

5 Extensões para Autoria no MPEG-4

Este capítulo apresenta a especificação de templates de composição hipermídia para a linguagem XMT-O. Com esse objetivo, o capítulo está organizado da forma a seguir. A Seção 5.1 propõe extensões para XMT-O, a partir dos recursos da linguagem NCL. A seguir, a Seção 5.2 aborda o perfil XT-SMIL, que corresponde à incorporação dos templates NCL à linguagem SMIL 2.0. Por fim, com base no perfil XT-SMIL, a Seção 5.3 apresenta o perfil XT-XMT-O.

5.1. Templates para Autoria no MPEG-4

Na apresentação da linguagem NCL no Capítulo 2, os seus módulos *XConnector* e o *XTemplate* foram abordados. O módulo *XConnector* permite a especificação de conectores hipermídia. Um conector hipermídia especifica uma relação que pode ser utilizada para a criação de elos em um documento. Na realidade, o conector especifica a semântica, sem mencionar os nós participantes do relacionamento. O módulo *XTemplate*, por sua vez, permite a especificação de templates de composição hipermídia. Templates hipermídia podem especificar diversos tipos de relacionamentos entre os componentes, definindo estruturas hipermídia genéricas que podem ser herdadas por composições. Um template especifica os componentes e os relacionamentos que uma composição pode possuir, sem especificar todos os componentes e relacionamentos propriamente ditos. Essa especificação é realizada pelas composições, que herdam a semântica definida pelo template.

Conectores e templates são elementos importantes na autoria declarativa de documentos multimídia/hipermídia, pois aumentam o reuso nesse processo. Os conectores permitem que as relações sejam definidas independente dos relacionamentos, o que favorece o reuso de relações em vários relacionamentos distintos. Os templates, por sua vez, permitem o reuso de especificações semiprontas.

Além de favorecer o reuso, conectores e templates são elementos importantes para a usabilidade. A linguagem NCL possui elevado poder de expressão (Muchaluat-Saade, 2003) e, conseqüentemente, é uma linguagem bastante complexa, quando usada em toda a sua plenitude. No entanto, a definição de uma base de conectores e de templates, por usuários especialistas dessa linguagem, torna o uso de NCL bem mais simples, pois as bases de conectores e de templates encapsulam a complexidade das especificações.

As vantagens obtidas através do uso de conectores e templates podem ser estendidas a outras linguagens de autoria hipermídia. Particularmente, este capítulo propõe a inclusão de templates de composição para a autoria de documentos MPEG-4. O uso de templates nesse padrão é apresentado em alguns trabalhos (Boughoufalah et al., 2000; Boughoufalah et al., 2001). No entanto esses trabalhos têm como foco o reuso do design da apresentação, principalmente dos componentes audiovisuais formados a partir das composições de objetos sintéticos simples, e não o reuso da especificação de relacionamentos complexos, que podem ser estabelecidos entre os objetos de uma apresentação.

Entre as linguagens para autoria de documentos do padrão MPEG-4, XMT-O, apresentada no Capítulo 2, destaca-se por permitir, através das suas construções, que a concepção dos documentos seja realizada com base nas intenções do autor. XMT-O é uma linguagem declarativa e modular, características que favorecem a incorporação de novas funcionalidades, através da adição de módulos definidos em outras linguagens, como *XConnector* e *XTemplate*, definidos na linguagem NCL.

Linguagens declarativas compostas por módulos podem ter suas funcionalidades agrupadas em perfis de linguagem. Cada perfil pode, de acordo com a necessidade, reunir um subconjunto dos módulos componentes da linguagem, gerando subconjuntos da própria linguagem, apropriados para casos específicos. Além disso, como essas linguagens podem incorporar módulos originalmente definidos em outras, perfis podem ser utilizados a fim de identificar essa incorporação.

A extensão da linguagem XMT-O, através da adição dos módulos *XConnector* e *XTemplate*, cria dois novos perfis de XMT-O, propostos nesta dissertação como: XC-XMT-O e XT-XMT-O. O primeiro perfil tem como

objetivo aplicar o conceito de conectores à linguagem XMT-O e, o segundo, introduzir as vantagens do uso de templates nessa linguagem.

A incorporação do módulo *XConnector* à linguagem XMT-O torna necessário distinguir, nos relacionamentos especificados nessa linguagem, seus aspectos semânticos, definidos como relações, dos seus participantes. As adaptações necessárias a esse primeiro perfil não serão abordadas nesta dissertação, porém, futuramente, esse perfil pode ser implementado, obtendo, por exemplo, um perfil único, que utilize tanto conectores quanto templates.

