

Model-Driven Engineering

Model Simulation

Relatório



**Mestrado Integrado em Engenharia Informática e
Computação**

Engenharia de Software

Turma 1:

Henrique Manuel Martins Ferrolho - 201202772
João Filipe Figueiredo Pereira - 201104203

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

25 De Novembro de 2014

Resumo

Ao longo da última semana de aulas, o grupo encontrou-se a desenvolver uma ficha que nos permitia executar a simulação de uma máquina de estados através do programa Enterprise Architect.

A ficha teve como objetivo aplicar os conhecimentos adquiridos nas aulas teóricas num ficheiro exemplo fornecido pelos docentes da cadeira. Nesse ficheiro foi nos pedido para realizar um conjunto de alterações para nos alertar das diversas funcionalidades que se podem aplicar neste ambiente, onde é o grupo a elaborar e esboçar um possível projecto.

As questões pedidas foram implementadas com sucesso e permitiram ao grupo uma visão mais ampla do que o Enterprise Architect pode fornecer.

Índice

Resumo	2
Índice	3
1 Como foram implementadas as funcionalidades pedidas.....	4
a) Garantir que dois dígitos são sempre mostrados no campo das horas e dos segundos.....	4
b) Introduzir botões para parar e continuar o relógio	5
c) Representar as horas, minutos, segundos e AM/PM em campos separados (sempre com segundos visíveis)	6
d) Implementar o primeiro modo (set/hide seconds), através da manipulação da propriedade “Show” da “Gui” para mostrar segundos.....	7
2 Dificuldades encontradas e erros a implementar as alterações.....	9
3 O que aprendemos	10
4 Pontos fortes e fraquezas no modelo de simulação	11
5 Tempo dedicado por elemento	12

1 Como foram implementadas as funcionalidades pedidas

a) Garantir que dois dígitos são sempre mostrados no campo das horas e dos segundos

A implementação desta funcionalidade baseou-se em mostrar um “0” nos campos das horas e dos segundos quando qualquer um destes mostre um valor menor que “10”. Para tal, na secção “Time Display”, estado “Seconds”, método “Display”, opção “Behavior” (Time Display->Seconds->Display->Behavior), o grupo executou as seguintes alterações.

- **Segundos:**

```
if(sim.seconds <= 9) {  
    sim.secondsChar = "0" + sim.seconds;  
} else {  
    sim.secondsChar = "" + sim.seconds;  
}
```

- **Horas:**

```
if(sim.hours <= 9) {  
    sim.hoursChar = "0" + sim.hours;  
} else {  
    sim.hoursChar = "" + sim.hours;  
}
```

