

Universidade do Minho

Escola de Engenharia

Departamento de Produção e Sistemas

Visio Basic for Simulations

Braga, Agosto de 2010

Índice

1	Introdução	3
2	Instalação	4
3	Utilização	7
3.1	New Simulation	9
3.2	Generate Simulation Code	10
3.3	Open With Stencils	13
4	Princípios e regras de desenho	14
5	Módulos	16
5.1	Main	16
5.2	Event	17
5.3	Queue	18
5.4	Resource	19
5.5	Expert	20
5.6	Links	21
6	Exemplo	22
6.1	Motor do Sistema	22
6.2	Acontecimento ClientArrival	23
6.3	Acontecimento EndOfService	24

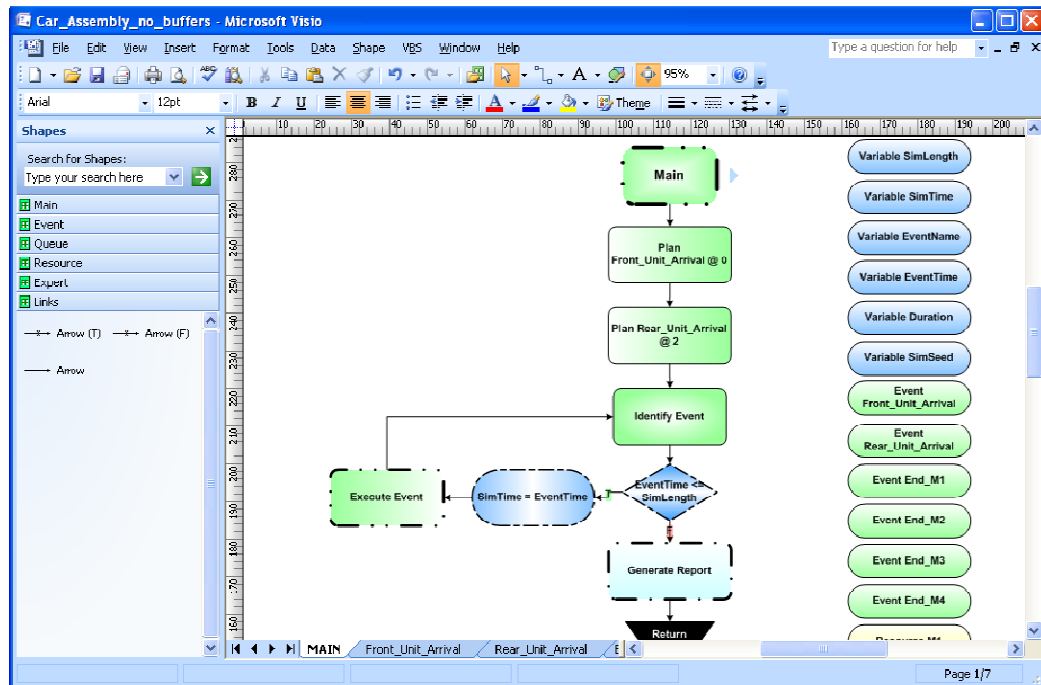
1 Introdução

O Visio Basic for Simulations (VBS) é uma aplicação, disponibilizada como *add-in* para o Microsoft® Office Visio® 2007, que permite a geração automática de programas de simulação, segundo a filosofia Planeamento de Acontecimentos (*Event Scheduling*).

A geração automática dos programas de simulação é efectuada através da análise de fluxogramas, previamente desenhados no Microsoft® Visio. O desenho desses

fluxogramas deverá recorrer a módulos (*shapes*) próprios, disponibilizados para o efeito e respeitar um conjunto de regras.

Uma vez que se trata de um *add-in*, a interacção com o VBS é feita através do Microsoft® Office Visio® e não através de um interface próprio, como acontece com as aplicações tradicionais.



2 Instalação

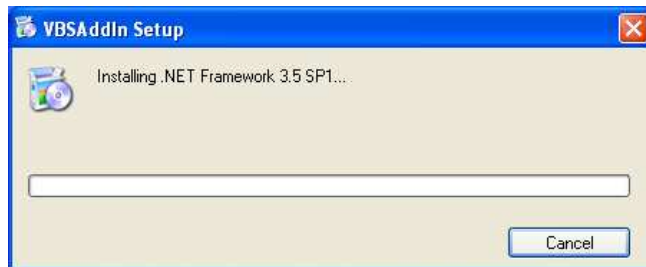
Uma vez que esta aplicação é disponibilizada como *add-in*, antes de efectuar o processo de instalação é necessário garantir que o **Microsoft® Office Visio® 2007** (e não qualquer outra versão!) já se encontra instalado.



