



Tópicos Avançados em Engenharia de Software

02/10/2012

Prof^o. Mario Godoy

mario.godoy@univasf.edu.br

www.univasf.edu.br/~mario.godoy/gis

Agenda

- a) Questionário grupo;
- b) Manic Time (*standard – free*): **USB -> Tools -> Backup;**
- c) NCL – parâmetro;
- d) Exercícios (5ª).

Questionário

Manic Time (*standard – free*):

<http://www.univasf.edu.br/~mario.godoy/ManicTimeUsb.zip>

USB -> Tools -> Backup

DESCRITORES

NCL

Estrutura de um Documento NCL

- ❑ Região **<regionBase>** - (onde)
- ❑ Descritores **<descriptorBase>** - (como)
- ❑ Conectores **<connectorBase>** - (comportamento dos links)

- ❑ Corpo do programa **<body>** são definidos os seguintes elementos:
 - ❑ Portas **<port>** – (quando)
 - ❑ Contextos necessários **<context>**
 - ❑ Nós de mídia **<media>** - (o que)
 - ❑ Elos **<link>** – (quando)

NCL

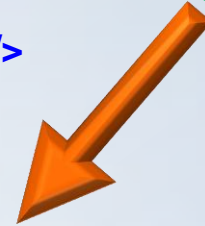
```
<head>
  <regionBase>
    <region id="regiaoFundo" width="100%" height="100%" />
  </regionBase>

  {
    <descriptorBase>
      <descriptor id="descRegiaoFundo" region="regiaoFundo"/>
    </descriptorBase>
  }

  <connectorBase>
    ...
  </connectorBase>
</head>

<body>
  <port id="plnicio" component="imgBart"/>
  <media id="imgBart" src="media/bart.gif" descriptor=" descRegiaoFundo " />
  <link>
    ...
  </link>
</body>
```

Controla a apresentação
do objeto de mídia?





NCL

Parâmetro de Descritor

O uso de parâmetros de descritor promove um alto grau de flexibilidade.

descriptorParam

descriptorParam: define um parâmetro do descritor como um par <propriedade, valor>. As propriedades e seus respectivos valores dependem do programa de exibição da mídia associada ao descritor.



Cada descritor pode conter diversos elementos **descriptorParam**

```
<descriptorParam name="nome_do_parametro" value="valor_do_parametro" />
```