O segundo perfil, definido como XT-XMT-O, permite aos autores a definição de estruturas hipermídia que, quando herdadas pelas composições, ofereçam novas semânticas temporais, além das obtidas com as composições tradicionais da linguagem XMT-O.

Na versão original do módulo *XTemplate* (versão 2.0), os relacionamentos, necessários para a definição das estruturas hipermídia nos templates são especificados através de elos, que necessitam dos conectores para definir sua semântica. Dessa forma, a utilização desse módulo por um determinado perfil de linguagem, exigiria a utilização do módulo *XConnector*.

Recentemente, uma nova versão da linguagem *XTemplate* (versão 2.1) tornou-a mais flexível, permitindo também a definição de relações de inclusão (Silva et al., 2004a). A nova linguagem XT-XMT-O proposta nesta dissertação utiliza, como já mencionado, apenas o perfil da linguagem *XTemplate* que permite relações de inclusão. Fica como trabalho futuro a definição de uma nova extensão para XMT-O utilizando o perfil completo *XTemplate*, incorporando relações de inclusão e as definidas por conectores.

Independente da linguagem onde o módulo *XTemplate* for incorporado, é importante mencionar que o conceito de template de composição é uma facilidade da linguagem de autoria. Dessa forma, um documento multimídia/hipermídia que, na sua especificação, faz referência a um template, deve ser previamente processado a fim de ser exibido. Um processador de templates para a linguagem NCL é proposto em (Muchaluat-Saade, 2003). Posteriormente esse processador foi estendido, para templates do perfil X-SMIL (Silva et al., 2004a). O processamento de templates para a linguagem XMT-O, que gera um novo documento XMT-O a partir de um documento especificado em XT-XMT-O e das definições dos templates relacionados, é abordado na seção final deste capítulo.

5.2. Perfil *XTemplate* de SMIL

Na linguagem NCL os relacionamentos são definidos através de elos e composições, sendo que as últimas representam apenas relacionamentos de inclusão. Nessa linguagem, o uso de templates tem como objetivo embutir semântica de relacionamentos, dos mais diversos tipos, nas composições, facilitando a tarefa de autoria do usuário. As composições com semântica embutida podem tornar, a princípio, mais fácil a tarefa de autoria, uma vez que, em uma única composição, podem ser estabelecidas relações para as quais seria necessário o uso de vários elos e outras composições.

Em algumas linguagens, como SMIL, as composições podem representar relações de sincronização temporal ou mesmo de escolha seletiva. No entanto, nessas linguagens, a existência de um conjunto limitado de composições pode restringir as facilidades para autoria, pois para definir relacionamentos complexos pode ser necessário estabelecer composições com vários níveis de aninhamento (Rodrigues et al., 2002).

Com o objetivo de facilitar a autoria em linguagens como SMIL, templates podem definir estruturas semânticas contendo relações de inclusão recursivas, que podem ser herdadas pelas composições. Essas relações permitem especificar composições com novas semânticas temporais. A especificação dos objetos relacionados fica a cargo das composições que herdam os templates.

A estrutura dos templates, especificados segundo o perfil XT-SMIL, é formada pelos nós de composição denominados *par*, *seq* e *excl*, todos definidos em SMIL. Além desses nós, que representam as relações de sincronização tradicionais da linguagem, nós *switch*, que agrupam um conjunto de componentes alternativos, também podem fazer parte das relações especificadas no template. A linguagem SMIL também permite a definição de relacionamentos através de eventos, especificados nos atributos dos seus elementos. Coerentemente, a estrutura dos templates da linguagem XMT-O também se utiliza de eventos.

A fim de possibilitar a definição do perfil XT-SMIL, alguns elementos devem estar presentes na definição da linguagem *XTemplate*, apresentada na Seção 2.3.3. A Tabela 5.1 apresenta os elementos, atributos e conteúdos, definidos na linguagem *XTemplate*, com as modificações propostas em sua versão 2.1 (Silva

et al., 2004a). Os símbolos da Tabela 5.1 possuem os seguintes significados: “?” opcional, “|” ou “*” zero ou mais ocorrências e “+” uma ou mais ocorrências.

