces, que foi desenvolvido com o propósito de oferecer recursos aos projetistas de interface para prototipagem e simulação de interfaces. Com base neste propósito foram identificadas as seguintes características como objetivos do sistema:

- Definir a interface através de uma linguagem interativa, baseada em menus e formulários.

- Simular a interface especificada, sem a necessidade de programação.

- Definir nas interfaces, mensagens de erro e de ajuda.

- Prover transparência na utilização de periféricos, ou seja, não é necessário para o projetista especificar os periféricos a serem utilizados nas suas interfaces. Na versão atual, podem ser utilizados o mouse e o teclado.

- Criar, facilmente, uma variedade de diálogos, no que diz respeito aos estilos de interação. Nesta versão são oferecidos quatro estilos: menu, comando, pergunta e resposta, e formulário.

. Combinar em um mesmo diálogo mais de um estilo de interação. Por exemplo, pode-se definir que em um diálogo de estilo formulário, determinados campos podem ser preenchidos através da seleção dos itens de um menu.

Para tanto, os componentes desenvolvidos foram:

. O módulo **Gerenciador de Diálogos**, que tem como função realizar o sequenciamento da interação de acordo com um protótipo de interface previamente especificado.

. A ferramenta **Editor de Interfaces**, que oferece recursos ao projetista, para especificar interativamente formas alternativas de interação, criando os protótipos de interface.

. A ferramenta **Gerador de Documentação**, a qual a partir de uma interface especificada, gera um arquivo documento correspondente.

As implementações mencionadas foram feitas, tomando como base a especificação feita para o sistema AGILE em [BRANCO 90], embora muitas modificações tenham sido introduzidas visando viabilizar o projeto, como por exemplo a alteração da interface do sistema, redução do conjunto de ferramentas e simplificação da documentação fornecida.

1.2 - Decisões de implementação

Na definição da linguagem de implementação, foi considerada a utilização de uma linguagem orientada a objeto, ou de

linguagens procedurais como C e Pascal. Optou-se pela linguagem de programação C, levando-se em consideração a portabilidade e capacidade de interfacear com outras linguagens, inclusive com a versão C orientada a objeto, pensando-se na expansão do sistema, ou seja o acoplamento de outros módulos, como por exemplo um editor gráfico.

Além do que, a utilização de uma linguagem orientada a objeto, exigiria um tempo de aprendizado inaceitável para o cronograma do trabalho.

Finalmente, o ambiente MS-DOS foi utilizado para desenvolvimento do sistema e também como o ambiente alvo para utilização de Agile. Esta escolha foi feita considerando-se a disseminação dos equipamentos PC.

1.3 - Escopo do trabalho

Para alcançarmos os objetivos do trabalho, definimos três fases básicas a serem cumpridas, foram elas:

- 1) Definição do paradigma da interface,
- 2) Desenvolvimento do módulo Gerenciador de Diálogos,
- 3) Desenvolvimento das ferramentas: Editor de Interfaces e Gerador de Documentação.

No paradigma da interface, descrito na seção 3.2.1, foi estabelecido que a primeira versão do sistema deveria permitir definições de janelas, mensagens e de diálogos dos quatro estilos de interação (menu, comando, pergunta e resposta, e formulário).

No entanto a modularidade do sistema desenvolvido permitirá a introdução de novos estilos que se façam necessários.

No desenvolvimento do Gerenciador de Diálogos, definimos para cada estilo de interação oferecido pelo sistema, os tipos de entradas que poderiam ser feitas pelo usuário, e quais as funções que deveriam ser executadas para se realizar o sequenciamento de interação. Com isso, concluímos que deveriam ser criados dois blocos de funções. No primeiro, para cada estilo de interação deveria existir uma função para analisar as entradas que lhe fossem pertinentes. No segundo bloco estariam as funções que executariam o sequenciamento da interação. Estes dois blocos correspondem respectivamente ao Interpretador de Diálogos e ao Sequenciador de Diálogos, descritos no Capítulo 4.

Como neste estágio já dispúnhamos de recursos para sequenciar e gerenciar o diálogo, porém restritos ao limite da interface, ampliamos a funcionalidade do ambiente Agile de modo que este pudesse gerenciar uma interface acoplada às funções da aplicação para o caso específico de suas ferramentas. Para isso foram definidas interfaces, uma para o Editor, e outra para o Gerador de Documentação, seguindo o paradigma de interface de Agile, e a seguir foram implementadas as funções do Editor e do Gerador, como aplicativos. No Capítulo 5 estas ferramentas são descritas. E, finalmente no Capítulo 6 são tecidas considerações sobre o trabalho como um todo.

2.0 CONCEITUAÇÃO

2.1 Sistemas Gerenciadores de Interface do Usuário

A visão clássica de um SGIU é um pedaço de software que controla toda comunicação entre o usuário e o programa da aplicação; a aplicação não deve se comunicar com o usuário final diretamente. Um SGIU também deve controlar a comunicação entre a aplicação e um gerenciador de janela ou sistema gráfico.

O principal objetivo de um SGIU é liberar o programador de aplicações dos detalhes de baixo nível dos periféricos, e dos detalhes das interfaces de alto nível, deixando isso como responsabilidade do projetista da interface.

Os SGIUs basicamente são um conjunto de ferramentas que oferecem recursos para:

- . prototipagem rápida de interfaces,
- . gerenciamento da interação entre o usuário e a aplicação.

Maiores detalhes sobre SGIU, podem ser encontrados em [BRANCO 90].

A seguir, apresentaremos algumas características relevantes ao estudo e caracterização dos SGIUs. Na exposição destas características, os SGIUs listados abaixo serão discutidos [PRIME 90].

- Tiger.
- SET.
- Dice.
- Blox/Template.
- Grape MMI.

1) **Separação entre a interface do usuário e a aplicação:** O código do programa da aplicação deve ser independente do código de sua interface com o usuário, de forma que seja possível escrever cada código separadamente.

Em Tiger, os códigos são totalmente separados, sendo possível escrever o código da interface antes ou depois do código da aplicação.

SET não consegue separar o código totalmente, o código da aplicação é distribuído, dentro do esqueleto da interface.

Em Dice, os códigos são bem interligados.

No Blox, a aplicação e a interface do usuário não são totalmente separados. É possível ter no código da aplicação, algumas funções da interface.

Grape tem um arquivo de sintaxe que contém todas as informações sobre a interface do usuário. Fica difícil ver a divisão entre o que seria colocado no arquivo de sintaxe e o que iria para o código da aplicação.

2) **O controle da interação é feito pelo SGIU ou pela aplicação:** O controle do sistema pode ser feito pelo SGIU via

entrada através da interface, ou pela aplicação. Se o SGIU é quem tem o controle, ele chama módulos da aplicação em resposta às entradas do usuário final. Este controle é chamado externo. Se o controle é sempre uma consequência do código da aplicação, então o controle é dito ser interno. Uma terceira forma de controle ocorre quando o usuário tem completo controle. Um verdadeiro SGIU deve possuir controle externo.

Em Tiger, o controle vem da aplicação, já que é ela que requisita as entradas do usuário final.

Em SET, o controle é do SGIU, este é quem solicita entradas dos usuários e decide o que fazer.

Em Dice, o fluxo de controle origina-se no usuário, segue via o SGIU para a aplicação.

O controle em Blox é totalmente externo.

Em Grape, o controle também é externo.

3) Comunicação entre a aplicação e o SGIU: Esta característica reflete a natureza da comunicação em tempo de execução, entre a interface e o usuário final, e entre a interface e os módulos da aplicação. É levado em consideração, se a comunicação entre os processos é feita através da passagem de mensagens, ou através de eventos.

O usuário final de uma interface deve poder interagir com múltiplos periféricos de entrada, e o SGIU é quem deve decidir

qual ele está usando (por exemplo mouse ou teclado).

A comunicação de informação entre a interface e os módulos da aplicação deve incluir a transmissão de solicitações de usuários, dados de usuário e resultados da aplicação.

Em Tiger, todas as entradas para uma aplicação estão na forma "padrão previsível", isto é, a entrada é traduzida em um formato uniforme de caracteres. Este mecanismo fornece dados em forma de token para a aplicação. Um exemplo seria: se um item "abrir arquivo" de um menu fosse selecionado e o nome do arquivo fosse solicitado, os tokens "ABRIR" e "nome do arquivo" seriam enviados para a aplicação.

SET suporta muitas formas de entrada. Após uma entrada do tipo evento ter sido recebida, a aplicação é notificada e deve então obter informações do evento através de uma chamada de rotina. Dependendo da rotina chamada, uma variedade de dados relevantes deve ser recuperada numa gama de formatos.

Em Dice, uma aplicação pode ser vista como uma árvore de chamadas de subrotinas. A raiz é o programa principal. Um diálogo é responsável pela chamada das subrotinas, que são as folhas da árvore. Cada subrotina de diálogo é responsável por receber um token de entrada do usuário; dar uma realimentação correspondente; e passar um valor de retorno apropriado para a subrotina da aplicação.

Em Blox, a comunicação é descrita por um modelo de transição de estados. O controle do usuário transita entre

estados causando ações que são associadas com rotinas escritas pelo projetista da interface do usuário.

Em Grape, a conexão entre a interface e a aplicação ocorre através das rotinas de ação, incluídas no arquivo sintaxe.

4) Ferramentas acopladas aos SGIUs: São ferramentas para desenvolvimento de interfaces, permitindo sua geração interativa.

Tiger tem muitas ferramentas para desenvolvimento das interfaces. Uma para projetar menus e formulários de entrada de dados, outra para layout de tela, editores de fontes e ícones, etc.

SET possui um utilitário que permite projetar caixas de dados, texto, etc.

Não tem informações sobre as ferramentas de Dice na bibliografia pesquisada.

Blox tem uma ferramenta para gerar a interface do usuário e os símbolos usados.

Grape possui uma ferramenta para projetar telas, que permite que telas, menus, formulários, ícones e rotinas de ação sejam projetados interativamente.

5) Portabilidade: No caso do SGIU rodar em determinado ambiente, a interface e a aplicação devem também rodar nele.

Tiger é escrito na linguagem "C" objetivando

portabilidade. Se a aplicação é levada para hardware diferente, então utilitários são usados para adequar detalhes da entrada e saída de periféricos nos arquivos de configuração.

SET está implementado para VAX, Sun, Apollo, etc.

Dice é portátil entre periféricos devido ao mapeamento reconfigurável de unidade de E/S, ou seja, ele é capaz de reconhecer o periférico que está sendo utilizado, sem ser necessário passar a informação para ele.

Blox usa Fortran, mas é possível usar C.

Grape roda em Sun, Apollo, VAX minis, etc.

2.2 Sistemas de Prototipagem Rápida de Interfaces

Prototipagem rápida de interfaces é o processo de construir e avaliar rapidamente uma série de protótipos - interfaces quase funcionais, ou seja, que permitem o teste de muitos aspectos de um diálogo antes de sua integração com a aplicação [LUOI 89].

A utilização de prototipagem rápida traz uma série de vantagens ao projeto de interfaces:

- rompe a barreira entre o usuário e o responsável pelo desenvolvimento na medida em que o usuário não precisa mais ler as especificações feitas pelo projetista e este não terá mais tantas dificuldades em entender as necessidades do usuário;

- permite uma validação do projeto através da experiência prática do usuário com o sistema;

- propicia uma abordagem evolucionária do processo de desenvolvimento baseado na realimentação do usuário.

O Oregon Speedcode Universe (OSU) [LEWIS 89] é um exemplo de um sistema de prototipagem rápida de interfaces.

2.3 Gerenciadores de Diálogos

Estes podem ser entendidos como um conjunto de funções responsáveis pela apresentação, coleta e análise dos dados diretamente relacionados com o tratamento da interação usuário-computador [BRITTS 87].

Como exemplos de gerenciadores de diálogos podemos citar:

- IDECS - Produzido na Universidade de Linköping.
- DESIGN - Parte do projeto Princess da Universidade de Estocolmo.
- FMS-11 - Produto da Digital Equipment
- TAXIS - Desenvolvido na Universidade de Toronto
- USE - Desenvolvido na Universidade da Califórnia.

2.4 Terminologia usada no trabalho

A seguir definiremos termos usados no decorrer deste trabalho, e que tem significados próprios no contexto do ambiente Agile.

2.4.1 Evento

Entradas geradas pelo usuário como por exemplo a opção selecionada de um menu, resposta a uma pergunta, etc.

Na versão atual do sistema Agile, os eventos podem ser gerados apenas através do teclado ou do mouse (ver seção 4.2.1).

2.4.2 Ações

Para o projetista de interface, uma ação é o efeito resultante de um dado evento. Estão disponíveis ao projetista doze tipos de ação que podem ser usadas na definição de um diálogo e que determinam o sequenciamento da interação. Estas ações estão detalhadas na seção 4.3.2.

2.4.3 Diálogo

Comunicação entre o usuário e o computador. Em Agile, um diálogo é a menor célula do processo de interação. É composto de informações de apresentação e de informações referentes ao sequenciamento. (ver seção 3.2.1). Como exemplo de diálogo, podemos citar um formulário para aquisição de dados cadastrais de uma empresa.

2.4.4 Diálogo Ativo

Diálogo com o qual o usuário está interagindo (gerando

eventos), em determinado ponto da interação. Por exemplo, supondo que a aplicação necessite do nome de um arquivo, e seja apresentado ao usuário, um diálogo do tipo menu com nomes de arquivos para que faça a seleção. Enquanto o usuário não selecionar, este diálogo é considerado o diálogo ativo.

2.4.5 Mensagens

Recurso oferecido aos projetistas, para que estes incorporem nos seus protótipos de interface, uma linha de texto para informações de erros e/ou orientação na utilização das interfaces. (ver seção 3.2.1).

2.4.6 Janelas

São áreas da tela, dimensionadas em um modelo de interface, e posteriormente associadas a diálogos e mensagens, como sendo os locais da tela onde estes devem ser apresentados (ver seção 3.2.1).

2.4.7 Tecla aplicável

Tecla que quando pressionada, e reconhecida dentro de um contexto da interação, ativa ações. (ver seção 4.2.2).

2.4.5 Estilos de Interação

Nesta versão do sistema Agile, quatro estilos de interação podem ser tratados, isoladamente ou combinados a critérios do

projetista. São eles: menu, comando, pergunta e resposta, e formulário.

Estes estilos de interação não são uma restrição à introdução de novos estilos que venham a surgir, eles refletem apenas o estado atual do sistema.

3.0 - O AMBIENTE AGILE

3.1 - Arquitetura do sistema

A arquitetura de Agile foi definida baseando-se no modelo de Seeheim [LOWGREN 88] [GREEN 85].

Este modelo representa um dos tipos de arquitetura de SGIU e foi proposto na "workshop", realizada em Seeheim em 1983. Como pode ser verificado na figura 3.1, ele coloca entre a aplicação e o usuário, uma interface composta de três módulos:

Apresentação: Módulo responsável pela apresentação externa da interface ao usuário.

Controle de diálogo: Gerencia a interação entre o usuário e a aplicação.

Modelo da interface com a aplicação: Representação da aplicação do ponto de vista da interface com o usuário.

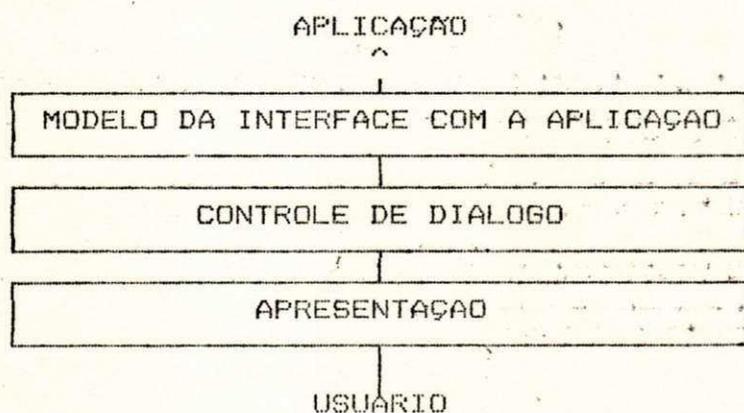


FIGURA 3.1 - Modelo de Seeheim

A seguir a figura 3.2 apresenta a arquitetura de Agile. Nela se verifica que os componentes do modelo de Seeheim: módulo de apresentação e módulo de controle de diálogo correspondem

respectivamente ao Gerenciador de Apresentação e ao Gerenciador de Diálogos.

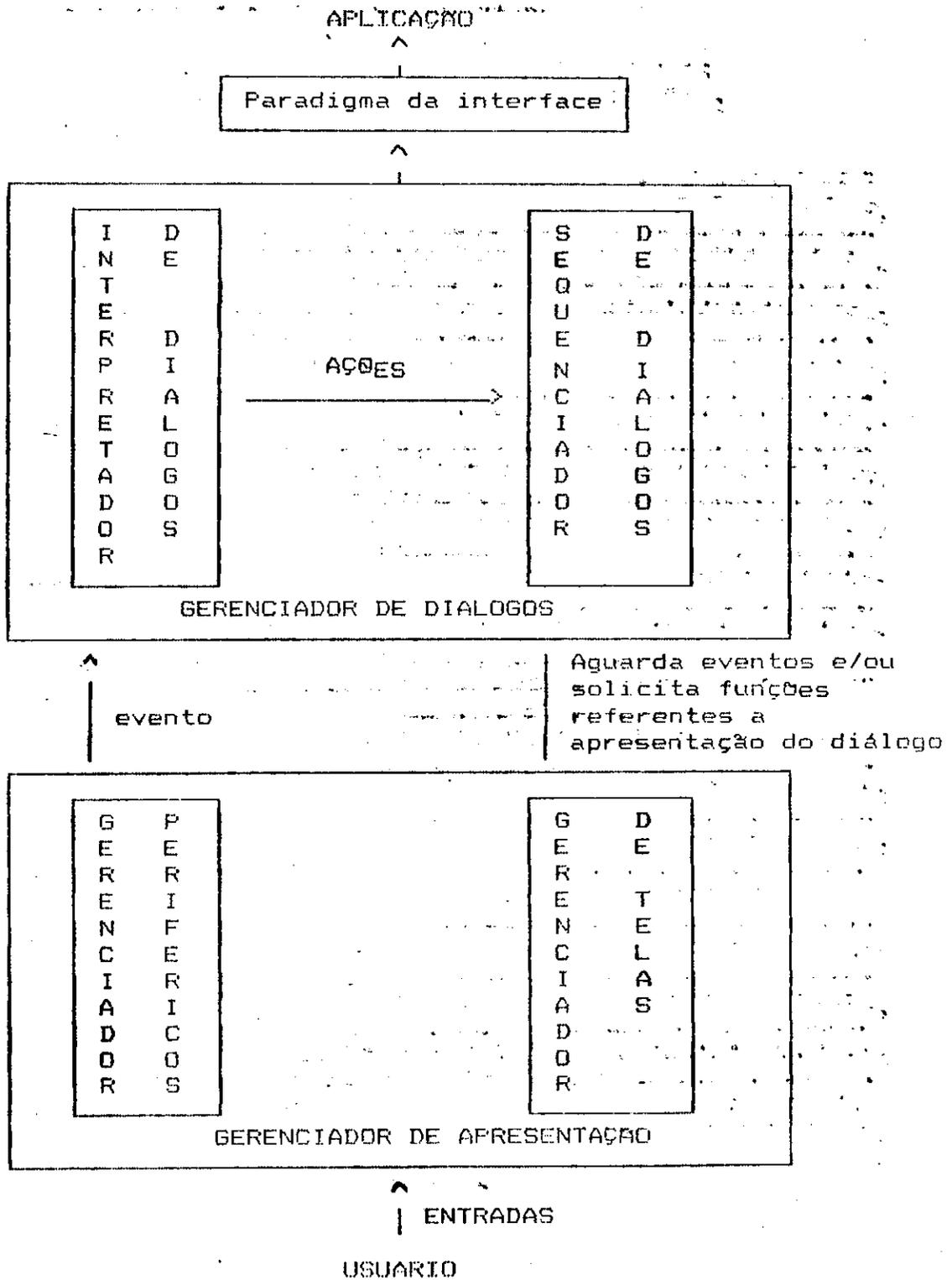


FIGURA 3.2 - Arquitetura de Agile

O Gerenciador de Diálogos(GD) tem como função controlar todo o sequenciamento da interação, a partir da definição de um protótipo de interface e dos eventos gerados pelo usuário. É formado por dois blocos, o "Interpretador de Diálogos", que é responsável pela análise de eventos, e o "Sequenciador de Diálogos", que tem a função de ativar ações decorrentes da análise dos eventos.