A instalação do VBS é muito simples, basta para isso executar o programa de instalação (setup.exe) e seguir os passos do assistente. **Durante o processo de instalação poderão ser descarregados alguns ficheiros (bibliotecas) necessários ao sistema, pelo que será necessário estar ligado à Internet.**

O instalador começa por detectar se já existem no sistema as bibliotecas necessárias, nomeadamente a *.NET Framework 3.5 SP1* e o *Visual Studio Tools for the Office system 3.0 SP1*. Se as bibliotecas necessárias já se encontrarem instaladas, o instalador avança para a instalação do VBS, caso contrário descarrega e instala as bibliotecas em falta.

Não existindo ligação à Internet, poderá não ser possível continuar com a instalação, visto que poderão faltar alguns componentes importantes.



Depois de instaladas as bibliotecas, ou se estas já existirem no sistema, o instalador procederá à instalação do VBS. A instalação é bastante simples e intuitiva, tendo o utilizador apenas que confirmar que pretende instalar a aplicação.

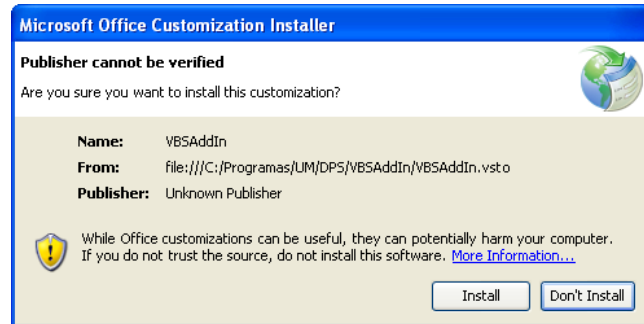
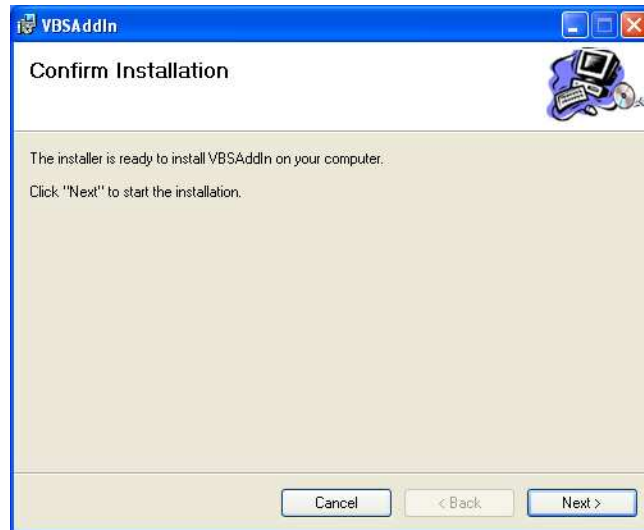


Concluída a instalação, será apenas necessário abrir o Microsoft® Office Visio® 2007 para que este incorpore o *add-in* do VBS.

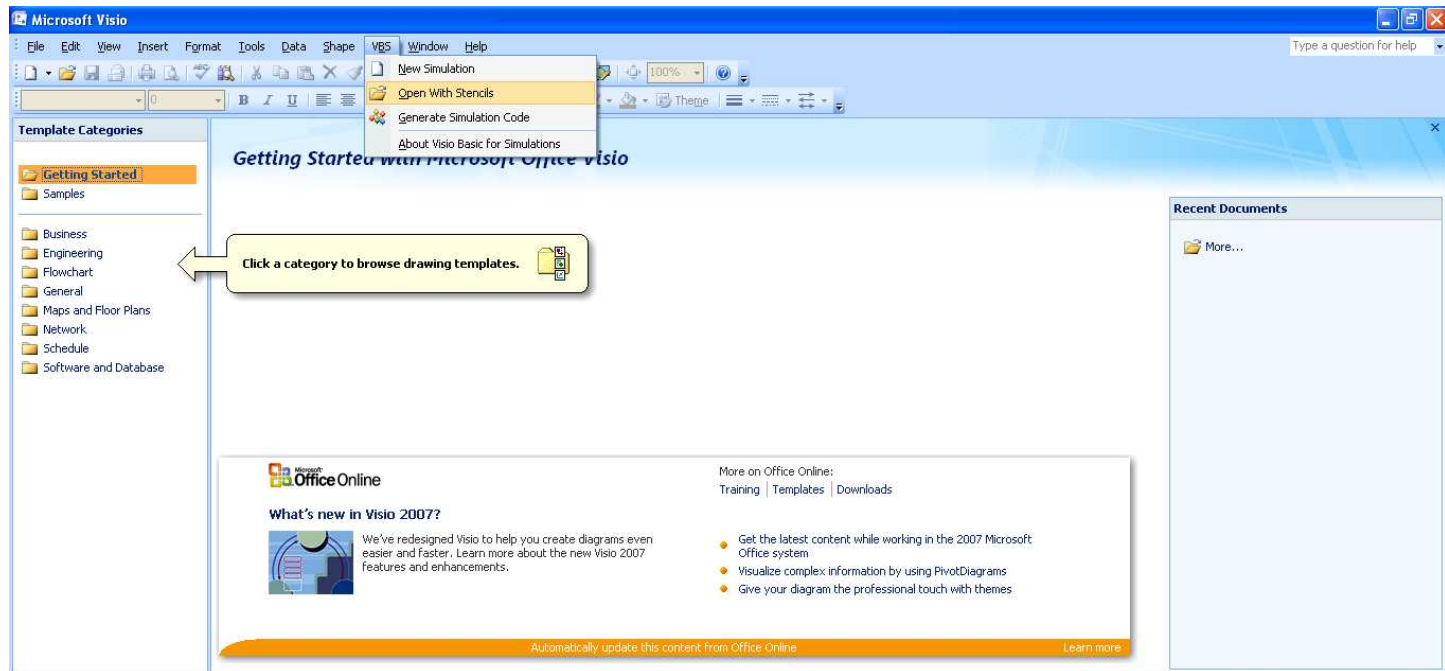
Por questões de segurança, e tal como acontece com as *macros*, os *add-ins* apenas serão incorporados com a autorização explícita do utilizador.

