

4 Componentes MPEG-4 em Documentos NCL

O padrão MPEG-4 é semelhante aos padrões MPEG anteriores, embora possua características peculiares, como a abordagem orientada a objeto, que o aproxima da modelagem de aplicações multimídia/hipermídia.

A fim de explorar como os componentes MPEG-4 podem se relacionar entre si, estando em diferentes cenas, e também com outros objetos, externos às cenas audiovisuais MPEG, este capítulo discute a definição desses componentes como nós de documentos multimídia/hipermídia especificados na linguagem NCL.

O capítulo está organizado da forma a seguir. A Seção 4.1 aborda os aspectos necessários à definição de componentes MPEG-4 em documentos hipermídia especificados em NCL. A seguir, na Seção 4.2 é proposta a representação desses componentes como objetos NCL de mídia contínua, similares aos obtidos pela codificação audiovisual utilizando os padrões MPEG-1 e MPEG-2. Por fim, a Seção 4.3 propõe que os componentes MPEG-4 sejam tratados como composições NCL.

4.1. Definição de Componentes MPEG-4 em Documentos NCL

Conforme definido no Capítulo 2, as apresentações MPEG-4 são formadas por cenas audiovisuais, onde cada cena corresponde a uma composição que pode conter vários objetos, bem como os relacionamentos entre eles. Dentro de cada cena não existem restrições para a definição de relacionamentos entre os seus objetos internos, no entanto, os relacionamentos entre cenas distintas são limitados.

Nos relacionamentos estabelecidos entre diferentes cenas a relação semântica é, normalmente, de referência: uma âncora é associada a um dos objetos da cena de origem, sendo acionada pela ação do clique do mouse. No entanto, algumas extensões XLink (W3C, 2001) podem ser utilizadas a fim de

estender a semântica desses relacionamentos, definindo relações com semântica de sincronização (ISO/IEC, 2001).

A limitação dos relacionamentos definidos entre cenas distintas não se deve somente a restrições existentes nas relações que podem ser estabelecidas, mas também ao fato dessas cenas serem apresentadas isoladamente nos exibidores MPEG-4. Em cada exibidor, durante um intervalo específico de tempo, somente uma cena pode ser apresentada. Dessa forma, quando relacionamentos entre cenas distintas são executados, a cena de origem tem que ser interrompida, para o início da cena de destino.

O modelo de referência dos exibidores MPEG-4 é definido na Parte 5 desse padrão (ISO/IEC, 2000b). Nesse modelo, as cenas audiovisuais são compostas por objetos de diferentes tipos, relacionados entre si, devendo o próprio exibidor MPEG-4 implementar todos os aspectos relacionados à apresentação desses objetos, como a descompressão, decodificação, controle da taxa de apresentação, entre outros, além de controlar, também, os relacionamentos estabelecidos entre eles.

Em outros modelos, mais genéricos, como o proposto pelo sistema HyperProp, o formatador (exibidor) dispõe de um conjunto de ferramentas de exibição a fim de tratar os diferentes tipos de objetos de mídia. Nesse modelo, é responsabilidade de cada ferramenta gerenciar os aspectos relacionados às características específicas dos tipos de objetos (padrões de codificação). Ao orquestrador (outro módulo do formatador) cabe a responsabilidade pelo controle da apresentação, preservando os relacionamentos definidos pelos autores (Rodrigues, 2003).

É importante notar que, inseridos no modelo proposto pelo sistema HyperProp, os exibidores MPEG-4 tornam-se apenas mais uma ferramenta de exibição, especializadas em componentes MPEG-4. Uma vez inseridos nesse sistema, esses exibidores favorecem a especificação de documentos NCL contendo componentes representando as cenas MPEG-4. Na realidade, a linguagem NCL não define restrições para os padrões de codificação adotados pelos objetos de mídia, no entanto cabe ao ambiente de execução, responsável pela apresentação dos documentos, oferecer aplicações capazes de exibir os objetos dos diferentes tipos de mídia, garantindo que os relacionamentos, concebidos na autoria, sejam obedecidos.

A fim de integrar os exibidores MPEG-4 ao sistema HyperProp, como parte do conjunto de ferramentas de exibição desse sistema, adaptações devem ser implementadas nesses exibidores, para que eles sigam o modelo de comunicação com o formatador, apresentado em (Rodrigues, 2003). Nesse modelo, uma interface padronizada para a troca de mensagens entre essas entidades é definida. Além dessas adaptações, os exibidores MPEG-4 devem ainda monitorar os eventos, que podem ser estabelecidos sobre os seus componentes.