O Gerenciador de Apresentação é o responsável pela representação externa da interface. Esta função, por sua vez, é executada por dois blocos. O gerenciamento de leiaute de tela é realizado pelo Gerenciador de Telas, e a interface com os periféricos, ou seja operações de E/S, estão a cargo do Gerenciador de Periféricos [SANTOS 92].

Nesta versão, estão integradas à arquitetura do sistema, duas ferramentas. O Editor de Interfaces, que tem como objetivo permitir que os projetistas de interface definam seus protótipos, através de uma linguagem de definição interativa. O Gerador de Documentação, que tem a função de fornecer documentos ao projetista, com a especificação dos seus protótipos.

3.2 - Filosofia de funcionamento

3.2.1 - Paradigma de interface

Para Agile, o paradigma de uma interface é um conjunto de especificações de elementos do tipo diálogo, janela e mensagem. A seguir passaremos a descrever estes elementos.

1) Diálogos

A definição de um diálogo consiste de informações de apresentação e de sequenciamento, conforme mostra a figura 3.3.

Dentre as **informações de apresentação** encontram-se as seguintes:

- Identificação da janela na qual o diálogo deverá ser apresentado,
- Posição da janela em coordenadas, onde o diálogo deverá aparecer,
- Texto do diálogo, associado a atributos de cor, fonte de caracteres, tamanho de caracteres,
- Atributos de tela, como por exemplo padrão de preenchimento, cor de preenchimento.

Como **informações de sequenciamento** podemos citar:

- Os eventos considerados pelo projetista como válidos e as ações a eles associados,
- As ações para o caso de eventos não reconhecidos,
- O código das teclas aplicáveis, no contexto do diálogo, e ações correspondentes.

I A N P R O E R S M E A N Ç T O A E Ç S A O	JANELA
	POSIÇÃO NA JANELA
	TEXTO DO DIALOGO
	ATRIBUTOS DE TELA
I S N E F Q O U R E M N A C Ç I O A E M S E N T O	EVENTOS A SEREM RECONHECIDOS E SUAS AÇÕES
	AÇÕES PARA EVENTOS NAO RECONHECIDOS
	TECLAS APLICAVEIS AO CONTEXTO E SUAS AÇÕES

FIGURA 3.3 - Informações referentes a um diálogo

Na definição de diálogos, as informações de apresentação referentes ao "TEXTO DO DIALOGO" e as informações de sequenciamento relativas aos "EVENTOS A SEREM RECONHECIDOS", são diferentes para cada estilo de interação. A seguir, mostramos na tabela 3.1 para cada estilo de interação, o que consideramos "TEXTO DO DIALOGO" e "EVENTOS A SEREM RECONHECIDOS".

ESTILO DE INTERAÇÃO	TEXTO DO DIALOGO	EVENTOS A SEREM RECONHECIDO
MENU	Texto dos itens	As possíveis opções do menu
COMANDO	Prompt para entrada dos comandos.	Cadeia de caracteres correspondente aos comandos a serem reconhecidos.
PERGUNTA E RESPOSTA	Texto da pergunta	Cadeia de caracteres correspondendo as respostas.
FORMULARIO	Texto dos campos	Possíveis valores dos campos.

TABELA 3.1 - Texto de Diálogos e Eventos a serem Reconhecidos

2) Janelas

As informações necessárias para definição de uma janela são as seguintes:

- Dimensões da janela,
- Localização da janela na tela,
- Definição de títulos e rodapés.

3) Mensagens

Na definição de mensagens, as informações necessárias são:

- Determinação da janela onde a mensagem deve aparecer,
- Posição dentro da janela,
- Texto da mensagem, associado com atributos de fonte e cor.

Este paradigma apresenta algumas restrições, principalmente na forma de apresentação dos diálogos. As restrições são

apresentadas a seguir:

- O número de diálogos que pode ser definido para cada estilo de interação, o número de janelas e o número de mensagens estão limitados a 60 cada. Esta limitação é decorrente do tipo de estrutura pela qual optamos. (ver seção 4.2.2).

- No estilo de interação menu, o número de itens está limitado ao tamanho da janela, já que não foi implementada nesta versão a técnica de rolamento ou paginação de menus. Esta restrição estende-se também ao estilo de interação por formulário, ou seja, um formulário deverá ter todos os seus campos contidos na janela onde ele será apresentado, já que não está disponível rolamento ou paginação de formulários.

- Uma mensagem está limitada a uma linha dentro da janela à qual está associada.

3.2.2 - Eventos/Ações

O funcionamento de Agile, é baseado na filosofia de eventos que ativam ações.

Eventos são gerados pelo usuário através dos periféricos, como por exemplo, o "click" de um mouse, uma tecla pressionada, etc.

Ações são rotinas que devem ser chamadas para executarem determinada função, que por sua vez foi definida como consequência de um evento.

A associação entre o evento e a ação é feito durante a especificação da interface, mais precisamente na definição de

sequenciamento da interface.

3.3 - Ambiente de instalação

A configuração de hardware mínima necessária para a instalação de Agile é a seguinte:

- Um PC-XT,
- 640K de memória RAM,
- Winchester de 20 MegaBytes.

Como recursos de entrada/saída, podem ser utilizados nesta versão:

- Monitor monocromático ou colorido,
- Teclado e mouse.

Quanto aos padrões de vídeo Agile suporta CGA, HERCULES, VGA e EGA.

3.4 - Ambientes de interação

Estão disponíveis dois ambientes de interação para o projetista, um deles, o ambiente de prototipagem, é onde se define o protótipo de uma interface. No outro, o ambiente de simulação, se pode fazer uma simulação da interface definida.

3.4.1 Prototipagem

Agile gerencia a interação do projetista de interface com o "Editor de Interfaces". Esta ferramenta auxiliará o projetista na definição dos diálogos, mensagens e janelas que irão compor os protótipos de sua interface.

O editor oferece, de forma interativa, recursos para criação, exclusão e alteração de qualquer dos elementos da interface que está sendo definida.

3.4.2 Simulação

Neste ambiente se processa a gerência da interação do usuário com uma aplicação qualquer. Este gerenciamento é feito a nível de prototipagem vazia, isto é, gerencia a interação entre o usuário e a interface da aplicação, sem acoplar as funções da aplicação, permitindo desta forma que o usuário final possa interagir com o protótipo da interface, de forma a validá-lo.

Durante a simulação, fica transparente para o usuário, a existência de Agile, já que toda a sequência da interação é determinada pelo protótipo que o projetista definiu.

O sistema permite a simulação tanto do todo, quanto trechos específicos do protótipo, dando assim mais flexibilidade ao projetista.

4.0 - GERENCIADOR DE DIALOGOS

O Gerenciador de Diálogos é o responsável pela análise dos eventos gerados pelo usuário final e o consequente sequenciamento do diálogo que se encontra ativo. A partir da interpretação de um evento reconhecido ou não, são ativadas as ações correspondentes para este contexto da interação.

O Gerenciador de Diálogos realiza a tarefa de sequenciamento da interação tanto no ambiente de prototipagem quanto no de simulação, diferenciando no entanto sobre qual protótipo de interface ele estará atuando. No ambiente de prototipagem, ele gerencia a interface das ferramentas, permitindo a edição e documentação dos protótipos de interface. No ambiente de simulação, ele gerencia a interface de qualquer outra aplicação, porém apenas até os limites dos recursos de interface, sem nenhuma comunicação com as funções da aplicação.

Apresentaremos a seguir, de acordo com o que foi feito no desenvolvimento das ferramentas "Editor de Interfaces" e "Gerador de Documentação", os passos a serem seguidos para que Agile gerencie qualquer outra aplicação. (ver detalhes na seção 4.3.2 e no apêndice "Guia do Usuário").

- Definir um protótipo de interface, definindo no sequenciamento, a chamada às funções e procedimentos da aplicação.

- Acessar a estrutura "Pilha de diálogos", já que é nessa estrutura onde são guardados os eventos gerados pelo usuário.

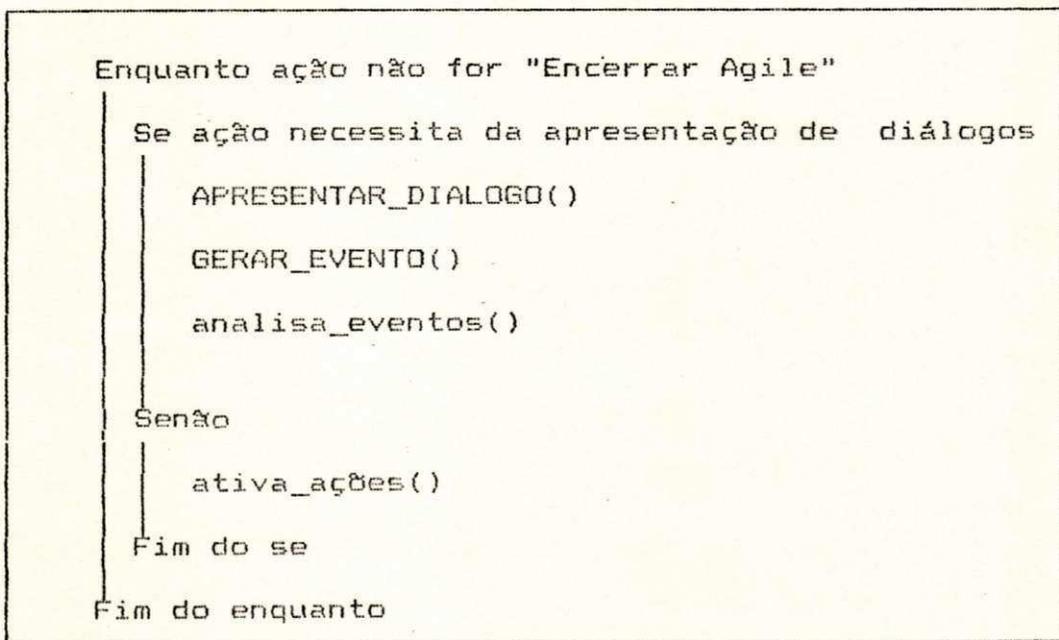
(ver seção 4.3.1).

. Definir as funções da aplicação, compilar e linkar com o código de Agile.

4.1 - Algoritmo de Gerência

Para que o Gerenciador de Diálogos desempenhe suas funções, ele necessita de apoio no que diz respeito à apresentação dos elementos da interface e recepção dos eventos. O Gerenciador de Apresentação (ver seção 3.1), é o responsável por estas funções.

No algoritmo apresentado abaixo, mostramos de forma sucinta como funciona o Gerenciador de Diálogos, e como é feita a comunicação entre ele e o Gerenciador de Apresentação. No algoritmo, as funções representadas em maiúsculo, correspondem às funções do Gerenciador de Apresentação.



Algoritmo de funcionamento do Gerenciador de Diálogos

4.2 - Interpretador de diálogos

A interpretação de um diálogo, corresponde a análise dos eventos gerados pelo usuário para o diálogo que se encontra ativo. Esta análise se fundamenta nas informações de sequenciamento, definidas pelo projetista.

O resultado da interpretação é uma lista de ações que, ao serem ativadas, realizam o sequenciamento da interação.

Na sessão 4.2.1 encontra-se uma descrição do caminho percorrido pelo evento desde a sua geração, a partir da entrada feita pelo usuário, até o momento de sua chegada para ser interpretado.

Na sessão 4.2.2, será detalhado como o Gerenciador de Diálogos, ao receber um evento do Gerenciador de Apresentação, procede para analisá-lo.

4.2.1 Geração do evento

A cada diálogo que se torna ativo o Gerenciador de Diálogos passa o controle para o Gerenciador de Apresentação, informando:

- o estilo de interação a que pertence o diálogo,
- a identificação do diálogo,
- um apontador para a estrutura EVENTO (fig. 4.1), onde o Gerenciador de Apresentação repassa as entradas do usuário.

O Gerenciador de Apresentação fica então recebendo as

entradas do usuário final, armazenando-as em um "buffer", até que uma das seguintes condições ocorra:

- chegada do terminador definido para o diálogo, ou
- a capacidade do "buffer" ter sido superada, ou
- chegada de caracteres não reconhecidos, isto é não definidos para o diálogo.

Quando uma destas condições ocorre a estrutura EVENTO é preenchida e repassada para o Gerenciador de Diálogos.

O conteúdo da estrutura EVENTO, mostrada na figura 4.1, é a seguinte:

- Um "flag", onde o Gerenciador de Apresentação informa que houve sucesso, pois foi teclado o terminador definido para aquele diálogo, ou insucesso quando foi pressionada uma tecla considerada pelo Gerenciador de Apresentação como tecla de controle. São teclas de controle: ESC, ENTER, teclas de função e as teclas combinadas com CTRL e ALT, desde que não tenham sido definidas como terminador do diálogo.

- Evento, que pode ser do tipo cadeia de caracteres, inteiro ou uma tecla. Quando o evento é uma tecla, são informados o "status" e o valor da tecla.

- Ultima tecla, que é o campo que deverá ser preenchido pela última tecla pressionada pelo usuário final, e que não foi reconhecida pelo Gerenciador de Apresentação.

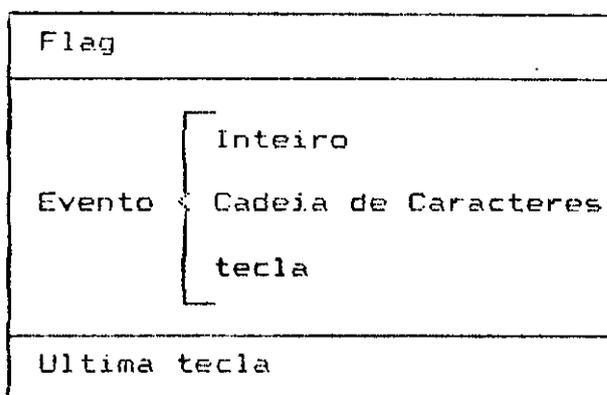


FIGURA 4.1 - Estrutura EVENTO

A seguir, para cada estilo de interação, apresentaremos como a estrutura EVENTO é preenchida.

- MENU:

Se for pressionada a tecla definida como terminador: o Gerenciador de Apresentação repassa a estrutura EVENTO com o campo "flag" preenchido com sucesso, e o campo evento preenchido com o número da opção selecionada.

Se a capacidade do "buffer" for superada, ou houver entrada de caracteres não reconhecidos: a estrutura EVENTO é passada com o "flag" indicando insucesso. O campo evento é preenchido com a cadeia de caracteres gerada pelo usuário, quando o tipo de seleção do menu for STRING (selecionar um item do menu, através da digitação da cadeia de caracteres associada ao item), e com a última tecla pressionada quando o tipo de seleção for BARRA (selecionar um item do menu, através do posicionamento da barra).

- COMANDO, PERGUNTA E RESPOSTA e FORMULARIO:

Se for pressionada a tecla definida como terminador do diálogo: o "flag" da estrutura EVENTO, é preenchido com

sucesso, e o campo evento com a cadeia de caracteres já digitado pelo usuário.

Caso seja superada a capacidade do "buffer", ou tenha ocorrido uma entrada de caracteres não reconhecidos: o "flag" é passado como insucesso, o campo evento é preenchido com a cadeia de caracteres já digitado pelo usuário até a entrada dessa tecla, e o campo da estrutura EVENTO, denominado última tecla, será preenchido com a tecla não reconhecida pelo Gerenciador de Apresentação.

Na tabela 4.1, apresentamos todas as possibilidades de eventos passados para o Gerenciador de Diálogos, de acordo com os estilos de interação atualmente disponíveis em Agile.

	Flag	Evento	Ultima tecla
MENU	sucesso	No. da opção	Não preenche
	insucesso - STRING BARRA	Cadeia de caracteres Tecla	Tecla
COMANDO	sucesso	Cadeia de caracteres	Não preenche
	insucesso	Cadeia de caracteres	Tecla não reconhecida
PERGUNTA E RESPOSTA	sucesso	Cadeia de caracteres	Não preenche
	insucesso	Cadeia de caracteres	Tecla não reconhecida
FORMULARIO	sucesso	Cadeia de caracteres	Não preenche
	insucesso	String	Tecla não reconhecida

TABELA 4.1 - Eventos repassados pelo Gerenciador de Apresentação

é necessário ressaltar que para o estilo de interação formulário, os eventos são gerados por campo do formulário.

4.2.2 Análise de eventos

Para analisar o evento repassado pelo Gerenciador de Apresentação, o Gerenciador de Diálogos necessita consultar as informações de sequenciamento do diálogo que se encontra ativo.

Para cada estilo de interação existe um vetor, onde cada elemento corresponde a um diálogo, e consiste de:

- . Identificação do diálogo,
- . Identificação da apresentação do diálogo,
- . Apontador para a estrutura, onde estão as informações de sequenciamento do diálogo,
- . Identificação da janela, onde o diálogo será apresentado,
- . Posição do diálogo, dentro da janela.

A estrutura com as informações de sequenciamento do diálogo, contém basicamente:

- 1) Uma lista de ações associadas aos eventos considerados como válidos para aquele diálogo,
- 2) Uma lista com as teclas aplicáveis ao diálogo,
- 3) Uma lista de ações para quando forem gerados eventos que não correspondem àqueles definidos pelo projetista.

Estes vetores tem 120 campos, sendo os primeiros 60 campos reservados para interface das ferramentas de Agile, como por

exemplo o "Editor de Interfaces". Os 60 campos restantes são reservados para interfaces que serão prototipadas e possivelmente gerenciadas por Agile.

A seguir, para cada estilo de interação, serão apresentadas com detalhes estas estruturas.

- MENU - Estruturas de dados

Cada elemento do vetor de menus (fig. 4.2) contém:

- identificação de um diálogo menu,
- identificação da apresentação de um diálogo menu,
- identificação da janela, onde este menu será apresentado,
- posição deste menu, dentro da janela,
- apontador para a estrutura com as informações de sequenciamento deste menu (INFO_SEQ_MENU).

	Ident. diálogo	Id. diálogo	Ap. Id. janela	Posição	Apontador para INFO_SEQ_MENU
0	menu1	apmen1	jan3	6,3	→ INFO_SEQ_MENU
1					
2					
120					

FIGURA 4.2 - Vetor dos menus

A estrutura com informações de sequenciamento de um menu (fig. 4.3) contém:

- um apontador para a estrutura ITENS,

- um apontador para a estrutura AÇÖES, onde estão as ações para os eventos não reconhecidos.

- um apontador para a estrutura TECLAS APLICAVEIS.

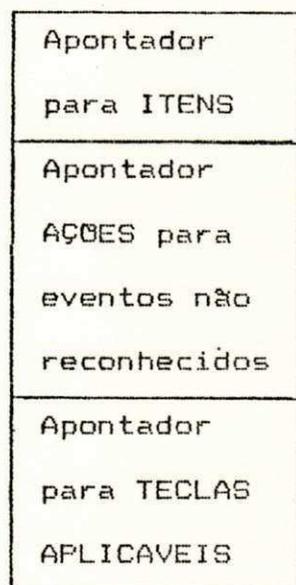


FIGURA 4.3 - Estrutura INFO_SEQ_MENU

A estrutura ITENS (fig. 4.4) é uma estrutura do tipo lista encadeada, onde cada elemento corresponde a um item do menu e contém:

- apontador para estrutura AÇÖES, que corresponde às ações a serem ativadas quando da seleção de um item.