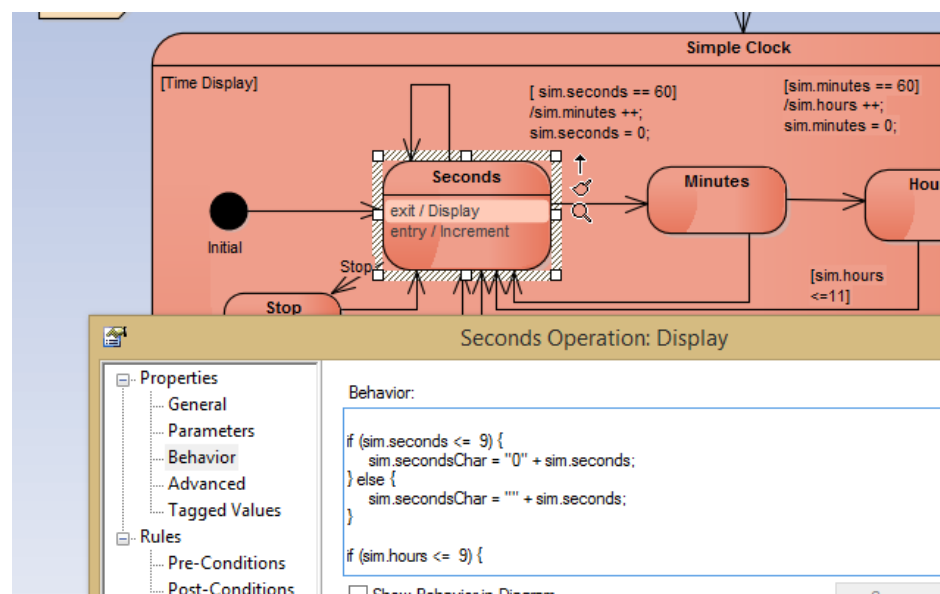


Figura 1 - Ilustração das alterações aplicadas no ficheiro

b) Introduzir botões para parar e continuar o relógio

Para satisfazer esta funcionalidade do relógio introduziu-se um novo estado, “Stop”, na máquina de estados “stm Clock” e ainda dois *triggers* cada um para representar as ações a tomar aquando das transições *stop* e *resume*. Foram adicionados dois “button” em “UIClockUserInterface”, “Stop” e “Resume”. Além de tudo isto, o grupo teve de adicionar dois *signals* ao *package* Clock, sendo eles denominados de “Stop” e “Resume”.

Para que estas componentes comunicassem entre si, o grupo teve de adicionar certas alterações que serão enumeradas de seguida:

- Ao se adicionar cada um dos botões na *user interface* é necessário seja emitido um sinal que servirá de interruptor para os *triggers* representados na máquina de estados. Para que os sinais fossem transmitidos o grupo teve de adicionar uma “Tag Value” na secção de “Tagged Values” em cada botão. O nome da *tag* terá de ser “OnClick” para que quando este botão for pressionado seja feita a operação definida em *Value* da *tag*, `BroadcastSignal(“nomeDoSinalCorrespondente”)`. Esta *value* emitirá um sinal que por sua vez irá ativar o trigger que o possuir.

Stop:

Nome: OnClick

Value: `BroadcastSignal(“Stop”);`

Resume:

Nome: OnClick

Value: `BroadcastSignal(“Resume”);`

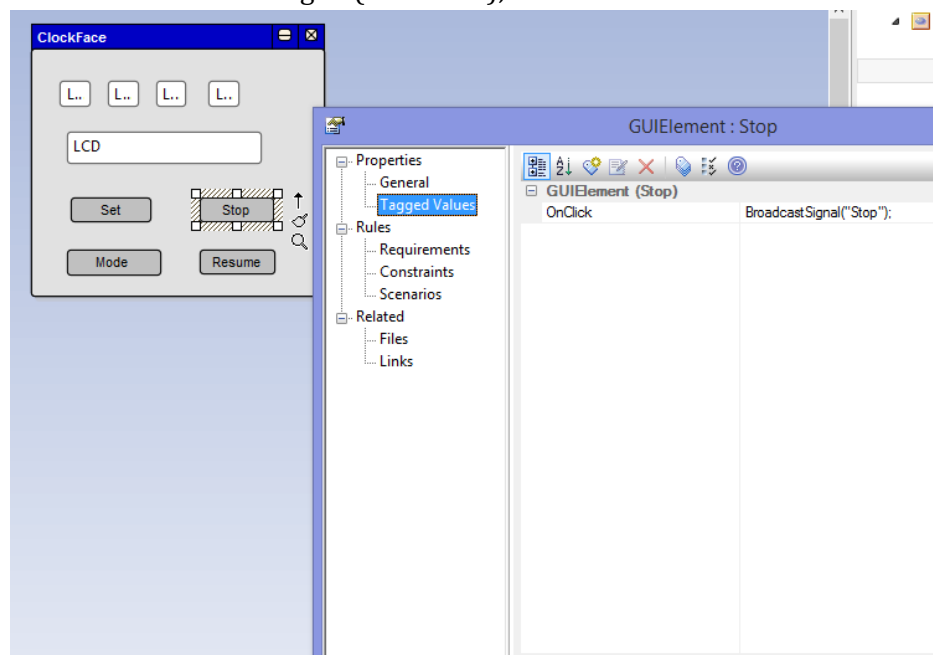


Figura 2 - Exemplo de aplicação em “Stop”

- Após aplicadas as acções nos botões da *user interface* é necessária assinalar no *trigger* o evento que este irá receber. Para isso bastaria ir apenas à secção Event e assinalar qual o sinal correspondente. No caso de “Stop” será o sinal “Stop” e no caso de “Resume” o sinal “Resume”.

