

UNIVERSIDADE DE SÃO PAULO

ASiAModV - Módulo Visual para o ASiA
Descrição de Implementação

WALTER GOMES PEDROSO JUNIOR
MARCOS JOSÉ SANTANA
NIVALDI CALÔNEGO JUNIOR
REGINA HELENA C. SANTANA

Nº 23

NOTAS DIDÁTICAS



Instituto de Ciências Matemáticas de São Carlos

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2577

ASiAModV - Módulo Visual para o ASiA
Descrição de Implementação

WALTER GOMES PEDROSO JUNIOR
MARCOS JOSÉ SANTANA
NIVALDI CALÓNEGO JUNIOR
REGINA HELENA C. SANTANA

Nº 23

NOTAS DIDÁTICAS DO ICMSC

São Carlos
Abr./1996

ASiAModV - Módulo Visual para o ASiA

Descrição de Implementação

**Walter Gomes Pedroso Junior
Marcos José Santana
Nivaldi Calônego Junior
Regina Helena C. Santana**

**Grupo de Sistemas Distribuídos e Programação Concorrente
Departamento de Ciências de Computação e Estatística
Instituto de Ciências Matemáticas de São Carlos
Universidade de São Paulo**

**São Carlos
1996**

ASiAModV - Módulo visual para o ASiA

Descrição de Implementação

**Walter Gomes Pedroso Junior
Marcos José Santana
Nivaldi Calonego Junior
Regina Helena C. Santana**

**Grupo de Sistemas Distribuídos e Programação Concorrente
Departamento de Ciências de Computação e Estatística
Instituto de Ciências Matemáticas de São Carlos
Universidade de São Paulo**

**São Carlos
1995**

Agradecimentos ao CNPq pelo apoio ao desenvolvimento deste trabalho.

Sumário

1. Introdução	02
2. Dicionário de Classes	04
3. Herança de Classes	06
4. Descrição de Classes	08
4.1 Classe TASiA	08
4.2 Classe TBaseWindow	10
4.3 Classe TBMEdit	11
4.4 Classe TConfDlg	11
4.5 Classe TDspMsg	12
4.6 Classe TGdiWindow	12
4.7 Classe TGrafBar	13
4.8 Classe TGrafCur	13
4.9 Classe TGrafFacI	14
4.10 Classe TGrafInt	14
4.11 Classe TGrafLine	15
4.12 Classe TGrafOpt	15
4.13 Classe TGrafParam	16
4.14 Classe TGrafPointf	16
4.15 Classe TGrafSel	17
4.16 Classe TGrafObsT	17
4.17 Classe TParamDlg	18
4.18 Classe TRptDlg	19
4.19 Classe TTblDlg	19
4.20 Classe TTblLmt	19
4.21 Classe TQtTblSel	20
5. Dicionário de Eventos	20
6. Relacionamento de Classes	21
7. Descrições dos Módulos e Principais Rotinas	27
7.1 Módulo asiaV.cpp	27
7.2 Módulo asiaB.cpp	30
7.3 Módulo asiaP.cpp	35
7.4 Módulo asiaT.cpp	36
7.5 Módulo asiaconf.h	39
8. Referências Bibliográficas	40

1. Introdução

Desenvolver programas de simulação é uma tarefa difícil que requer conhecimentos em relação ao problema em estudo, às técnicas de modelagem e análise. Essas dificuldades podem ser minimizadas quando ferramentas automáticas para modelagem e programação são utilizadas. Com esse intuito foi desenvolvido no ICMSC-USP (pelo grupo de Sistemas Distribuídos e Programação Concorrente) o ASiA (Ambiente de Simulação Automático). Esse ambiente permite que usuários com pouco conhecimento de programação especifiquem modelos de simulação e também que profissionais com experiência tenham o tempo de programação reduzido.

O ASiA foi desenvolvido em três módulos:

- Editor Gráfico: responsável pela captura do modelo em estudo;
- Gerador de aplicações: responsável pela geração do código fonte do programa que representa o modelo;
- Módulo Visual: responsável pela apresentação dos resultados do modelo simulado.

O Módulo Visual é parte integrante do programa gerado pelo Gerador, ou seja, no programa fonte são incluídas chamadas a funções que tratam da análise das informações geradas pelo modelo e valores de parâmetros em tempo de simulação. Este módulo permite o rastreamento de "tokens" e eventos ocorridos no modelo, a verificação dos valores dos parâmetros do modelo, a geração de gráficos dos parâmetros e "facilities" e a alteração dos valores dos parâmetros, dentre outras funções. Quando um programa de simulação (que usa o módulo visual) entra em execução, uma janela windows é aberta apresentando todas as opções do módulo visual. Nesse momento a simulação é iniciada e só é interrompida se a condição de parada for atingida ou se o usuário selecionar a parada do programa.

A utilização do módulo visual não é exclusividade do Gerador de aplicações do ASiA. Para utilizá-lo separadamente basta que o usuário conheça o compilador e programação orientada a objetos e que siga a estrutura definida pelo módulo visual (maiores detalhes no manual de usuário [PED]).

O módulo visual foi desenvolvido segundo o paradigma de orientação a objetos. Esse módulo é uma composição e herança de classes existentes no Object Windows Library (OWL), conjunto de classes que acompanha o compilador Borland C++ 4.0. O módulo foi implementado usando-se o compilador Borland C++ versão 4.02 em um micro computador compatível com o IBM-PC 386 equipado com 8Mb de memória RAM, disco rígido de 500Mb, monitor VGA colorido e mouse.

Este documento tem por finalidade apresentar os aspectos de implementação do módulo visual. Não será discutido o paradigma de orientação a objetos como também não serão tratados a implementação e configuração do EdGraf [SPOa] e do Gerador de aplicações [SPOb]. Informações de uso do módulo visual são encontradas no manual do usuário. Detalhes sobre programação Windows pode ser obtido em bibliografia complementar.