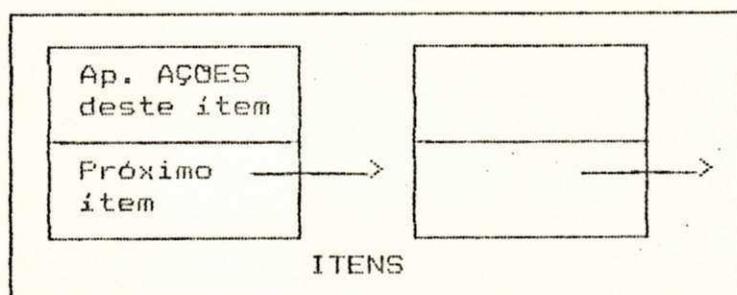


FIGURA 4.4 - Estrutura ITENS

AÇÖES é uma estrutura do tipo lista encadeada, onde cada

elemento da lista corresponde à descrição da ação a ser ativada. O conteúdo de cada elemento da lista AÇÕES está descrito na seção 4.3.2.

TECLAS APLICAVEIS é uma lista encadeada, onde cada elemento possui:

- identificação da tecla,
- apontador para a estrutura AÇÕES, que descreve as ações a serem ativadas caso a tecla seja pressionada.

- **COMANDO** - Estrutura de dados

Cada elemento do vetor de comandos (fig. 4.5) contém:

- Identificação de um diálogo comando,
- identificação da apresentação de um diálogo comando,
- identificação da janela, onde os comandos deverão ser digitados,
- posição da janela, onde serão digitados os comandos,
- apontador para a estrutura com as informações de sequenciamento dos comandos (INFO_SEQ_COMANDO).

	Ident. diálogo	Id. diálogo	Ap. janela	Id. janela	Posi ção	Apontador para INFO_SEQ_COMANDO
0	cmd1	apcmd1	jan3	2,4	—	→ INFO_SEQ_COMANDO
1						
2						
120						

FIGURA 4.5 - Vetor dos comandos

A estrutura com as informações de sequenciamento (fig. 4.6) para um diálogo do estilo comando contém:

- um apontador para a estrutura COMANDOS,
- apontador para a estrutura AÇÕES, que indicam as ações a serem ativadas para os eventos não reconhecidos,
- apontador para TECLAS APLICAVEIS.

Ap. para COMANDOS
Ap. para AÇÕES para comandos não reconhecidos
Ap. p/ TECLAS APLICAVEIS

FIGURA 4.6 - Estrutura INFO_SEQ_COMANDO

A estrutura COMANDOS (fig. 4.7) é uma lista encadeada, onde cada elemento corresponde à descrição de um comando e contém as seguintes informações:

- sintaxe do comando: nome, separador de parâmetros, delimitador inicial e final dos parâmetros,
- apontador para a estrutura PARAMETROS, onde estão descritos todos os parâmetros deste comando,
- apontador para a estrutura AÇÕES, onde estão as ações a serem ativadas, caso este comando seja gerado pelo usuário final,
- apontador para a estrutura TECLAS APLICAVEIS, que deve

conter as teclas a serem reconhecidas, caso o usuário final esteja editando este comando,

- apontador para a estrutura AÇÕES, onde está a sequência de ações que deve ser ativada, caso o usuário gere eventos não reconhecidos durante a edição deste comando.

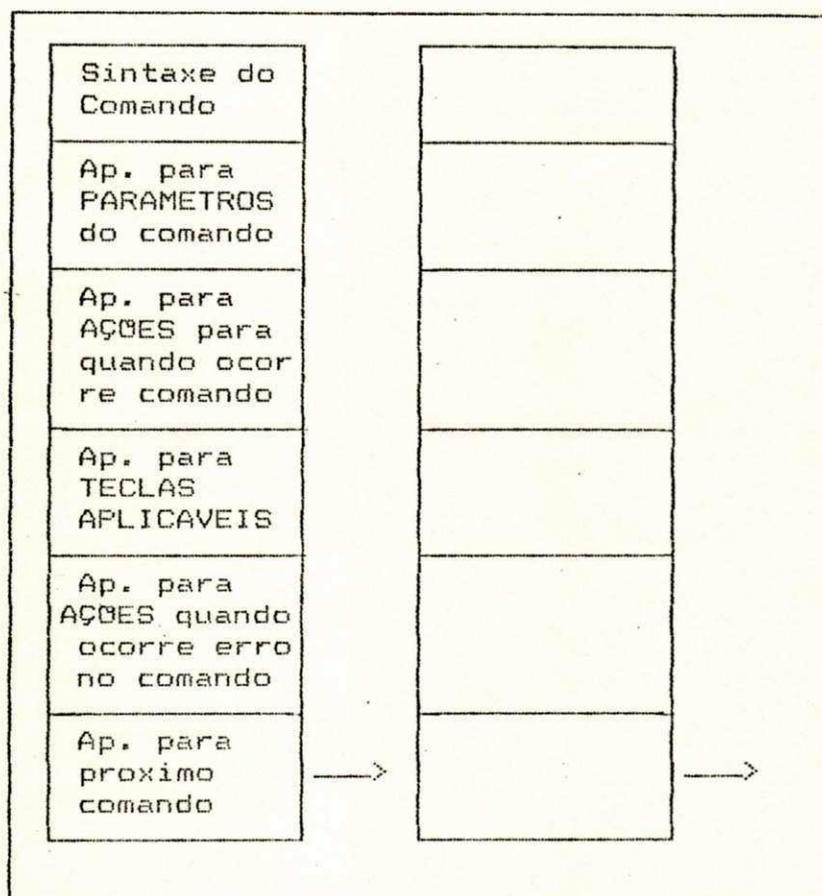


FIGURA 4.7 - Estrutura COMANDOS

PARAMETROS é uma estrutura do tipo lista encadeada, e cada elemento contém:

- um "flag", onde o projetista informa se o parâmetro deve estar presente quando da edição do comando ou não,
- valor do parâmetro,

- apontador para a estrutura AÇÕES, onde se encontram as ações que devem ser ativadas, caso tenha sido informado pelo usuário final um parâmetro errado.

- PERGUNTA E RESPOSTA - Estrutura de dados

Cada elemento do vetor de perguntas e respostas - P/R (fig. 4.8) contém:

- identificação de um diálogo P/R,
- identificação da apresentação de um diálogo P/R,
- identificação da janela, onde esta P/R será apresentada,
- posição desta P/R, dentro da janela,
- apontador para a estrutura com as informações de sequenciamento desta P/R (INFO_SEQ_PERGUNTA).

	Ident. diálogo	Id. Ap. diálogo	Id. janela	Posição	Apontador para INFO_SEQ_PERGUNTA
0	pr1	appr1	jan3	6,3	→
1					
2					
120					

INFO_SEQ_PERGUNTA

FIGURA 4.8 - Vetor das perguntas e respostas

A estrutura com as informações de sequenciamento para diálogos, do estilo pergunta e resposta (fig. 4.9), apresenta:

- apontador para estrutura RESPOSTAS,

- apontador para estrutura AÇÕES, que representam as ações a serem ativadas caso os eventos não correspondam a nenhuma das respostas,

- apontador para estrutura TECLAS APLICAVEIS, informando as teclas que podem ser reconhecidas quando o diálogo estiver ativo.

Ap. para RESPOSTAS associada a pergunta
Ap. para AÇÕES para resposta errada
Ap. para TECLAS APLICAVEIS

FIGURA 4.9 - Estrutura INFO_SEQ_PERGUNTA

RESPOSTAS (fig. 4.10) é uma estrutura do tipo lista encadeada, onde cada elemento corresponde a uma resposta que é considerada como válida para a pergunta. O conteúdo de cada elemento é o seguinte:

- cadeia de caracteres correspondendo ao texto da resposta,
- apontador para estrutura AÇÕES, informando a sequência de ações a serem ativadas, caso o evento gerado pelo usuário final, corresponda a esta resposta.

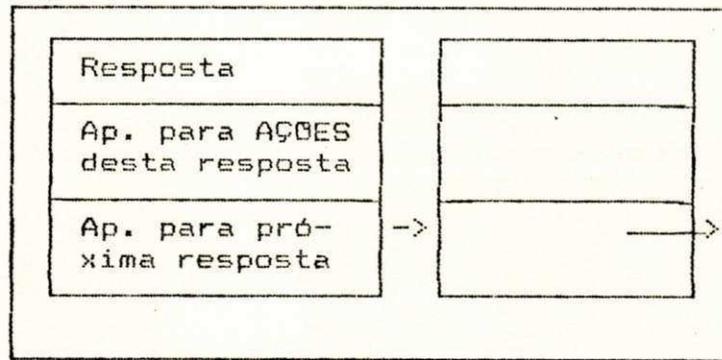


FIGURA 4.10 - Estrutura RESPOSTAS

- FORMULARIO - Estrutura de dados

Cada elemento do vetor de formulários (fig. 4.11) contém:

- identificação de um diálogo formulário,
- identificação da apresentação de um diálogo formulário,
- identificação da janela, onde este formulário será apresentado,
- apontador para a estrutura com as informações de sequenciamento deste formulário (INFO_SEQ_FORMULARIO).

	Ident. diálogo	Id. Ap. diálogo	Id. janela	Apontador para INFO_SEQ_FORMULARIO
0	form1	apform	jan3	→ INFO_SEQ_FORMULARIO
1				
2				
120				

FIGURA 4.11 - Vetor dos formulários

A estrutura contendo as informações de sequenciamento de um formulário (fig. 4.12) apresenta:

- apontador para a estrutura CAMPOS,
- apontador para a estrutura AÇÕES, que indica as ações a serem ativadas quando do término de preenchimento do formulário, por parte do usuário final,
- apontador para TECLAS APLICAVEIS, informando as teclas que devem ser reconhecidas quando este diálogo estiver ativo.

Ap. para CAMPOS do formulário
Ap. para AÇÕES quando término preenchimento do formulário
Ap. para TECLAS APLICAVEIS

FIGURA 4.12 - Estrutura INFO_SEQ_FORMULARIO

CAMPOS (fig. 4.13) é uma estrutura do tipo lista encadeada, onde, cada elemento representa um campo do formulário. Cada elemento contém:

- "flag", onde o projetista da interface informa se o campo deve ser preenchido para depois ser analisado, e em seguida ativadas as ações correspondentes; ou se devem ser automaticamente ativadas as ações,

- valor do campo,
- apontador para estrutura AÇÕES, indicando as ações a serem ativadas quando o usuário final gerar o evento correspondente ao valor do campo,
- apontador para estrutura AÇÕES, informando as ações que devem ser ativadas caso, o evento gerado para este campo não seja reconhecido,
- apontador para estrutura TECLAS APLICAVEIS, onde se encontram as teclas que devem ser reconhecidas quando o usuário final estiver gerando eventos para este campo do formulário.

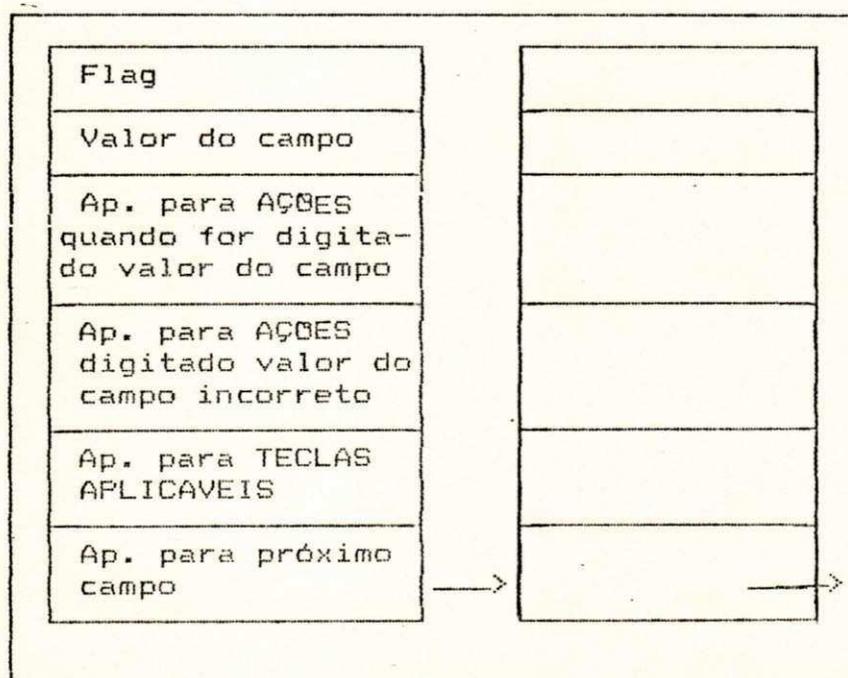


FIGURA 4.13 - Estrutura CAMPOS

Conhecidas as estruturas necessárias para análise de eventos, podemos iniciar a descrição de como é feita esta análise, mostrando o algoritmo analisa_evento, que é o responsável pela chamada das funções que analisam os eventos de cada estilo de interação.

```

analisa_evento()
início
  se estilo de interação do diálogo ativo for menu
    analisa_evento_menu();
  senão se estilo de interação do diálogo ativo for comando
    analisa_evento_comando();
  senão se estilo de interação do diálogo ativo for pergunta
    analisa_evento_pergunta();
  senão se estilo de interação do diálogo ativo for formulário
    analisa_evento_formulário();
  fim_se
fim

```

As funções para analisar os eventos de cada estilo de interação basicamente fazem o seguinte:

Se o flag da estrutura EVENTO for igual a sucesso:

- Percorre a lista de eventos definidos pelo projetista de interface como válidos até encontrar um evento que case com o evento gerado ou chegue-se ao final da lista. Caso encontre-se o evento, é passado para o "Sequenciador de Diálogos", o apontador da lista de ações que estão associadas a este evento. Caso chegue-se ao final da lista, o evento gerado é considerado inválido e é passado para o bloco "Sequenciador de Diálogos", o apontador da lista de ações definidas pelo projetista, para eventos não reconhecidos.

Se o flag da estrutura EVENTO estiver como insucesso:

- Será varrida a lista de teclas aplicáveis, tentando casar a tecla pressionada pelo usuário e não reconhecida pelo Gerenciador de Apresentação, com as definidas pelo projetista. Caso encontre a combinação, serão ativadas as ações associadas a esta tecla, senão serão ativadas as ações para eventos não reconhecidos.

- COMANDO - Algoritmo de análise de eventos

analise_evento_comando()

início

```
se o flag da estrutura EVENTO for igual a sucesso
decompõe o evento em nome do comando e parâmetros
faça para cada elemento da lista COMANDOS
| se o nome do comando for igual ao nome da lista COMANDOS
| | se os parâmetros são iguais aos da lista de PARAMETROS
| | | ativa_ações( AÇÕES de comando reconhecido );
| | senão
| | | ativa_ações( AÇÕES para erros de parâmetros );
| | fim_se
| fim_se
fim_faça
ativa_ações( AÇÕES para comando não reconhecido );
senão
retira do evento a parte referente ao nome do comando;
faça para cada elemento da lista COMANDOS
| se nome do comando for igual a nome da lista COMANDOS
| | se foram definidas teclas aplicáveis para o comando
| | | enquanto evento não corresponder as teclas aplicáveis
| | | | varre a lista TECLAS APLICAVEIS;
| | | | fim_enquanto
| | | se evento corresponder a uma das teclas aplicáveis
| | | | ativa_ações( AÇÕES da tecla aplicável );
| | | | fim_se
| | | fim_se
| | fim_se
fim_faça
se não casou o evento com nenhum dos comandos
| se existem teclas aplicáveis
| | enquanto não achar o evento na lista TECLAS APLICAVEIS
| | | varre a lista de TECLAS APLICAVEIS;
| | | fim_enquanto
| | se evento corresponder a uma das teclas aplicáveis
| | | ativa_ações( AÇÕES associada a tecla aplicável );
| | | fim_se
| | fim_se
fim_se
ativa_ações( AÇÕES para comando não reconhecido );
fim_se
```

fim

- PERGUNTA E RESPOSTA - Algoritmo de análise de eventos

analisa_evento_pergunta()

```
início
| se o flag de EVENTO for igual a sucesso
|   faça para cada elemento da lista RESPOSTAS
|     se evento for igual a resposta da lista RESPOSTAS
|       | ativa_ações( AÇÕES associadas a resposta );
|       fim_se
|     fim_faça
|   ativa_ações( AÇÕES associadas as respostas não reconhecidas);
senão
|   faça para cada elemento da lista TECLAS APLICAVEIS
|     se o evento corresponder a tecla aplicável
|       | ativa_ações( AÇÕES associadas a tecla aplicável );
|       fim_se
|     fim_faça
|   ativa_ações( AÇÕES para respostas não reconhecidas );
fim_se
fim
```

- FORMULARIO - Algoritmo de análise de eventos

analisa_evento_formulário()

```
início
| se o flag da estrutura EVENTO for igual a sucesso
|   se evento corresponde ao valor do campo
|     | ativa_ações( AÇÕES decorrentes do campo );
|     senão
|       | ativa_ações( AÇÕES para valor do campo não reconhecido );
|       fim_se
|   senão
|     enquanto não achar evento na lista TECLAS APLICAVEIS de campo
|       | varre a lista de TECLAS APLICAVEIS ao campo
|       fim_enquanto
|     se o evento não corresponde a nenhuma das teclas aplicáveis
|       | ignora o evento;
|       | fica esperando a geração de outro evento;
|     senão
|       | ativa_ações( AÇÕES decorrentes da tecla reconhecida );
|       fim_se
|     fim_se
fim
```

4.3 Sequenciador de diálogos

Entende-se por sequenciamento a especificação do que deve ser feito, estando-se em determinado ponto da interação e recebendo-se um evento.

Cientes do que é sequenciamento, padronizaram-se ações, de forma que o projetista, ao especificar o sequenciamento de sua interface, pudesse definir que o diálogo a se tornar ativo a seguir é:

- um diálogo anterior ao que está ativo no momento,
- continue sendo o mesmo diálogo que já se encontra ativo,
- outro diálogo qualquer.

Além das ações que definem o sequenciamento, foram padronizadas ações para o projetista gerenciar mensagens dentro de sua interface, como também dois tipos de ação que permitem ao projetista informar ao Gerenciador de Diálogos, que ele deve chamar uma determinada função da aplicação.

Todas estas ações serão explicadas com mais detalhes mais adiante na seção 4.3.2.

4.3.1 Filha de diálogos

Antes de apresentarmos as ações usadas para definir o sequenciamento, passamos a apresentar a estrutura de dados que nos auxilia na execução destas ações.

Esta estrutura tem como objetivo guardar o caminho percorrido para se chegar ao diálogo que se encontra ativo

atualmente. Desta forma podemos reativar diálogos ativados anteriormente.

Cada elemento da pilha corresponde a um diálogo, e contém as seguintes informações:

- identificação do diálogo,
- apontador para a estrutura EVENTO, que foi alocada quando da ativação do diálogo.

- número do campo do formulário que se encontrava sendo preenchido quando este diálogo era o ativo. Esta informação só é preenchida para o diálogo de estilo formulário.

- apontador para uma lista, onde cada elemento contém o dado digitado pelo usuário final, para cada campo do formulário que está sendo sequenciado.

As operações sobre esta pilha serão apresentadas juntamente com a descrição das ações.