Elemento	Atributo	Conteúdo
<i>xtemplate</i>		<i>(vocabulary, constraints?)</i>
<i>vocabulary</i>		<i>(component+, connectors*)</i>
<i>component</i>	<i>type, ctype, maxOccurs, minOccurs</i>	<i>(port component)*</i>
<i>port</i>	<i>type, maxOccurs, minOccurs</i>	
<i>constraints</i>		<i>(constraint resource linkBase XSLT)*</i>
<i>constraint</i>	<i>select, description</i>	
<i>resource</i>	<i>src, type, label</i>	<i>resource</i>
<i>connectors</i>	<i>src, type, maxOccurs, minOccurs</i>	
<i>linkBase</i>	<i>id</i>	<i>(link XSLT)*</i>
<i>link</i>	<i>type</i>	<i>bind*</i>
<i>bind</i>	<i>role, select</i>	

Tabela 5.1 – Elementos, atributos e conteúdo da linguagem *XTemplate*

Conforme comentado na seção 2.3.3, a especificação de um template é composta por duas partes: o vocabulário, definido pelo elemento *vocabulary*, e as restrições, definidas pelo elemento *constraints*.

No vocabulário, onde os componentes do template são especificados, esses podem conter, além de pontos de interface definidos pelo elemento *ports*, outros componentes (*component*) recursivamente. Conectores não são permitidos no perfil *XTemplate* da linguagem XT-SMIL, bem como outros elementos, apontados ainda nesta seção.

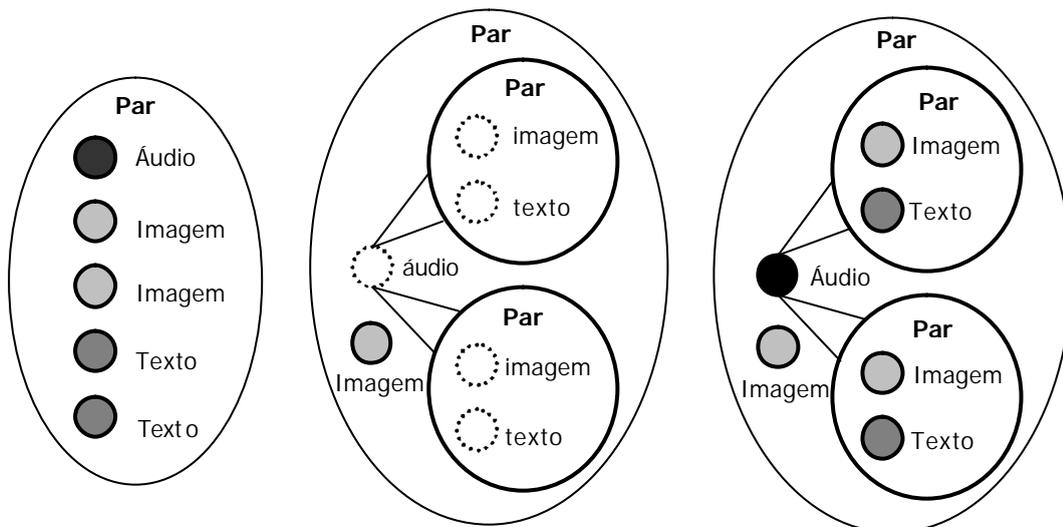
Nas restrições são definidas as regras sobre os elementos especificados no vocabulário, bem como instâncias de componentes, definidas através do elemento *resource*, que também podem conter outras instâncias de componentes (*resource*) recursivamente.

Para incluir atributos nos elementos de uma composição, os templates, especificados para a linguagem SMIL, utilizam as instruções da linguagem XSLT (W3C, 2005c) em conjunto com as expressões da linguagem XPath. As linguagens XPath e XSLT são, sobretudo, empregadas para a especificação de restrições sobre os componentes e pontos de interface, mas também permitem a especificação de atributos e de conteúdos para as instâncias de componentes, representadas pelo elemento *resource*.

Ainda na Tabela 5.1, alguns elementos não são utilizados para a especificação de templates SMIL. Como os relacionamentos, especificados nesses templates, a princípio, não são estabelecidos através de elos e conectores, os elementos *connectors*, *linkBase*, *link* e *bind*, definidos na parte final da Tabela 5.1,

não são empregados por esse perfil. Esses elementos fazem parte do perfil completo de *XTemplate*.

Os templates SMIL, depois de definidos, podem ser diretamente referenciados por composições XT-SMIL. A Figura 5.1 (1) apresenta a visão estrutural de uma composição que referencia um template. Nessa composição, que possuirá nova semântica oferecida pelo template, são definidos objetos de mídia imagem e texto, além de um objeto de áudio com vários pontos de interface associados (âncoras).