Na primeira execução do Microsoft® Office Visio® 2007, e após concluída a instalação, o utilizador terá que autorizar a incorporação do VBS AddIn na aplicação da Microsoft®.

Depois de autorizada a incorporação do *add-in*, é adicionado



um novo menu ao Microsoft[®] Visio. É a partir deste menu que é feita toda a interacção com o VBS.



Esse menu permite criar novos diagramas, abrir diagramas existentes, bem como efectuar a geração de código fonte a partir dos diagramas.

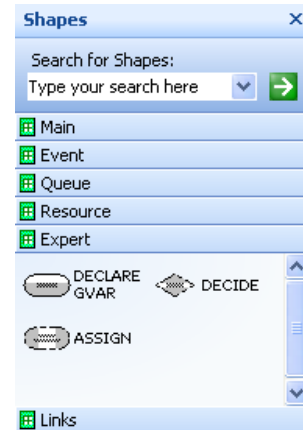
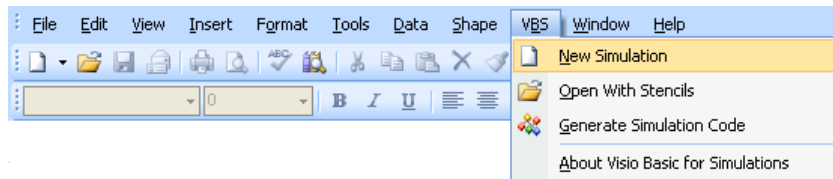
3 Utilização

O Visio Basic for Simulations (VBS) é uma aplicação para a geração automática de programas de simulação, segundo a filosofia Planenamento de Acontecimentos (*Event Scheduling*). A geração automática dos programas de simulação é efectuada através da análise de fluxogramas, previamente desenhados. Uma vez que seria necessário lidar com a complexidade da criação de diagramas e face às potencialidades de integração do Microsoft® Office® Visio 2007, optou-se por desenvolver uma aplicação que interagisse directamente com o Visio, razão pela qual esta é disponibilizada em forma de *add-in*.

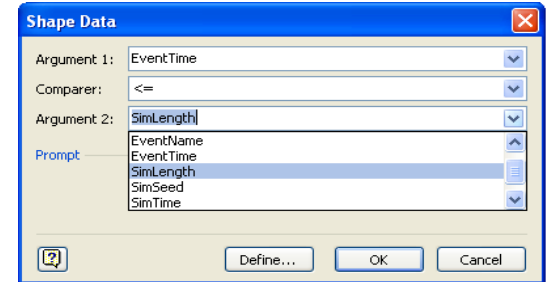
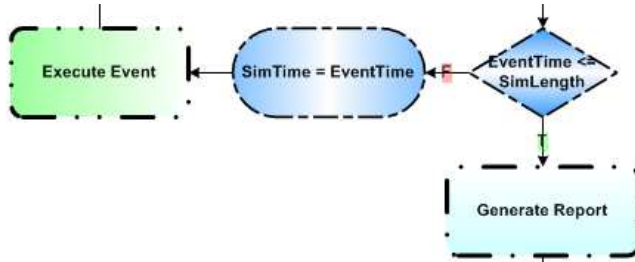
Toda a interacção com a aplicação, assim como a abertura e criação de novas simulações deverá ser efectuada através do menu do VBS em vez dos menus tradicionais do Visio!