Um evento especifica uma ocorrência no tempo resultante de alguma ação sobre um atributo ou sobre uma região do conteúdo (âncora) de um objeto de representação¹⁴ (Rodrigues et al., 2001). Como exemplo de um evento, considere a seleção de uma âncora pelo usuário (evento de seleção). Essa seleção pode corresponder à ação do clique do mouse sobre uma região da apresentação de um vídeo, por exemplo. Caso a ferramenta de exibição, destinada a exibir esse vídeo, não monitore esse evento, relacionamentos causais, disparados por ele, ainda que especificados no documento multimídia/hipermídia, não serão considerados durante a apresentação.

Os eventos que podem ser estabelecidos sobre os componentes MPEG-4 variam de acordo com a forma de representação desses componentes. Em um dos cenários possíveis, esses eventos correspondem aos mesmos que podem ser definidos sobre os objetos NCL de mídia contínua (Rodrigues, 2003). Em outro cenário, onde os componentes MPEG-4 são definidos como composições NCL, esses eventos correspondem aos ocorridos internamente nessas composições, devendo ser exportados através do mapeamento dessas ocorrências internas para as portas da composição.

No primeiro cenário, os eventos podem estar associados a âncoras espaciais sobre a região de apresentação do componente MPEG-4. Nesse caso compete à ferramenta identificar a região estabelecida e controlar as ações nessa área. Ainda nesse cenário, âncoras temporais, relativas ao tempo da apresentação do componente podem ser definidas, devendo a ferramenta monitorar o tempo decorrido sobre a apresentação do componente.

¹⁴ No modelo NCM, o objeto de representação corresponde ao componente apresentado, agregando os atributos do objeto e do seu descritor.

No segundo cenário, os componentes são apresentados como composições NCL, cuja estrutura semântica é estabelecida pelos comandos BIFS, contidos no fluxo descritor de cenas do componente em questão. Nesse cenário, além dos eventos apresentados no cenário anterior, outros podem ser estabelecidos, através do mapeamento dos eventos definidos internamente a essas composições.

Para exemplificar os diferentes cenários, considere uma apresentação audiovisual, onde os relacionamentos são definidos com a participação de um componente MPEG-4. Esse componente, apresentado na Figura 4.1, é formado por três objetos principais (1): o plano de fundo, correspondente a um objeto de imagem (VOP1); um objeto de vídeo (VOP2); e outro objeto de imagem (VOP3). Esses três componentes, definem planos de objeto de vídeo individuais (VOP - *video object plane*) que são sobrepostos para formar o componente MPEG-4 (2).

Suponha que quando o usuário selecionar com o mouse o objeto de vídeo VOP2, pertencente ao componente MPEG-4, um objeto de imagem, relativo a uma fotografia sobre o desmatamento, deva ser exibido e, quando o objeto de imagem VOP3, também pertencente a esse componente, for selecionado, um objeto de vídeo, contendo informações sobre a Amazônia, deva ser exibido.