Composição XT-SMIL (1) Template SMIL (2) Composição Processada (3)

Figura 5.1 – Composição SMIL com semântica definida por um template hipermídia

Na Figura 5.1 (2), o template, referenciado pela composição XT-SMIL (1), especifica uma estrutura semântica formada por uma composição paralela, que contém outras composições, também com semântica paralela. Cada uma dessas composições terminais contém duas referências para objetos, uma para o objeto do tipo imagem e outra para o objeto do tipo texto. Na composição paralela mais externa, também se encontram definidas uma instância de um objeto de mídia imagem e uma referência para um objeto do tipo áudio. Esse objeto de áudio contém vários pontos de interface que se relacionam com as composições paralelas mais internas do template, através dos atributos definidos nessas composições. Esses relacionamentos possuem semântica de sincronização temporal, onde os pontos de interface acionam eventos de apresentação dessas composições, relacionados às transições de início e término de cada apresentação. Note que nesse template somente o objeto de mídia imagem, pertencente à composição mais externa, é instanciado. Os demais objetos correspondem a

referências, a serem instanciadas pelas composições onde esse template for herdado. O mesmo se aplica aos pontos de interface do objeto do tipo áudio. Essas âncoras participam de relacionamentos, porém somente serão especificadas pela composição que vier a herdar esse template.

Ainda na Figura 5.1, após o processamento do template (2), referenciado por essa composição, uma nova semântica para a apresentação é obtida (3). A nova apresentação da composição exhibe, inicialmente, os objetos do tipo áudio e do tipo imagem (definida pelo template). Cada ponto de interface do áudio é relacionado a uma composição paralela, de forma que, quando esse ponto é alcançado, a composição é executada. Internamente às composições paralelas, são definidos um objeto do tipo imagem e um objeto do tipo texto, que são apresentados simultaneamente, quando cada composição é executada.

5.3. Perfil *XTemplate* de XMT-O

Para a especificação de templates no perfil XT-XMT-O, não são necessárias modificações na estrutura do perfil *XTemplate*, apresentado na seção anterior e utilizado por XT-SMIL, pois os relacionamentos em XMT-O, de forma idêntica à SMIL, são definidos através de eventos, especificados nos atributos dos seus elementos, ou pelo aninhamento das suas composições. Como exemplo, a Figura 5.2 apresenta um template especificado nesse perfil.

```
<?xml version="1.0" ?>
<xtemplate name="XTemplate Example" id="audio-with-subtitles"
xmlns="http://www.telemidia.puc-rio.br/specs/xml/XTemplate" ... >

<vocabulary>
  <component type="audio" ctype="audio" maxOccurs="1" minOccurs="1"/>
  <component type="logoResource" ctype="img" maxOccurs="1" />

  <component type="seq" ctype="seq">
    <component type="par" ctype="par">
      <component type="image" ctype="img" maxOccurs="1" minOccurs="1" />
      <component type="text" ctype="string" maxOccurs="1" minOccurs="1" />
    </component >
  </component >
</vocabulary>

<constraints>
  <resource type="logoResource" label="logo" src="../../img/BackDrop.jpg" />

  <xsl:template match="/*/*/*">
    <xsl:if test="not(./@type='image') and not(./@type='text')">
      <xsl:copy-of select="." />
    </xsl:if>
```

```

</xsl:template>

<xsl:template name="imageElement">
  <xsl:param name="i"></xsl:param>
  <xsl:for-each select="/*/*/child::*[@type='image']" >
    <xsl:copy>
      <xsl:for-each select="text()|@"><xsl:copy/></xsl:for-each>
      <xsl:copy-of select="*/"/>
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:for-each>
</xsl:template>

<xsl:template name="textElement">
  <xsl:param name="i"></xsl:param>
  <xsl:for-each select="/*/*/child::*[@type='text']" >
    <xsl:copy>
      <xsl:for-each select="text()|@"><xsl:copy/></xsl:for-each>
      <xsl:copy-of select="*/"/>
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:for-each>
</xsl:template>

<resource type="seq">
  <xsl:for-each select="child::*[@type='image']" >
    <resource type="par">
      <xsl:variable name="i" select="position()"/>
      <xsl:attribute name="id">par<xsl:value-of select="$i"/></xsl:attribute>
      <xsl:call-template name="imageElement">
        <xsl:with-param name="i" select="$i"></xsl:with-param>
      </xsl:call-template>
      <xsl:call-template name="textElement">
        <xsl:with-param name="i" select="$i"></xsl:with-param>
      </xsl:call-template>
    </resource>
  </xsl:for-each>
</resource>

</constraints>
</xtemplate>