2. Dicionário de Dados: Classes

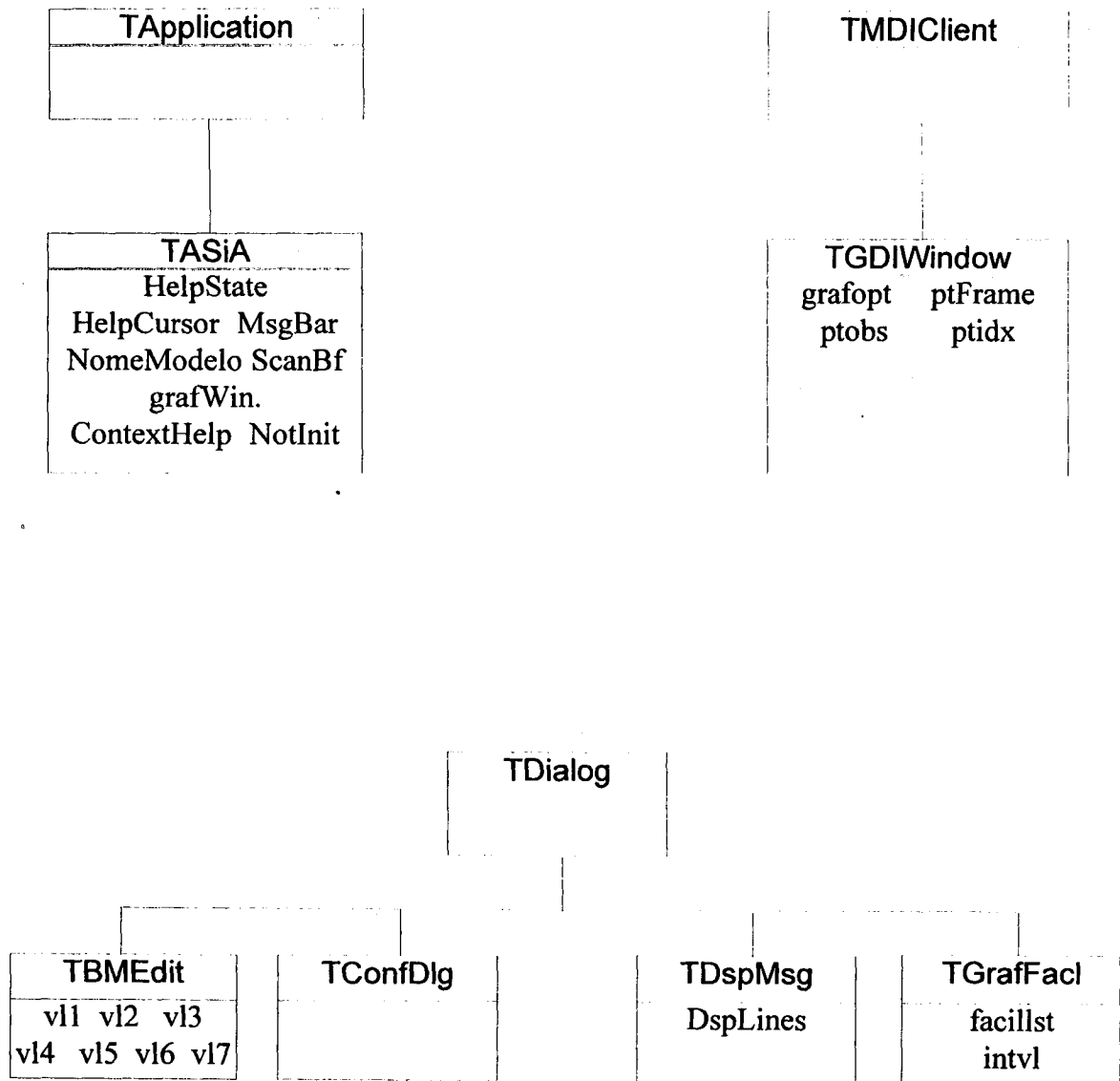
Um dicionário de dados é um repositório central para definição de termos e conceitos. Tem sua importância na facilidade fornecida para checagem interna dos modelos com relação a consistência e completude. As definições são facilmente encontradas pelo nome da entrada. A notação usada nesse relatório foi a notação tabular.

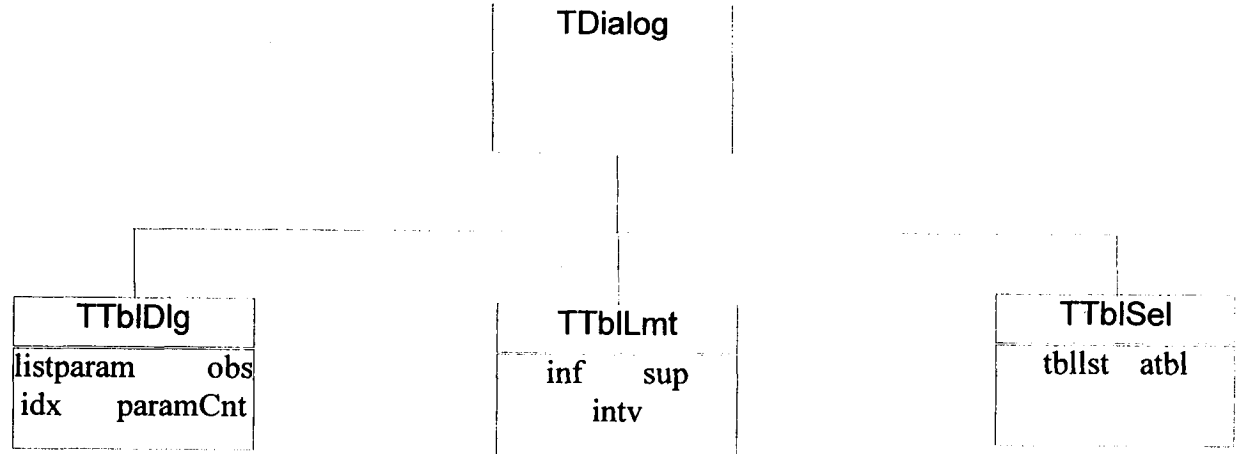
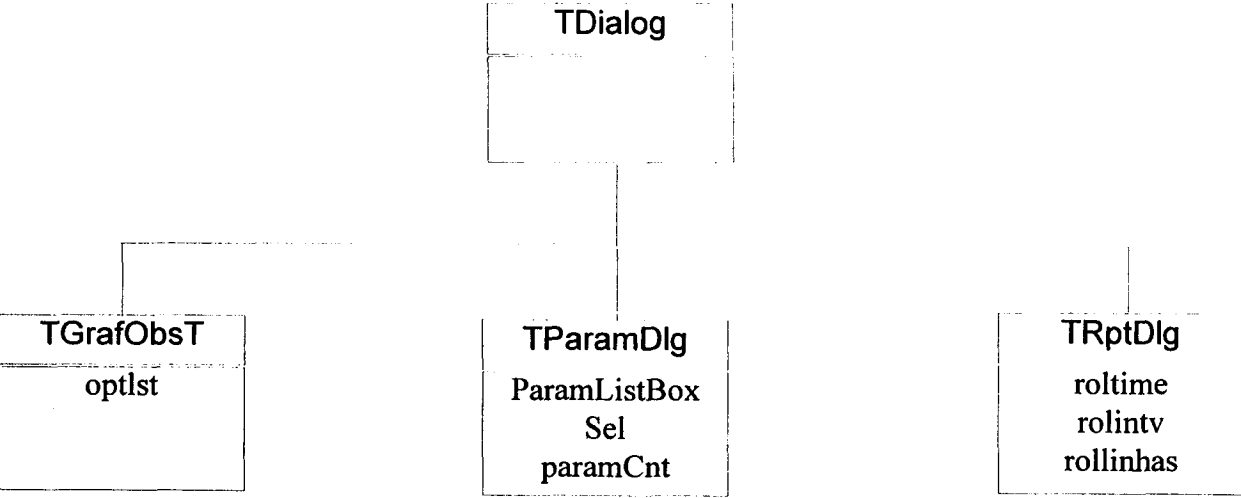
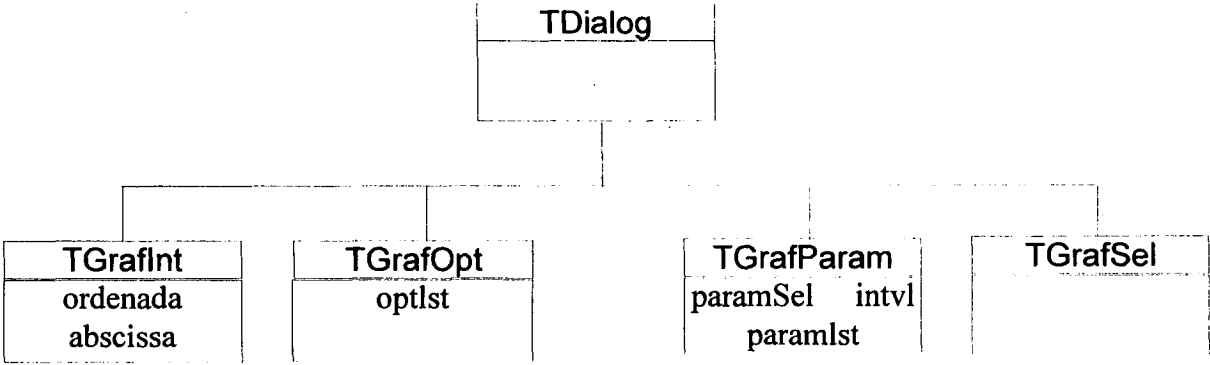
NOME	DESCRIÇÃO
TASiA	Derivada TApplication, implementa o suporte para a geração de aplicação de simulação que utiliza o módulo visual do ASiA.
TBaseWindow	Derivada de TWindow, Implementa a base para as janelas gráficas do módulo visual.
TBMEdit	Derivada de TDialog, implementa meios para a leitura de valores para Batch Means e ativa o cálculo do mesmo.
TConfDlg	Derivada de TDialog, implementa diálogo para a apresentação das configurações do módulo de simulação.
TDspMsg	Derivada de TDialog, implementa diálogo para a apresentação de mensagens de rastreamento do modelo implementado.
TGdiWindow	Derivada de TMDIClient, é responsável pelo gerenciamento das janelas gráficas em exposição no módulo.
TGrafBar	Derivada de TBaseWindow, implementa uma janela gráfica que desenha um gráfico de Barras.
TGrafCur	Derivada de TBaseWindow, implementa uma janela gráfica que desenha um gráfico de Curvas.
tGrafFacI	Derivada de TDialog, implementa diálogo para a seleção de "facility" definida no modelo que será usada na geração de gráfico.
tGrafInt	Derivada de TDialog, implementa diálogo para leitura de intervalos de geração de gráficos.
TGrafLine	Derivada de TBaseWindow, implementa uma janela gráfica que desenha um gráfico de linhas.
tGrafOpt	Derivada de TDialog, implementa diálogo para seleção dos parâmetros do gráfico que será gerado em uma janela gráfica.
TGrafParam	Derivada de TDialog, implementa diálogo para edição dos valores dos parâmetros do gráfico de uma janela gráfica.
TGrafPoint	Derivada de TBaseWindow, implementa uma janela gráfica que desenha um gráfico de Pontos.

