Instituto Tecnológico de Aeronáutica - Departamento de Pós Graduação

Engenharia Eletrônica e Computação

CE-235 Sistemas Embarcados de Tempo Real

Prova

1º. Bimestre

2º Semestre 2008

(Tipo Home Take Lab Take Test)

Aluno: Rafael Ferreira Conrado

Professor: Adílson Marques Cunha

Data:

15/09/2007

Versão 1.0

Prova do 1oBim CES-63 e CE-235 do Tipo *Take Home / Take Lab Test*

(1ª Parte –Warm Ups 1-7)

- Realizar os Tutoriais do Rational Rose Real Time – RRRT (Warm-Ups) 1, 2, 3, 4, 5, 6 e 7;
- Demonstrar a realização dos Tutoriais (Warm-Ups) 1, 2, 3, 4, 5, 6 e 7 no dia 15/09/08, das 13:30 às 16:30h, nos Módulos1 e/ou 4 do LABTECdaFCMF, para os Professores e Monitores designados para as Disciplinas; e
- Publicar, nas suas respectivas Homepages e na Ferramenta IBM – ClearCase do LABTEC da FCMF no ITA, até as 16:30h do dia 15/09/07, um Relatório comprobatório sobre a realização dos Tutoriais (Warm-Ups) 1, 2, 3, 4, 5, 6 e 7, em 2 itens:
- Relatando as principais funcionalidades do RRRT praticada sem cada um dos Tutoriais realizados; e
- Sintetizando a sua visão sobre a aplicação desses 7 (sete) Tutoriais (Warm-Ups) no contexto da sua Unidade de Software de Computador – USC, inserida no seu Componente de Software de Computador – CSC e a ser inserida mais tarde no seu Item de Configuração de Software de Computador – ICSC.

Warm-Up 1: Hello, World

O objetivo deste *Warm-Up* é criar um modelo contendo uma cápsula, na qual seu comportamento é escrever "Hello, World!" na tela.

Este *Warm-Up* é simples, porém fundamental para todos que nunca tiveram nenhum tipo de experiência com o conceito de cápsula e com a ferramenta *Rational Rose Real Time*.

As principais funcionalidades deste exercício são:

- Ele nos apresenta passo a passo cada um dos elementos fundamentais para a construção deste e dos demais Warm-Up, como a modelagem através do *Logical View*, *Componente View* e *Deployment View*.
- Assim como a criação de uma classe dinâmica (cápsula) que possui comportamento através do *State Diagram* e *Structure Diagram*;
- O uso de transações onde podemos inserir código em sua ação, no caso deste *Warm-Up* 1 foi inserido o código C++ responsável por mostrar a frase "Hello, World!" na tela.

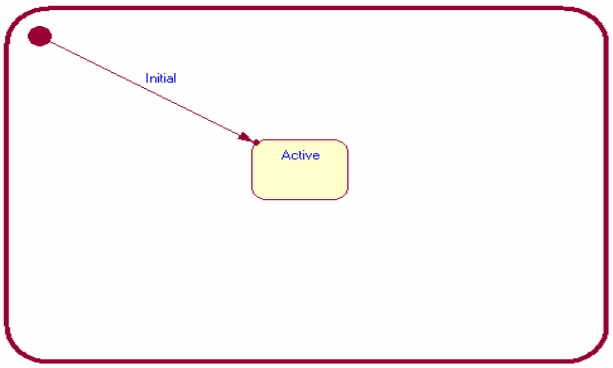

Figura 1.1 - Diagrama de Estado Warm-Up 1.

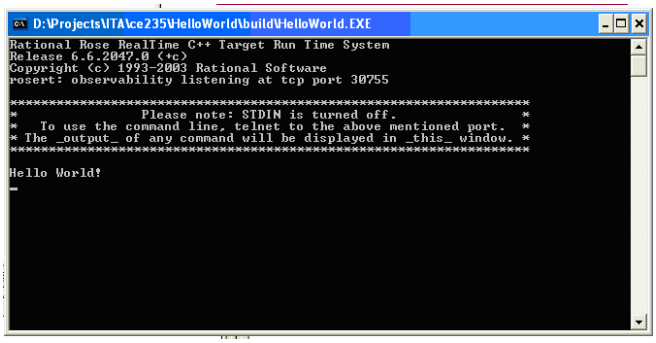

Figura 1.2 - Execução do Warm-Up 1.