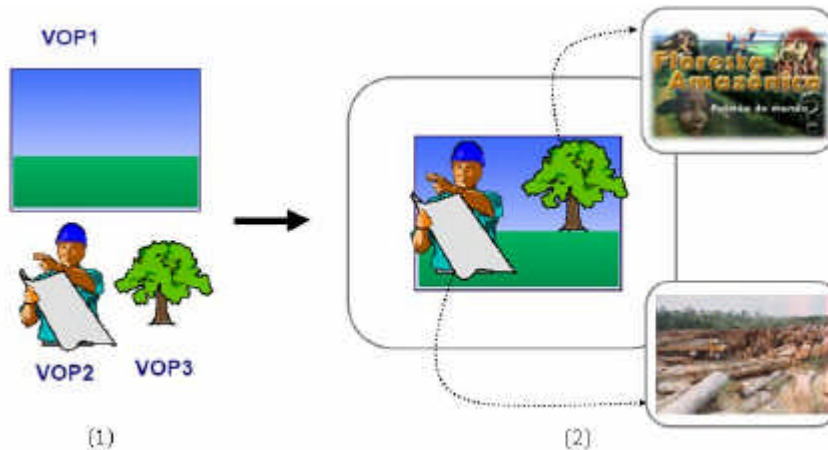


Figura 4.1 – Apresentação MPEG-4 com sincronização espacial e temporal

No cenário inicial, em relação ao exemplo apresentado, o componente MPEG-4 é tratado como um objeto NCL de vídeo. Nesse cenário, a fim de alcançar a concepção desejada, um documento NCL é especificado onde, além do componente MPEG-4, dois outros objetos são definidos, um objeto NCL de imagem, relativo à fotografia sobre o desmatamento, e um objeto NCL de vídeo, contendo informações sobre a Amazônia. Nesse documento, duas âncoras são definidas sobre o componente MPEG-4, cada uma sobre as áreas de exibição dos objetos de vídeo VOP2 e de imagem VOP3. As relações nesse documento são

definidas através de um único conector NCL com semântica causal. Esse conector estabelece como condição o evento de clique do mouse, e como ação a transição de início do evento de apresentação. Dois relacionamentos distintos são estabelecidos referenciando esse conector. No primeiro, um elo associa o componente MPEG-4, na sua âncora definida sobre VOP2, ao objeto NCL de imagem. No segundo, a associação é definida por um elo entre a âncora definida sobre VOP3, no componente MPEG-4, e o objeto NCL de vídeo.

É importante ressaltar que nesse primeiro cenário, os relacionamentos NCL são definidos sobre as âncoras estabelecidas no componente MPEG-4, como mencionado, áreas espaciais cujo posicionamento varia no tempo, e não diretamente sobre os objetos que compõem esse componente. Portanto, algumas dificuldades podem ocorrer na definição desses relacionamentos, como, no exemplo, a definição de âncoras como função do tempo.

No segundo cenário, os eventos são definidos internamente ao componente MPEG-4, usando as facilidades definidas pelo padrão. Na Figura 4.1, os eventos de clique do mouse podem ser diretamente especificados sobre os objetos de vídeo VOP2 e de imagem VOP3, através de comandos BIFS. Nesse cenário, cabe à ferramenta de exibição capturar esses eventos, quando disparados, e ativar interfaces atribuídas ao componente MPEG-4. Essas interfaces, por sua vez, podem estar relacionadas a outros componentes MPEG-4, composições ou objetos de diversos tipos, definidos no documento NCL. Portanto, nesse segundo cenário, ao contrário do primeiro, os elos encontram-se associados, através do mapeamento de interfaces, diretamente aos objetos que compõem o componente MPEG-4. O componente MPEG-4 é tratado como uma composição NCL, com cada âncora interna associada a uma de suas portas.

Voltando ao exemplo da Figura 4.1, aplicado agora a esse segundo cenário, a especificação do documento NCL, destinado a representar a concepção desejada, contém dois relacionamentos. No primeiro, um elo NCL associa a interface do componente (composição) MPEG-4, mapeada ao evento do clique do mouse sobre o objeto de imagem VOP3. No segundo, através de outro elo NCL, uma associação é definida entre a interface do componente (composição) MPEG-4, relativa ao evento do clique do mouse sobre o objeto de vídeo VOP2, e o objeto NCL de imagem.

Independente do cenário considerado, a especificação de documentos NCL contendo componentes MPEG-4 permite que relacionamentos, anteriormente não definidos, com especialização semântica causal ou de restrição possam ser estabelecidos entre esses componentes. Nessa proposta, os documentos MPEG-4 concebidos pelo autor podem, a qualquer momento, ser relacionados entre si, ou participar de relacionamentos com outros componentes e objetos, externos às cenas audiovisuais, tais como: composições NCL ou outros objetos de mídia.

4.2. Objetos MPEG-4

Uma das possíveis representações de componentes MPEG-4, em documentos NCL, consiste em defini-los como objetos NCL de mídia contínua.

No ambiente de autoria, a representação declarativa dos componentes MPEG-4 pode ser definida pelos tipos básicos de objetos de mídia definidos em NCL (Muchaluat-Saade, 2003). Nesse ambiente, além da definição do tipo do objeto, a ferramenta de exibição, destinada ao componente MPEG-4, deve ser identificada. Essa identificação pode ser estabelecida pelo nome da classe que contém a implementação da ferramenta, descrita através do atributo *player*, no descritor de objetos destinado ao componente MPEG-4.