TGrafSel	Derivada de TDialog, implementa diálogo para seleção do tipo de gráfico selecionado pelo usuário do modelo.
tGrafObsT	Derivada de TDialog, implementa diálogo para leitura do parâmetro de observação de um gráfico.
TParmDlg	Derivada de TDialog, implementa diálogo para a edição dos valores de parâmetros definidos pelo modelo simulado.
TRptDlg	Derivada de TDialog, implementa diálogo para apresentação de relatório de simulação do modelo.
TTblDlg	Derivada de TDialog, implementa diálogo para leitura de tabelas para o modelo simulado.
TTblLmt	Derivada de TDialog, implementa diálogo para leitura dos intervalos da tabela.
TTblSel	Derivada de TDialog, implementa diálogo para seleção de tabela de usuário para uso.

3.Herança de Classes

A notação de caixas divididas interligadas por linhas define a relação de herança entre classes de objetos. As classes definidas neste trabalho herdaram características das classes definidas pelo Object Windows Library (OWL). A seguir estão as classes com seus ancestrais diretos.





4. Dicionário de dados: Atributos e Métodos

No paradigma de orientação a objetos uma classe é definida como sendo uma abstração que representa a idéia ou noção geral de um conjunto de objetos similares. Um atributo é definido como sendo uma característica de um dado objeto. Métodos de um objeto são definidos como sendo a forma de comunicação com o objeto referido.

Esta seção descreve as classes implementadas pelo módulo visual do ASiA.

4.1 Classe: TASiA

Atributos

Atributo	Tipo	Descrição
NomeModelo	char [20]	Nome do modelo simulado
HelpState	BOOL	Se "help" em uso
ContextHelp	BOOL	Estado da tecla SHIFT-F1
HelpCursor	HCURSOR	"help" sensível a contexto
ScanBf	TDspMsgXfer*	buffer p/ apresentação de linhas
NotInit	BOOL	verifica se o modelo já foi inicializado
MsgBar	TMessageBar*	apresenta tempo de simulação
grafWin	TGdiWindow*	controle das janelas de gráficos

Funções Membro

Função	Tipo	Descrição
Modelo	void	Implementa o modelo simulado.
ParadaModelo	int	Condição de parada do modelo.
IniciaModelo	void	Inicialisa o modelo.
IdleAction	int	Executa o modelo e módulo visual.
Controle	void	Controla as opções ativas do módulo visual.
CmHelpAbout	void	Ativa o diálogo de informação do modelo.
CmHelpContents	void	Ativa a ajuda do módulo visual.
CmHelpUsing	void	verifica se a ajuda está em uso.
CmGrafico	void	Ativa seleção de gráficos.
CmScanLin	void	Ativa rastreamento.
CmScanCont	void	Ativa rastreamento.

CmScanTela	void	Ativa rastreamento.
CmPause	void	Realisa uma pausa na execução do modelo.
CmStopParam	void	Ativa parada por valor de parâmetro e lê valor.
CmStopTime	void	Ativa parada por tempo e lê tempo de parada.
CmParametro	void	Apresenta diálogo com parâmetros do modelo.
CmReset	void	Reinicializa modelo de simulação.
CmStream	void	Altera semente de números aleatórios.
CmBMeans	void	Lê valores de "Batch-Means".
CmQuantiles	void	Lê valores de "Quantiles".
CmDump	void	Gera relatório do status das "facilities" do modelo.
CmRelat	void	Gera relatório dos parâmetros do modelo.
CmTblShow	void	Apresenta a tabela gerada no modelo.
CmConf	void	Apresenta a configuração do módulo visual.
CmBMEnd	long	Informa término do Batch-Means.
DoParamSel	int	Realiza seleção de parâmetros para vários funções.
DisplayErro	void	Apresenta mensagem de erro.
DisplayMsg	int	Apresenta uma mensagem ao usuário.
QuantileInit	int	Inicializa a captura de "Quantiles".
DefineTable	void	Define uma tabela no modelo.
MakeTblDsp	void	Apresenta a tabela do modelo.
TblDistr	void	Lê a distribuição da tabela.
HdrDump	void	Monta o cabeçalho do relatório de Dump.
DumpFacil	void	Gera linhas de "facilities" do Dump .
DumpQueue	void	Gera linhas de filas do Dump.
DumpEvent	void	Gera linhas de eventos do Dump.
BMReport	void	Executa relatório de batch-means
DisplayBreak	void	Apresenta mensagem de parada por valor de parâmetro.
ShowTime	void	Apresenta mensagem de parada por tempo.
BmRptLine	char*	Monta as linhas do relatório de parâmetros.
BMDoReport	void	Executa o relatório.
QtIDoReport	void	Executa o relatório de "quantiles".