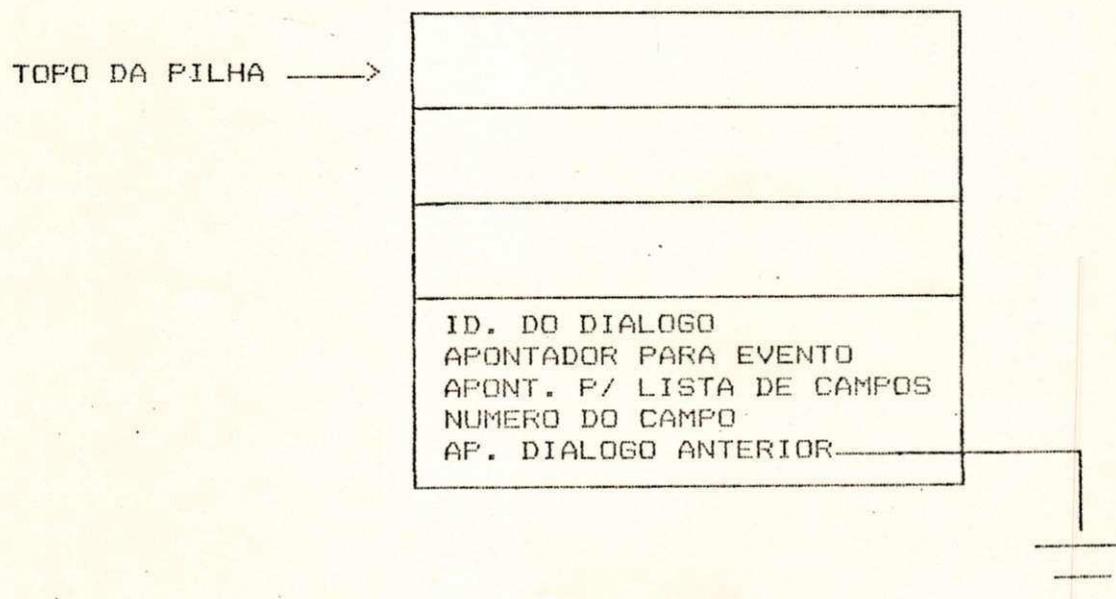


FIGURA 4.14 - Estrutura da pilha de diálogos

4.3.2 Descrição das ações

Como já foi dito anteriormente, a estrutura onde as ações são descritas corresponde a uma lista encadeada, onde cada elemento contém:

1. tipo da ação a ser executada,
2. atributo da ação, onde estes atributos podem ser:
 - i. informações de um diálogo:
 - estilo de interação,
 - identificação do diálogo,
 - número do campo do formulário, caso o estilo de interação seja formulário.
 - ii. informações de uma rotina da aplicação:
 - número da rotina,
 - apontador para a estrutura RETORNOS, que será apresentada mais adiante.
 - iii. valor inteiro.

A figura 4.15 mostra a estrutura correspondente a uma ação:

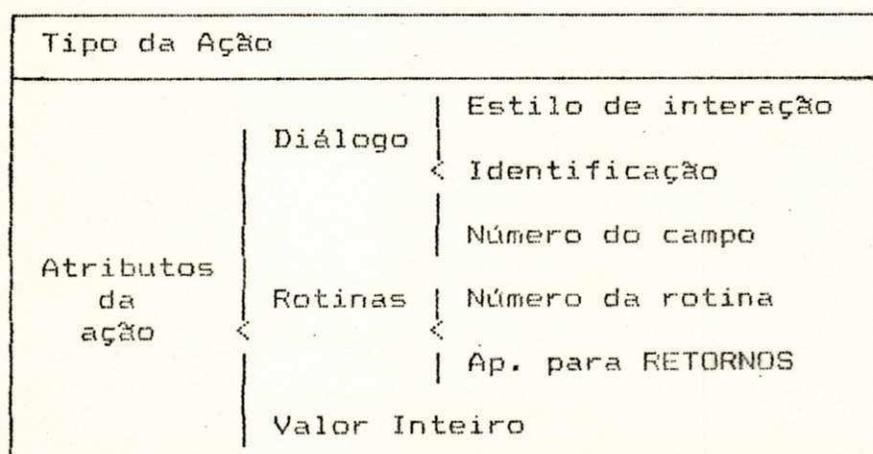


FIGURA 4.15 - Estrutura de uma ação

As ações definidas são:

- **ATIVAR DIALOGO** -> Tornar ativo um novo diálogo, ou seja, associar os eventos (entradas do usuário) com o diálogo ativado.

Os atributos desta ação são:

- . Estilo de interação: - Menu, comando, pergunta e resposta, e formulário.

- . Identificação do diálogo.

- . Número do campo que deve receber eventos, caso o estilo de interação seja formulário. Caso este dado não seja fornecido, o campo considerado será o primeiro campo do formulário.

- **VOLTAR DIALOGO** -> Ativar um diálogo que já foi ativado anteriormente.

Os atributos desta ação são:

- . Estilo de interação

- . Identificação do diálogo.

- . Número do campo que deve receber eventos, caso o estilo de interação seja formulário. Caso este campo não seja fornecido, os eventos serão gerados para o campo posterior ao último campo preenchido.

- **CONTINUAR DIALOGO** -> Continuar a receber os eventos (entradas dos usuários) do diálogo que se encontra ativo no instante da chamada desta ação.

Os atributos desta ação:

. Número do campo que deve receber eventos, caso o estilo de interação seja formulário. Caso esta informação não seja passada, o campo a receber eventos será o campo posterior ao último campo que recebeu eventos.

- CONTINUAR A LER DIALOGO -> Terminar de digitar uma entrada de um diálogo que por algum evento foi interrompido.

Os atributos desta ação:

. Estilo de interação
. Identificação do diálogo.
. Número do campo que deve receber eventos, caso o estilo de interação seja formulário. Se esta informação não foi definida, o campo a receber eventos será o último que os tinha recebido anteriormente.

- ENCERRAR AGILE -> Encerra a interação com Agile e volta ao Sistema Operacional.

- ENCERRAR SIMULAÇÃO -> Encerra a simulação que está sendo realizada, retornando o controle para Agile.

- ENCERRAR DIALOGO -> Fecha o diálogo que se encontra ativo.

- APRESENTAR MENSAGEM -> Apresenta em determinada posição

da tela uma mensagem.

- RETIRAR MENSAGEM -> Retira da tela uma mensagem.

- SUSPENDER EXECUÇÃO -> Suspende a simulação da interface por um intervalo de tempo.

O atributo desta ação é:

• Valor inteiro, informando o tempo em segundos que a simulação deve ser suspensa.

Como já foi dito na seção 4.3, foram desenvolvidas duas ações para funcionar como elo de comunicação entre a interface e a aplicação. Estas ações são:

- CHAMAR PROCEDIMENTO -> Permite que sejam chamadas as rotinas da aplicação as quais não retornam nenhum valor.

- CHAMAR FUNÇÃO -> Chama as rotinas da aplicação as quais retornam um valor, e cujo sequenciamento dado, após a chamada da função, é dependente do valor retornado.

Para se utilizar estas ações, é necessário:

- Editar o arquivo "rotaplic.c", e inserir na definição de "vet_proc" e "vet_func", os procedimentos e funções da aplicação que devem ser chamados como ações de algum evento gerado. Um exemplo da definição destes vetores é apresentado a seguir:

```

void vet_proc          int vet_func
    proc1,             func1,
    proc2              func2

```

- Informar o índice do vetor de procedimentos ou de funções, onde se encontra o procedimento ou função que deve ser chamada quando uma destas ações for ativada.

- Preencher o campo "número da rotina" da estrutura ação, com o índice do vetor de procedimento (vet_proc) ou vetor de função (vet_func) correspondente ao procedimento ou função a ser chamada.

- Ligar com o código de Agile as funções e procedimentos da aplicação.

A estrutura RETORNOS, é uma estrutura do tipo lista encadeada, onde cada elemento contém os possíveis valores retornados por uma função da aplicação, associando a cada um dos valores as ações que devem ser ativadas quando um deles for retornado. A figura 4.16 apresenta esta estrutura.

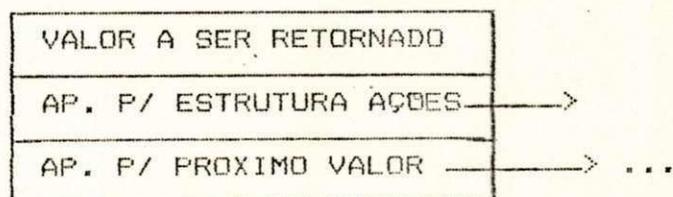


FIGURA 4.16 - Estrutura RETORNOS

4.3.3 - Ativação das ações

Uma vez descrita as ações disponíveis em Agile, passamos a apresentar como elas são executadas.

Inicialmente apresentamos o algoritmo que mostra como as ações decorrentes de um evento são ativadas.

```
ativa_ações()
início
  enquanto existe ação a ser ativada
  | se tipo da ação for Ativar Diálogo
  | | ativar_diálogo()
  | fim-se
  | se tipo da ação for Voltar Diálogo
  | | voltar_diálogo()
  | fim-se
  | se tipo da ação for Continuar Diálogo
  | | continuar_diálogo()
  | fim-se
  | se tipo da ação for Continuar a ler Diálogo
  | | continuar_ler_diálogo()
  | fim-se
  | se tipo da ação for Encerrar Agile
  | | parar()
  | fim-se
  | se tipo da ação for Encerrar Simulação
  | | encerrar_simulação()
  | fim-se
  | se tipo da ação for Encerrar Diálogo
  | | encerrar_diálogo()
  | fim-se
  | se tipo da ação for Suspende Execução
  | | suspender_execução()
  | fim-se
  | se tipo da ação for Apresentar Mensagem
  | | apresentar_mensagem()
  | fim-se
  | se tipo da ação for Retirar Mensagem
  | | retirar_mensagem()
  | fim-se
  | se tipo da ação for Chamar Procedure
  | | chamar_procedure()
  | fim-se
  | se tipo da ação for Chamar Função
  | | chamar_função()
  | fim-se
  fim-enquanto
fim
```

Apresentamos a seguir, o algoritmo mostrando como cada uma destas ações foram implementadas.

ATIVAR DIALOGO

```
ativar_dialogo()
início
| Chama a função ABRIR_DIALOGO;
| Aloca espaço para a estrutura EVENTO;
| Empilha o diálogo na estrutura PILHA DE DIALOGOS;
| Fica esperando a geração de eventos;
fim
```

VOLTAR DIALOGO

```
voltar_dialogo()
início
| se não foi definido o estilo de interação
|   | chama a função FECHAR_DIALOGO();
|   | desempilha um diálogo da estrutura PILHA DE DIALOGOS
|   | chama a função ABRIR_DIALOGO();
|   | zera a estrutura EVENTO;
|   | fica esperando a geração de eventos;
| senão
|   enquanto não achar o diálogo que deverá se tornar ativo
|   | percorre a estrutura PILHA DE DIALOGOS;
|   fim do enquanto
|   se não encontrar
|   | apresenta uma mensagem de erro;
|   senão
|   | enquanto não chegar ao diálogo que se tornará ativo
|   |   | chama a função FECHAR_DIALOGO();
|   |   | desempilha um diálogo da estrutura PILHA DE DIALOGOS;
|   |   fim do enquanto
|   |   chama a função ABRIR_DIALOGO();
|   |   zera a estrutura EVENTO;
|   |   fica esperando a geração de eventos;
|   fim_se
| fim_se
fim
```

CONTINUAR DIALOGO

```
continuar_dialogo()
início
| zerar a estrutura EVENTO;
| fica esperando a geração de eventos;
fim
```

CONTINUAR A LER DIALOGO

continuar_a_ler_dialogo()

início

```
| se não foi definido o estilo de interação
|   chama a função FECHAR_DIALOGO();
|   desempilha um diálogo da estrutura PILHA DE DIALOGOS
|   chama a função ABRIR_DIALOGO();
|   fica esperando a geração de eventos;
senão
  enquanto não achar o diálogo que deverá se tornar ativo
  | percorre a estrutura PILHA DE DIALOGOS;
  fim do enquanto
  se não encontrar
  | apresenta uma mensagem de erro;
  senão
  | enquanto não chegar ao diálogo que se tornará ativo
  |   chama a função FECHAR_DIALOGO();
  |   desempilha um diálogo da estrutura PILHA DE DIALOGOS;
  fim do enquanto
  chama a função ABRIR_DIALOGO();
  fica esperando a geração de eventos;
  fim_se
fim_se
fim
```

ENCERRAR DIALOGO

encerrar_dialogo()

início

```
| chama a função FECHAR_DIALOGO()
| desempilha um diálogo da estrutura PILHA DE DIALOGOS
fim
```

APRESENTAR OU RETIRAR MENSAGEM

Para executar estas ações, foi necessário criar um vetor, onde cada elemento, refere-se a uma mensagem definida pelo projetista de interface no seu protótipo, e contém um "flag" informando se a mensagem está sendo apresentada ou não, em determinado ponto da interação, para o caso de ocorrer uma ação de retirar esta mensagem.

```
apresentar_mensagem()  
início  
| se a mensagem ainda não está sendo apresentada  
| | ABRE_MENSAGEM()  
| fim_se  
fim
```

```
retirar_mensagem()  
início  
| se a mensagem se encontra na tela  
| | FECHA_MENSAGEM()  
| fim_se  
fim
```

CHAMAR PROCEDURE

```
chamar_procedure()  
início  
| execute vet_proc( índice da procedure a ser chamada )  
fim
```

CHAMAR FUNÇÃO

```
chamar_função  
início  
| retorno = vet_func( índice da função a ser chamada )  
| enquanto não for fim da lista de RETORNOS ou não achar o retorno  
| | se retorno for igual a um dos valores esperados como retorno  
| | | ative ações associadas ao retorno  
| | | fim_se  
| fim do enquanto  
| ative ações para o caso de retornos não esperados  
fim
```

5.0 - FERRAMENTAS DE PROTOTIPAGEM

No ambiente Agile estão disponíveis duas ferramentas para prototipagem de interfaces, uma para especificação do protótipo - Editor de Interfaces, e outra para geração de documentação das interfaces prototipadas - Gerador de Documentação. Além destas ferramentas, estão previstas entre outras, um editor gráfico, para a definição dos símbolos gráficos necessários à modelagem de interfaces gráficas, e um módulo de avaliação de interfaces.

5.1 - Filosofia de desenvolvimento

Como já foi ressaltado, as ferramentas de prototipagem foram desenvolvidas como aplicações, cujas interfaces são gerenciadas por Agile.

Para desenvolver estas ferramentas segundo esta filosofia, foi necessário:

1. Expandir o Gerenciador de Diálogos, no sentido de implementar duas ações que funcionassem como elo de comunicação entre a aplicação e a interface (ver seção 4.3.2).

2. Definir interfaces para as ferramentas, através da criação de protótipos, baseados no paradigma de interface de Agile.

3. Implementar as funções particulares a cada uma das ferramentas.

4. Acoplar aos protótipos de interface dessas ferramentas, as funções desenvolvidas no passo 3, através das ações CHAMAR FUNÇÃO e CHAMAR PROCEDIMENTO (ver seção 4.3.3).

5.2 - Editor de Interfaces

Desenvolvido com o objetivo, de permitir que os projetistas de interface definam seus protótipos, através de uma linguagem de definição interativa. (ver Guia do usuário).

Estes protótipos são gravados em arquivos sequenciais, com extensão "FTP", de forma a serem usados para a simulação, ou para reedição.

A interação com o editor se processa através de formulários e menus. Os formulários são usados para entrada dos dados que definem os protótipos. Os menus são utilizados para guiar o projetista na navegação da interface e para fornecer as alternativas de preenchimento de alguns dos campos dos formulários - menus do editor.

Para chegar aos formulários, o projetista de interface informa o que deseja especificar, através da seleção em menus, repassando ao sistema o "path" dentro de uma árvore de sequenciamento das especificações.

5.2.1 - Árvore de sequenciamento

A figura 5.1 apresenta a árvore de sequenciamento para chegar aos formulários. Os nós que se encontram em um mesmo nível

da árvore, são itens de um menu, logo cada nível da árvore representa um diálogo de estilo de interação menu. O deslocamento entre os níveis se dá através das ações ATIVAR DIALOGO e VOLTAR DIALOGO (ver seção 4.3.2).

O sequenciamento dado a partir dos nós terminais dessa árvore consiste em ativar os diálogos de estilo formulário, para aquisição dos dados que irão compor o protótipo.

Foi na definição do sequenciamento destes formulários, que houve a integração das funções da aplicação "Editor de Interfaces" com a sua interface.

Nas seções 5.2.2 e 5.2.3, apresentaremos respectivamente um exemplo da definição do sequenciamento de um diálogo de estilo menu, e do sequenciamento de um diálogo de estilo formulário.