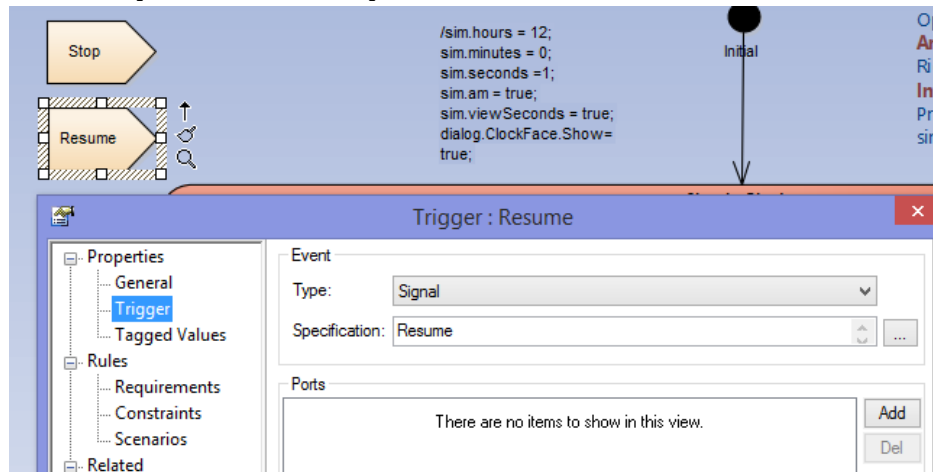


Figura 3 - Exemplo de aplicação de sinal a trigger

- Por fim, mas não menos importante é necessário as transições a efectuar. O grupo colocou duas transições, uma de “Seconds” para “Stop” denominada de “Stop”, e outra de “Stop” para “Seconds” denominada de “Resume”. Para que as transições sejam efetuadas elas devem estar associadas a um *trigger* que por sua vez se associa a um *signal*. A aplicação de como fazer isso está representada na imagem abaixo.

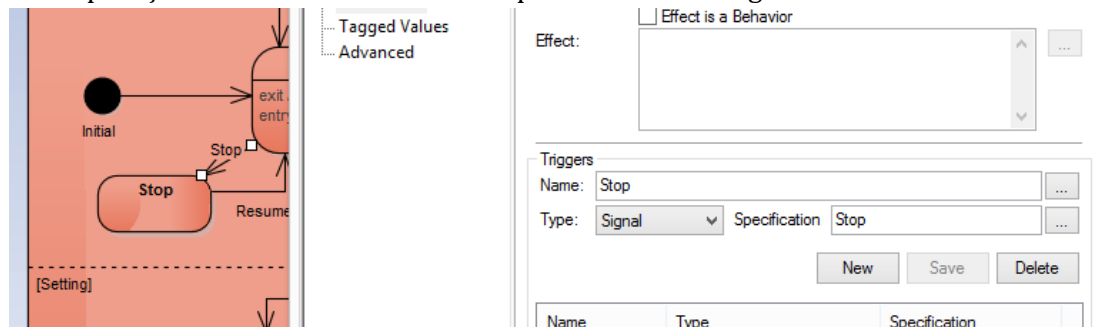


Figura 4 - Exemplo de como aplicar trigger a transição

c) Representar as horas, minutos, segundos e AM/PM em campos separados (sempre com segundos visíveis)

Um primeiro requisito para adicionar esta funcionalidade ao projeto era adicionar quatro objetos do tipo “Edit Control” à “UIClockUserInterface”. Estes objetos servem para a representação dos campos de tempo e terão de ser acedidos na secção “Behavior” do método “Display” do estado

“Seconds”, onde terá que se alterar o conteúdo de cada campo com a informação recebida das atualizações feitas ao tempo.