4.2 Classe: TBaseWindow

Atributos

Atributo	Tipo	Descrição
Iconized	BOOL	se janela iconizada
PontoMax	BOOL	se ponto máximo em x foi atingido
abscissa	char [50]	rótulo do eixo X
minY	double	menor valor em y
maxY	double	maior valor em y
minX	double	menor valor em X
maxX	double	maior valor em X
intX	double	intervalos em X
UltIdx	double	último valor do índice de observação
ObsIdx	double*	ponteiro para índice de observação
ObsParam	double*	ponteiro para parâmetro de observação
escX	int	escala no eixo X
escY	int	escala no eixo Y

Funções Membro

Função	Tipo	Descrição
RePlot	void	redesenha o gráfico gerenciado.
NewPoint	void	adiciona novo ponto ao gráfico gerenciado.

4.3 Classe: TBMEdit

Atributos

Atributo	Tipo	Descrição
vl1	TEdit	Número de observações iniciais para descartar.
vl2	TEdit	Número máximo de observações
vl3	TEdit	Tamanho inicial do "Batch".
vl4	TEdit	Número inicial de "Batches".
vl3	TEdit	Acuracidade desejada.
vl5	TEdit	Intervalo de confiança.
vl6	TEdit	Número máximo de "lags".
vl8	TEdit	Nível de significância de autocorreção.

Função Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.

4.4 Classe: TConfDlg

Função Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.

4.5 Classe: TDspMsg

Atributos

Atributo	Tipo	Descrição
DspLines	TListBox*	Linhas de mensagens para apresentar.

Funções Membro

Função	Tipo	Descrição
AddStr	void	Adciona uma nova String na lista de apresentação.
ClearList	void	Limpa a lista de apresentação.
CmCancel	void	Desativa o diálogo de apresentação.

4.6 Classe: TGdiWindow

Atributos

Atributo	Tipo	Descrição
grafopt	int	opção de gráfico selecionada.
ptFrame	TModule	ponteiro para frame de janela.
ptobs	double*	ponteiro para parâmetro de observação.
ptidx	double*	ponteiro para índice de observação.

Funções Membro

Função	Tipo	Descrição
Barra2D	void	Cria gráfico de barras segundo especificações .
Curva	void	Cria gráfico de curvas segundo especificações.
Ponto	void	Cria gráfico de pontos segundo especificações.
Linha	void	Cria gráfico de linhas segundo especificações.
Plot	void	Efetua o desenho de cada janela gráfica ativa.
SetupWindow	void	Configura a janela de acordo com o objeto.
EvTimer	void	responde a eventos de atualizações do Windows.
EvDestroy	void	responsável pela destruição do objeto.

4.7 Classe: TGrafBar

Atributos

Atributo	Tipo	Descrição
Points	TRPoint [MaxPoints]	Conjunto de pontos para desenho do gráfico.
PointsCount	int	Contador do numero de pontos no gráfico.

Funções Membro

Função	Tipo	Descrição
Paint	void	Desenha o gráfico .
NewPoint	void	Acrescenta um novo ponto ao gráfico .
CanClose	BOOL	Verifica se pode fecha a janela gráfica .
EvSize	void	Trata o efeito "Zoom" na janela.

4.8 Classe: TGrafCur

Atributos

Atributo	Tipo	Descrição
Points	TRPoint [MaxPoints]	Conjunto de pontos para desenho do gráfico.
PointsCount	int	Contador do numero de pontos no gráfico.

Funções Membro

Função	Tipo	Descrição
Paint	void	Desenha o gráfico .
NewPoint	void	Acrescenta um novo ponto ao gráfico .
CanClose	BOOL	Verifica se pode fecha a janela gráfica .
EvSize	void	Trata o efeito "Zoom" na janela.

4.9 Classe: TGrafFacI

Atributos

Atributo	Tipo	Descrição
facilist	TListBox*	Lista de "facilities".
intvl	int	"facility" selecionada.

Funções Membro

Função	Tipo	Descrição
ClkOk	void	Trata o evento clique do botão OK.
DisplayErro	void	Apresenta mensagem de erro.
ClkGrafInt	void	Trata evento de clique do botão Intervalo (ativa leitura do intervalo de um gráfico.

4.10 Classe: TGrafInt

Atributos

Atributo	Tipo	Descrição
ordenada	TEdit*	Contém nome do parâmetro para ordenada do gráfico.
abscissa	TEdit*	Contém nome do parâmetro para abscissa do gráfico.
minY	TEdit*	Contém o menor valor para o eixo das ordenadas.
mxY	TEdit*	Contém o maior valor para o eixo das ordenadas.
intX	TEdit*	Contém o máximo de intervalos no eixo das abscissas.

Funções Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.
CanClose	int	verifica se intervalos estão corretos e se pode fechar o diálogo.

4.11 Classe: TGrafLine

Atributos

Atributo	Tipo	Descrição
Points	TRPoint [MaxPoints]	Conjunto de pontos para desenho do gráfico
PointsCount	int	Contador do numero de pontos no gráfico

Funções Membro

Função	Tipo	Descrição
Paint	void	Desenha o gráfico
NewPoint	void	Acrescenta um novo ponto ao gráfico
CanClose	BOOL	Verifica se pode fecha a janela gráfica
EvSize	void	Trata o efeito "Zoom" na janela

4.12 Classe: TGrafOpt

Atributos

Atributo	Tipo	Descrição
optlst	TListBox*	Lista de opções de gráficos disponíveis.

Funções Membro

Função	Tipo	Descrição
ClkOk	void	Trata o evento clique do botão OK
DisplayErro	void	Apresenta mensagem de erro.
ClkList	void	Trata evento de clique do botão seleção.

4.13 Classe: TGrafParam

Atributos

Atributo	Tipo	Descrição
facillst	TListBox*	Lista de "facilities".
intvl	int	"facility" selecionada.

Funções Membro

Função	Tipo	Descrição
ClkOk	void	Trata o evento clique do botão OK.
DisplayErro	void	Apresenta mensagem de erro.
ClkGrafInt	void	Trata evento de clique do botão Intervalo (ativa leitura do intervalo de um gráfico.

4.14 Classe: TGrafPoint

Atributos

Atributo	Tipo	Descrição
Points	TRPoint [MaxPoints]	Conjunto de pontos para desenho do gráfico
PointsCount	int	Contador do numero de pontos no gráfico

Funções Membro

Função	Tipo	Descrição
Paint	void	Desenha o gráfico
NewPoint	void	Acrescenta um novo ponto ao gráfico
CanClose	BOOL	Verifica se pode fecha a janela gráfica
EvSize	void	Trata o efeito "Zoom" na janela

4.15 Classe: TGrafSel

Funções Membro

Função	Tipo	Descrição
ClkBarra2D	void	Ativa seleção de gráfico de barra.
ClkBarra3D	void	Ativa seleção de gráfico de curvas.
ClkLinha	void	Ativa seleção de gráfico de linha.
ClkPonto	void	Ativa seleção de gráfico de ponto.

4.16 Classe: TGrafObsT

Atributos

Atributo	Tipo	Descrição
paramlst	TListBox*	Lista de parâmetros para observação.
intvl	int	Item selecionado.