Esta aplicação permite a geração de programas de simulação, sem que o utilizador tenha de escrever uma única linha de código! O utilizador terá apenas que focar a sua atenção na construção dos fluxogramas que melhor descrevam os acontecimentos da simulação.

Para a construção dos fluxogramas, existem na aplicação seis conjuntos de módulos: **Main**, **Event**, **Queue**, **Resource**, **Expert** e **Links**.



Cada conjunto (*stencil*) inclui os módulos (*shapes*) necessários para lidar com diferentes aspectos do sistema, sendo que cada módulo representa uma operação distinta, podendo esta estar preparada para receber parâmetros de entrada e/ou saída.



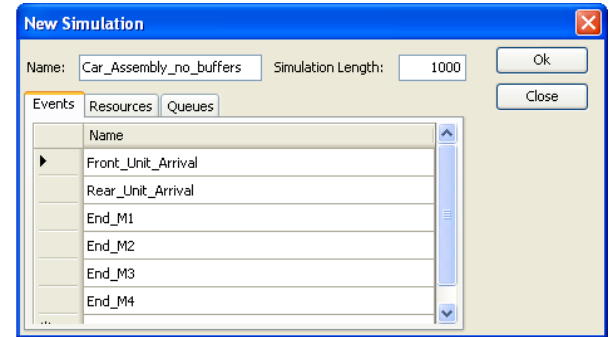
Apesar de ser possível alterar o modo como o “motor” do sistema funciona, o utilizador terá apenas que modelar os diferentes acontecimentos que compõem a simulação. Para agilizar o processo de criação de uma nova simulação, foi criado um assistente.

De uma forma geral, podemos afirmar que o VBS é uma aplicação que cobre todo o processo da geração de programas de simulação, uma vez que permite a **criação dos diagramas**; efectua a **interpretação dos diagramas, gerando o respectivo código fonte**; interpreta o código gerado, mostrando os **resultados da simulação**; permitindo também a exportação do código gerado para o Microsoft® Visual Basic® 2008, onde este poderá ser compilado.

3.1 New Simulation

O assistente, além de modelar automaticamente o mecanismo de simulação, permite de uma forma rápida declarar quais os acontecimentos, recursos e filas que serão utilizados na simulação. Para chamar o assistente, basta ir ao menu do VBS e seleccionar a opção *New*.

Com base nos acontecimentos declarados, o assistente criará uma página com o template adequado para a modelação de cada acontecimento.



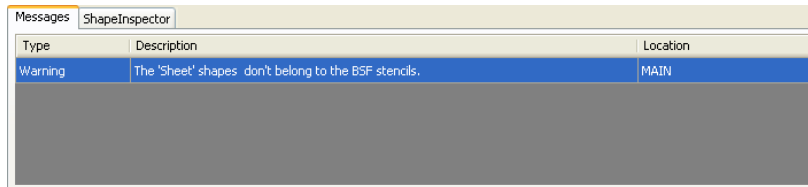
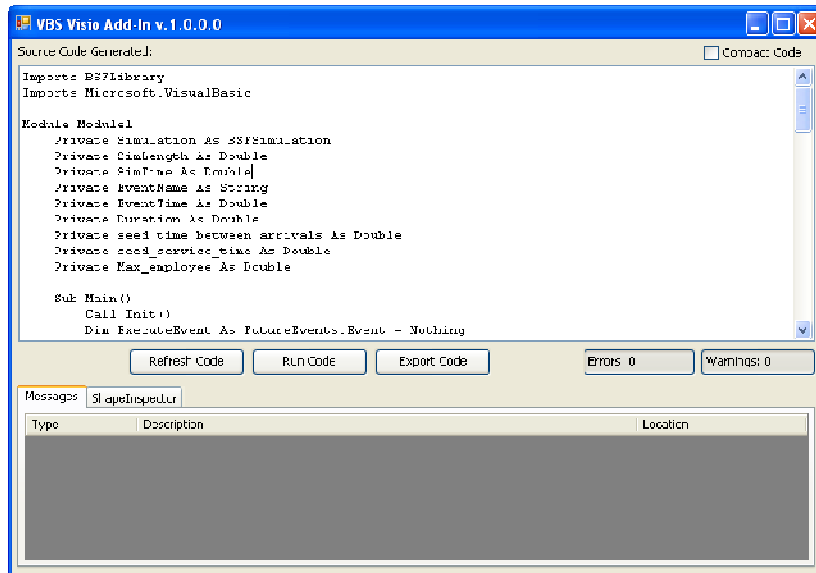
Se o utilizador optar por não utilizar o assistente, terá que efectuar manualmente todo o processo de declaração, modelação de acontecimentos e do mecanismo do sistema!