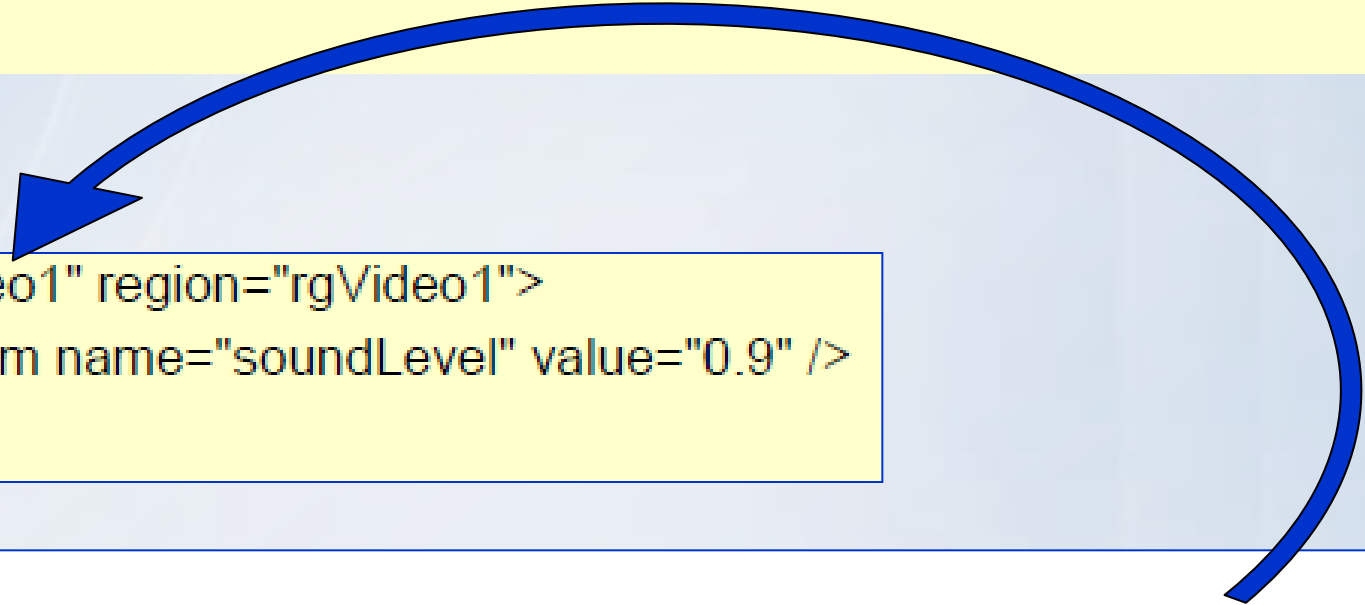
descriptorParam

Exemplo:

para indicar que a mídia correspondente deve ser reproduzida com o volume a 90% do máximo:

```
<descriptor id="dVideo1" region="rgVideo1">  
  <descriptorParam name="soundLevel" value="0.9" />  
</descriptor>
```

```
<body>  
  <media id="video1" type="video/mpeg" srcmedia/video1.mpg" descriptor="dVideo1"/>  
  ...  
</body>
```



descriptorParam – Tipos de Parâmetros



Dois descritores distintos reusando a mesma região.

```
<descriptor id="dVideo1" region="rgVideo1" />  
↕  
<descriptor id="dVideo2" region="rgVideo1">  
  <descriptorParam name="visible" value="false" />  
  <descriptorParam name="soundLevel" value="0"/>  
</descriptor>
```

descriptorParam – Tipos de Parâmetros



Dois descritores distintos reusando a mesma região.

```
<media type="video/mpeg" id="video1" src="media/video1.mpg" descriptor="dVideo1">  
  <property name="visible" />  
  <property name="soundLevel" />  
</media>
```



```
<media type="video/mpeg" id="video2" src="media/video2.mpg" descriptor="dVideo2">  
  <property name="visible" />  
  <property name="soundLevel" />  
</media>
```

descriptorParam

Tipos de Parâmetros

descriptorParam – Tipos de Parâmetros

Tabela 2. Parâmetros que podem ser utilizados em descriptorParam, conforme a mídia.

tipo de mídia	parâmetro	descrição
objetos com áudio	soundLevel, balanceLevel, trebleLevel, bassLevel	valores entre 0 e 1 . No caso de soundLevel, 0 = mute; 0.5 = volume a 50%; e 1 = volume máximo)

descriptorParam – Tipos de Parâmetros

objetos visuais	location	posição do objeto de mídia. Trata-se de dois números separados por vírgula, na ordem <left, top> num dos seguintes formatos: a) números reais entre 0 e 100, seguidos do sinal de porcentagem; ou b) números inteiros não negativos que especifiquem um valor em <i>pixels</i> .
objetos visuais	size	dimensões do objeto de mídia. Trata-se de dois números separados por vírgula, na ordem <width, height>, num dos seguintes formatos: a) números reais entre 0 e 100, seguidos do sinal de porcentagem; ou b) números inteiros não negativos que especifiquem um valor em <i>pixels</i> .

descriptorParam – Tipos de Parâmetros

objetos visuais	size	<p>dimensões do objeto de mídia. Trata-se de dois números separados por vírgula, na ordem <width, height>, num dos seguintes formatos:</p> <p>a) números reais entre 0 e 100, seguidos do sinal de porcentagem; ou</p> <p>b) números inteiros não negativos que especifiquem um valor em <i>pixels</i>.</p>
objetos visuais	bounds	<p>posição e dimensões do objeto de mídia. Trata-se de quatro números separados por vírgula, na ordem <left, top, width, height>, num dos seguintes formatos:</p> <p>a) números reais entre 0 e 100, seguidos do sinal de porcentagem; ou</p> <p>b) números inteiros não negativos que especifiquem um valor em <i>pixels</i>.</p>