Funções Membro

Função	Tipo	Descrição
ClkOk	void	Confirma o Diálogo.
DisplayErro	void	Apresenta mensagem de erro.
ClkGrafObsT	void	Lê intervalos do gráfico de parâmetros.

4.17 Classe: TParamDlg

Atributos

Atributo	Tipo	Descrição
Sel	unsigned int	Tipo de leitura.
paramCnt	int	Número de parâmetros selecionados.

Funções Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.
HandleEditMsg	void	Manipula mensagens de edição.
EditParam	void	Edita parâmetros do modelo.
EditStop	void	Lê valor de parada para modelo.
SelBMPParam	void	Seleciona parâmetros do Batch-Means.
HandleBMEdit	void	Manipula Edição do Batch-Means.
DisplayErro	void	Apresenta Mensagem de Erro.

4.18 Classe: TRptDlg

Atributos

Atributo	Tipo	Descrição
roltime	TEdit*	Apresenta o tempo de simulação.
rolintv	TEdit*	Apresenta o intervalo de simulação.
rollinhas	TEdit*	Apresenta o relatório.

4.19 Classe: TTbIDlg

Atributos

Atributo	Tipo	Descrição
listparam	TListBox*	Lista de parâmetros do modelo.
obs	TEdit*	Valor do parâmetro de observação.
idx	TEdit*	Valor do parâmetro de índice.
paramCnt	int	Número de parâmetros selecionados.

Funções Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.
ParamSelect	void	Seleciona parâmetro para "Quantile".
EditValue	void	Edita valor do intervalo.
DisplayErro	void	Apresenta Mensagem de Erro.

4.20 Classe: TTbLLmt

Atributos

Atributo	Tipo	Descrição
inf	TEdit*	Limite inferior da tabela.
sup	TEdit*	Limite superior da tabela.
intv	TEdit*	intervalo entre os limites.

4.21 Classe: TQtTblSel

Atributos

Atributo	Tipo	Descrição
tblst	TListBox*	Lista de parâmetros do modelo.
atbl	TEdit*	Tabela selecionada.

Funções Membro

Função	Tipo	Descrição
SetUpWindow	void	Dimensiona campos de leitura.
TblSelect	void	Seleciona parâmetro para "Quantile".
DisplayErro	void	Apresenta Mensagem de Erro.

5. Dicionário de Dados: Eventos

No contexto de programação orientada a eventos define-se um evento como um comando gerado (pelo usuário, objetos, dispositivos de hardware e outros) para a aplicação em execução. A tabela a seguir apresenta os eventos definidos e tratados pelo módulo visual.

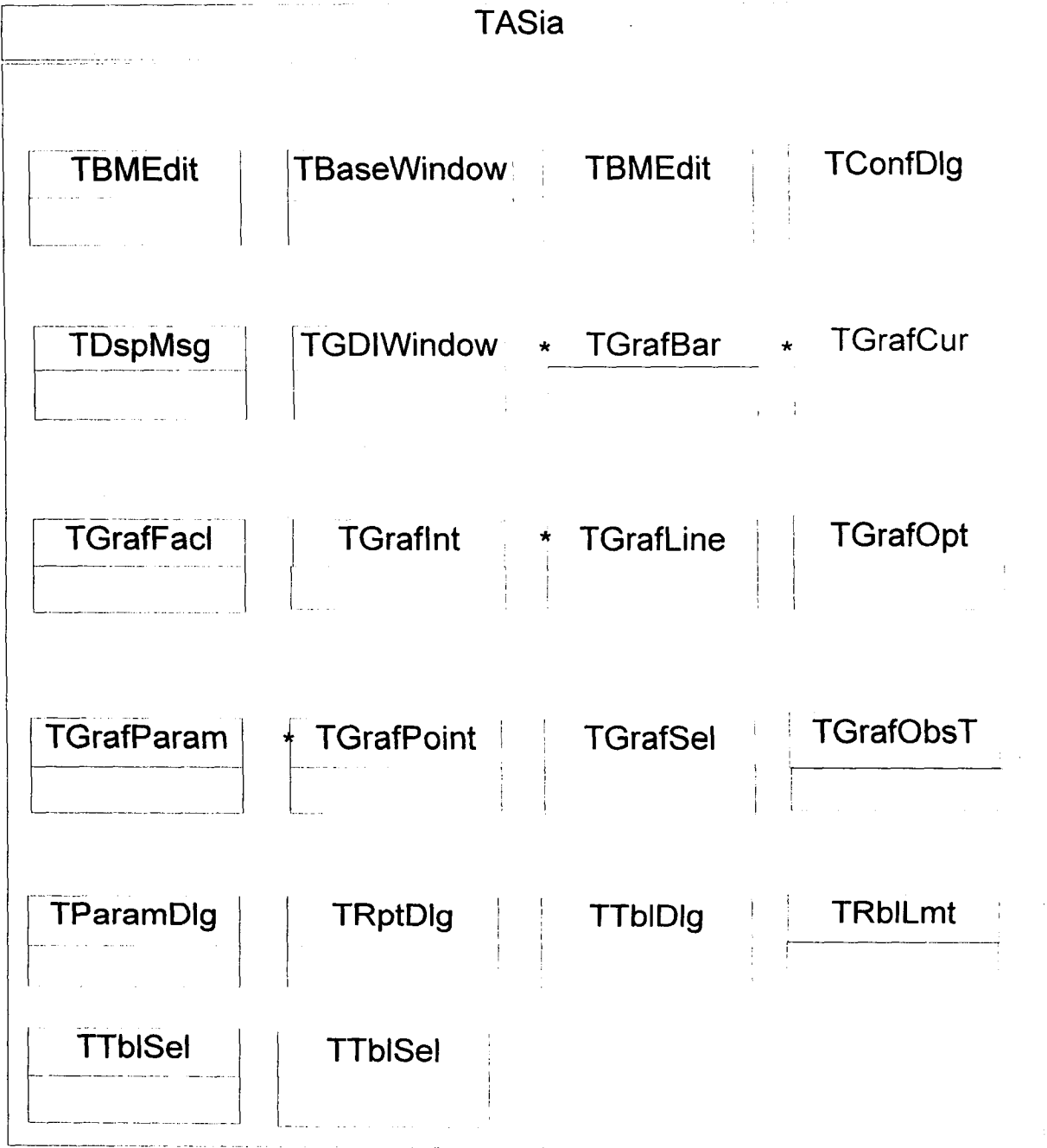
Evento	Descrição
CM_BMEND	Informa o término da análise de Batch-Means
CM_GRAFICO	Ativa a seleção de gráficos do módulo visual
CMHELPABOUT	Ativa o diálogo de informações sobre o sistema
CM_HELPCONTENTES	Ativa a ajuda do sistema
CM_HELPUSING	Ativa a ajuda sensível ao contexto
CM_PARAMETRO	Ativa a edição dos parâmetros definidos no módulo
CM_PARADASPAUSA	Efetua uma pausa no sistema
CM_SCANCONTINUO	Ativa o rasteramento contínuo do módulo visual
CM_SCANLINHA	Ativa o rasteramento linha a linha do módulo visual
CM_SCANTELA	Ativa o rasteramento por tela do módulo visual