3.2 Generate Simulation Code

Depois de modelados os acontecimentos da simulação, estamos em condições de gerar o código fonte do que estamos a construir. Para gerar o código fonte, devemos seleccionar a opção *Generate Simulation Code* no menu do VBS.

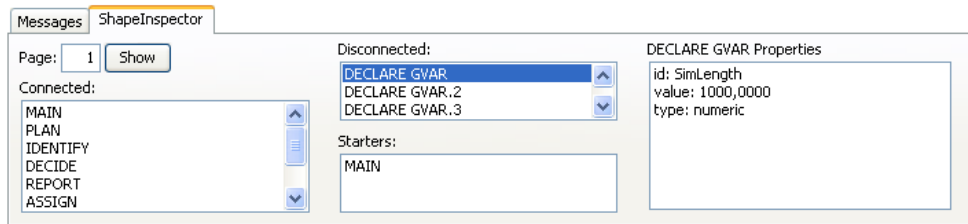
A geração automática de código, pela simples interpretação dos fluxogramas, poderá produzir código redundante. No entanto essa redundância poderá ser benéfica, ajudando o utilizador a melhor compreender a relação existente entre o fluxograma e o código gerado. Os utilizadores mais avançados poderão eliminar a redundância, activando para isso a opção **Compact Code**.

A aplicação inclui também uma espécie de depurador de erros, que permite verificar se existem erros no desenho dos diagramas ou erros de programação.



Por razões pedagógicas, a geração (e sua execução) está a ser efectuada mesmo que existam erros no desenho dos diagramas.

Por ilusão de óptica, podemos por vezes ser levados a pensar que duas formas estão ligadas quando efectivamente não estão. Nas situações em que o depurador acusar erros que possam aparecer desajustados, poderá recorrer ao ***ShapeInspector*** para uma análise mais aprofundada dos módulos e suas conexões.



Depois de detectadas e corrigidas as anomalias, o código do programa poderá ser actualizado, bastando para isso pressionar o botão ***Refresh Code***.

Apesar do código produzido pelo VBS, estar escrito na linguagem Visual Basic 2008, não é necessário recorrer a nenhum compilador para o executar! Convém referir que o código é interpretado e não compilado, razão pela qual o VBS não necessita de qualquer compilador. No entanto, se por qualquer razão desejar compilar o código, poderá fazê-lo, deverá para isso exportá-lo (***Export Code***) para o Visual Basic 2008 e compilá-lo através daí.

Para executar o código, através do VBS, deverá pressionar o botão ***Run Code*** e aguardar que a simulação termine. Uma vez terminada a simulação, e conforme o mecanismo do simulador, será apresentado um *Report* com as estatísticas da simulação e o histórico do planeamento de acontecimentos.

É possível exportar os valores das estatísticas para um ficheiro externo, de forma a poderem ser trabalhados numa outra aplicação, como por exemplo o Microsoft® Excel.

Para efectuar a exportação, deverá primeiro seleccionar o separador com o tipo de dados a exportar (*Statistics* ou *History*) e depois clicar no botão **Export**. Esta operação irá criar um ficheiro no formato CSV (Comma-Separated Values) que depois deverá ser importado através da aplicação externa (ex: Microsoft® Excel).

No histórico os acontecimentos encontram-se ordenados pela sua ordem de agendamento (*EventID*), no entanto é possível ordenar os acontecimentos por qualquer outra coluna. A coluna *ExecutedAt* refere-se ao instante de tempo em que este foi executado.

Simulation Report

Statistics History

Name	Type	IN	OUT	Now	AvgStay	AvgLen
q1	Queue	352	254	98	116,0118	40,8362
q2	Queue	211	199	12	28,5075	6,0151
b1	Queue	253	165	88	171,3394	43,3489
b2	Queue	198	165	33	77,1152	15,2688
b3	Queue	164	164	0	0	0
m1	Resource	254	253	1	3,9209	0,9959
m2	Resource	199	198	1	5	0,995
m3	Resource	165	164	1	6,0244	0,994
m4	Resource	164	163	1	1,9018	0,3119

Export (.csv) Close

Simulation Report

Statistics History

Event ID	EventName	ExecutedAt
1	Front_Unit_Arrival	0
2	Rear_Unit_Arrival	2
3	End_M1	2
4	Front_Unit_Arrival	2
5	Front_Unit_Arrival	4
6	End_M2	6
7	Rear_Unit_Arrival	4
8	End_M1	6
9	Front_Unit_Arrival	15

3.3 Open With Stencils

Sempre que quiser abrir um documento Visio, previamente construído no VBS, deverá fazê-lo através desta opção. Se o fizer de outra forma, os *stencils* poderão não ser carregados automaticamente!