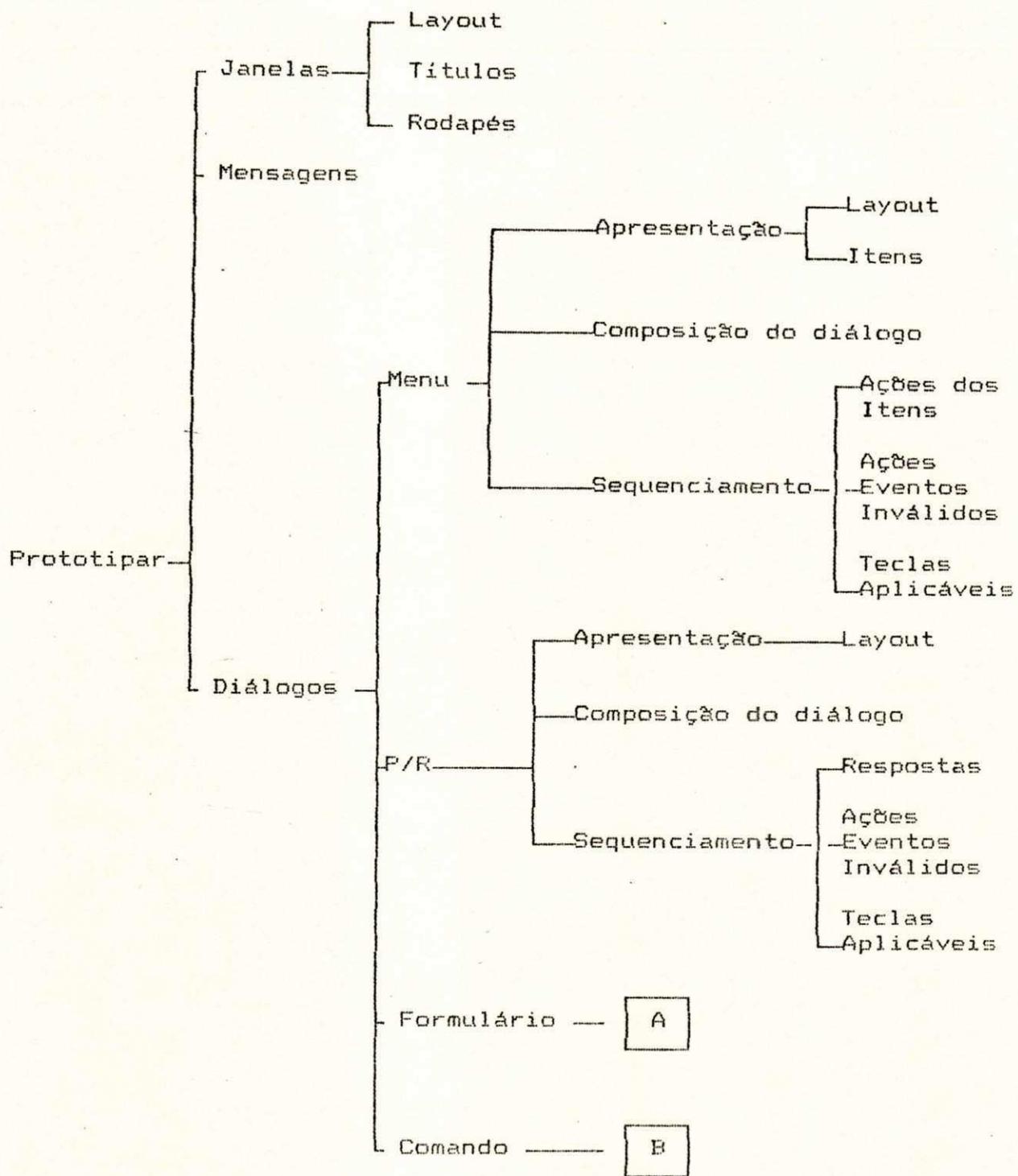


FIGURA 5.1a - Árvore de sequenciamento dos menus

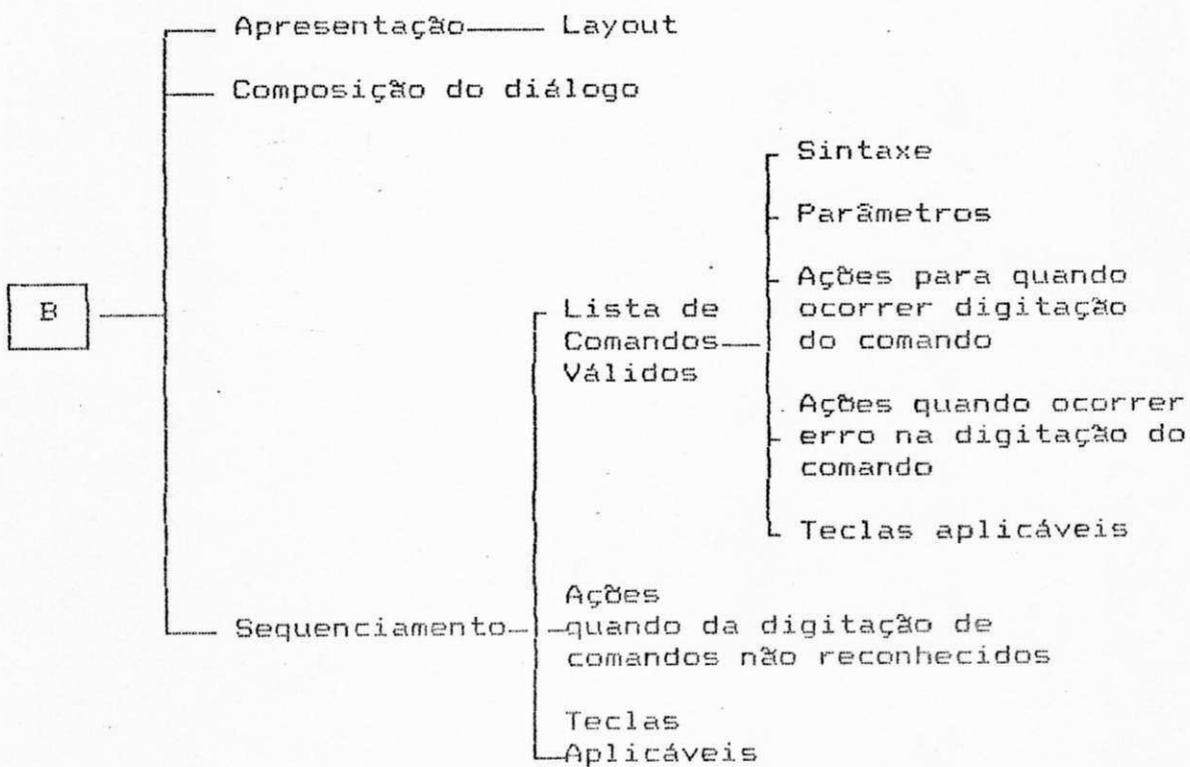
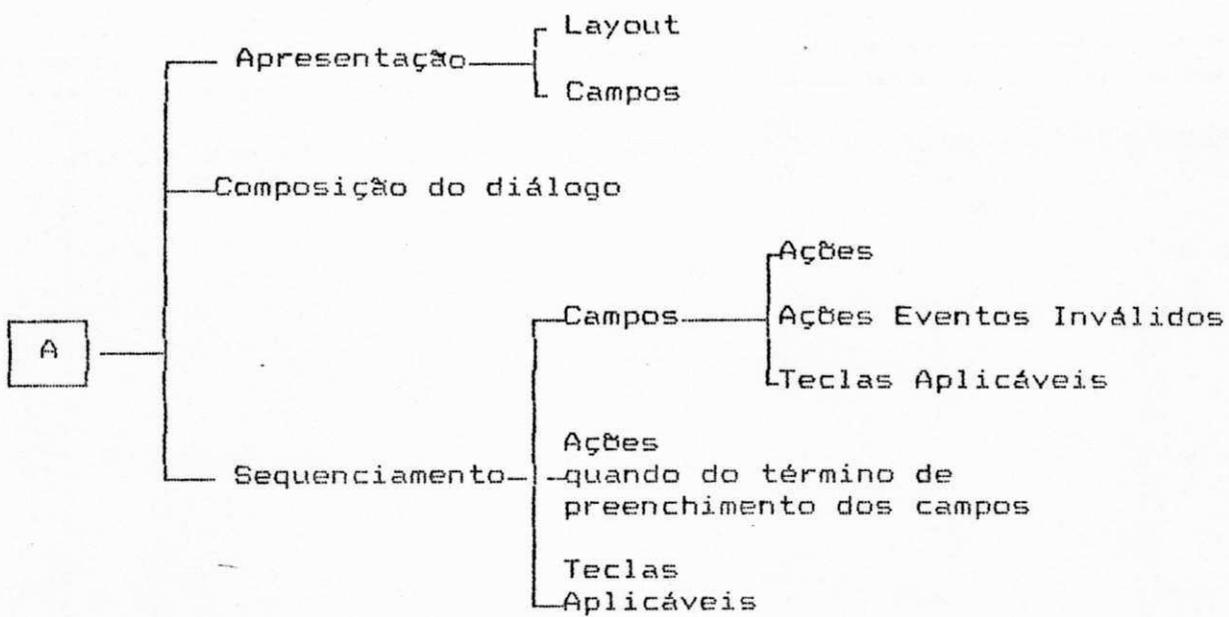
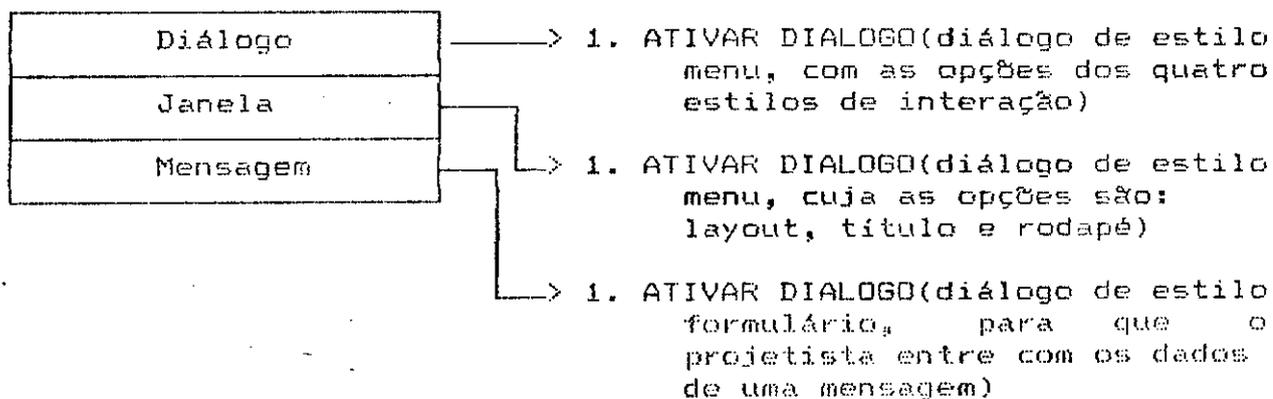


FIGURA 5.1b - Árvore de sequenciamento dos menus

5.2.2 Exemplo do sequenciamento de um menu

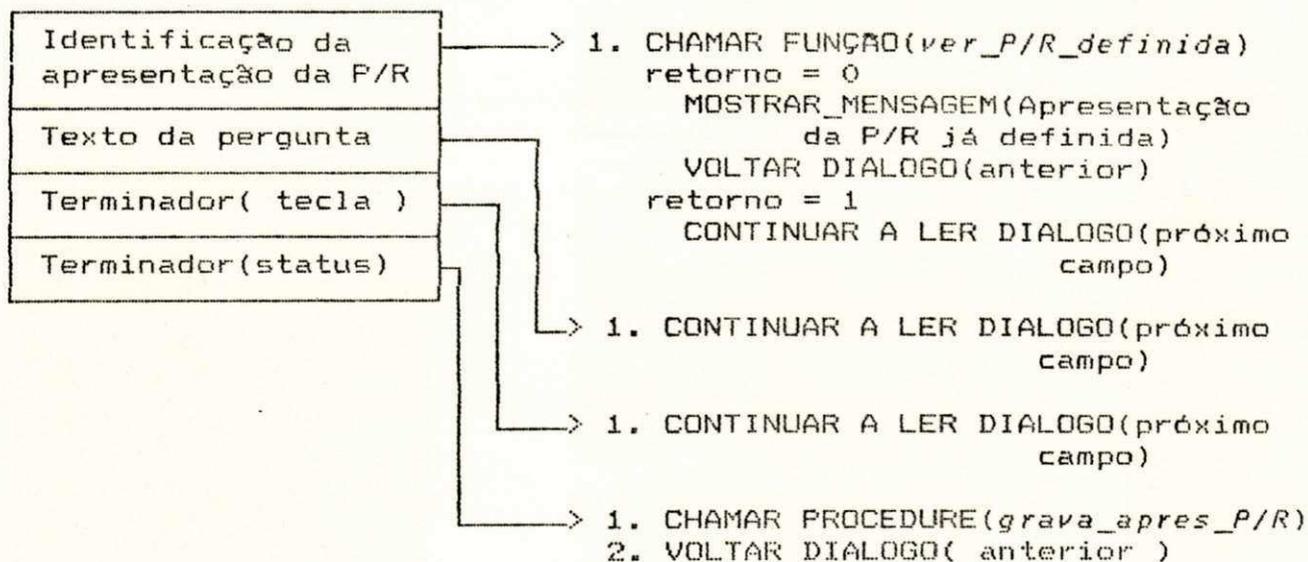
Como já foi apresentado no capítulo 4, para se definir o sequenciamento de um menu, é necessário se definir para cada item, as ações que deverão ser ativadas quando o item for selecionado.

O diálogo de estilo menu da interface do Editor de Interfaces, que foi selecionado para este exemplo, é o menu onde o projetista seleciona, qual dos elementos que compõem o modelo de interface de Agile (diálogo, janela e mensagem), ele deseja prototipar.



5.2.3 Exemplo do sequenciamento de um formulário

O diálogo da interface do "Editor de Interfaces", que foi usado para apresentação deste exemplo, foi o formulário que é utilizado, para adquirir os dados de apresentação de um diálogo de estilo de interação pergunta e resposta.



As estruturas usadas para edição da apresentação de P/R são as seguintes:

- . A estrutura definida no Gerenciador de Apresentação, para guardar os dados da apresentação de uma P/R. (Estrutura APRES_PR, [SANTOS 92]).

- . Uma estrutura do tipo vetor, onde para cada apresentação de P/R, guarda informações sobre a situação das mesmas: se a apresentação da P/R está definida ou não, e se está sendo referenciada na composição de algum diálogo de estilo P/R. (Estrutura VET_APRES_PR).

- . A estrutura FILHA_DE_DIALOGOS (ver seção 4.3.1), onde são guardados os eventos gerados para cada campo deste formulário.

Algoritmos das funções de edição das apresentações de P/R:

```
int ver_P/R_definida
```

```
início
```

```
| Enquanto não achar identificação da P/R
```

```
  varre o VET_APRES_P/R
```

```
  se achar
```

```
    retorna(0)
```

```
  senão
```

```
    retorna(1)
```

```
  Fim_enquanto
```

```
fim
```

```
grava_apres_P/R
```

```
início
```

```
| aloca espaço na estrutura APRES_P/R
```

```
| varre a lista de campos da estrutura PILHA_DE_DIALOGOS, gravando  
os dados na estrutura APRES_P/R
```

```
| coloca no VET_APRES_P/R a informação de que a apresentação da  
P/R já está definida
```

```
fim
```

5.3 - Gerador de documentação

Esta outra ferramenta, tem como objetivo, fornecer documentos ao projetista, com a especificação dos seus protótipos.

A partir do arquivo que contém a especificação do protótipo da interface, é gerado um arquivo documento, descrevendo de forma detalhada e organizada todas as informações que compõem o paradigma de uma interface, descrito na seção 3.2.1. Este arquivo poderá em seguida ser editado, em qualquer editor de texto, permitindo ao projetista acrescentar comentários e/ou organizá-lo da maneira que melhor lhe convier. Este documento formal tem como objetivo principal, fornecer ao projetista uma referência para que ele possa utilizar durante a implementação da interface especificada, caso esta aplicação tenha outro ambiente alvo que

não o ambiente IBM-PC.

O arquivo gerado tem o mesmo nome dado ao protótipo, mas com a extensão "DOC".

A interface do gerador se resume a um diálogo de estilo "pergunta e resposta", que é utilizado para obter do projetista, a informação sobre qual dos seus protótipos ele deseja adquirir a documentação.

O sequenciamento dado a partir da digitação do nome do arquivo que contém o protótipo, é o seguinte:

1. CHAMAR FUNÇÃO(*ver_arquivo_existe*)

retorno = 0

CHAMAR PROCEDURE(*ler_arquivo_protótipo*)

CHAMAR PROCEDURE(*gerar_documento_protótipo*)

retorno = 1

MOstrar MENSAGEM(protótipo inexistente)

```
int ver_arquivo_existe
início
| se o arquivo existir
| | retorna(0)
| senão
| | retorna(1)
| fim_se
fim
```

```
ler_arquivo_protótipo
início
| abrir o arquivo onde está o protótipo
| carregar o protótipo nas estruturas do Gerenciador de
| Apresentação e de Diálogos
| fechar o arquivo do protótipo
fim
```

gerar_documento_protótipo

início

```
| abrir o arquivo com extensão "DOC" para guardar a documentação  
| para cada janela definida no protótipo  
|   gera e grava no arquivo um documento de definição da janela  
fim_para  
| para cada mensagem definida no protótipo  
|   gera e grava no arquivo um documento de definição da mensagem  
fim_para  
| para cada apresentação de diálogo definida no protótipo  
|   gera e grava no arquivo um documento de definição da  
|   apresentação do diálogo  
fim_para  
| para cada sequenciamento de diálogo definido no protótipo  
|   gera e grava no arquivo um documento de definição do  
|   sequenciamento do diálogo  
fim_para  
| fechar o arquivo  
fim
```

As estruturas utilizadas pelo "Gerador de Documentação" são as seguintes:

- Estruturas definidas no Gerenciador de Apresentação, que guardam a definição da apresentação dos protótipos (ver [SANTOS 92]).

- Estruturas do Gerenciador de Diálogos, onde são guardadas a definição do sequenciamento da interface prototipada.

Concluimos este capítulo com a apresentação de um exemplo de documento gerado por esta ferramenta. Este documento pode ser enriquecido com uma impressão das telas criadas pelo projetista, e que pode ser obtido através de software aplicativos já existentes.

JANELAS

Id. da janela.....: "jm2"
Dimensao.....: 111, 2, 9, 191
Padrao de preenchimento: 1
Cor de preenchimento...: Branca
Borda.....: Nao

TITULOS.....: Nao tem

RODAFES.....: Nao tem

MENSAGENS

Id. da mensagem.....: "msg2"
Fonte.....: DEFAULT_FONT
Tamanho do caracter: 1
Cor.....: BRANCA
Texto da mensagem...: "Prototipar"

DIALOGOS - APRESENTACAO

MENUS

Id. do menu.....: "m2"
Padrao de preenchimento: 1
Cor de preenchimento...: Branca
Wrap.....: Sim
Direcao.....: Vertical
Tipo de selecao.....: Barra
Terminador.....: Tecla: 1c0d Status: 0

ITENS.....: Texto do item.....: "Dialogo"
Mnemonico.....:
Fonte.....: DEFAULT_FONT
Tamanho do caracter: 1
Cor.....: Branca

 Texto do item.....: "Janela"
 Mnemonico.....:
 Fonte.....: DEFAULT_FONT
 Tamanho do caracter: 1
 Cor.....: Branca

 Texto do item.....: "Mensagem"
 Mnemonico.....:
 Fonte.....: DEFAULT_FONT
 Tamanho do caracter: 1
 Cor.....: Branca

PERGUNTA E RESPOSTA

Id. da pergunta...: "pa1"
Terminador.....: Tecla: 1c0d Status: 0
Texto da pergunta: "Prototipo: "

FORMULARIO

Id. do formulario: "fa3"
Terminador.....: Tecla: 1c0d Status: 0

CAMPOS: Texto do campo.....: "Id. Menu.....:"
Posicao.....: 10 - 10
Fonte.....: DEFAULT_FONT
Tamanho do caracter.....: 1
Cor.....: Preto
Posicao do dado.....: 10 - 186
Fonte do dado.....: DEFAULT_FONT
Tamanho do caracter do dado: 1
Cor do dado.....: Preto
Largura do dado.....: 5

Texto do campo.....: "Id. Apresentacao.....:"
Posicao.....: 28 - 10
Fonte.....: DEFAULT_FONT
Tamanho do caracter.....: 1
Cor.....: Preto
Posicao do dado.....: 28 - 186
Fonte do dado.....: DEFAULT_FONT
Tamanho do caracter do dado: 1
Cor do valor.....: Preto
Largura do dado.....: 1

Texto do campo.....: "Janela.....:"
Posicao.....: 46 - 10
Fonte.....: DEFAULT_FONT
Tamanho do caracter.....: 1
Cor.....: Preto
Posicao do dado.....: 46 - 186
Fonte do dado.....: DEFAULT_FONT
Tamanho do caracter do dado: 1
Cor do dado.....: Preto
Largura do dado.....: 1

Texto do campo.....: "Posicao.....:"
Posicao.....: 64 - 10
Fonte.....: DEFAULT_FONT
Tamanho do caracter.....: 1
Cor.....: Preto
Posicao do dado.....: 64 - 186
Fonte do dado.....: DEFAULT_FONT
Tamanho do caracter do dado: 1
Cor do dado.....: Preto
Largura do dado.....: 1

DIALOGOS - SEQUENCIAMENTO

M E N U S

Id. do dialogo.....: "m2"
Id. da apresentacao.....: "m2"
Id. da janela.....: "jm2"
Posicao na janela.....: 8, 8

ITENS:-No. do item: 0

ACOES: 0)MOSTRAR MENSAGEM msg2, janela jcam, posicao (93,1)

1)ATIVAR DIALOGO tipo MENU e identificacao m3

-- No. do item: 1

ACOES: 0) ATIVAR DIALOGO tipo MENU e identificacao m4

- No. do item: 2

ACOES: 0) ATIVAR DIALOGO tipo FORMULARIO e identificacao f10

ACOES PARA EVENTOS NAO RECONHECIDOS: Nao tem

TECLAS APLICAVEIS: Nao tem

P E R G U N T A E R E S P O S T A

Id do dialogo.....: "p1"
Id. da apresentacao.: "pa1"
Id. da janela.....: "j2"
Posicao na janela....: 10 - 5

RESPOSTAS: Id. da resposta...: "r1"
Texto da resposta: ""

ACOES: 0)RETIRAR MENSAGEM msg7, janela jmsg, posicao (8,2)

1)ENCERRAR O DIALOGO

2)MOSTRAR MENSAGEM msg8, janela jmsg, posicao (8,2)

3)ATIVAR DIALOGO tipo MENU e identificacao m2

TECLAS APLICAVEIS: Nao tem

ACOES PARA EVENTOS NAO RECONHECIDOS: Nao tem

6.0 CONCLUSOES

6.1 Considerações

Consideramos que o principal objetivo deste trabalho, que foi desenvolver o módulo **Gerenciador de Diálogos**, e integrá-lo com o **Gerenciador de Apresentação** desenvolvido em [SANTOS 92], para compor o sistema de prototipagem rápida de interfaces - Agile foi alcançado.