descriptorParam – Tipos de Parâmetros

objetos visuais	background	nomes de cores reservados: <i>white, black, silver, gray, red, maroon, fuchsia, purple, lime, green, yellow, olive, blue, navy, aqua</i> , ou <i>teal</i> . Também pode ser <i>transparent</i> , para o caso de imagens com transparência, como alguns GIFs.
objetos visuais	visible	<i>true</i> ou <i>false</i>
objetos visuais	transparency	número real entre 0 e 1 indicando transparência: 0 significa totalmente opaco e 1 significa totalmente transparente

descriptorParam – Tipos de Parâmetros



objetos
visuais

fit

um dos seguintes valores: *fill*, *hidden*, *meet*, *meetBest* ou *slice*,
onde:

- *fill* = redimensiona o conteúdo do objeto de mídia para que toque todas as bordas da região;
- *hidden*: se a altura intrínseca ao conteúdo da mídia for menor que o atributo **height**, o objeto precisa ser renderizado a partir do topo e ter sua altura restante preenchida com a cor de **background**; caso seja maior, o restante deve ser cortado. Idem para a largura e esquerda.
- *meet* = redimensiona o conteúdo do objeto de mídia mantendo suas proporções até atingir uma das bordas da região. Caso haja um espaço vazio à direita ou na parte de baixo, deve ser preenchido com a cor de **background**.
- *meetBest* = semelhante ao *meet*, mas o objeto de mídia não é ampliado em mais do que o dobro das dimensões originais
- *slice* = redimensiona o conteúdo do objeto de mídia mantendo suas proporções até que toda a região seja preenchida. Parte do conteúdo pode ser cortado à direita ou na parte de baixo do conteúdo.

descriptorParam – Tipos de Parâmetros



objetos visuais	scroll	um dos seguintes valores: <i>none</i> , <i>horizontal</i> , <i>vertical</i> , <i>both</i> ou <i>automatic</i>
-----------------	---------------	---------------------------------------------------------------------------------------------------------------

objetos visuais	style	localização de um arquivo de folha de estilo
-----------------	--------------	----------------------------------------------

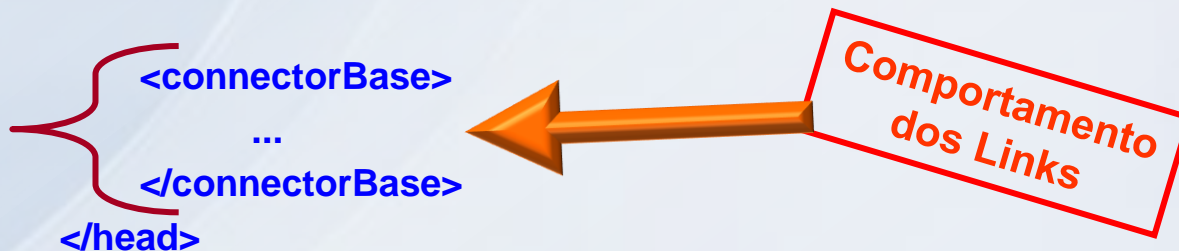
Conectores

Comportamento dos Links

NCL

```
<head>  
  <regionBase>  
    <region id="regiaoFundo" width="100%" height="100%" />  
  </regionBase>  
  
  <descriptorBase>  
    <descriptor id="descRegiaoFundo" region="regiaoFundo"/>  
  </descriptorBase>
```