```

Figura 5.2 – Template para o perfil XT-XMT-O

Na Figura 5.2, dentro do vocabulário (*vocabulary*) são declarados três componentes (*component*), um componente do tipo *audio*, relativo a um objeto de mídia áudio; um componente do tipo *logoResource*, relativo a um objeto de mídia imagem e um componente composto do tipo *seq*. Dentro desse componente composto é declarado um componente do tipo *par*. Na realidade, pelo fato desse componente, do tipo *par*, não possuir restrições sobre a sua quantidade de ocorrências (*minOccurs*, *maxOccurs*), vários desses componentes podem ser instanciados. Cada componente do tipo *par*, por sua vez, contém declarados sempre dois outros componentes, um do tipo *image*, relativo a um objeto de mídia imagem e outro do tipo *text*, relativo a um objeto sintético *string* (texto).

Ainda na Figura 5.2, dentro das restrições (*constraints*), é especificada uma instância (*resource*) para o componente do tipo *logoResource*, declarado no vocabulário. Essa instância atribui o objeto de mídia *Backdrop.jpg* a esse componente. A seguir, uma instrução XSLT é especificada (`<xsl:template match="/*/*/*">`). Essa instrução especifica que os componentes do tipo *image* e *text* devem estar contidos em outros componentes. Isso significa que as instâncias desses componentes serão retiradas do conteúdo da composição, onde esse template for referenciado, a fim de serem inseridas em outros componentes.

Ao final das restrições no template, apresentado na Figura 5.2, o componente composto do tipo *seq* é instanciado através do elemento *resource*, tendo como conteúdo instâncias dos componentes do tipo *par*. Cada componente do tipo *par* instanciado recebe, para o seu atributo *id*, um valor diferente, representando a ordem da sua criação. Dentro de cada componente do tipo *par*, são realizadas chamadas a instruções XSLT, previamente declaradas (`xsl:call-template`). Essas instruções têm como objetivo instanciar os objetos de mídia, relativos aos componentes do tipo *image* e *text*, a partir do conteúdo da composição onde esse template for referenciado. Através dessas instruções são estabelecidas as relações de inclusão das composições, relacionadas aos objetos de mídia e suas propriedades, obtidas na composição XT-XMT-O que referencia esse template.

A Figura 5.3 apresenta um documento especificado no perfil XT-XMT-O, que faz referência ao template apresentado na Figura 5.2. Nos elementos desse perfil são adicionados atributos, em relação aos elementos originais da linguagem XMT-O. Para as composições que herdam as estruturas semânticas especificadas nos templates, o atributo *xtemplate*, adicionado nesse perfil, define o endereço (URI) do template utilizado. Complementarmente, nos elementos que podem estar contidos nessas composições, o atributo *type*, também adicionado no perfil, define a referência a um tipo de componente declarado no vocabulário do template referenciado.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XMT-O xmlns="urn:mpeg:mpeg4:xmto:schema:2002" ...>
  <head>
    <layout metrics="pixel" type="xmt/xmt -basic-layout">
      <topLayout width="400" height="300" backgroundColor="#CCCCCC ">
        </topLayout>
      </layout>
    </head>
```

```

<body>
  <par id="propaganda" xtemplate="file:../templates/templateaudiocomlegenda.xml">
    <audio type="audio" region="audioRegion1" id="samba" src="../img/aquarela.mp3"/>
    