Além do módulo **Gerenciador de Diálogos**, foram desenvolvidas também as ferramentas **Editor de Interfaces** e **Gerador de Documentação**, que auxiliam os projetistas de interface na definição dos seus protótipos, cumprindo assim com os demais objetivos propostos para o trabalho.

O desenvolvimento destas ferramentas, levou-nos a expandir o sistema de prototipagem vazia para um sistema que fizesse o gerenciamento não apenas de interface, mas também da aplicação, dando-lhe assim características de um SGIU.

Analisando o sistema implementado, segundo critérios, sugeridos por [PRIME 90], e comparando-o com outros SGIU's, concluímos que o seu enquadramento nesta classe é possível. No entanto ainda são necessários testes específicos e a utilização por outras aplicações além das ferramentas do Agile, para formalizarmos esta afirmativa.

A seguir, descreveremos os critérios utilizados na análise de SGIUs segundo [PRIME 90], para que possamos situar o sistema

Agile dentro desta perspectiva.

1) **Separação entre a interface do usuário e a aplicação:** O código do programa da aplicação deve ser independente do código de sua interface com o usuário, de forma que seja possível escrever cada código separadamente.

Esta característica em um SGIU é bastante importante, em virtude de permitir que se faça modificações na interface, sem necessidade de alteração no código da aplicação.

Em Agile, se define inicialmente a interface, podendo simular a mesma, sem ter implementado nenhuma função da sua aplicação, e depois desenvolver as funções da sua aplicação, associando estas a interface definida, através das ações CHAMAR FUNÇÃO e CHAMAR PROCEDIMENTO.

2) **O controle da interação é feito pelo SGIU ou pela aplicação:** O controle do sistema pode ser feito pelo SGIU via entrada através da interface, ou pela aplicação. Se o SGIU tem o controle, ele chama módulos da aplicação em resposta às entradas do usuário final. Este controle é chamado externo. Se o controle é sempre uma consequência do código da aplicação, então o controle é dito ser interno. Uma terceira forma de controle ocorre quando o usuário tem completo controle. Um verdadeiro SGIU deve possuir controle externo.

Não observou-se nenhuma vantagem neste critério, mas no caso de Agile, o controle é externo, ou seja, é ele quem solicita

entradas dos usuários e decide o que fazer.

3) **Comunicação entre a aplicação e o SGIU:** Esta característica reflete a natureza da comunicação em tempo de execução, entre a interface e o usuário final, e entre a interface e os módulos da aplicação. É levado em consideração, se a comunicação entre os processos é feita através da passagem de mensagens, ou através de eventos.

O usuário final de uma interface deve poder interagir com múltiplos periféricos de entrada, e o SGIU é quem deve decidir qual ele está usando (por exemplo mouse ou teclado).

A comunicação de informação entre a interface e os módulos da aplicação deve incluir a transmissão de solicitações de usuários, dados de usuário e resultados da aplicação.

Em Agile, a comunicação entre a interface e o usuário final é através de eventos, e a comunicação entre a interface e os módulos da aplicação é feita através das ações CHAMAR FUNÇÃO e CHAMAR PROCEDIMENTO.

4) **Ferramentas acopladas aos SGIUs:** São ferramentas para desenvolvimento de interfaces, permitindo sua geração interativa.

Agile possui duas ferramentas, uma para edição interativa dos protótipos de interface (Editor de Interfaces), e outra que gera um documento do protótipo da interface (Gerador de Documentação).

Com base nestes critérios, acrescentamos à tabela original de comparação de SGIUs proposta em [PRIME 90], o sistema Agile.

CRITÉRIOS	Agile	Tiger	SET	Dice	Blox	Grape MMI
SEPARAÇÃO DA INTERFACE DA APLICAÇÃO	Sim	Sim	Não	Não	Não	Não
CONTROLE PELO SGIU OU APLICAÇÃO	SGIU	Apliação	SGIU	Usuário	SGIU	SGIU
COMUNICAÇÃO ENTRE SGIU E APLICAÇÃO	Evento	String	Evento	Mensagem	Estado	"lexemes"
FERRAMENTAS	Duas	Muitas	Poucas	Nenhuma	Muitas	Poucas

TABELA 6.1 - Tabela de Critérios

6.2 Testes

Para testar Agile como um sistema de prototipagem vazia foram criados alguns protótipos, por exemplo o protótipo da interface do Editor e do Gerador de Documentação, os quais antes de terem suas funções desenvolvidas e acopladas às interfaces, funcionavam como protótipos vazios, isto é sem conexão com as funções da aplicação.

O teste de gerenciamento de interação entre o usuário e a aplicação foi feito também com estas ferramentas, já com suas funções acopladas às respectivas interfaces.

6.3 Dificuldades

Uma das dificuldades encontradas no decorrer deste trabalho, foi a ausência do módulo responsável pelo gerenciamento dos dados.

A dificuldade se deu, em virtude de na proposta inicial do trabalho haver um módulo que iria gerenciar os dados, o qual ficaria a cargo de outro membro da equipe de desenvolvimento. Como o desenvolvimento deste módulo não aconteceu, ao chegarmos no desenvolvimento das ferramentas, tivemos que repensar toda a manipulação dos dados, e desenvolver as funções para tratamento dos mesmos.

Estas dificuldades só foram sentidas durante a implementação das ferramentas, porque na fase de desenvolvimento do Gerenciador de Diálogos, considerávamos os dados como já carregados nas estruturas.

Outra dificuldade, foi não se ter adotado a filosofia de orientação a objetos na codificação. Esta dificuldade foi sentida principalmente, na implementação das funções que tratavam do diálogo, pois como este é tratado, de acordo com o estilo de interação, foi necessário implementar para cada estilo, uma função. Caso, tivéssemos utilizado, uma linguagem orientada a objeto, o diálogo seria um objeto, sendo necessário apenas implementar as funções de manipulação de diálogo, sem levar em consideração o estilo, já que as características relevantes para cada estilo já estariam definidas no objeto correspondente.

6.4 Sugestões para trabalhos futuros

Existem trabalhos a curto e longo prazo que podem ser realizados neste sistema. A curto prazo podemos citar:

- o acréscimo de mais recursos nos estilos de interação atualmente já desenvolvidos, como por exemplo, permitir que formulários ultrapassem os limites da janela,

- uma documentação mais sofisticada, ou mais interessante, mostrando por exemplo a árvore de sequenciamento do protótipo de interface,

- desenvolvimento de uma função para realizar teste de consistência no protótipo especificado pelo projetista, antes de sua simulação, como por exemplo, verificar se existe algum diálogo que não foi definido, e o que deve ser feito, quando o usuário final estiver interagindo com ele durante a simulação,

- associar ao Editor de Interfaces, uma função que permita que o projetista, ao terminar de definir a apresentação de um diálogo, possa ver antecipadamente como ficou a sua apresentação,

- fazer testes para verificar até que ponto se pode usar Agile para gerenciar uma aplicação.

A longo prazo sugerimos:

- desenvolver e acoplar à Agile uma ferramenta para validação de interfaces,

- acrescentar recursos para geração e manipulação de

interfaces gráficas,

- criar uma segunda versão, dando uma visão de orientada a objeto.

Este trabalho, como pode ser verificado pela lista de trabalhos futuros, foi a concretização apenas da idéia básica de se ter um sistema que auxiliasse os programadores, na definição das interfaces de suas aplicações. E como tal, parece suprir a lacuna associada à dificuldade de se ter ferramentas que permitam a definição interativa de interfaces, sem a utilização de linguagem de programação, e de recursos para simular a interface projetada, sem a necessidade de desenvolvimento das funções da aplicação.

BIBLIOGRAFIA

[BRANCO 90] Branco S. O. "Especificação de um Sistema de Prototipagem Rápida de Interfaces Usuário-Computador", dissertação de Mestrado, UFPB/COPIN, 1990.

[BRITTS 87] Britts S. "Dialog Management in Interactive Systems: A Comparative Survey", SIGCHI Bulletin, 1987, vol. 18, no. 3, pp. 30-42.

[FLECCHIA 87] Flecchia M. A., e Bergeron R. D. "Specifying Complex Dialogs in ALGAE", ACM, 1987, pp. 229-234.

[GREEN 85] Green M. "The University of Alberta User Interface Management System", Computer Graphics, 19, 3, 1985, pp. 205-213.

[HILL 87] Hill R. D. "Event-Response Systems - A Technique for Specifying Multi-Threaded Dialogues", ACM, 1987, pp. 241-248.

[HUCKLE 81] Huckle B. A., "The Man-Machine Interface: Guidelines for the Design of End-user/System conversation", Savant Research Studies, Savant Institute, Jan 1981.

[KERNIGHAN 78] Kernighan B. W., Ritchie D. M. "The C Programming Language", Prentice-Hall, Inc., 1978.

[LEWIS 89] Lewis T. G., Handloser F., Bose S., e Yang S. "Prototypes form Standard User Interface Management Systems".

Computer, May, 1989, pp. 51-60.

[LOWGREN 88] Lowgren J. "History, State and Future of User Interface Management Systems". SIGCHI Bulletin, 20, 1, 1988, pp. 38-44.

[LUIGI 89] Luigi. "Software Evolution through Rapid Prototyping". Computer, May, 1989, pp. 13-25.

[MANHEIMER 89] Manheimer J. M., Burnett R. C., e Wallers J. A. "A Case Study of User Interface Management System Development and Application". Proceedings of the Conference on Human Factors in Computer Systems, May, 1989, pp. 127-132.

[OLSEN 90] Olsen D. R. "Propositional Production Systems for Dialog Description". Proceedings of the Conference on Human Factors in Computer Systems, April, 1990, pp. 57-63.

[OLSEN 89] Olsen D. R. "A Programming Language Basis for User Interface Management". Proceedings of the Conference on Human Factors in Computer Systems, May, 1989, pp. 171-176.

[PERRY 89] Perry T. S. "Of Mice and Menus: Designing the User-Friendly Interface", IEEE Spectrum, Vol. 26, No. 9, September 1989, pp. 46-51.

[PRIME 90] Prime M. "User Interface Management Systems - A Current Product Review". Computer Graphics, 1990, pp 53-76.

[SANTOS 92] Santos R. "Um Gerenciador de Apresentação para o Sistema Agile de Prototipagem Rápida Usuário-Computador", dissertação de Mestrado, UFFB, 1992.

[SHNEIDERMAN 87] Shneiderman B. "Designing the User Interface: Strategies for Effective Human-Computer Interaction", Addison-Wesley, Inc., 1987.

[SIBERT 86] Sibert J. L., Hurley W. D., e Bleser T. W. "An Object-Oriented User Interface Management System", ACM, vol. 20, no. 4, August, 1986, pp. 259-268.

[WELLNER 89] Wellner P. D. "Statemaster: A UIMS based on Statecharts for Prototyping and Target Implementation", ACM, May, 1989, pp. 177-182.

GUIA DO USUARIO

PROPOSITO

AGILE é um sistema de prototipagem rápida de interfaces, que oferece facilidades para criação, simulação e documentação de interfaces de comunicação entre o usuário e um aplicativo.

Você usa o Agile para:

- . Projetar janelas, mensagens e diálogos que irão compor a interface.
- . Definir o sequenciamento da interface.
- . Simular a operação da interface projetada.
- . Gerar a documentação da interface projetada.

O propósito deste manual é orientar o usuário na criação, simulação e documentação das interfaces.

AUDIENCIA

Este manual é voltado para o projetista de interface, que é o usuário que se utiliza de Agile para definir a interface das aplicações que está desenvolvendo, podendo ser um analista de sistemas, um programador ou um outro profissional qualquer.

ORGANIZAÇÃO

Este manual está dividido nas seguintes seções.

Seção 1. Conceituação

Esta seção apresenta a definição de alguns termos utilizados neste manual, que tem significados próprios para Agile.

Seção 2. Inicializando o Sistema

Esta seção descreve como carregar o sistema Agile.

Seção 3. Prototipando Interfaces

Esta seção descreve como usar Agile para criar e modificar interfaces usando o Editor de Interfaces disponível no sistema.

Seção 4. Simulando Interfaces

Esta seção explica como simular uma interface no ambiente Agile.

Seção 5. Documentando Interfaces

Esta seção informa como utilizar a função de documentar as interfaces que são prototipadas em Agile.

Seção 6. Como ligar a aplicação à interface

Esta seção explica como o usuário deve interligar as funções da aplicação com a interface prototipada.

Seção 7. Mensagens de erro

Esta seção apresenta uma listagem das mensagens de erro, que podem ocorrer durante a prototipagem ou simulação.

INDICE DE CONTEUDOS

Seção 1. Conceituação	05
Seção 2. Inicializando o sistema	08
Seção 3. Prototipando Interfaces	
Layout da Tela do Ambiente de Prototipagem.....	10
Prototipando Janelas	11
Prototipando Diálogos	13
Definindo um sequenciamento.....	13
Menu	17
Apresentação	17
Composição dos Diálogos	20
Sequenciamento	21
Comando	22
Apresentação	22
Composição dos Diálogos.....	23
Sequenciamento	24
Pergunta e Resposta	26
Apresentação	26
Composição dos Diálogos.....	27
Sequenciamento	28
Formulário	29
Apresentação	29
Composição dos Diálogos.....	32
Sequenciamento	33

Prototipando Mensagens	34
Seção 4. Simulando Interfaces	
Inicializando a Simulação	36
Encerrando a Simulação	37
Seção 5. Documentando Interfaces	38
Seção 6. Como ligar a aplicação à interface	40
Seção 7. Mensagens de erro	
Mensagens de erro durante a prototipagem.....	42
Mensagens de erro durante a simulação.....	44

SEÇÃO 1 - Conceituação

Nesta seção definiremos termos usados no decorrer deste manual, e que tem significados próprios no contexto do ambiente Agile.

Evento

Entradas geradas pelo usuário, como por exemplo a opção selecionada de um menu, resposta (string) a uma pergunta, etc.

Na versão atual do sistema Agile, os eventos podem ser gerados apenas através do teclado ou do mouse.

Ações

Para o projetista de interface, uma ação é o efeito resultante de um dado evento. Estão disponíveis ao projetista doze tipos de ação que podem ser usados na definição de um diálogo e que determinam o sequenciamento da interação.

Diálogo

Comunicação entre o usuário e o computador. Em Agile, um diálogo é a menor célula do processo de interação. É composto de informações de apresentação e de informações referentes ao sequenciamento. Ex.: Um menu de cores.

Diálogo Ativo

Diálogo com o qual o usuário está interagindo (gerando eventos), em determinado ponto da interação. Por exemplo, supondo que a aplicação necessite do nome de um arquivo, e seja apresentado ao usuário, um diálogo do tipo menu com nomes de arquivos para que faça a seleção. Enquanto o usuário não selecionar, este diálogo será considerado o diálogo ativo.

Mensagens

Recurso oferecido aos projetistas, para que estes incorporem nos seus protótipos de interface, uma linha de texto para informações de erros e/ou orientação na utilização das interfaces.

Janelas

São áreas da tela, dimensionadas em um modelo de interface, e posteriormente associadas a diálogos e mensagens, como sendo os locais da tela onde estes devem ser apresentados.

Tecla aplicável

Tecla, que quando pressionada, é reconhecida dentro de um contexto da interação, ativa ações.

Estilos de Interação

Nesta versão do sistema, quatro estilos de interação podem

ser tratados, isoladamente ou combinados a critérios do projetista. São eles: menu, comando, pergunta e resposta, e formulário.

SEÇÃO 2 - Inicializando o Sistema

Deve ser digitada no prompt do Sistema Operacional uma das seguintes opções:

1a. opção: AGILE

2a. opção: AGILE ?

3a. opção: AGILE ambiente para apresentação

A 1a. opção deve ser usada quando o usuário desejar rodar sua interface no mesmo ambiente onde a está especificando.

A 2a. opção quando o usuário não souber informar o ambiente onde a interface irá rodar. Ao ser digitada esta opção, todos os ambientes suportados por Agile serão mostrados na tela.

Na 3a. opção, o usuário já informa o ambiente onde a interface irá rodar.

SEÇÃO 3 - Prototipando Interfaces

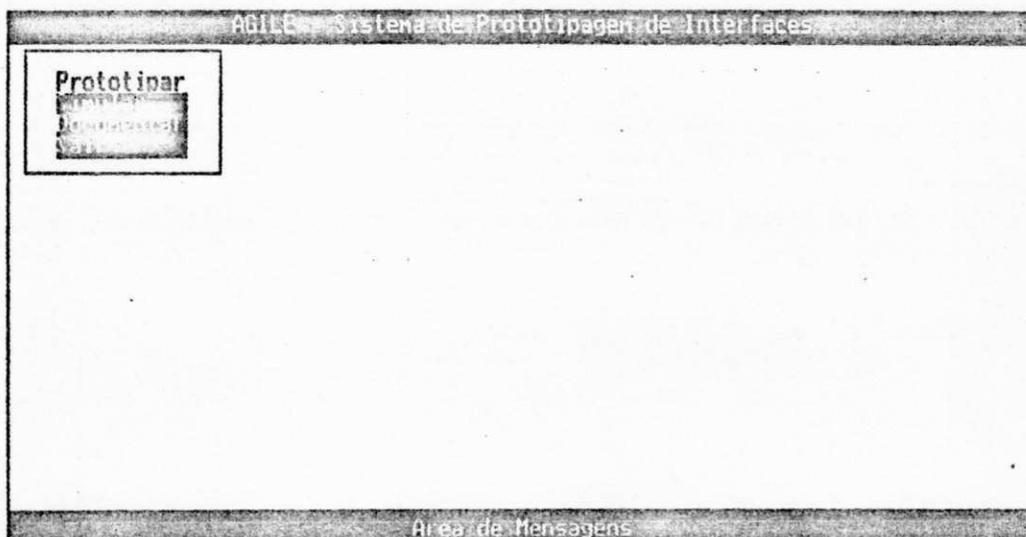
Nesta seção, é apresentado o método de como se editar os protótipos de interface. Estes protótipos seguem o paradigma de interface de Agile, e portanto devem conter informações sobre janelas, diálogos e mensagens.

A prototipagem de uma interface deve seguir a seguinte ordem cronológica:

1. Definir janelas.
2. Definir mensagens.
3. Definir a apresentação dos diálogos.
4. Definir o sequenciamento dos diálogos.

Apesar da interface do editor permitir que o projetista defina os elementos do seu protótipo (janelas, mensagens e diálogos), em qualquer sequência, é necessário seguir uma ordem cronológica para se chegar na definição dos diálogos. Esta sequência se faz necessária devido um diálogo consistir da composição da definição da sua apresentação, com a definição da janela onde ele será apresentado, e por fim com o sequenciamento dado a partir deste diálogo.

Layout da Tela do Ambiente de Prototipagem



A tela para prototipagem é dividida em três janelas:

- Janela de status, que indica o caminho percorrido pelo usuário para chegar no ponto em que ele se encontra.

- Janela de trabalho, onde são apresentados os menus, formulários, e perguntas e respostas, para aquisição dos dados do protótipo.

- Janela para apresentação de mensagens de orientação e/ou de erros.

Prototipando Janela

Para se editar uma janela é necessário:

1. Dar uma identificação a janela, de no máximo cinco caracteres.

Caso a operação seja **alterar ou eliminar** janelas, será apresentado ao usuário, um menu com as identificações de todas as janelas que já foram definidas, para que ele selecione aquela que deseja alterar ou eliminar.

Se a operação for eliminar uma janela, que por sua vez está associada a algum diálogo, será mostrada uma mensagem para o usuário, informando que a janela que ele vai eliminar está sendo referenciada por um diálogo.