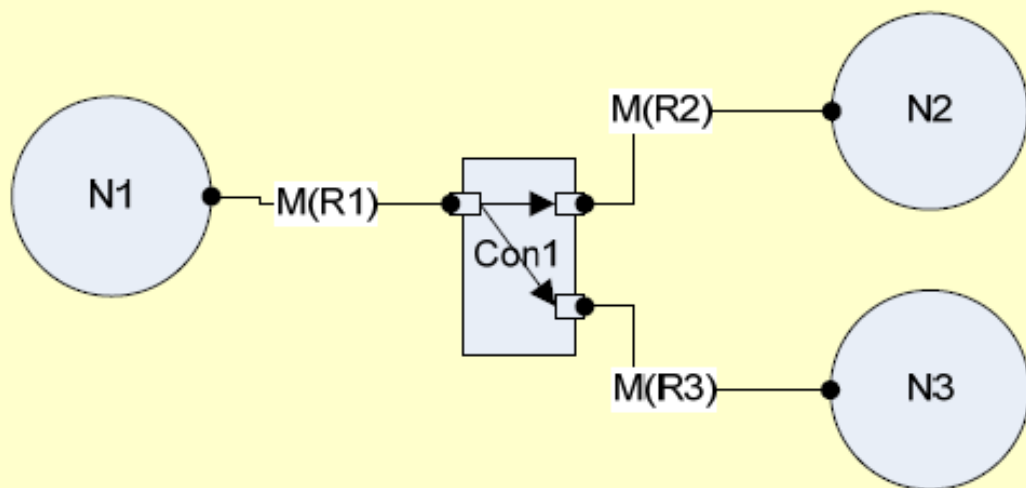
```
  <connectorBase>  
    ...  
  </connectorBase>  
</head>
```



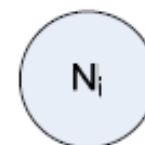
Comportamento dos Links

```
<body>  
  <port id="plnicio" component="imgBart"/>  
  <media id="imgBart" src="media/bart.gif" descriptor=" descRegiaoFundo " />  
  <link>  
    ...  
  </link>  
</body>
```

Conectores (básico)



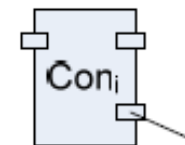
Legenda:



nó de mídia ou
conteúdo



mapeamento M
(papel R_i)



conector

papel de cada nó
relacionado

Figura 56. Ilustração de um conector ligando três nós.

Conectores

onBeginSetStartN

Conectores (básico)

Para manipular as propriedades **top**, **left**, **width** e **height** da região à qual o vídeo está associado, é necessário definir explicitamente essas propriedades como âncoras da mídia. A versão atual do Composer não permite definir âncoras de propriedade.

No caso específico das propriedades desse exemplo, pode-se especificar uma âncora para a propriedade **bounds** para manipulá-las em conjunto, na ordem **left, top, width, height**:

```
<media type="video/mpeg" id="video1" src="media/video1.mpg" descriptor="dVideo1">  
  <!-- âncora no vídeo que deve sincronizar a imagem -->  
  <area id="aVideo1Imagem1" begin="3s" end="8s"/>  
  <!-- propriedade que será manipulada pelos elos -->  
  <property name="bounds" />  
</media>
```


Conectores (básico)

- Ação:
- altera propriedades dos nós associados ao papel *set*, conforme os valores passados pelo mapeamento (**bindParam**) para o parâmetro do conector (**connectorParam**) denominado *var*;
 - inicia a apresentação da(s) mídia(s) no papel *start*.

Código

```
<causalConnector id="onBeginSetStartN">
```

```
  <connectorParam name="var"/>
```

```
  <simpleCondition role="onBegin"/>
```

```
  <compoundAction operator="seq">
```

```
    <simpleAction role="set" value="$var"/>
```

```
    <simpleAction role="start" max="unbounded" qualifier="par"/>
```

```
  </compoundAction>
```

```
</causalConnector>
```

Observação: Os elos que utilizarem este conector deverão identificar, como parâmetro adicional do mapeamento do nó (**bindParam**), as novas coordenadas e dimensões (x,y,w,h). Por exemplo:

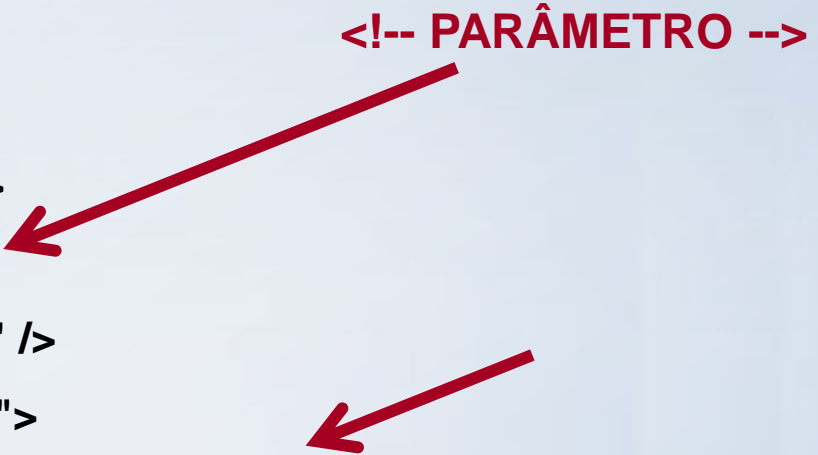
```
<bind component="video1" interface="bounds" role="set">
```

```
  <bindParam name="var" value="0,0,100%,100%" />
```

```
</bind>
```

Conectores (básico)

```
<!-- PARÂMETRO -->  
<causalConnector id="onBeginSetStartN">  
  <connectorParam name="var"/>  
  <simpleCondition role="onBegin" />  
  <compoundAction operator="par">  
    <simpleAction role="set" value="$var" />  
    <simpleAction role="start" max="unbounded" qualifier="par"/>  
  </compoundAction>  
</causalConnector>
```



Conectores (básico)

```
<media type="video/mpeg" id="video1" src="media/video1.mpg" descriptor="dVideo1">  
  <!-- âncora no vídeo que deve sincronizar a imagem -->  
  <area id="aVideo1Imagem1" begin="3s" end="8s"/>  
  <!-- propriedade que será manipulada pelos elos -->  
  <property name="bounds" />  
</media>
```

A manipulação é feita através dos elos *Video1Imagem1_start* e *Video1Imagem1_stop*:

```
<!-- âncora do vídeo 1 deve iniciar a exibição da imagem -->  
<link id="Video1Imagem1_start" xconnector="meusConectores#onBeginSetStartN">  
  <bind component="video1" interface="aVideo1Imagem1" role="onBegin" />  
  <bind component="video1" interface="bounds" role="set">  
    <bindParam name="bounds" value="0,0,50%,50%" />  
  </bind>  
  <bind component="imagem1" role="start" />  
</link>
```

Conectores

onKeySelectionStarNStopNAbortN

Conectores (básico)



Código

```
<causalConnector id="onKeySelectionStartNStopNAbortN">
```

```
  <connectorParam name="keyCode" />
```

NCL:

```
  <simpleCondition role="onSelection" key="$keyCode"/>
```

```
  <compoundAction operator="par">
```

```
    <simpleAction role="start" max="unbounded" qualifier="par"/>
```

```
    <simpleAction role="stop" max="unbounded" qualifier="par"/>
```

```
    <simpleAction role="abort" max="unbounded" qualifier="par"/>
```

```
  </compoundAction>
```

```
</causalConnector>
```

Leitura:

Quando a tecla <keyCode> for selecionada, inicia as mídias <start>, pára as mídias <stop> e aborta as mídias <abort>.

Observação:

Os elos que utilizarem este conector deverão identificar, como parâmetro adicional da ligação com o nó de origem (**bindParam** ou **linkParam**), o código virtual da tecla do controle remoto associada à seleção, através do parâmetro **keyCode**. Por exemplo:

```
<bindParam name="keyCode" value="GREEN" />
```

Conectores (básico)



Os códigos de teclas definidos no formatador atual são:

- RED, GREEN, YELLOW, BLUE
- VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN
- VK_ENTER
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 0
- MENU
- e outros dependentes do dispositivo de entrada: INTERACTION, *, #, letras A-Z