Warm-Up 2: Passive Classes

O objetivo deste *Warm-Up* é criar um modelo contendo uma classe passiva, que não contém comportamento e escrever a frase "Greeting, Earthing" na tela.

As principais funcionalidades deste exercício são:

- Este Warm-Up tem a mesma finalidade do Warm-Up 1, porém devemos usar classes passivas com seus atributos e métodos, por isso criamos o método main() onde irá o código C++ que deverá escrever a frase "Greeting, Earthing" na tela, veja figura 2.2;
- Após executarmos com sucesso este *Warm-Up*, o tutorial nos mostra como podemos acessar o código gerado automaticamente pela ferramenta e fazermos alteração, veja figura 2.2;
- Assim que alteramos o código manualmente, compilamos usando o comando *NMAKE* na linha de comando no *prompt* do DOS, conforme figura 2.3;
- Executamos o código na linha de comando do *prompt* do DOS;
- E sincronizamos as mudanças feitas manualmente no código utilizando a ferramenta RRRT através da opção *Build* > *CodeSync* no *Component View*;

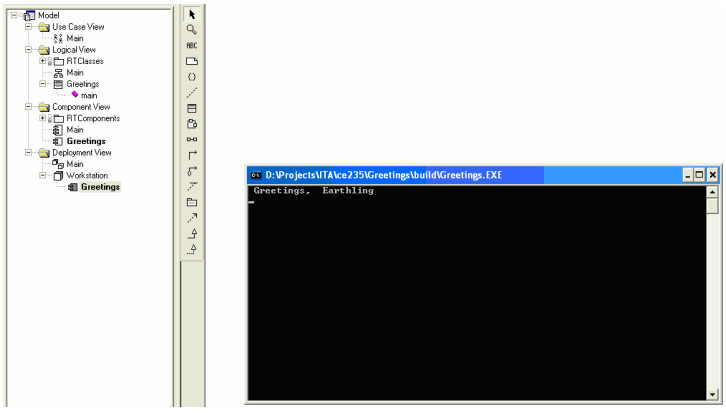

Figura 2.1 - Resultado do Warm-Up 2

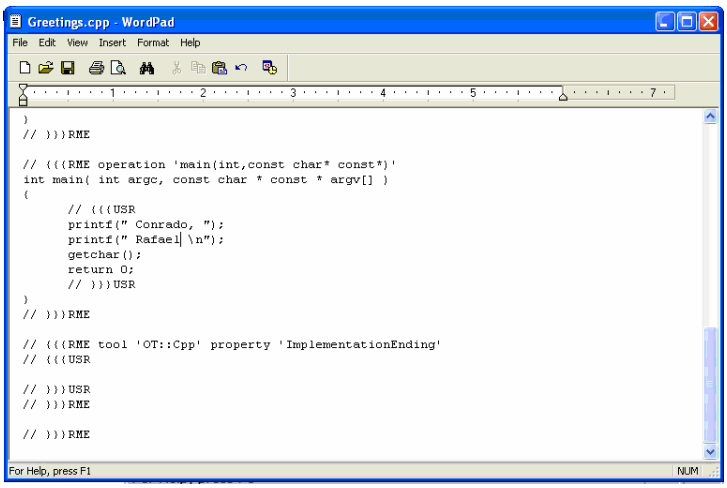

Figura 2.2 - Código C++ gerado automaticamente pela ferramenta

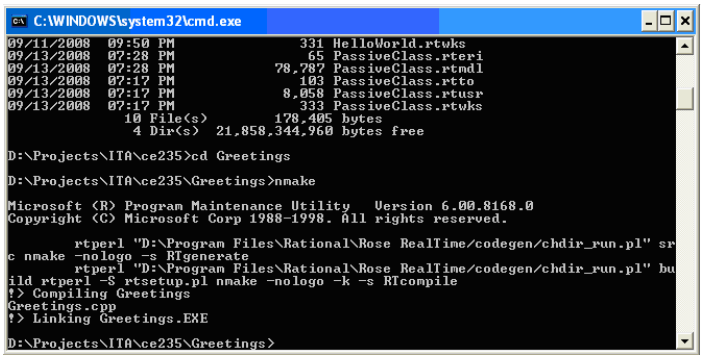