2. Determinar a dimensão e a localização da janela na tela. Esta informação é dada através da manipulação de uma "caixa elástica". A movimentação da caixa elástica é feita utilizando-se as setas de controle do cursor.

3. Informar se a janela deve ter ou não borda. Este dado é fornecido através da seleção de um menu, cujos itens são "SIM" e "NÃO".

4. Uma janela pode ter títulos e rodapés. A seguir, fazemos algumas considerações sobre eles, e apresentamos como o usuário deve defini-los.

Título e rodapé correspondem a uma linha de texto, com tamanho igual ou inferior a largura da janela.

As linhas de títulos são posicionadas nas primeiras

linhas da janela, e as linhas de rodapé nas últimas. Caso uma janela tenha mais de um título definido, a determinação de qual título será colocado na primeira, segunda ou n-ésima linha da janela, será feita pela sequência em que os títulos foram definidos. A mesma regra se aplica a rodapés.

Para definir um título ou rodapé é necessário:

1. Informar o texto.

2. Definir os atributos do texto:

. Fonte: A informação de fonte dos caracteres é adquirida através da seleção de um menu, cujos itens correspondem aos possíveis tipos de fontes suportados por Agile.

. Tamanho do caracter: É preenchido pelo projetista com um número inteiro no intervalo entre 0 e 9.

. Cor: É indicada pelo projetista, através da seleção de um menu de cores fornecido por Agile.

3. Padrão de preenchimento das linhas da janela, onde os títulos/rodapés serão apresentadas será indicado pelo projetista de interface, através da seleção de um menu que contém todos os possíveis padrões de preenchimento.

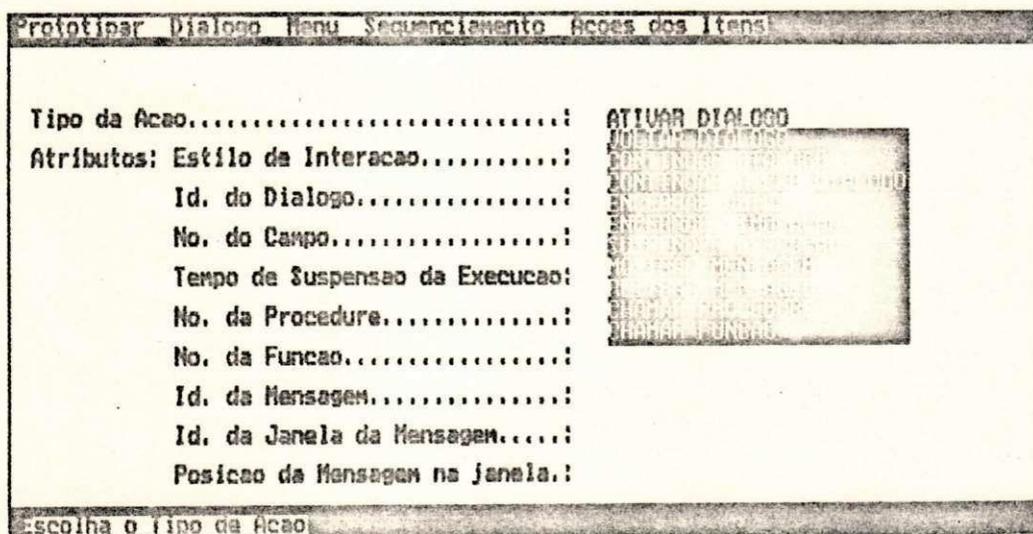
4. Cor de Preenchimento das linhas de título e rodapé é fornecida da mesma forma que padrão de preenchimento.

Prototipando Diálogos

Um diálogo é formado por informações de apresentação e de sequenciamento. Informações de apresentação englobam os dados que definem a forma como o diálogo será mostrado na tela. Informações de sequenciamento indicam as ações que devem ser ativadas de acordo com os eventos gerados pelo usuário.

Definindo o Sequenciamento

Como a definição de ações, é a mesma para qualquer um dos estilos de interação, apresentaremos como o projetista deverá fazer a definição, antes de descermos aos detalhes de cada estilo.



Prototipar Dialogo Menu Sequenciamento Acoes dos Itens	
Tipo da Ação.....:	ATIVAR DIALOGO
Atributos: Estilo da Interação.....:	
Id. do Dialogo.....:	
No. do Campo.....:	
Tempo de Suspensão da Execução:	
No. da Procedure.....:	
No. da Função.....:	
Id. da Mensagem.....:	
Id. da Janela da Mensagem.....:	
Posição da Mensagem na janela.:	

Escolha o tipo da Ação

Para definir ações é necessário:

1. Informar o tipo da ação. Esta informação é dada pela seleção em um menu, onde cada item corresponde a um tipo de ação definida em Agile.

2. Para cada tipo de ação existem atributos. Então ao selecionar um tipo de ação, o usuário deverá informar os

atributos da mesma. A seguir, apresentaremos uma tabela (TABELA 1), onde para cada tipo de ação, colocaremos uma descrição de sua função dentro do sequenciamento, assim como os atributos a serem informados.

<p>Tipo da ação: ATIVAR DIALOGO</p> <p>Descrição...: Torna ativo um novo diálogo.</p> <p>Atributos...: Dados do diálogo a ser ativado</p> <ul style="list-style-type: none">. Estilo de interação. Identificação do diálogo. No. do primeiro campo a receber evento. <p>Obs.: Este atributo só deve ser informado se o estilo for formulário.</p> <p>Caso o estilo seja formulário, e este atributo não tenha sido informado, o 1o. campo a receber evento, será o 1o. campo definido do formulário.</p>
<p>Tipo da ação: VOLTAR DIALOGO</p> <p>Descrição...: Ativa um diálogo que já foi ativado anteriormente.</p> <p>Atributos...: Dados do diálogo</p> <ul style="list-style-type: none">. Estilo de interação. Identificação do diálogo. No. do primeiro campo a receber evento. <p>Obs.: - Este atributo só deve ser informado se o estilo for formulário.</p> <p>- Caso o estilo seja formulário, e este atributo não tenha sido informado, o 1o. campo a receber evento, será o 1o. campo definido do formulário.</p>

TABELA 1.a - Tabela de ações

<p>Tipo da ação: CONTINUAR A LER DIALOGO</p> <p>Descrição...: Continua a receber uma entrada de um diálogo, que por algum evento foi interrompido.</p> <p>Atributos...: Dados do diálogo</p> <ul style="list-style-type: none"> . Estilo de interação . Identificação do diálogo . No. do primeiro campo a receber evento. <p>Obs.: - Este atributo só deve ser informado se o estilo for formulário. - Caso o estilo seja formulário, e este atributo não tenha sido informado, o 1o. campo a receber evento, será o último que tinha recebido evento.</p>
<p>Tipo da ação: CONTINUAR DIALOGO</p> <p>Descrição...: Continua a receber os eventos do diálogo que se encontra ativo no instante da chamada desta ação</p> <p>Atributos...: No. do campo que deve receber eventos, para o estilo de interação formulário.</p> <p>Obs.: Caso esta informação não seja preenchida, o campo a receber eventos será o campo posterior ao último campo que recebeu eventos.</p>
<p>Tipo da ação: MOSTRAR MENSAGEM</p> <p>Descrição...: Apresenta em determinada posição da tela, uma mensagem.</p> <p>Atributos...: . Identificação da mensagem . Identificação da janela, onde a mensagem deve ser apresentada. . Posição da mensagem dentro da janela.</p>
<p>Tipo da ação: RETIRAR MENSAGEM</p> <p>Descrição...: Retira da tela uma mensagem.</p> <p>Atributos...: Identificação da mensagem</p>

TABELA 1.b - Tabela de ações

<p>Tipo da ação: ENCERRAR AGILE</p> <p>Descrição...: Encerra a interação com o ambiente e volta ao Sistema Operacional.</p> <p>Atributos...: Desnecessários</p>
<p>Tipo da ação: ENCERRAR SIMULAÇÃO</p> <p>Descrição...: Encerra a simulação que está sendo realizada, retornando o controle para Agile.</p> <p>Atributos...: Desnecessários</p>
<p>Tipo da ação: SUSPENDER EXECUÇÃO</p> <p>Descrição...: Suspende a execução, por um intervalo de tempo.</p> <p>Atributos...: Tempo em segundos durante o qual a execução deve ser suspensa.</p>
<p>Tipo da ação: ENCERRAR DIALOGO</p> <p>Descrição...: Encerra o diálogo que se encontra ativo.</p> <p>Atributos...: Desnecessários</p>
<p>Tipo da ação: CHAMAR PROCEDURE</p> <p>Descrição...: Chama rotinas da aplicação, que não retornam valor.</p> <p>Atributos...: No. que identifique a rotina (índice do vet_proc).</p> <p>Obs.: Para maiores detalhes ver seção 6.</p>
<p>Tipo da ação: CHAMAR FUNÇÃO</p> <p>Descrição...: Chama rotinas da aplicação que retornam um valor e cujo sequenciamento após a chamada da função, é dependente do valor retornado.</p> <p>Atributos...: . No. que identifica a função (índice do vet_func) . Lista de valores retornados, associados à ações.</p> <p>Obs.: Para maiores detalhes ver seção 6.</p>

TABELA 1.c - Tabela de ações

A seguir, para cada estilo de interação será apresentada a sequência de passos necessárias para definir: a apresentação, composição e o sequenciamento de um diálogo.

Prototipando Menu

Apresentação

Prototipar Dialogo Menu Apresentacao Layout

Id. da Apresentacao do Menu..: []

Padrao da Preenchimento.....:

Cor da Preenchimento.....:

Wrap.....:

Direcao.....:

Selecao.....:

Terminador - Tecla.....:

Status.....:

Digite identificacao da Apresentacao do Menu, com no maximo 5 caracteres!

1. Definir a identificação da apresentação do menu, com no máximo cinco caracteres.

No caso da operação, ser alteração ou exclusão, será apresentado um menu com as identificações da apresentação dos menus já especificados, para que o projetista escolha o menu que ele deseja alterar ou excluir .

Caso o usuário selecione a apresentação de um menu que está sendo usado na composição dos diálogos de estilo menu, para exclusão, será apresentada ao usuário uma mensagem, informando que o menu que ele está querendo eliminar está sendo usado, e que ele não poderá eliminá-lo.

2. Indicar o padrão de preenchimento do menu, através da seleção em um menu que contém todos os possíveis padrões de preenchimento.

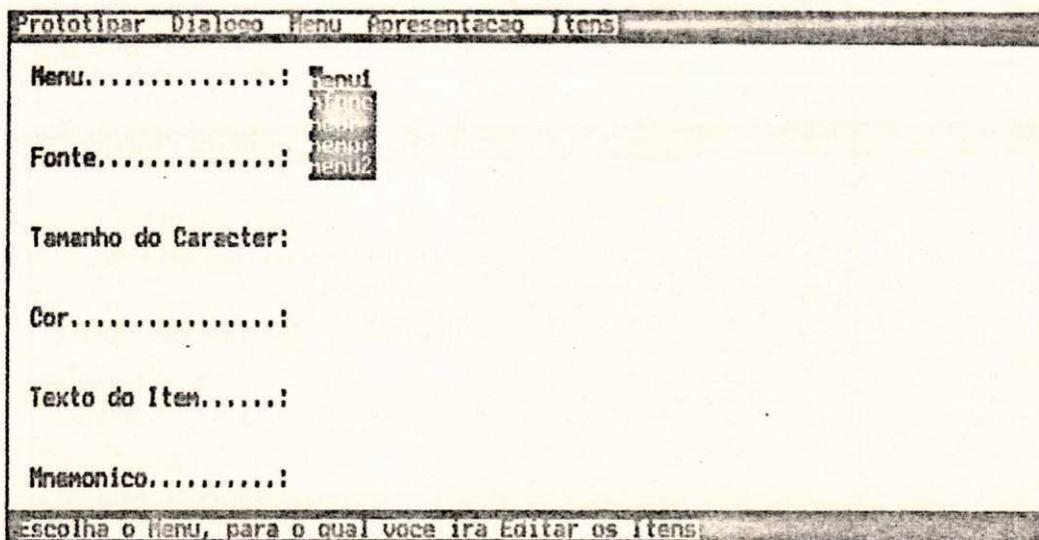
3. Informar a cor de preenchimento do menu, através da seleção em um menu de cores.

4. Definir se o menu deve ter "wrap" ou não. O projetista define esta opção através da seleção em um menu, cujos itens são "sim" e "não".

5. Definir se a seleção no menu, será através das setas (BARRA), ou da digitação de um mnemônico (STRING). Esta informação também é dada através da seleção em um menu, cujos itens representam os dois tipos de seleção.

6. Determinar a tecla que deverá ser pressionada para confirmar a seleção de um item do menu. Este campo deve ser preenchido com o valor do status e da tecla em hexadecimal.

7. Definir os itens do menu, especificando para cada um deles, as seguintes informações:



1. Texto do item.

2. Definir os atributos do texto:

. Fonte do carácter: A informação de fonte é adquirida através da seleção em um menu, cujos itens correspondem aos possíveis tipos de fontes suportados por Agile.

. Tamanho do carácter: é preenchido pelo projetista com um número inteiro no intervalo entre 0 e 9.

. Cor: é indicada pelo projetista, através da seleção em um menu de cores fornecido por Agile.

3. No caso do tipo de seleção no menu ser através da digitação de mnemônicos, o usuário deve informar o mnemônico correspondente ao item.

A ordem de apresentação dos itens é determinada pela ordem de definição dos mesmos, ou seja, o primeiro item definido será o primeiro item do menu.

3. Escolher através de um menu, a identificação da janela, onde este menu será apresentado.

4. Definir a localização do menu dentro da janela. Para fazer esta definição, será colocado para o usuário, a janela onde o menu será apresentado, e dentro desta janela, uma caixa correspondendo às dimensões do menu, de forma que através da movimentação das setas, o usuário faça o posicionamento do menu.

Sequenciamento

Para definir o sequenciamento de um menu é necessário:

1. Definir para cada item do menu uma lista de ações.
2. Definir uma lista de ações para o caso de eventos inválidos.
3. Definir as teclas aplicáveis durante a interação com este menu.

Para cada tecla a ser reconhecida durante o sequenciamento do menu, devem ser dadas as seguintes informações:

- i. Informar o valor da tecla e do status da tecla em hexadecimal.
- ii. Definir uma lista de ações que devem ser ativadas quando da digitação desta tecla.

Prototipando Comando

Apresentação

A apresentação de um diálogo de estilo comando, corresponde a definição do "prompt" para a digitação dos comandos, e da tecla que deve ser reconhecida como o terminador do comando.

Para especificar a apresentação de um diálogo de estilo comando, o usuário deve informar:

Prototipar Dialogo Comando Apresentacao Layout

Id. da Apresentacao do Comando: []

Terminador - Tecla.....:

- Status.....:

Prompt.....:

Digite identificacao da Apresentacao, com no maximo 5 caracteres

1. Identificação da apresentação do comando, com no máximo cinco caracteres.

No caso da operação, ser alteração ou exclusão, será apresentado um menu com as identificações das apresentações de comando já definidas, para que o projetista escolha aquela que ele deseja alterar ou excluir .

Caso o usuário selecione para exclusão, uma apresentação que está sendo usada na composição de algum diálogo de estilo comando, será apresentada ao usuário uma mensagem, informando que

a apresentação que ele pretende eliminar está sendo usada, e que ele não vai poder eliminá-la.

2. Informar a tecla que deverá ser pressionada para confirmar a digitação de um comando. Este dado deve ser preenchido com o valor do status e da tecla em hexadecimal.

3. Definir o "prompt" para entrada dos comandos. Ex.: "a:", "a>".

Composição do Diálogo

Para definir a composição dos diálogos de estilo comando é necessário:

The image shows a screenshot of a graphical user interface window titled "Prototipar Dialogo Comando Composicao do Dialogo". The window contains four labeled input fields:

- Id. do Dialogo.....:
- Id. da Apresentacao do Dialogo:
- Janela.....:
- Posicao na Janela.....:

At the bottom of the window, there is a status bar that reads: "Digite Identificacao do Dialogo, com no maximo 5 caracteres".

1. Informar a identificação do diálogo, com no máximo cinco caracteres.

2. Informar a identificação da apresentação do comando, através da seleção em um menu, cujos itens relacionam todas as

apresentações de estilo comando já definidas.

3. Informar a janela, onde o diálogo será apresentado. Esta informação é dada pela seleção em um menu, com a identificação de todas as janelas que já foram definidas.

4. Definir a posição do diálogo dentro da janela. O posicionamento é feito pela movimentação de uma caixa, que corresponde a apresentação do comando, dentro da janela.

Sequenciamento

Como para diálogo de estilo comando, existe uma lista de comandos, que podem ser digitados pelo usuário da interface. Apresentamos a seguir, o que deve ser informado pelo projetista para cada um dos comandos que compõem esta lista.

1. A sintaxe do comando, que deve conter:

i. Nome do comando, que deve ser formado por um número qualquer de caracteres.

ii. Separador de parâmetros, que deve ser um caracter, como por exemplo uma vírgula (,).

iii. Delimitador inicial e final dos parâmetros, que devem consistir de um caracter.

2. Lista de parâmetros, onde para cada um deve ser definido:

i. Um flag, informando se o parâmetro deve estar presente quando da digitação do comando ou não. Esta informação é dada através da seleção em um menu, cujos itens são "SIM" e

"NAO".

ii. Um string de caracteres, correspondendo ao valor do parâmetro.

iii. Uma lista de ações, que devem ser ativadas, caso haja erro na digitação do parâmetro.

3. Lista de ações, a serem ativadas, quando este comando for gerado durante a interação.

4. Lista de ações, para o caso de serem encontrados erros no comando gerado.

5. Teclas aplicáveis durante a edição deste comando.

Para cada tecla a ser reconhecida durante a edição do comando, devem ser dadas as seguintes informações:

i. O valor da tecla e do status da tecla em hexadecimal.

ii. Uma lista das ações que devem ser ativadas quando da digitação desta tecla.

Além destas definições para cada comando da lista, é necessário definir também:

1. A lista de ações, a serem ativadas caso o evento gerado não corresponda a nenhum dos comandos desta lista.

2. As teclas aplicáveis durante a interação, independentemente do comando que esteja sendo digitado. Para cada tecla a ser reconhecida devem ser dadas as seguintes informações:

i. Informar o valor da tecla e do status da tecla em hexadecimal.

a apresentação de diálogos já referenciadas.

2. Determinar o texto da pergunta.

3. Informar a tecla que finaliza a digitação de uma resposta. Devem ser fornecidos: o valor do status e o valor da tecla em hexadecimal.

Composição do Diálogo

Para compor os diálogos de estilo "pergunta e resposta" é necessário:

Prototipar Dialogo P/R Composicao do Dialogo

Id. do Dialogo.....: pes1

Id. da Apresentacao do Dialogo: Tri

Janela.....: [icon]

Posicao na Janela.....:

Escolha a Apresentacao do seu Dialogo

1. Dar uma identificação, de no máximo cinco caracteres, para o diálogo.

Caso o usuário deseje alterar ou excluir, os dados da composição de um diálogo, será apresentado um menu com a identificação de todos os diálogos definidos, para que ele selecione aquela que deseja alterar ou excluir.

No caso do usuário escolher para exclusão, um diálogo que está definido como atributo de alguma ação, será enviada uma mensagem para o usuário, informando que ele não pode eliminá-lo.

2. Informar a identificação da apresentação da "pergunta e resposta". Esta informação é dada, através da seleção em um menu, cujos itens representam as identificações das apresentações de "pergunta e resposta" já definidas.

3. Selecionar a janela, onde o diálogo deve ser apresentado. Esta seleção é feita através de um menu, cujos itens correspondem a identificação de todas as janelas já definidas.

4. Definir a posição da "pergunta e resposta" dentro da janela. Para esta definição, será colocada para o usuário, a janela onde a pergunta e resposta será apresentada, e dentro desta janela, uma caixa correspondendo a pergunta, de forma que através da movimentação das setas, o usuário faça o posicionamento da "pergunta e resposta".