De seguida é demonstrado como é feita a adição de mais campos à *user interface*.

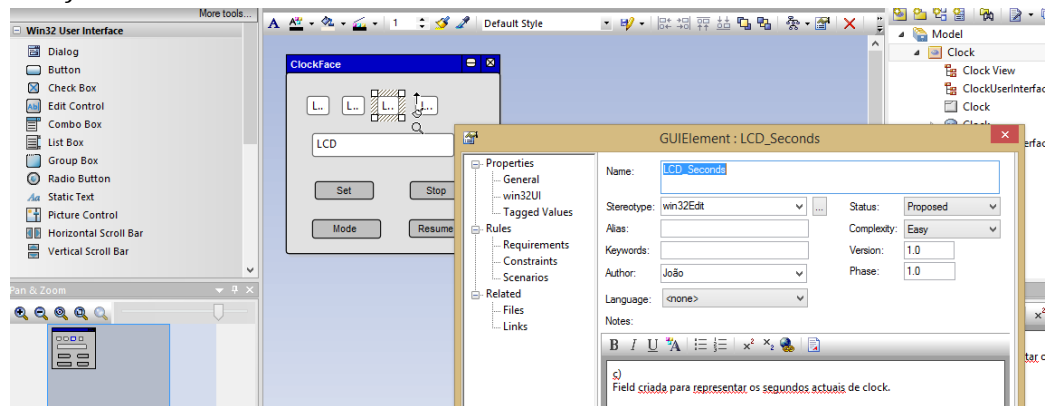


Figura 5 - Exemplo de inserção de campos

As alterações feitas na secção “Behavior” do método “Display” do estado “Seconds” estão anexadas abaixo:

```
if (sim.viewSeconds) {
    sim.lcd = sim.hoursChar + ":" + sim.minutesChar + ":" + sim.secondsChar +
    " " + sim.ampm;
    dialog.ClockFace.LCD.Text = sim.lcd;
    dialog.ClockFace.LCD_Hours.Text = sim.hoursChar;
    dialog.ClockFace.LCD_Minutes.Text = sim.minutesChar;
    dialog.ClockFace.LCD_Seconds.Text = sim.secondsChar;
    dialog.ClockFace.LCD_AM_PM.Text = sim.ampm;
} else {
    sim.lcd = sim.hoursChar + ":" + sim.minutesChar + " " + sim.ampm;
}
```

d) Implementar o primeiro modo (set/hide seconds), através da manipulação da propriedade “Show” da “Gui” para mostrar segundos

O objetivo desta funcionalidade consiste em esconder o campo dos segundos na *user interface* quando a sua visibilidade é alterada pelo utilizador. Na máquina de estado do Clock, na secção “Setting”, no estado “Set Display” existe uma transição para ele mesmo em que ao clicar no botão “Set” desativa o modo de visualização dos segundos na “GUI” de Clock.

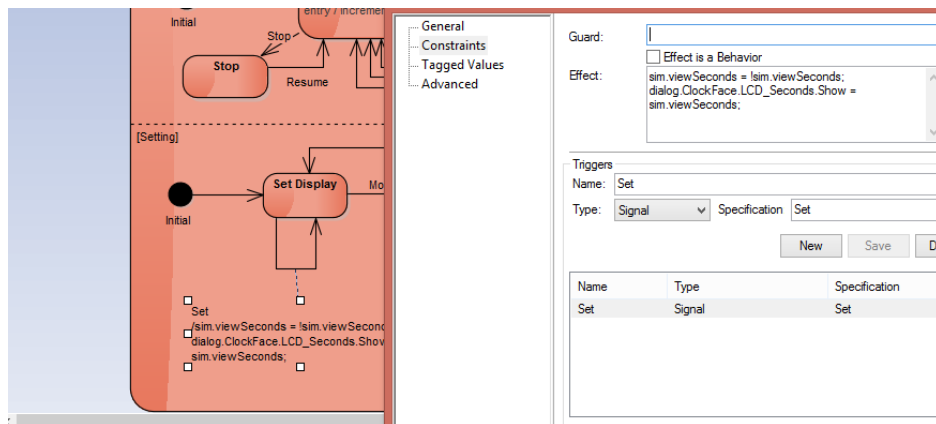


Figura 6 - Alteração feita no estado "Set Display"