Figura 2.3 – Código compilado manualmente.

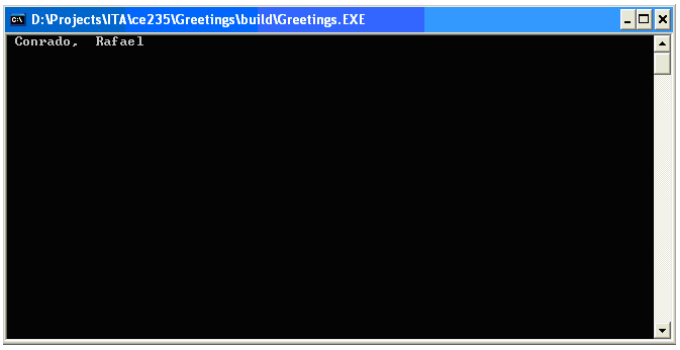

Figura 2.4 – Código executado manualmente.

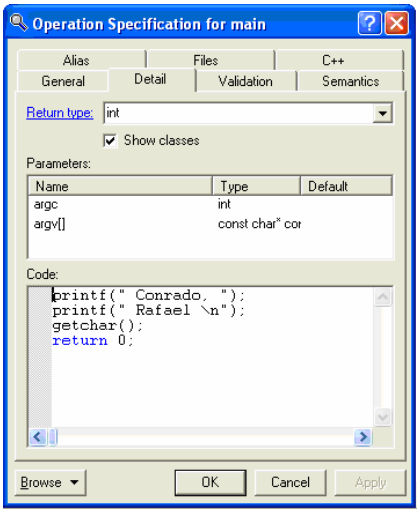

Figura 2.5 – Código sincronizado.

Warm-Up 3: Traffic Light

O objetivo deste *Warm-Up* é criar uma aplicação de sinal de trânsito. Também nos ensina o conceito de porta e máquina de estado finito com três estados (verde, amarelo e vermelho).

As principais funcionalidades deste exercício são:

- Construir e comparar nosso modelo e seu comportamento baseado no diagrama de seqüência mostrado no exercício.
- Explicar uma importante funcionalidade da ferramenta RRRT que é a possibilidade de simularmos a execução da aplicação fazendo um debug no diagrama de estado.
- Criarmos um protocolo de nome LightControl, que deverá conter os sinais de saída green, yellow e red;
- Criar uma cápsula TrafficLight que conterá uma porta final conjugada de nome Control para o protocolo LightControl no diagrama de estrutura;
- Inserimos transações com trigger entre os estados, que utilizam a porta Control e seus sinais de saída do protocolo LightControl;

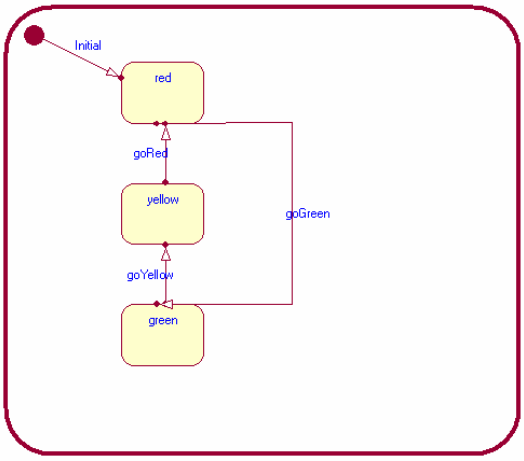

Figura 3.1 – Diagrama de estado do Sinal de Trânsito.