      <transformation translation="0 16"/>
    </img>
    
      <transformation translation="0 16"/>
    </img>
    
      <transformation translation="0 16"/>
    </img>
    
      <transformation translation="0 16"/>
    </img>
    <string type="text" id="text1" textLines="&quot;Rio de Janeiro&quot;">
      <material color="black" filled="true"/>
      <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
      <transformation translation="0 -126"/>
    </string>
    <string type="text" id="text2" textLines="&quot;Sao Paulo&quot;">
      <material color="black" filled="true"/>
      <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
      <transformation translation="0 -126"/>
    </string>
    <string type="text" id="text3" textLines="&quot;Belo Horizonte&quot;">
      <material color="black" filled="true"/>
      <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
      <transformation translation="0 -126"/>
    </string>
    <string type="text" id="text4" textLines="&quot;Florianopolis&quot;">
      <material color="black" filled="true"/>
      <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
      <transformation translation="0 -126"/>
    </string>
  </par>
</body>
</XMT-O>

```

Figura 5.3 – Documento especificado segundo o perfil XT-XMT-O

Na Figura 5.3 uma composição paralela herda a estrutura semântica do template apresentado na Figura 5.2. Essa composição declara um objeto de mídia áudio, quatro objetos de mídia imagem e quatro objetos sintéticos do tipo *string*. Pela semântica herdada do template, após o processamento do documento e do template referenciado pela composição, um novo documento XMT-O é obtido. Nesse novo documento, uma composição paralela contém, além do objeto de mídia áudio, um objeto de mídia imagem, instanciado pelo template, e uma composição seqüencial. Nessa composição seqüencial são definidas outras quatro composições paralelas, onde, em cada uma dessas composições, estão dispostos um objeto de mídia imagem e um objeto sintético *string*. Esses objetos, declarados na composição do documento XT-XMT-O apresentado na Figura 5.3, são agrupados de acordo com a ordem dessa declaração. Além disso, todas as

propriedades originais desses objetos são preservadas no documento obtido. A Figura 5.4 apresenta o documento XMT-O resultante do processamento.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XMT-O xmlns="urn:mpeg:mpeg4:xmto:schema:2002" ...>
<head>
  <layout metrics="pixel" type="xmt/xmt-basic-layout">
    <topLayout backgroundColor="#CCCCCC" height="300" width="400"/>
  </layout>
</head>
<body>
  <par id="propaganda">
    <audio id="samba" src="img/aquarela.mp3"/>
    
    <seq>
      <par id="par1">
        
          <transformation translation="0 16"/>
        </img>
        <string id="text1" textLines="&quot;Rio de Janeiro&quot;">
          <material color="black" filled="true"/>
          <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
          <transformation translation="0 -126"/>
        </string>
      </par>
      <par id="par2">
        
          <transformation translation="0 16"/>
        </img>
        <string id="text2" textLines="&quot;Sao Paulo&quot;">
          <material color="black" filled="true"/>
          <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
          <transformation translation="0 -126"/>
        </string>
      </par>
      <par id="par3">
        
          <transformation translation="0 16"/>
        </img>
        <string id="text3" textLines="&quot;Belo Horizonte&quot;">
          <material color="black" filled="true"/>
          <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
          <transformation translation="0 -126"/>
        </string>
      </par>
      <par id="par4">
        
          <transformation translation="0 16"/>
        </img>
        <string id="text4" textLines="&quot;Florianopolis&quot;">
          <material color="black" filled="true"/>
          <fontStyle justify="MIDDLE; MIDDLE" size="18" style="BOLD"/>
          <transformation translation="0 -126"/>
        </string>
      </par>
    </seq>
  </par>
</body>
</XMT-O>
```

Figura 5.4 – Documento XMT-O obtido através do processador de templates

Para realizar a apresentação do documento XMT-O obtido, ele deve ser convertido para o formato BIFS. A Figura 5.5 apresenta duas visões da exibição do documento MPEG-4, correspondente ao documento XMT-O apresentado na Figura 5.4, em dois instantes distintos de tempo. No instante de tempo relativo a oito segundos do início da apresentação, a imagem inicial da composição XMT-O está sendo apresentada, com uma legenda que corresponde ao conteúdo do primeiro objeto sintético *string* declarado na composição. Posteriormente, decorridos 22 segundos do início da apresentação, a imagem apresentada corresponde à terceira imagem da composição, também com a legenda correspondente ao terceiro objeto sintético declarado na composição.



Figura 5.5 – Apresentação do documento MPEG-4 obtido através do template

Devido às semelhanças entre o perfil XT-XMT-O e o perfil XT-SMIL, o processador de templates proposto em (Silva et al., 2004a) pôde ser estendido a fim de ser aplicado também ao novo perfil.

Em (Silva et al., 2004a), a classe *TemplateProcessor* é a responsável pelo processamento de uma composição, que herda um template, gerando uma nova composição. Essa nova composição contém a estrutura semântica especificada pelo template. Nessa classe, o processamento de templates é iniciado através de uma cópia da composição original. Após essa cópia, instâncias de componentes, representadas pelo elemento *resource*, definidas sem o uso de instruções XSLT, são adicionadas à nova composição. Posteriormente, os recursos definidos através das instruções XSLT, como as relações de inclusão, são aplicados à nova composição, através de um processador XSLT (JAXP, 2003). Por fim, as restrições em XSLT são verificadas (Silva et al., 2004a).