Além desta pequena alteração, o grupo reestruturou o código do método "Behavior" de "Seconds", isto na secção "Time Display". Deixamos em anexo em baixo o resultado final:

```
sim.ampm = sim.am?"AM":"PM";
if (sim.minutes <= 9) {
    sim.minutesChar = "0" + sim.minutes;
} else {
    sim.minutesChar = "" + sim.minutes;
}

if (sim.seconds <= 9) {
    sim.secondsChar = "0" + sim.seconds;
} else {
    sim.secondsChar = "" + sim.seconds;
}

if (sim.hours <= 9) {
    sim.hoursChar = "0" + sim.hours;
} else {
    sim.hoursChar = "" + sim.hours;
}

if (sim.viewSeconds) {
    sim.lcd = sim.hoursChar + ":" + sim.minutesChar + ":" + sim.secondsChar + " " +
sim.ampm;
} else {
    sim.lcd = sim.hoursChar + ":" + sim.minutesChar + " " + sim.ampm;
}

dialog.ClockFace.LCD.Text = sim.lcd;
dialog.ClockFace.LCD_Hours.Text = sim.hoursChar;
dialog.ClockFace.LCD_Minutes.Text = sim.minutesChar;
dialog.ClockFace.LCD_Seconds.Text = sim.secondsChar;
dialog.ClockFace.LCD_AM_PM.Text = sim.ampm;
```


2 Dificuldades encontradas e erros a implementar as alterações

Ao longo da implementação dos vários requisitos não surgiram dificuldades, porém o grupo deparou-se com um erro que demorou bastante tempo a ser resolvido. Num dos campos adicionados o nome do campo era “LCD_AM/PM” e verificamos que os campos deixaram de mostrar as informações pedidas. O grupo com esforço descobriu o erro e verificou que os nomes não podem conter ‘/’ (barras).

Esta foi o único impasse que o grupo enfrentou e teve bastante tempo a resolver.

3 O que aprendemos

Neste modo de simulação, o grupo aprendeu como esquematizar os vários estados para uma aplicação, a representa-los em forma de máquina de estados e materializa-los numa interface. Essencialmente, é possível criar uma aplicação com bastante rapidez enquanto se faz a sua projeção com pouco esforço.

Com a realização desta simulação de um relógio o grupo interiorizou diversos aspetos relacionados com o tema.

Através do ficheiro exemplo o grupo teve um exemplo de criação de um *script* de inicialização e da sua configuração numa máquina de estados com uso de transições e operações a realizar em cada estado. O uso eficaz de *triggers* e a sua respetiva ativação através de sinais para a mudança de estados, sendo estes obtidos a partir de cliques nos botões da *interface* fornecida ao utilizador.

Embora já estivesse implementado, foi possível verificar como executar dois fluxos paralelos um ao outro na mesma máquina de estados (Time Display e Settings). Aprendemos a utilizar componentes Win32 para interação com o utilizador e devida associação com *triggers* e sinais.

O grupo pode afirmar uma grande produtividade e interiorização de conhecimentos após a realização desta ficha.

4 Pontos fortes e fraquezas no modelo de simulação

Os pontos fortes desta capacidade são a rápida criação, eficiência de simulação, pragmatismo, organização e funcionalidades básicas e coerentes.

Os pontos fracos neste tipo de modelo é que possui bastantes limitações, não permitindo explorar para situações mais complexas e extensas.

5 Tempo dedicado por elemento

O tempo dedicado a esta ficha por cada elemento foi similar, visto que a ficha foi realizada em conjunto, sempre com a presença de ambos os elementos.

Assim sendo, o elemento Henrique Ferrolho disponibilizou duas horas, e o elemento João Pereira disponibilizou duas horas também.