Execução do debug após inserirmos um ponto de prova que simula os sinais de entrada.

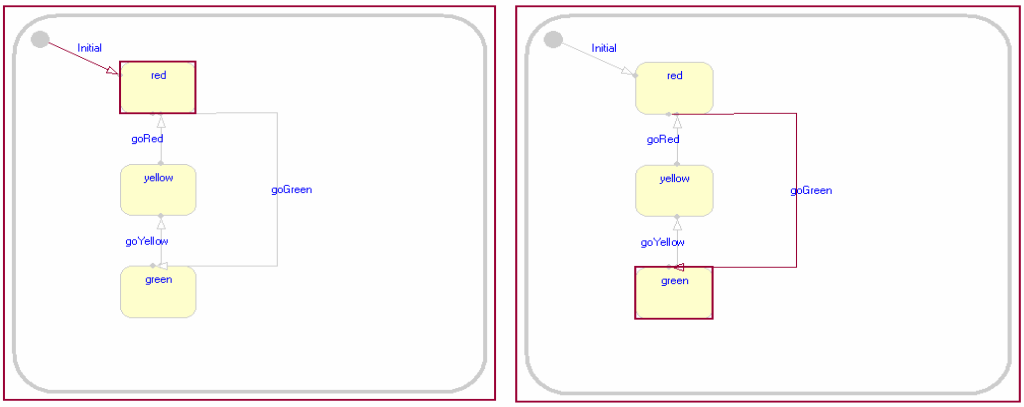

Figura 3.2 – Debug do diagrama de estado.

Warm-Up 4: Electronic Lock

O objetivo deste *Warm-Up* é modelar o comportamento de uma trava eletrônica. Ele é composto por uma classe passiva com uma máquina de estado que provê o comportamento e possui três estados: *Locked, Unlocking* e Unlocked.

O estado Unlocking espera pelo segundo caracter para fazer a combinação, mudando para o estado Locked ou Unlocked dependendo do caracter recebido usando o item ponto de escolha que nos é apresentado neste exercício.

As principais funcionalidades deste exercício são:

- Adicionar uma máquina de estados a uma classe passiva de nome EletronicLock;
- Criar uma operação de nome Number e tipo Trigger para esta máquina de estado, que recebe como parâmetro o tipo de operação realizado;
- Adicionar as condições através dos dos pontos de escolha, por exemplo, se o usuário pressionar 1, o sistema irá para o estado Unlocking e testará uma condição para o próximo valor selecionado;

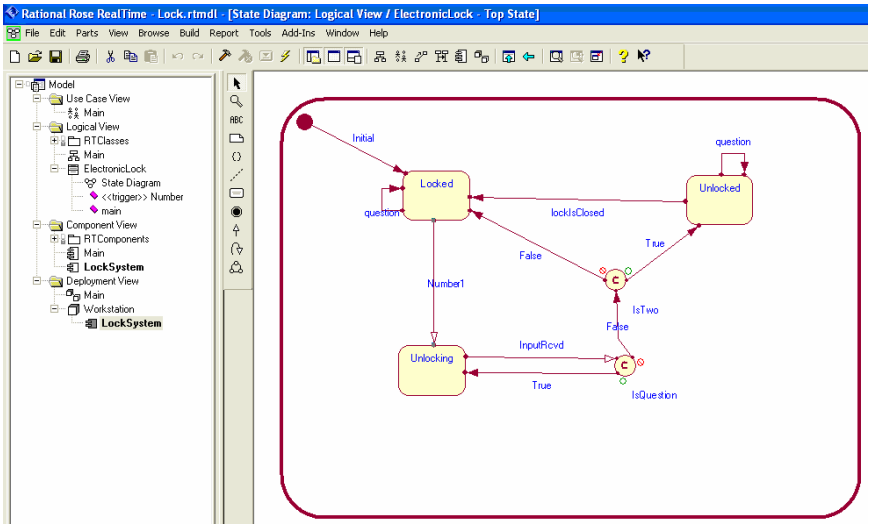

Figura 4.1 – Diagrama de estado de uma Trava Eletrônica.