4 Princípios e regras de desenho

O código do programa gerado assenta num paradigma procedimental e é composto por $N+2$ procedimentos, representando N o número de acontecimentos a simular. Além dos procedimentos que representam os acontecimentos da simulação, existem mais dois procedimentos, o principal (**Main**) que coordena todo o sistema, e o de inicialização de entidades (**Init**), que por incluírem toda a lógica de funcionamento do sistema, não deverão ser alterados!

As páginas do documento do Microsoft® Visio® representam os procedimentos do sistema - a primeira é normalmente ocupada pelo procedimento principal, e as restantes pelos acontecimentos da simulação. Todos os procedimentos deverão estar representados em páginas separadas, iniciar com o módulo **Event** (ou Main no caso do procedimento principal) e terminar com o módulo **Return**. Todos os módulos que estiverem compreendidos entre esses, representam o corpo do procedimento.

Todas as entidades que fazem parte da simulação (variáveis, acontecimentos, recursos e filas), antes de poderem ser utilizadas, deverão primeiro ser declaradas, através dos módulos criados para o efeito (Declare GVar, Declare Event, Declare Resource e Declare Queue). A declaração de entidades, por ser uma operação global a todo o sistema, deverá ser efectuada na página principal do documento. Nas outras páginas, apenas é possível declarar variáveis, que serão tratadas como sendo locais e não globais.

Ao contrário do que acontece com os outros módulos, os módulos de declaração de entidades (por não terem uma ordem de execução), deverão estar desligados de outros módulos.

Os módulos que desempenham operações lógicas (Decide, ResourceFree e QueueEmpty), deverão ter dois pontos de saída e para tal deverão usar ligadores lógicos, um do tipo True e outro do tipo False. A função de ligação automática do Visio não é aconselhada neste tipo de moldes, uma vez que esta não recorre aos decisores lógicos referidos.

5 Módulos

5.1 Main

Main

Indica o início do procedimento principal que comanda a simulação (Main). Este módulo não aceita parâmetros.

Event

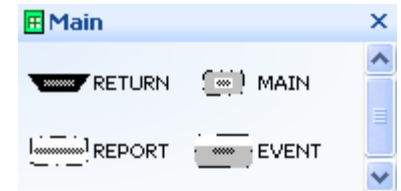
Indica o início do procedimento do tipo acontecimento. Este módulo aceita um parâmetro que é o nome do acontecimento.

Return

Indica o fim de um qualquer procedimento, seja o procedimento principal ou um procedimento do tipo acontecimento. Este módulo não aceita parâmetros.

Report

Mostra as estatísticas resultantes da execução da simulação. Este módulo não aceita parâmetros.



5.2 Event

DeclareEvent

Declara um novo acontecimento na simulação. Recebe um parâmetro que define o nome do acontecimento (que deverá ser único).

Plan

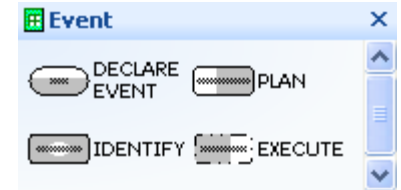
Agenda (planeia na lista de acontecimentos futuros) a execução de um determinado acontecimento. Recebe dois parâmetros, o nome do acontecimento a agendar e o instante de tempo em que este irá ocorrer.

Identify

Identifica, com base na lista de acontecimentos futuros, qual o próximo acontecimento a ocorrer. Recebe dois parâmetros que corresponderão ao valor de duas variáveis - EventName, com o nome do acontecimento identificado, e EvenTime, com o instante de tempo em que o acontecimento irá ocorrer.

Execute

Executa o próximo acontecimento na lista de acontecimentos futuros. Não recebe nenhum parâmetro.



5.3 Queue

DeclareQueue

Declara uma nova fila na simulação. Recebe um parâmetro que define o nome da fila (que deverá ser único).

Insert

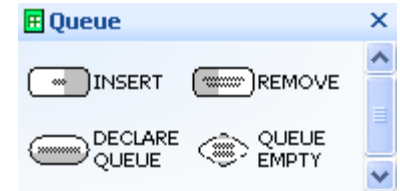
Insere um novo elemento numa fila. Recebe três parâmetros, o nome da fila, a sua prioridade (caso não exista deverá ser igual para todos), e o instante de tempo em que foi inserido. Os elementos serão ordenados pela prioridade e pelo instante de tempo de inserção na fila.