Link

e a passagem de parâmetro

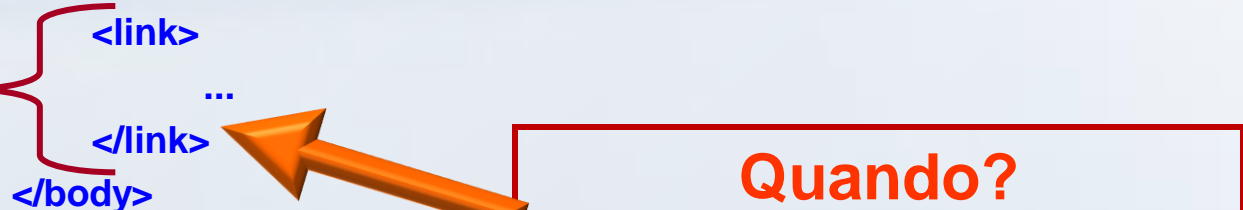
NCL

```
<head>
  <regionBase>
    <region id="regiaoFundo" width="100%" height="100%" />
  </regionBase>

  <descriptorBase>
    <descriptor id="descRegiaoFundo" region="regiaoFundo"/>
  </descriptorBase>

  <connectorBase>
    ...
  </connectorBase>
</head>

<body>
  <port id="plnicio" component="imgBart"/>
  <media id="imgBart" src="media/bart.gif" descriptor=" descRegiaoFundo " />
  <link>
    ...
  </link>
</body>
```



Link e o descriptorParam

A NCL define os seguintes elementos contidos num elemento de elo:

- **linkParam**: define um parâmetro do elo como um par <propriedade, valor>. As propriedades e seus respectivos valores dependem da definição do conector ao qual o elo está associado. Um elo pode conter diversos elementos **linkParam**.
- **bind**: indica um componente (**component**, nó de mídia ou de contexto) envolvido no elo, indicando o seu papel (**role**) no elo, conforme a semântica do conector. Em alguns casos deve-se indicar também o ponto de interface (**interface**)
- **bindParam**: define um parâmetro específico do *bind* como um par <propriedade, valor>. As propriedades e seus respectivos valores dependem da definição do conector ao qual o elo está associado.

Exercícios

Manic Time (*standard – free*):

USB -> Tools -> Backup;

Exercício (entregar 5ª, 04/10, antes da aula)

Perguntas sobre os exercícios (5ª).

Exercício 1



Exemplo 08 – Alternando entre mídias em resposta a ações do usuário

O objetivo deste exemplo é permitir ao usuário alternar entre dois vídeos, através da seleção das teclas vermelha e verde do controle remoto da TV digital, como em uma mudança de câmera em uma partida de futebol, por exemplo. Ambos os vídeos devem ser apresentados na mesma posição da tela. Para alternar entre os vídeos, algumas modificações precisam ser feitas com relação ao Exemplo 03.

Os dois vídeos devem ser iniciados de forma sincronizada, sendo que um deles deve iniciar invisível e sem som. Quando o usuário fizer uma seleção com os botões do controle remoto, os valores das propriedades dos dois vídeos devem ser invertidos.

Nesse exemplo, os dois botões ficarão visíveis o tempo todo. Isso é uma decisão ruim de projeto, pois acionar um dos botões não terá efeito nenhum. No exemplo 09 resolveremos este problema: somente o botão que trocar o vídeo atual será exibido.



Exercício 2



Exemplo 11 – Simulação de um menu de DVD com feedback

No exemplo anterior, ao selecionar um idioma, o usuário não recebe qualquer *feedback* imediato sobre qual opção foi selecionada. Ele precisa esperar o início da reprodução do áudio, após o *video2*, para saber se fez a seleção correta. Como boa prática de projeto de programa audiovisual interativo, deve-se fornecer um *feedback* para toda ação do usuário. Uma forma de se fazer isso é ocultar momentaneamente as opções que não foram selecionadas, deixando apenas a seleção do usuário visível, mesmo que por uma fração de segundo. Para atingir esse efeito, pode-se definir um atributo *delay* no mapeamento de ativação dos papéis do conector.





Obrigado!

www.univasf.edu.br/~mario.godoy/gis

Referências

- ❑ www.telemidia.puc-rio.br
- ❑ www.ncl.org.br
- ❑ <http://www.univasf.edu.br/~mario.godoy/Disciplina-Top-Av-Eng-Software.htm>
- ❑ <http://www.univasf.edu.br/~mario.godoy/gis/>