Warm-Up 5: Battleship

Neste *Warm-Up* foi criada uma simulação de um Navio de guerra que envia sinais de sonar (*pings*) no oceano em intervalos regulares de tempo e detecta e mostra o sinal recebido (*echo*).

<u>As principais funcionalidades deste exercício são:</u>

- Este *Warm-Up* nos apresenta o serviço Timer que possibilita a execução automática dos sinais, através da bibliotecas de serviços;
- Criar uma nova cápsula chamada World que abrigará as demais cápsulas, sendo esta a *TopCapsule*;

- Por utilizarmos um serviço, temos que mostrar as dependências. No diagrama de classe *Main*, inserimos as classes *Battleship*, *Ocean*, and *RTTimespec* e adicionamos as dependências *Battleship* para *RTTimespec* e *Ocean* para *RTTimespec*;
- Também é criado um diagrama de seqüência em tempo de execução para compararmos com o diagrama apresentado no tutorial.

O nível de dificuldade deste *Warm-Up* é maior porque ele somente nos apresenta um diagrama de seqüência como base para a modelagem e diminui no passo a passo de todos os conceitos. Colocando a prova o conhecimento adquirido nos *Warm-Ups* anteriores.

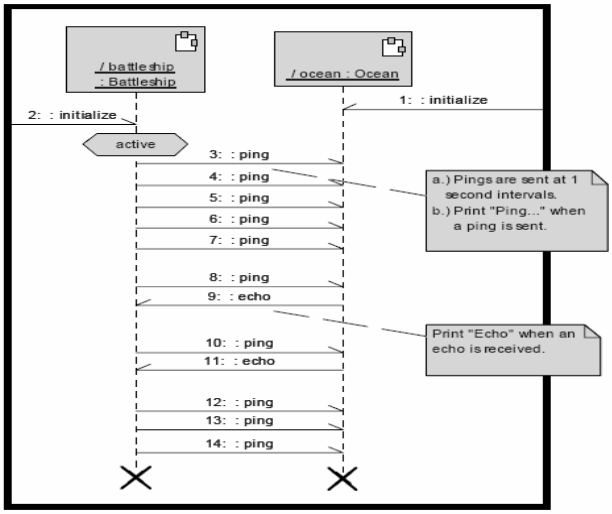

Figura 5.2 – Diagrama de seqüência apresentado no Warm-Up.

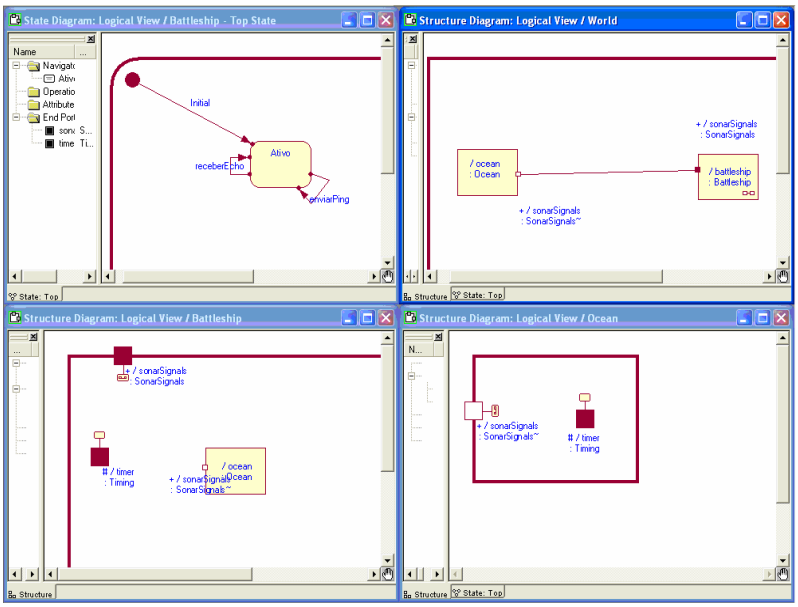

Figura 5.2 – Diagramas de estado e de estrutura do Battleship.