CM_STOPPRAM	Efetua leitura d e parada por valor de parâmetro
CM_STOPTEMP	Efetua leitura de parada por tempo de simulação
CM_TBLAPRES	Apresenta uma tabela definida
CM_TBLDEF	Define uma tabela de usuário
CM_UTILBMEANS	Define os critérios de Batch-Means
CM_UTILCONF	Apresenta a configuração do módulo visual
CM_UTILDUMP	Gera relatório das filas do SMPL
CM_UTILRESET	Zera os valores dos filas e facilities do módulo
CM_UTILROL	Gera relatório das facilities definidas
CM_UTILSTREAM	Altera a semente de números aleatórios
CM_UTILQUANT	Ativa a seleção de quantiles
EV_WM_DESTROY	Fecha um janela MDI ativa
EV_WM_SIZE	Modifica o tamanho de uma janela MDI
IDCANCEL	Cancela a operação em andamento
IDC_GRAFBARRA2D	Ativa seleção de gráfico de barras
IDC_GRAFBARRA3D	Ativa seleção de gráfico de curvas
IDC_GRAFINT	Ativa a leitura de intervalos para os gráficos
IDC_GRAFLINHA	Ativa seleção de gráfico de linha
IDC_GRAFPARAMINT	Seleciona intervalo para gráficos de parâmetros
IDC_GRAFPARAMSEL	Seleciona parâmetros para gráficos
IDC_GRAFPONTO	Ativa seleção de gráfico de pontos
IDC_GROPTLST	Apresenta lista de opções para gráficos
IDOK	Confirma a operação em andamento
IDC_PARAMEDIT	Permite seleção de parâmetro para edição
IDC_SELPARAM	Edita valores dos parâmetros
IDC_TBLSEL	Permite seleção de tabelas

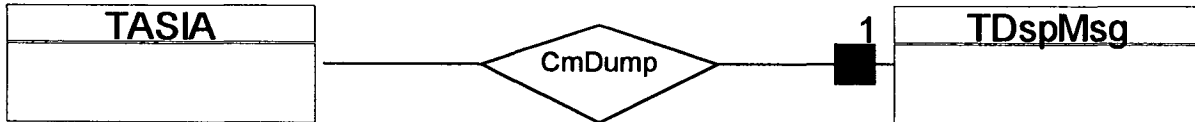
6.Relacionamento de Classes

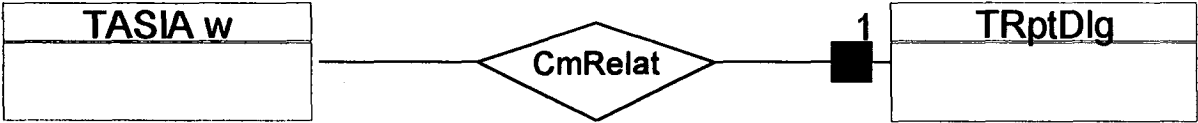
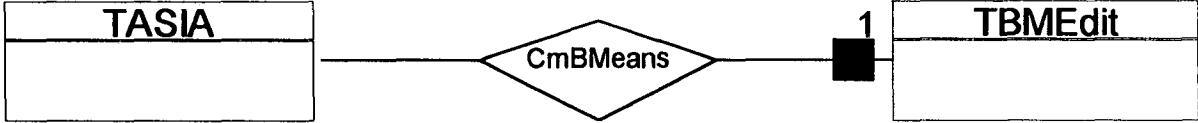
Assim como no mundo real existem relacionamentos entre os objetos, ocorrem relacionamentos no paradigma de orientação a objetos. Um relacionamento é uma tupla de objetos que dependendo do tipo de relacionamento, um número qualquer de objetos pode estar envolvido. Uma relação entre classes expressa apenas a *possibilidade* de um relacionamento entre objetos das classes envolvidas. Não implica que cada objeto de uma classe está relacionada com todos objetos de outra classe [FUS].

Os esquemas a seguir apresentam os relacionamentos entre as classes definidas no Módulo Visual.











7. Descrições dos Módulos e Principais Rotinas

Esta seção descreve cada módulo e as principais rotinas implementadas no projeto do Módulo visual.

7.1 Módulo asiaV.cpp

Este módulo contém rotinas preliminares para criação do módulo visual do ASiA permitindo a visualização dos resultados em tempo de execução do programa de simulação.

Variáveis

Nome	Tipo	Descrição
asiaTraceFlag	static int	controle de rastreamento do modelo simulado
bkpTime	static int	flags de parada por tempo. Valores: 0 - não ativo; 1 - parada ativada; 2 - valor atingido
bkpptParam	static int	flags de parada por parametro. Valores: 0 - não ativo; 1 - parada ativada; 2 - valor atingido
bkpParamIdx	static int	índice de parada de parâmetro.
bktime	static real	valor de parada por tempo.
bkParam	static real*	ponteiro para parâmetro de parada.
bkParamVal	static real	valor de parada por parâmetro.
call [mxpoll]	static struct call_parms	Conjunto de rotinas a serem executadas pela função asiaVCtrl.
NbrPoll	static int	número de operações ativas no poll.
glMsgWin	TDspMsg*	global para controle de mensagens de rastreamento.
po	double*	ponteiro para parâmetro de observação.
pi	double*	ponteiro para parâmetro de índice.

Funções

int asiaVCtrl ()

Processa o controle do módulo visual.

void asiaVRequestPoll (double *v,double *c, int (*f)(double))

Através desta função os módulos de análise podem adicionar suas funções e parâmetros de análise no “poll” de funções executadas pela asiaVCtrl().

Parâmetros:

v :aponta para parâmetro de observação.

c :aponta para parâmetro de índice.

f :aponta para a função que será executada cada vez que os valores apontados por v e c forem atingidos.

void asiaVParamStop (int i, char* vl)

Estabelece o valor de parâmetro e o parâmetro para realizar uma parada na execução do modelo.

Parâmetros:

i: índice do parâmetro a ser observado.

vl: valor para parada.

double asiaVStopVal (void)

Retorna o valor do parâmetro de parada do modelo.

int asiaVStopIdx (void)

Retorna o índice do parâmetro de parada do modelo.

void asiaVTimeStop (double i)

Ativa e estabelece o valor da parada por tempo do modelo.

Parâmetro:

i: tempo de parada do modelo.

void asiaVSetTrace (int st)

Estabelece o tipo de rastreamento para o modelo em execução.

Parâmetro:

st: novo estado para o rastreamento do modelo.

int asiaVGetTrace (void)

Retorna o tipo de rastreamento ativo no módulo.

void asiaVMsgWin (TWindow* win, int init = 0)

Cria diálogo para apresentação do rastreamento.

int asiaVMsgWinState ()

Retorna o estado do diálogo de rastreamento.

1 - se ativo;

0 - caso contrário.

TDspMsg* asiaVGetMsgWin ()

Retorna um ponteiro para o diálogo de rastreamento.

void asiaVMsg (char* s)

Adiciona no diálogo de rastreamento uma mensagem.

double* asiaVpi (void)

Retorna ponteiro de índice do módulo.

void asiaVSetpi (double *v)

Altera o ponteiro de índice do módulo.

double* asiaVpo (void)

Retorna o ponteiro de parâmetro do módulo.

void asiaVSetpo (double *v)

Altera o ponteiro de parâmetro do módulo.