O diagrama de classes para estender o processador de templates é apresentado nas Figura 5.6 e 5.7. A primeira apresenta as classes necessárias à instanciação desse processador para o perfil XT-XMT-O. A segunda apresenta uma instância do *framework* para compiladores XMT-O, discutido no Capítulo 3, que tem como objetivo percorrer os documentos especificados pelo perfil XT-XMT-O, gerando as cópias das composições originais.

Na Figura 5.6 são definidas duas novas classes: *XxmtoStructureParser* e *XxmtoTemplateProcessor*. Na primeira, o corpo (*body*) de um documento especificado segundo o perfil XT-XMT-O, onde se encontram definidas as composições que podem referenciar os templates, é analisado. Para isso, essa classe herda os métodos da classe *XmtoStructureParser*, especificada no *framework* para compiladores XMT-O. Nessa classe também é definida uma instância para a classe *XxmtoTemplateProcessor*, que herda as funcionalidades definidas na classe *TemplateProcessor* para o processamento de templates.

Na classe *XxmtoStructureParser*, o método *parseBody*, utilizado para analisar o corpo de um documento, chama o método *processComposite*, herdado pela classe *XxmtoTemplateProcessor*. Esse método realiza as operações necessárias ao processamento de templates, definidas em (Silva et al., 2004a), caso, no corpo do documento, existam composições que façam referências a templates.

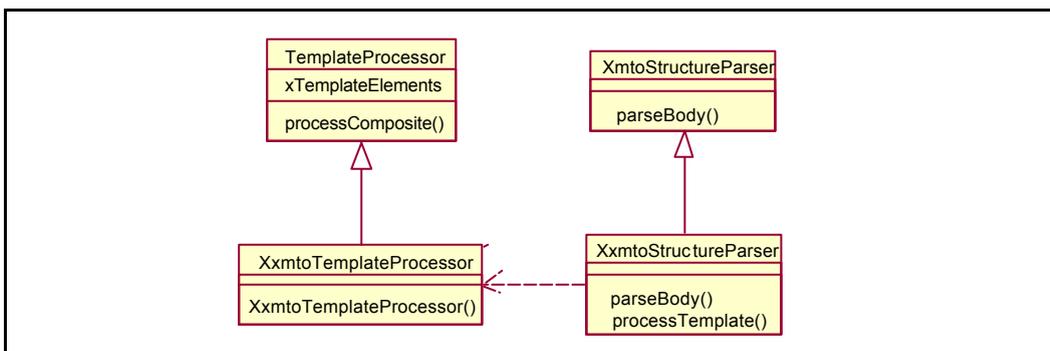


Figura 5.6 – Diagrama de classes para o processador de templates XMT-O

Na Figura 5.7 são definidas várias classes, relativas às áreas funcionais da linguagem XMT-O. Essas classes instanciam o *framework* para compiladores XMT-O, com o objetivo de realizar a tradução das especificações no documento, do perfil XT-XMT-O para a linguagem XMT-O. Na realidade, essas classes percorrem o documento XT-XMT-O, fazendo uma cópia do documento original. Nessa cópia, para as composições que fazem referência a templates, a adição de

instâncias de componentes e a análise das instruções XSLT é realizada pela classe *XxmtoTemplateProcessor*, apresentada na Figura 5.6. Para as demais composições, bem como para as outras partes do documento, como, por exemplo, o *layout*, é realizada uma simples cópia para o novo documento na linguagem XMT-O.