Remove

Remove de uma fila um elemento já existente. Recebe um parâmetro que identifica a fila de onde o elemento será eliminado.

QueueEmpty

Verifica se uma dada fila se encontra vazia. Recebe um parâmetro que indica qual a fila a analisar.



5.4 Resource

DeclareResource

Declara um novo recurso na simulação. Recebe dois parâmetros, o primeiro define o nome do recurso (que deverá ser único) e o segundo, o número de unidades disponíveis no sistema.

Seize

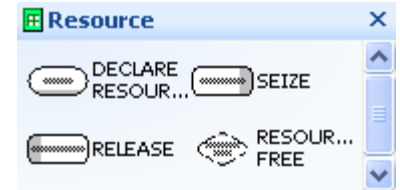
Ocupa um recurso durante um período de tempo especificado. Recebe dois parâmetros, o primeiro identifica o nome do recurso, e o segundo, o instante de tempo futuro em que o recurso será libertado.

Release

Liberta um recurso previamente ocupado. Recebe um parâmetro que identifica o recurso a libertar.

ResourceFree

Verifica se um dado recurso está disponível. Recebe um parâmetro que identifica o recurso a analisar.



5.5 Expert

DeclareGVar

Declara uma nova variável na simulação. Recebe três parâmetros, o primeiro define o nome da variável (que deverá ser único), o segundo define o valor predefinido, e o terceiro define o tipo de dados (numérico ou alfanumérico).

Decide

Verifica o valor lógico de uma dada expressão. Recebe três parâmetros, o primeiro argumento da expressão, o comparador lógico e o segundo argumento da expressão. Quando usar este decisor lógico, deverá usar os ligadores lógicos Arrow (T) e Arrow (F).

Assign

Permite atribuir valores absolutos a variáveis, ou valores resultantes de funções pré-definidas, como por exemplo valores aleatórios. Recebe até seis parâmetros, no entanto o número de valores obrigatório varia consoante a operação a executar. Dois parâmetros obrigatórios são o nome da variável onde o valor será armazenado, e o segundo parâmetro é o que indica a operação a executar. Não deverá esquecer (quando recorrer a estas operações), que a geração de números aleatórios necessita de uma variável semente.



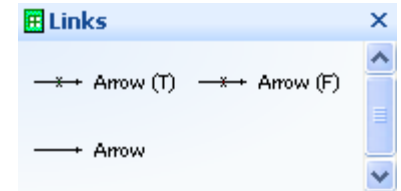
5.6 Links

Arrow

Este é o ligador que normalmente será utilizado para ligar os diferentes módulos - em substituição deste ligador poderá ser usado o ligador automático que o Microsoft® Visio® disponibiliza. Nenhum ligador recebe parâmetros.

Arrow (T) e Arrow (F)

Estes ligadores apenas deverão ser usados como ponto de saída de decisores lógicos. O primeiro define o caminho a executar quando a condição é verdadeira, e o segundo define o caminho quando a condição é falsa.