7.2_Módulo asiaB.cpp

Contém rotinas e estruturas para cálculo de beatch-means.

Variáveis

Nome	Tipo	Descrição
nb0	static int	número inicial de batches
nb	static int	número corrente de batchs - deve estar pelo - na fase 0
nlags	static int	número de lags para teste de autocorrelação.
phase	static int	fase da análise: 0 - bma inativo 1 - encontra tamanho do batch 2 - encontra no. de batches 3 - analise completa
dt	static int	flag de decisão de rastreamento de análise
mr	static int	flag de status do monitor (1 se ativo)
bs0	static long	tamanho inicial
bs	static long	tamanho corrente do batch
n0	static long	número inicial de observações para descartar
nv	static long	número de observações coletadas (inclui n0)
nt	static long	número de observações para coletar: $nb \cdot bs + n0$
nx	static long	máximo número de observações para coletar
y[mxb]	static double	$y[i]$ =i-ésima soma de batch de valores ou media
re	static double	intervalo de confiança relativo de half-width desejado
cl	static double	nível de confiança especificado (1 - α)
al	static double	nível de significância de autocorrelação
sm	static double	grand mean
sv	static double	variância do batch means
hw	static double	intervalo de confiança estimado half-width
tc	static double	tempo de simulação do último batch completado
vn	char [asParamNameSp]	nome da variável sendo analisada
BMMsgWin	TDspMsg*	local para controle de mensagens de rastreamento.

Funções

int asiaBInit (double *o, double *j, int x)

Inicializa a análise de Batch-Means.

Parâmetros:

o: ponteiro para parâmetro de observação;

j: ponteiro para parâmetro de índice;

x: decisão de rastreamento.

void asiaBSet (int i1, int i2, long l1, long l2, long l3, real r1, real r2, real r3)

Chamada para alterar 1 ou mais dos parâmetros padrões de análise de saída. Pode ser chamada antes ou depois de 'asiaBInit', mas deve ser chamada antes da primeira chamada a 'asiaBAccum' (uma vez que nenhuma checagem é feita).

Parâmetros:

i1 - nb0 número de batches

i2 - nlags número de lags

l1 - bs0 tamanho inicial do batch

l2 - n0 número inicial de observações para descartar

l3 - nx máximo de observações para coletar antes de parar

r1 - re intervalo de confiança relativo half-width

r2 - cl intervalo de confiança especificado (1 - a)

r3 - al nível de significância especificado para testes

Se o valor de um argumento for 0, o parâmetro de análise correspondente tem seu valor padrão associado na declaração exetuo n0.

int asiaBState (void)

Retorna o estado de asiaB.

void asiaBSetState (int st)

Altera o estado de asiaB.

Parâmetro:

st: novo estado.

int asiaBAccum (real v)

Chamada pelo modelo de simulação ou pelo asiaVCtrl para processar a observação corrente; v é o valor da observação corrente. No fim da análise, 'asiaBAccum' retorna 1; caso contrario retorna 0.

void asiaBMeans (void)

Computa o Batch means, Media total e variância.

real asiaBR (int l)

Computa o i-ésimo lag de autocorrelação.

void asiaBDisplayR (void)

Apresenta a autocorrelação

void asiaBResults (real* mean, real* ci, long* nobs)

Retorna o a média mean, intervalo de confiança half width ci, e o número total de observações nobs. Permite ao modelo obter os resultados de Batch-Means sem gerar relatório para tanto.

void asiaBRpt (void)

Gera evento fim de análise.

int asiaBAnalyze (void)

Chamada quando o número de observações especificadas em nt é atingida completando o batch. Chama 'means' para calcular batch means da soma do batch, e então chama ou o 'bsize', se a determinação do tamanho do batch está no processo, ou 'asiaBNumberBatches', se a determinação do número de batches está no processo.

int asiaBSize (void)

Computa o coeficiente de correlação serial do batch sobre o número de lags especificados e determina se todos os coeficientes satisfazem a hipótese de autocorrelação Zero. Se não satisfaz dobra o tamanho do batch e coleta batches adicionais.

int asiaBNumberBatches (void)

Computa o intervalo de confiança de half-width e compara com o valor desejado; se ainda não obtido, verifica se a coleta de um batch adicional exederá o número total de batch ou o limite total de observações. se não coleta outro batch. Se o comprimento desejado foi atingindo, reduz o número de batchs de 30 ou menos, checa o comprimento outra vez e continua coletando até aceito.

void asiaBHalfWidth (void)

Calcula o intervalo de confiança Half-Width.

int asiaBObsLimit(int i)

Chamada quando a análise de batch-means dobra o tamanho do batch ou coleta um batch adicional.

void asiaBBatchDbl(void)

Dobra a tamanho do batch e divide o número de batches.

int asiaBReduceBatch (void)

Reduz nb a 30 ou menos dobrando o tamanho do batch size e dividindo por 2 o número de batches; Isto requer que primeiro seja coletado um outro batch quando nb for impar. Note que um máximo de duas metades pode ter lugar aqui (dado mxb = 100).

void asiaBReconvert (void)

Reconverte y [i]s de media para soma.

double asiaBSmITime (void)

Retorna o tempo de simulação.

char* asiaBVarName (void)

Retorna o nome da variável sendo analisada.

void asiaBMsgWin (TWindow* win, int init)

Inicializa ou desativa a janela de mensagens do Batch-Means.

int asiaBMsgWinState ()

Retona o estado da janela de mensagens de Batch-Means.

1 - se janela ativa

0 - caso contrário

TDspMsg* asiaBGetMsgWin ()

Retorna um ponteiro para a janela de mensagens de Batch-Means.

void asiaBMsg (char* s)

Adiciona a mensagem 's' a janela de mensagens Batch-Means.

void asiaBClrMsg (void)

Limpa as mensagens existentes na janela de mensagens Batch-Means.

```
void asiaBGetDefault ( int *NB0, int *NLags, long *N0, long *BS0,  
                      long *NX, double *Al, double *Re,  
                      double *Cl )
```

Retorna os parâmetros de entrada com os valores dos parâmetros de análise de batch-Means.

Parâmetros de Saída:

NB0	número de inicial de batches
Nlags	número de lags para teste de autocorrelação
N0	número inicial de observações para descartar
BS0	tamanho inicial e corrente do batch
NX	máximo número de observações para coletar
al	nível de significância de autocorrelação
re	Intervalo de confiança relativo desejado de half-width.
cl	nível de confiança especificado (1 - a)

```
void asiaBSetDefault ( int NB0, int NLags, long N0, long BS0, long  
                      NX, double Al, double Re, double Cl )
```

Altera os parâmetros de entrada com os valores dos parâmetros de análise de batch-Means.

Parâmetros de Entrada:

NB0	número de inicial de batches
Nlags	número de lags para teste de autocorrelação
N0	número inicial de observações para descartar
BS0	tamanho inicial & corrente do batch
NX	máximo número de observações para coletar
al	nível de significância de autocorrelação
re	Intervalo de confiança relativo desejado de half-width.
cl	nível de confiança especificado (1 - a)

```
int asiaBPhase ( void )
```

Retorna a fase atual do Batch-Means.

```
int asiaBCurrent ( void )
```

Retorna o número do batch atual.

```
int asiaBCurrentLap ( void )
```

Retorna o número do lap atual.

```
long asiaBCurrentLen ( void )
```

Retorna o tamanho atual do batch.