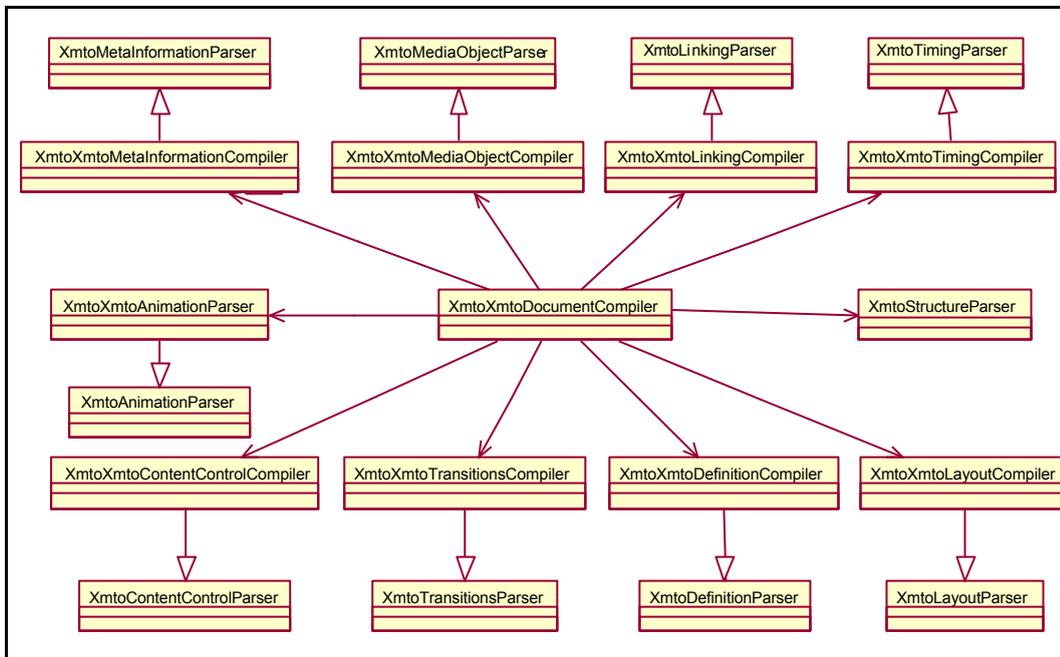


Figura 5.7 – Diagrama de classes para percorrer composições XMT-O

Por fim, é importante destacar que a proximidade entre as linguagens XMT-O e SMIL favorece a interoperabilidade entre documentos especificados nessas linguagens (Kim & Wood, 2004). No caso dos templates para composições hipermídia, essa proximidade permite que templates possam ser herdados, com algumas restrições, pelas composições definidas tanto no perfil XT-XMT-O quanto no perfil XT-SMIL. Templates interoperáveis entre esses perfis podem ser especificados utilizando elementos pertencentes a vários módulos dessas linguagens, entre os quais destacam-se:

- *BasicTimeContainers*, *ExclTimeContainers* e *BasicContentControl*. Todos esses módulos são definidos em XMT-O a partir de módulos SMIL. Esses módulos especificam as composições da linguagem XMT-O (*par*, *seq*, *excl* e *switch*).
- *BasicInlineTime* e *EventTiming*. Esses módulos, definidos em XMT-O a partir de módulos SMIL, especificam os atributos *begin*, *end* e *dur*, que podem ser qualificados através de eventos. A única exceção é o atributo *dur*, que somente pode conter valores absolutos em XMT-O.

Existem, no entanto, algumas diferenças entre os eventos que podem ser especificados nas linguagens XMT-O e SMIL. Em XMT-O, os eventos são, sobretudo, especificados através do módulo *XMT Events*, onde alguns eventos, que correspondem a extensões destinadas à interação dos objetos MPEG-4, não possuem representação em SMIL. Em (ISO/IEC, 2001) são apresentados os principais eventos de XMT-O, comparativamente aos definidos em SMIL. Entre os eventos definidos em ambas as linguagens incluem-se: clique do mouse (*click*), pressionamento do mouse (*mousedown*), mouse fora do elemento (*mouseout*), início da apresentação (*begin*) e término da apresentação (*end*). Os seguintes eventos de XMT-O não possuem representação em SMIL: despressionamento do mouse (*mouseup*), mouse sobre o elemento (*mouseover*), visibilidade do elemento (*viewable*), proximidade do elemento (*near*), colisão de elementos (*collide*).

A principal restrição do emprego de templates, especificados segundo o perfil XT-SMIL nos documentos do perfil XT-XMT-O, corresponde à definição de pontos de interface sobre os componentes do template. A linguagem XMT-O, ao contrário de SMIL, não define pontos de interface nos nós de conteúdo.

No sentido contrário, isto é, na utilização de templates especificados segundo o perfil XT-XMT-O em documentos do perfil XT-SMIL, a principal restrição é relativa à diversidade de componentes definidos pela linguagem XMT-O. Nessa linguagem, o módulo *xMedia* especifica, além dos objetos de mídia tradicionais, tais como: imagens, vídeos e áudio (*img*, *video*, *audio*), definidos pelo padrão como objetos naturais, objetos sintéticos, produzidos por unidades computacionais. Na linguagem SMIL somente os objetos naturais podem ser representados.

Como exemplo de interoperabilidade, o template do perfil XT-XMT-O, apresentado na Figura 5.2, à exceção do objeto sintético denominado *string*, pode ser utilizado por documentos do perfil XT-SMIL.