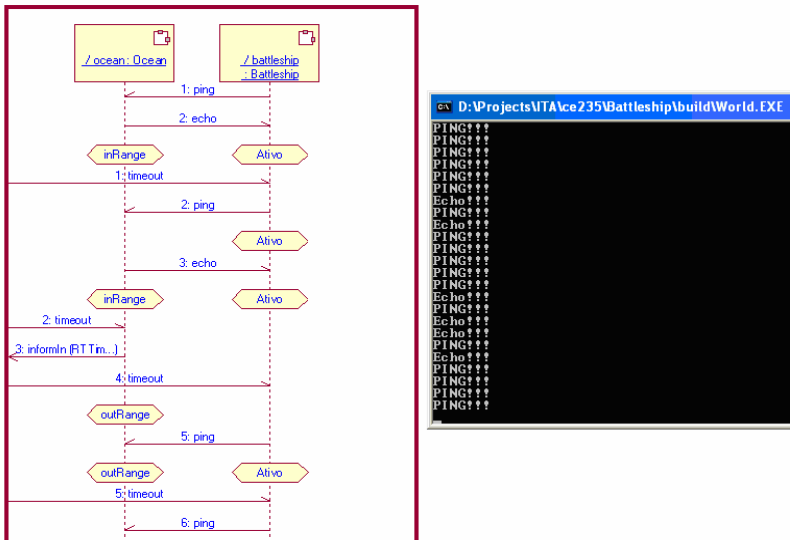

Figura 5.3 – Execução do Battleship e o diagrama de seqüência.

Warm-Up 6: Traffic System

Neste *Warm-Up* foi criada uma simulação de cruzamento de trânsito onde devemos modelar o sinal de trânsito para este cruzamento.

Este exercício utiliza o que já foi criado no *Warm-Up* 3 e nos desafia a criar um modelo em cima de outro modelo existente e seus elementos (classes e protocolos).

As principais funcionalidades deste exercício são:

- Foi criada uma cápsula denominada *Controller* que contém quatro *"capsule roles",* baseadas na cápsula *TrafficLight*, chamadas de norte, sul, leste e oeste.
- Adicionamos um *"container capsule"* chamada de *Intersection*, que contém uma *"capsule role"* baseada na *Controller* e novamente as quatro *"capsule role"* norte, sul, leste e oeste. Estas quatro cápsulas foram conectadas ao *Controller* que enviará os sinais para cada par norte/sul e leste/oeste.
- Quando o par norte/sul estiver verde o *Controller* enviará o sinal vermelho para o par leste/oeste, e vice-versa.

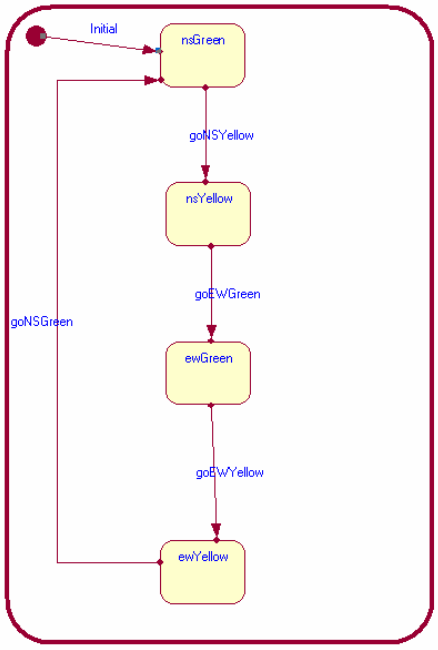

Figura 6.1 – Diagrama de estado da cápsula *Controller*.