7.3 Módulo asiaP.cpp

Este módulo implementa rotinas para tratamento de parâmetros definidos por usuário no modelo de simulação.

Variáveis

Nome	Tipo	Descrição
asiaParam[asParamMax]	TASiAParam	Parâmetros definidos
asiaVParamCount	static int	número de parâmetros definidos

Funções

int asiaPCount (void)
Retorna o número de parâmetros definidos no modelo de simulação.

void asiaPInit (void)
Chamada por asiaPDefine na primeira definição de parâmetro.
inicializa a tabela de parâmetros.

char *asiaPName (int i)
Retorna o nome do i-ésimo parâmetro.

double asiaPValue (int i)
Retorna o valor do i-ésimo parâmetro.

double* asiaPptValue (int i)
Retorna o endereço do i-ésimo parâmetro.

void asiaPNewValue (int i, double v)
Atualiza o valor do i-ésimo parâmetro com o valor de v.

void asiaPDefine (real* p, char* s)
Define o i-ésimo parâmetro como um ponteiro para uma variável do tipo real apontada por 'p'. Armazena o ponteiro e o nome (s).

7.4 Módulo asiaT.cpp

Constante

Nome	Valor	Descrição
mxn	51	Tamanho max. do nome de uma tabela

Variáveis

Nome	Tipo	Descrição
tbldef	static int	número de tabelas definidas.
tblnxn	static int	posição do próximo espaço de nome disponível.
tblavc	static int	índice da próxima célula livre.
tblmtbl	static int	índice da tabela definida pelo módulo visual.
tblno	static int	número de observações de parâmetro para tabela definida.
tblni	static int	número de índice de parâmetro para tabela definida.
tblCell[nc+1]	static int	pool de células de tabela.
*tblpo	static real	ponteiro para parâmetro de obs.
*tblpi	static real	ponteiro para parâmetro de índice
tblL1[3*ntbls+1]	static int	elemento descritor de tabela pool (elemento 0 não usado)
tblL2[3*ntbls+1]	static int	elemento descritor de tabela pool (elemento 0 não usado)
tblL3[3*ntbls+1]	static int	elemento descritor de tabela pool (elemento 0 não usado)
tblL4[3*ntbls+1]	static real	elemento descritor de tabela pool (elemento 0 não usado)
tblL5[3*ntbls+1]	static real	elemento descritor de tabela pool (elemento 0 não usado)
tblNames[mxn*ntbls+1]	static char	espaço para nomes de tabelas

int asiaTCreate (real from, real to, int n, char* p)

Esta função define uma tabela, cria uma estrutura de dados e retorna um índice para esta estrutura para uso subsequente. 'from' e 'to' especificam o intervalo da tabela e 'n' especifica o número de intervalos (células) no qual este intervalo deve ser dividido. Um contador retém o número de valores que se encaixam em cada intervalo; Quando o relatório da tabela é gerado, o número e proporção de valores armazenados em cada intervalo é relatado. também são

relatados a média, desvio padrão, mínimo e máximo. (se 'n' for 0, apenas estes valores são relatados.). 'p' aponta para o nome da tabela (ver asiaTName).

int asiaTStoreName (char* p)
Armazena o nome da tabela.

void asiaTEnter (real v, int tbl)
Insere o valor 'v' na tabela 'tbl'.

int asiaTEnterVal (real v)
Armazena valor 'v' na tabela definida.

void asiaTClearAll (void)
Limpa as tabelas.

void asiaTClear (int tbl)
Limpa a tabela tbl.

int asiaTFracDig (real x)
Pega o número de dígitos fracionários de x.

int asiaTWidth (real v, int nf)
Retorna o tamanho do campo.

int asiaTGreater (int i, int j)
Retorna o maior enter 'i', 'j'.

void asiaTSetno (int x)
Modifica o valor de no para o valor de x.

void asiaTSetni (int x)
Modifica o valor de ni para o valor de x.

void asiaTSetpi (double* x)
Modifica o valor de pi para o valor de x.

void asiaTSetpo (double* x)
Modifica o valor de po para o valor de x.

double* asiaTGetpo (void)
Retorna o valor de po.

double* asiaTGetpi (void)
Retorna o valor de pi.

int asiaTGetni (void)
Retorna o valor de ni.

int asiaTGetno (void)

Retorna o valor de no.

void asiaTSetmtbl (int x)

Modifica o valor de mtbl para o valor de x.

int asiaTNbrOftbl (void)

Retorna o número de tabelas definidas.

char* asiaTTblName (int t)

Retorna o nome da tabela definida na célula t.

int asiaTTbl1 (int t)

Retorna o elemento descritor da tabela em tbl1 [t].

int asiaTTbl2 (int t)

Retorna o elemento descritor da tabela em tbl2 [t].

int asiaTTbl3 (int t)

Retorna o elemento descritor da tabela em tbl3[t].

double asiaTTbl4 (int t)

Retorna o elemento descritor da tabela em tbl4[t].

double asiaTTbl5 (int t)

Retorna o elemento descritor da tabela em tbl5 [t].

int asiaTGetCell (int c)

Retorna o elemento c do pool de células de tabelas.

7.5 Módulo asiaconf.h

Contém definição de constantes e estruturas usadas no módulo visual.

Tamanho das estruturas de dados do 'smpl'

nome	Tamanho	Descrição
nl	256	Tamanho do pool de elementos.
ns	256	Tamanho do espaço de nomes do 'smpl'.
asParamMax	30	Número máximo de parâmetros em um modelo.
asParamNameSp	20	Comprimento do nome do parâmetro.
ntbls	4	Comprimento do pool de elementos de 'table'
nc	100	Comprimento da célula do pool 'table'.
mxpoll	4	Número máximo de conjunto de pedidos de tabelas.
mxb	100	máximo número de batches.
mxh	22	Tamanho máximo de histograma + 2.
mxs	51	Tamanho máximo do nome da distribuição.

Estruturas globais

Tabela descritiva de pedidos

```
struct call_parms
{
    double *obs; // ponteiro para parâmetro de observação
    double *inx; // ponteiro para parâmetro de índice
    double last; // valor de 'inx' na última chamada do Controle ()
    int (*callee)( double ); // função a ser chamada quando 'inx' > 'last'
};
```

Estrutura para conter os parâmetros do modelo

```
typedef struct TASiAParam {
    char Name [ asParamNameSp + 1 ]; // nomes dos parâmetros
    double* Value; // valores dos parâmetros
};
```

8. Referências Bibliográficas

- [PED95] PEDROSO, Walter G. Jr. **ASiAModV - Módulo Visual do ASiA. Manual de usuário.** - ICMSC-USP. Setembro 1995.
- [SPO93a] SPOLON, Renata. **Um Editor Gráfico para Simulação de Sistemas Discretos.** Minidissertação (Mestrado) -ICMSC-USP. Junho 1993.
- [SPO93b] SPOLON, Roberta. **Um Gerador de Aplicação Para Um Ambiente de Simulação Automático.** Minidissertação (Mestrado) -ICMSC-USP. Junho 1993.