Sequenciamento

1. Definir a lista de respostas que estão associadas à pergunta.

Para cada resposta é necessário:

i. Definir o texto da resposta.

ii. Definir a lista de ações, que devem ser disparadas

quando for gerado um evento correspondente a resposta.

2. Definir uma lista de ações, a serem ativadas, caso sejam gerados eventos que não correspondam a nenhuma das respostas definidas.

3. Definir a lista de teclas aplicáveis, a serem reconhecidas, durante a interação com este diálogo. Para cada tecla a ser reconhecida devem ser fornecidas as seguintes informações:

i. Informar o valor da tecla e do status da tecla em hexadecimal.

ii. Definir uma lista de ações que devem ser ativadas quando da digitação desta tecla.

Prototipando Formulário

Apresentação

1. Definir um nome que identifique a apresentação do formulário. Este nome não deve ultrapassar cinco caracteres.

No caso da operação, ser **alteração** ou **exclusão**, será apresentado um menu com as identificações das apresentações de formulário, que já foram definidas, para que o usuário possa escolher aquela que ele deseja alterar ou excluir.

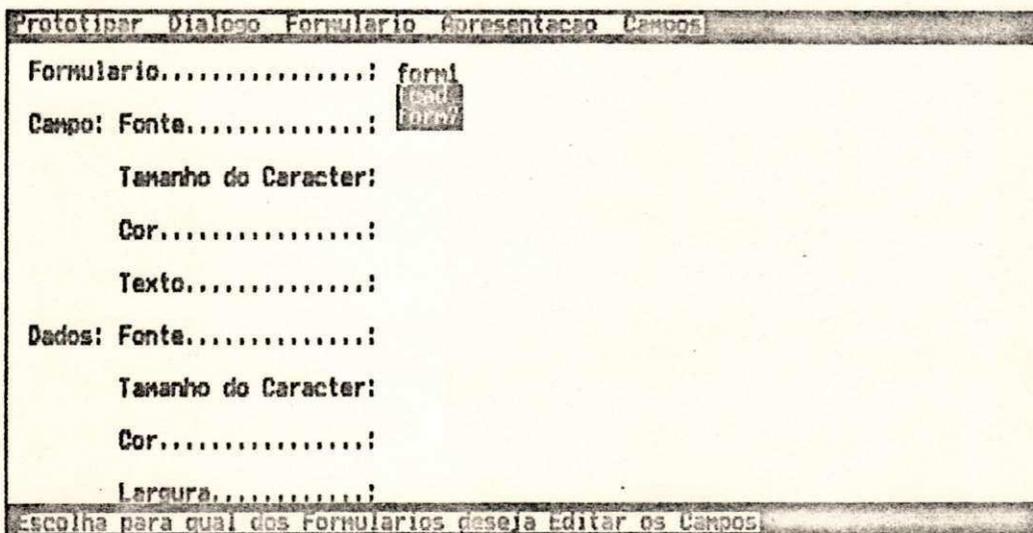
Caso o usuário selecione para exclusão, uma apresentação que já foi referenciada na composição de diálogos de estilo formulário será apresentada uma mensagem, informando que ele não

pode excluir esta apresentação.

2. Definir a tecla, que indicará a finalização da digitação de um campo. Deve ser informado o valor do status e da tecla em hexadecimal.

3. Definir a lista de campos que irão compor este formulário.

Para cada campo deve ser definido:



i. Texto do campo, que deve ser inferior a largura da janela, onde este formulário será apresentado.

ii. Definir os atributos do texto:

• Fonte: A informação de fonte é adquirida através da seleção em um menu, cujos itens correspondem aos possíveis tipos de fontes suportados por Agile.

• Tamanho do caracter: é preenchido pelo projetista com um número inteiro no intervalo entre 0 e 9.

. Cor: é indicada pelo projetista, através da seleção em um menu de cores.

iii. Definir os atributos dos dados, que irão ser preenchidos pelo usuário final, quando da interação com o formulário.

. Fonte: A informação de fonte é adquirida através da seleção em um menu, cujos itens correspondem aos possíveis tipos de fontes suportados por Agile.

. Tamanho do caracter: é preenchido pelo projetista com um número inteiro no intervalo entre 0 e 9.

. Cor: é indicada pelo projetista, através da seleção de um menu de cores fornecido por Agile.

iv. Definir a largura máxima dos dados do campo.

Composição do Diálogo

Para compor os diálogos de estilo formulário é necessário:

Prototipar Dialogo Formulário Composicao do Dialogo
Id. do Dialogo.....:
Id. da Apresentacao do Dialogo:
Janela.....:
Posicao na Janela.....:

Digite Identificacao do Dialogo, com no maximo 5 caracteres

1. Dar uma identificação, de no máximo cinco caracteres, para o diálogo.

Caso o usuário deseje alterar ou excluir, um diálogo, será apresentado um menu com a identificação de todos os diálogos definidos, para que ele selecione o que deseja alterar ou excluir.

No caso do usuário escolher para exclusão, um diálogo que está definido como atributo de alguma ação, será enviada uma mensagem, informando que ele não pode excluir-lo.

2. Informar a identificação da apresentação do formulário. Esta informação é fornecida, através da seleção em um menu, cujos itens representam as identificações das apresentações de formulário que já foram previamente definidas.

3. Selecionar a janela, onde o formulário, deve ser apresentado. Esta seleção é feita através de um menu, cujos itens correspondem a identificação de todas as janelas já definidas.

4. Para cada campo do formulário, definir a localização deste dentro da janela. Para fazer esta definição, será colocada para o usuário, a janela onde o formulário será apresentado, e dentro desta janela, uma caixa correspondendo ao campo, de forma que através da movimentação das setas, o usuário faça o posicionamento do campo.

Sequenciamento

1. Definir o sequenciamento a partir de cada campo do formulário. Para cada campo do formulário devem ser definidos:

i. O valor do campo.

ii. Um flag informando se as ações decorrentes do preenchimento do campo devem ser ativadas sem a necessidade de geração de evento para ele, ou se só serão disparadas quando for gerado um evento para o campo. Este dado é fornecido, através da seleção em um menu, cujos itens são "Sim" e "Nao".

iii. Lista de ações, a serem ativadas a partir deste campo do formulário.

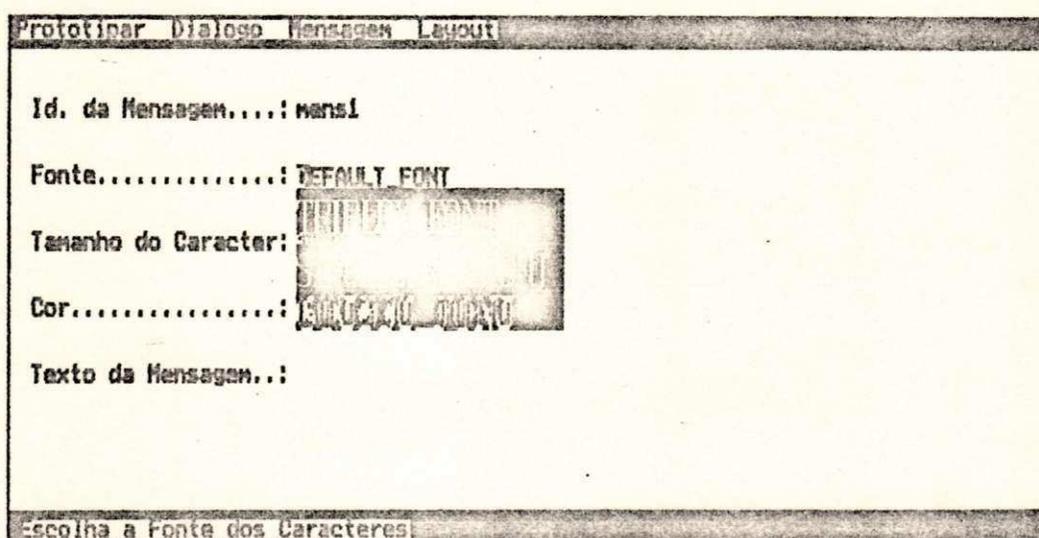
iv. Definir as teclas aplicáveis, a serem reconhecidas durante a digitação do campo.

v. Definir a lista de ações, a serem ativadas caso o evento gerado não corresponda ao valor esperado para o campo.

2. Definir as teclas aplicáveis, que devem ser reconhecidas durante a interação com o formulário.

3. Definir a lista de ações, a serem ativadas no término do preenchimento do formulário.

Prototipando Mensagens



1. Definir um nome que identifique a mensagem. Este nome deve ter no máximo cinco caracteres.

Caso se deseje alterar ou excluir uma mensagem, será apresentado um menu, cujos itens representam a identificação de todas as mensagens que o usuário já definiu.

No caso do usuário selecionar, para exclusão, uma mensagem que já está sendo referenciada, será apresentada uma

mensagem informando que ele não pode excluí-la.

2. Texto da mensagem.

3. Definir os atributos do texto:

i. Fonte: A informação de fonte é adquirida através da seleção em um menu, cujos itens correspondem aos possíveis tipos de fontes de caracter suportados por Agile.

ii. Tamanho do Caracter: é preenchido pelo projetista com um número inteiro no intervalo entre 0 e 9.

iii. Cor: é indicada pelo projetista, através da seleção em um menu de cores fornecido por Agile.

SEÇÃO 4 - Simulando Interfaces

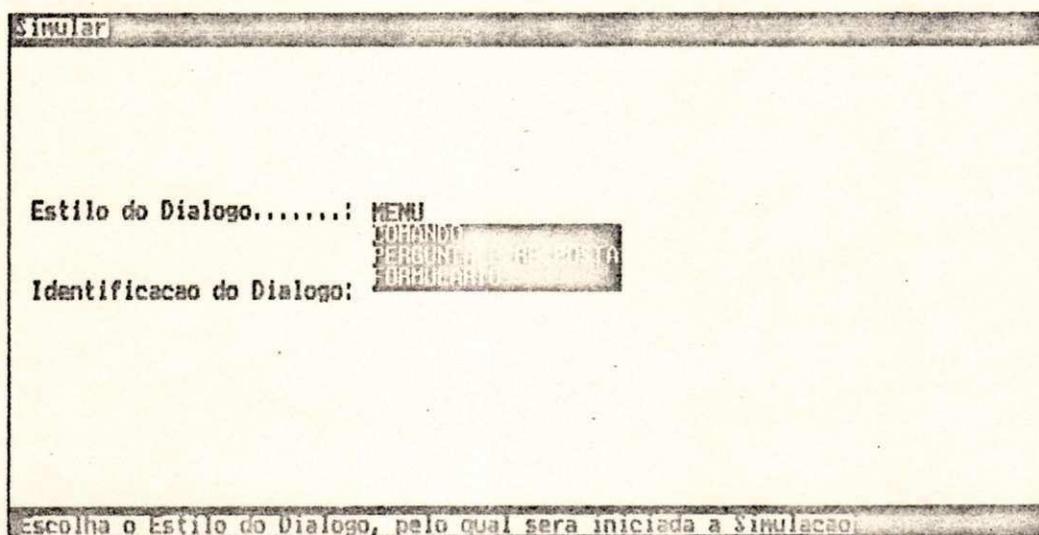
é na simulação que se processa a gerência da interação do usuário com uma aplicação qualquer. Este gerenciamento é feito a nível de prototipagem vazia, isto é, o sistema gerencia a interação entre o usuário e a interface da aplicação, sem acoplar as funções da aplicação, permitindo desta forma que o usuário final possa interagir com o protótipo da interface, de forma a validá-lo.

Inicializando a simulação

1. Informar o nome do arquivo onde se encontram os dados do protótipo que deve ser simulado.

Se o arquivo existir:

é solicitada ao usuário, a identificação do diálogo, por onde deve ser iniciada a simulação, isto é, o sistema permite a simulação tanto do todo, quanto trechos específicos do protótipo, dando assim mais flexibilidade ao projetista.



O usuário repassa esta informação ao sistema da seguinte forma:

1. Informa o estilo de interação do diálogo, a partir da seleção em um menu.

2. Informa a identificação do diálogo, através da seleção em um menu, cujos itens apresentam a identificação de todos os diálogos do estilo de interação informado no passo anterior.

Se o arquivo não existir:

Será apresentada uma mensagem, informando a ausência do arquivo.

Encerrando a Simulação

O término da simulação ocorre em um dos seguintes casos:

1. Erro detectado, durante o sequenciamento do protótipo. Neste caso é emitida uma mensagem de erro, e o controle retorna para o ambiente Agile.

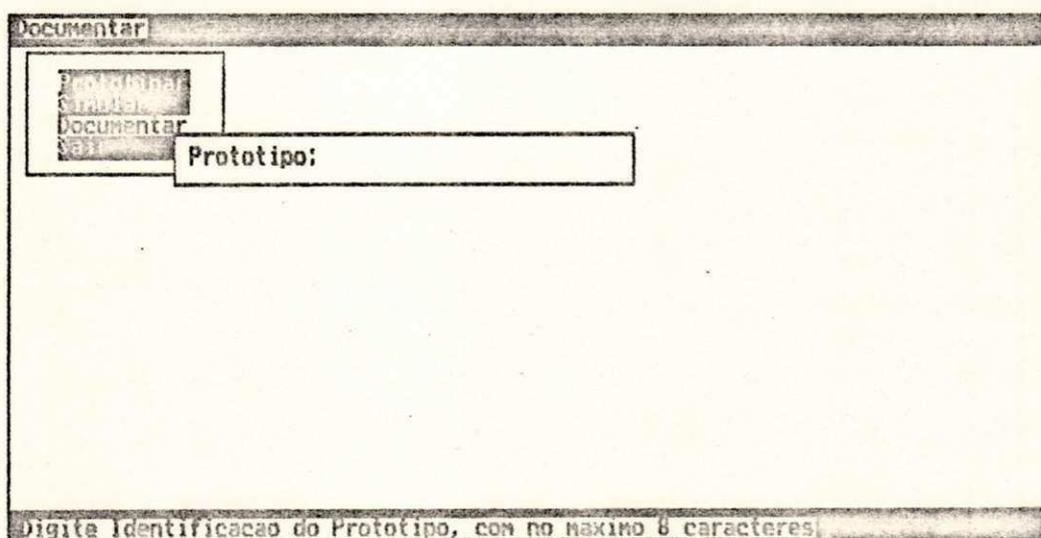
2. Chegada a um determinado ponto da interação com o protótipo, onde a ação é retornar para o ambiente Agile, ou voltar ao Sistema Operacional.

SEÇÃO 5 - Documentando interfaces

Este recurso do sistema fornece documentos ao projetista, com a especificação dos seus protótipos.

A partir do arquivo que contém a especificação do protótipo da interface, é gerado um arquivo documento, descrevendo de forma detalhada todas as informações que compõem o paradigma de uma interface (janelas, mensagens e diálogos). Este documento tem como objetivo principal, fornecer ao projetista uma referência para que ele possa se guiar, durante a implementação da interface especificada.

Para gerar este documento é solicitado ao projetista o nome do arquivo onde se encontra a definição do protótipo, do qual ele deseja a documentação.



Se o arquivo existir, será gerado um arquivo com o mesmo nome do protótipo, e com a extensão ".DOC".

Atualmente, para edição e impressão deste arquivo, é necessário sair do ambiente Agile, e utilizar outros ambientes/aplicativos.

SEÇÃO 6 - Como ligar a aplicação à interface

Caso o projetista deseje que a simulação de seu protótipo não se limite aos recursos de interface, mas integre funções da aplicação, ele deve seguir os seguintes passos:

- . Definir um protótipo de interface, definindo durante o sequenciamento, a chamada às funções e procedures da aplicação.
- . Acessar a estrutura "Filha de diálogos", já que é nessa estrutura onde são guardados os eventos gerados pelo usuário.
- . Definir as funções e procedures.

Para que o projetista defina no sequenciamento do seu protótipo, a chamada às funções e procedures da sua aplicação, foram desenvolvidas duas ações para funcionar como elo de comunicação entre a interface e a aplicação. Estas ações são:

- CHAMAR PROCEDURE -> Permite que sejam chamadas as rotinas da aplicação que não retornam valor.

- CHAMAR FUNÇÃO -> Permite chamar as rotinas da aplicação que retornam valor; do qual depende o sequenciamento dado, após a chamada da função.

Para se utilizar estas ações, é necessário:

1. Editar o arquivo "rotaplic.c", e inserir na definição de "vet_proc" e "vet_func", os procedures e funções da aplicação que devem ser chamados como ações de algum evento gerado. Um exemplo da definição destes vetores é apresentado a seguir:

```
void vet_proc
```

```
    proc1,
```

```
    proc2
```

```
void vet_func
```

```
    func1,
```

```
    func2
```

2. Informar como atributos destas ações, o índice do vetor de procedure (vet_proc) ou vetor de função (vet_func) correspondente ao procedure ou função a ser chamado.

3. Linkar com o código de Agile as funções e procedures da aplicação.

SEÇÃO 7 - Mensagens de Erro

Mensagens de Erro durante a Prototipagem

JANELA JA DEFINIDA

Causa: Tentar definir uma janela que já se encontra definida no protótipo.

NAO EXISTEM JANELAS DEFINIDAS

Causa: Tentar alterar ou excluir janelas, que não foram previamente definidas, ou tentar compor um diálogo, com janelas ainda não definidas.

EXCLUSÃO DE JANELA NAO PERMITIDA

Causa: Tentar excluir uma janela, que já foi referenciada na composição de algum diálogo.

APRESENTAÇÃO JA DEFINIDA

Causa: Tentar definir uma apresentação que já se encontra definida.

NAO EXISTEM APRESENTAÇÕES DEFINIDAS

Causa: Tentar alterar ou excluir apresentações, sem haver definido, ou tentar compor um diálogo, sem ter apresentação previamente definida.

EXCLUSÃO DE APRESENTAÇÃO NAO PERMITIDA

Causa: Tentar excluir uma apresentação, que já foi referenciada na composição de algum diálogo.

COMPOSIÇÃO DO DIALOGO JA REALIZADA

Causa: Tentar compor um diálogo, que já existe.

NAO EXISTEM DIALOGOS DEFINIDOS

Causa: Tentar alterar ou excluir diálogos, sem havê-los definido, ou tentar definir o sequenciamento de diálogos, sem haver diálogo previamente definido.

EXCLUSÃO DE DIALOGO NAO PERMITIDA

Causa: Tentar excluir um diálogo, que está sendo referenciado na definição de sequenciamento de algum outro diálogo.

MENSAGEM JA DEFINIDA

Causa: Tentar definir uma mensagem, que já se encontra definida.

NAO EXISTEM MENSAGENS DEFINIDAS

Causa: Tentar alterar ou excluir mensagens, sem haver mensagens definidas.

EXCLUSÃO DE MENSAGEM NAO PERMITIDA

Causa: Tentar excluir uma mensagem, que já foi usada como atributo da ação MOSTRAR MENSAGEM, na definição de sequenciamento de algum diálogo.

Mensagens de Erro durante a Simulação

PILHA DE DIALOGOS VAZIA

Causa: Este erro ocorre, quando são ativadas ações para retornar a diálogos que deveriam ter sido ativados anteriormente, e não o foram.

MENSAGEM NAO FOI DEFINIDA

Causa: Este erro ocorre quando é ativada a ação de mostrar uma mensagem, a qual não foi definida.

APRESENTACAO DO DIALOGO NAO FOI DEFINIDA

Causa: Ativação de um diálogo, cuja apresentação não foi definida.

SEQUENCIAMENTO DO DIALOGO NAO FOI DEFINIDO

Causa: O diálogo a ser ativado não está com as informações de sequenciamento definidas.

NAO FORAM DEFINIDAS AÇÕES PARA ESTE RETORNO DA FUNÇÃO

Causa: Não haver ações definidas para o valor que a função retornou.