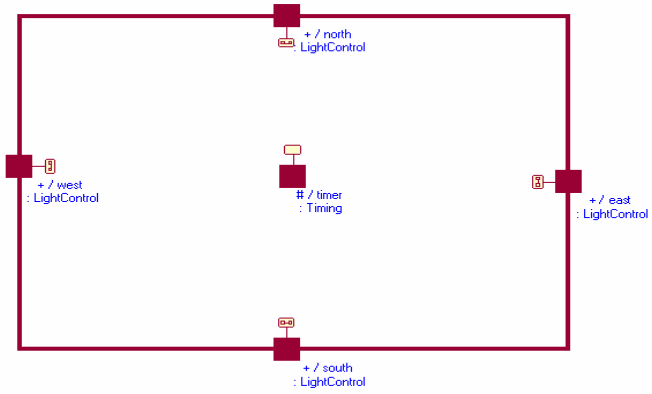

Figura 6.2 – Diagrama de estrutura da cápsula *Controller*.

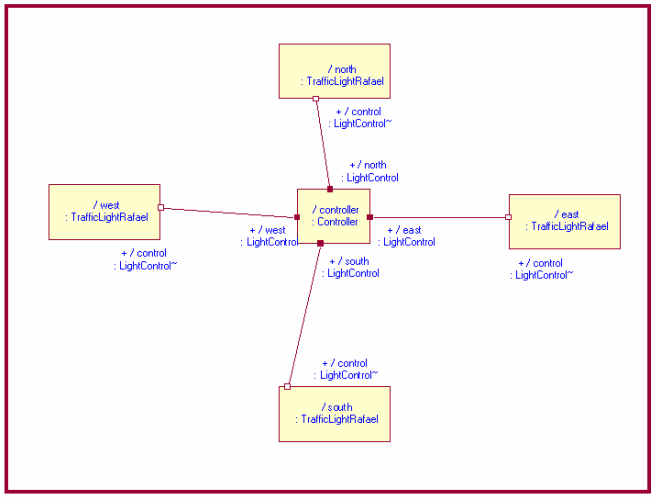

Figura 6.2 – Diagrama de estrutura da cápsula *Intersection.*

Figura 6.3 – Execução com simulação do *Traffic System.*

Warm-Up 7: Client/Server

Neste *Warm-Up* foram criadas duas aplicações do tipo Cliente/Servidor. Na primeira aplicação o cliente é dinamicamente criado pela cápsula do container em tempo de execução. Na segunda, o cliente é dinamicamente importado em tempo de execução.

As principais funcionalidades deste exercício são:

Primeira parte: Client será criado em tempo de execução.

- Configurar a cápsula *Client* de *Fixed* para *Optional*;
- Criar uma porta de nome *Frame* protegida e sem fio;
- Criar uma transação inicial no Diagrama de Estado da cápsula *TheSystem*, colocando uma ação onde a porta Frame cria a cápsula *Client* (*frame.incarnate(client)*);
- Após isso nós compilamos e testamos;

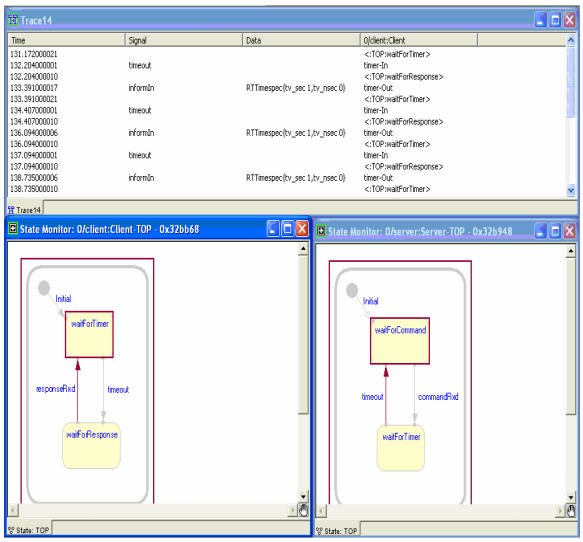

Figura 7.1 – Execução e teste da primeira parte do Client/Server.