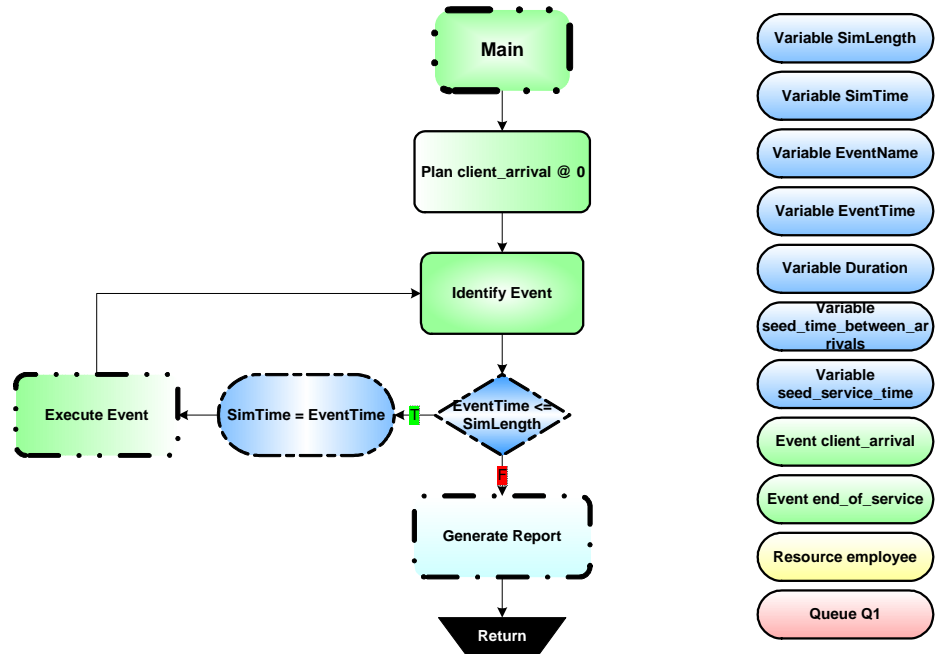


6 Exemplo

Para ilustrar os conceitos que neste manual têm vindo a ser expostos, a seguir será apresentado um exemplo de uma simulação (*BankExampleSimple*), que será seguido de uma breve explicação.

6.1 Motor do Sistema

A imagem ao lado ilustra a lógica de funcionamento de todo o sistema, bem como as variáveis, os acontecimentos, recursos e filas (no exemplo concreto apenas existe um recurso e uma fila), que asseguram o correcto funcionamento do mesmo.



O funcionamento do sistema é bastante simples e começa com o agendamento do primeiro acontecimento. A seguir é identificado qual o acontecimento a ser executado. Essa identificação devolve o nome do acontecimento (*EventName*) e o instante de tempo em que o mesmo irá ocorrer (*EventTime*).

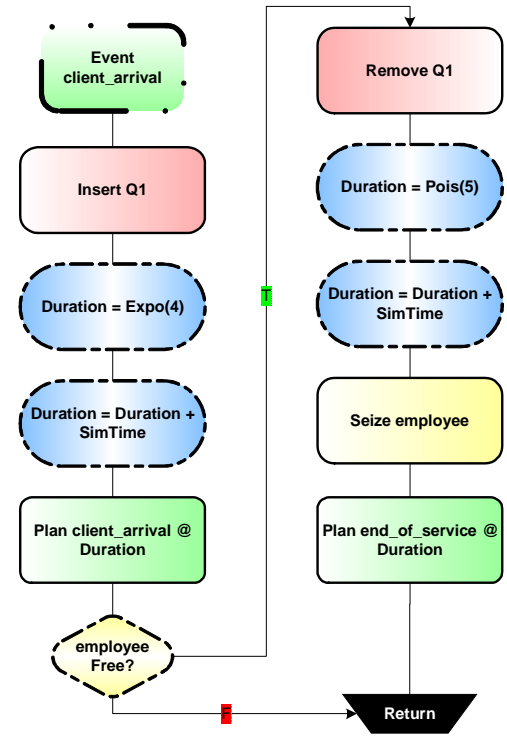
Se o acontecimento ocorrer antes da simulação terminar (*SimLength*), o relógio (*SimTime*) é actualizado, o acontecimento é executado, sendo depois identificado o próximo acontecimento a ser executado. Como se poderá constatar, este bloco descreve um ciclo que irá ocorrer até que seja atingido o fim da simulação. Se o acontecimento ocorrer depois do fim da simulação, é mostrado um quadro com as estatísticas da simulação, e a simulação termina.

6.2 Acontecimento ClientArrival

Este acontecimento representa o instante de tempo em que um qualquer cliente chega ao sistema e executa todas as tarefas correspondentes a essa alteração no estado do sistema, correspondendo sempre à lógica de funcionamento do sistema real a simular.

Assim, este acontecimento incluirá o planeamento da chegada do próximo cliente ao sistema, bem como a integração no sistema do cliente que acaba de chegar, isto é:

- Colocação do cliente em fila de espera se o servidor estiver ocupado
- Ocupação do servidor (com indicação do instante de tempo futuro em que o recurso será libertado), planeamento do acontecimento futuro fim de serviço (com o instante de tempo futuro em que este novo acontecimento ocorrerá), se o servidor estiver livre.



Nota: a inserção e eventual remoção do cliente na fila de espera ocorrem, na lógica deste acontecimento, na sequência exacta que permite ao sistema detectar a presença deste cliente, mesmo que não necessite de esperar pelo serviço na fila de espera. Esta questão será debatida nas aulas.

6.3 Acontecimento EndOfService

Quando este acontecimento é executado, o funcionário que anteriormente estava ocupado (a atender o cliente), passa agora a estar livre. No momento em que este é liberto, é verificado se existe algum cliente em fila de espera. Se existir algum cliente em espera, este é removido da fila de espera, o funcionário que estava livre passa a estar novamente ocupado, identificando-se o instante de tempo futuro em que será de novo libertado e agenda-se novo acontecimento fim de serviço a ocorrer nesse mesmo instante de tempo futuro.