Segunda parte: Client será importado em tempo de execução.

- Configurar a cápsula *Client PlugIn*;
- Criar uma nova cápsula chamada *realClient* do tipo *Client* configurada como *Fixed;*
- Mudar o código colocado na transação inicial no Diagrama de Estado da cápsula *TheSystem*, fazendo com que a cápsula *realClient* seja importada para cápsula *client*, *frame.import(realClient, client);*

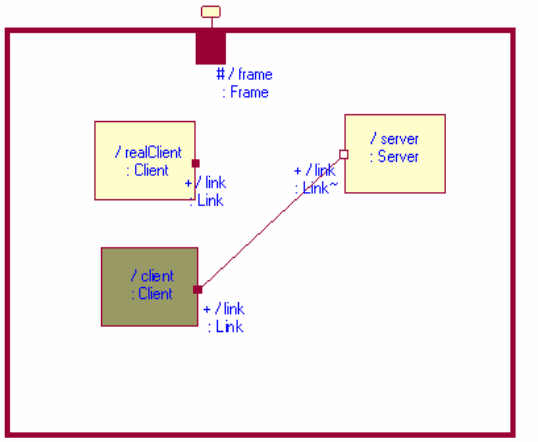

Figura 7.2 – Nova cápsula *Client* chamada *realClient.*

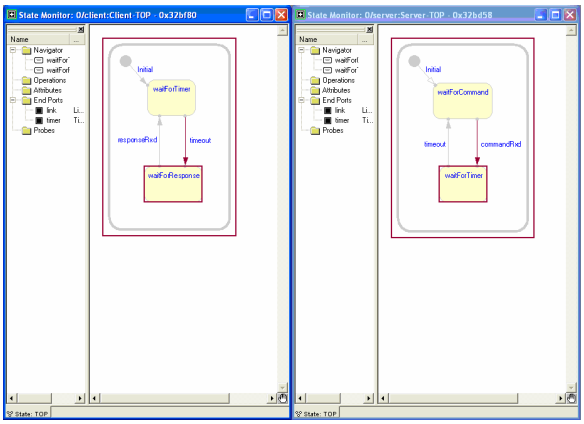

Figura 7.3 – Execução e teste da segunda parte do *Client/Server*.

Síntese dos 7 tutoriais na Unidade de Software de Computador – Visualização de Informações em Tempo Real (C-VTR)

A USC C-VTR deverá exibir informações em tempo real fornecidas pelas demais USC que compõem o Componente de Software de Computador (CSC), Gerenciamento do Monitoramento e Controle (GMC):

- Carga e Descarga de Dados Hidro/Operacionais (C-CDD);
- Interrogação da PCD (C-IDP);
- Configuração da PCD (C-CDP);
- Visualização de Informações em Tempo Real (C-VT).

Estas informações se constituem basicamente da visualização de dados operacionais, hidrológicos, meteorológicos e eventos de falha,

visando melhorar o acompanhamento do status de funcionamento da PCD em tempo real, e reduzir a defasagem da aquisição para o monitoramento.

A aplicação dos 7 tutoriais ao C-VTR se dará principalmente nas seguintes funcionalidades utilizados durante estes tutorias:

- Comunicação entre cápsulas através de protocolos comuns de entrada e de saída;
- Na utilização de transações com *triggers* e ações que mostrem as mensagens recebidas dos demais aplicativos;
- Assim como, a capacidade de esperar por um comando vindo do usuário, onde ele poderá estar solicitando algum tipo de informação que deseja visualizar naquele momento.

Em suma, estes Warm-Up são a base para conceitos importantes de sistemas embarcados em tempo real e nos ensina a utilizarmos as facilidades que a ferramenta *Case Rational Rose RealTime* nos proporciona, sem termos que escrever muitas linhas de código e aplicando o conceito de desenvolvimento a nível de modelo MDD (Model Design Development).