A Figura 4.2 apresenta um documento NCL contendo um componente MPEG-4. Nela, o documento MPEG-4, denominado *foodVideo320.mp4*, é atribuído ao componente MPEG-4 desse documento, através do tipo de objeto vídeo (*video*). Ainda na Figura 4.2, o descritor para o componente MPEG-4, denominado *videoMP4*, é especificado. Esse descritor estabelece, através do seu atributo *player*, a classe que instancia a ferramenta de apresentação destinada a esse componente, denominada *HYPF_MPEG4Player*.

```

<ncl>
  <head>
    ...
    <descriptorBase>
      <descriptor id="logoDescriptor" region="logo"/>
      <descriptor id="textD1" region="text"/>
      <descriptor id="videoMP4" region="image" player="HYPF_MPEG4Player"/>
      <descriptor id="imagtextD2" region="image"/>
      <descriptor enableTimeBar="on" id="audio" region="time"/>
    </descriptorBase>
    ...
  </head>
  <body>
    ...
    <video descriptor="videoMP4" id="foodVideo" src="foodVideo320.mp4"/>
    ...
  </body>
</ncl>

```

Figura 4.2 – Documento NCL contendo um componente MPEG-4

No ambiente de execução, a representação de componentes MPEG-4 como objetos NCL de mídia contínua aproxima as funcionalidades desejadas das ferramentas de exibição desse padrão, às atuais funcionalidades das ferramentas de exibição destinadas aos componentes dos padrões MPEG anteriores. Nas ferramentas, o controle de eventos (controle de âncoras), estabelecido sobre os objetos, é similar, característica que facilita a integração da ferramenta de exibição MPEG-4 ao conjunto de ferramentas de exibição do sistema HyperProp, onde adaptadores para exibidores MPEG-1 e MPEG-2 já haviam sido incluídos (Rodrigues, 2003).

Uma importante limitação para a inserção de uma ferramenta de exibição MPEG-4 no conjunto de ferramentas de exibição do sistema HyperProp deve-se ao fato do modelo de referência dos exibidores MPEG-4 disponibilizar apenas uma versão do exibidor MPEG-4, implementado, em grande parte, na linguagem C. Essa implementação impõe limitações operacionais para a integração com aplicações implementadas em outras linguagens, particularmente, o formatador do sistema HyperProp, que na sua versão atual é implementado na linguagem Java.

Com o objetivo de favorecer a interoperabilidade entre a ferramenta de exibição e o formatador, nesta dissertação, o projeto para desenvolvimento de aplicações relacionadas ao MPEG-4, denominado *IBM Toolkit for MPEG-4*¹⁵, foi adotado. Esse projeto especifica um conjunto de ferramentas, implementadas na linguagem Java, cujas principais aplicações são:

¹⁵ <http://www.alphaworks.ibm.com/tech/tk4mpeg4>.

- AVgen: aplicação destinada a criar arquivos MPEG-4, a partir de mídias de áudio e vídeo. Além da multiplexação, permite que os arquivos criados sejam transmitidos através de servidores de fluxos utilizando RTP/RTSP;
- XMTBatch: aplicação destinada a conversão de documentos no formato XMT-O para XMT-A ou de XMT-A para arquivos MPEG-4, especificados no formato binário e contendo as mídias relacionadas multiplexadas;
- M4Play: exibidor MPEG-4 destinado a exibir documentos desse padrão.

Cada aplicação desse projeto oferece um conjunto de APIs (*Application Program Interface*) que podem ser utilizadas no desenvolvimento de novas aplicações voltadas para o padrão MPEG-4 (*Software Developer Kit for MPEG-4 Systems*). Particularmente, as classes e métodos pertencentes à API da aplicação M4Play são diretamente utilizadas nesta dissertação no desenvolvimento de uma ferramenta de exibição para componentes MPEG-4 no sistema HyperProp.

A Figura 4.3 apresenta o diagrama de classes para a ferramenta de exibição MPEG-4 desta dissertação. A definição dessa ferramenta segue o *framework* proposto em (Rodrigues et al., 2001). A classe principal dessa ferramenta é definida em *HF_MPEG4PlayerAdapter*. Da mesma forma que a classe das ferramentas de exibição JMF (*Java Media Framework*), voltadas para a exibição de objetos MPEG-1, denominada *HF_JmfPlayerAdapter*, a classe *HF_MPEG4PlayerAdapter* herda das classes *HF_PlayerAdapter* e *HF_GenericPresentationTool* os métodos básicos oferecidos pelo formatador, implementados a partir da definição da interface *TF_PresentationTool*.

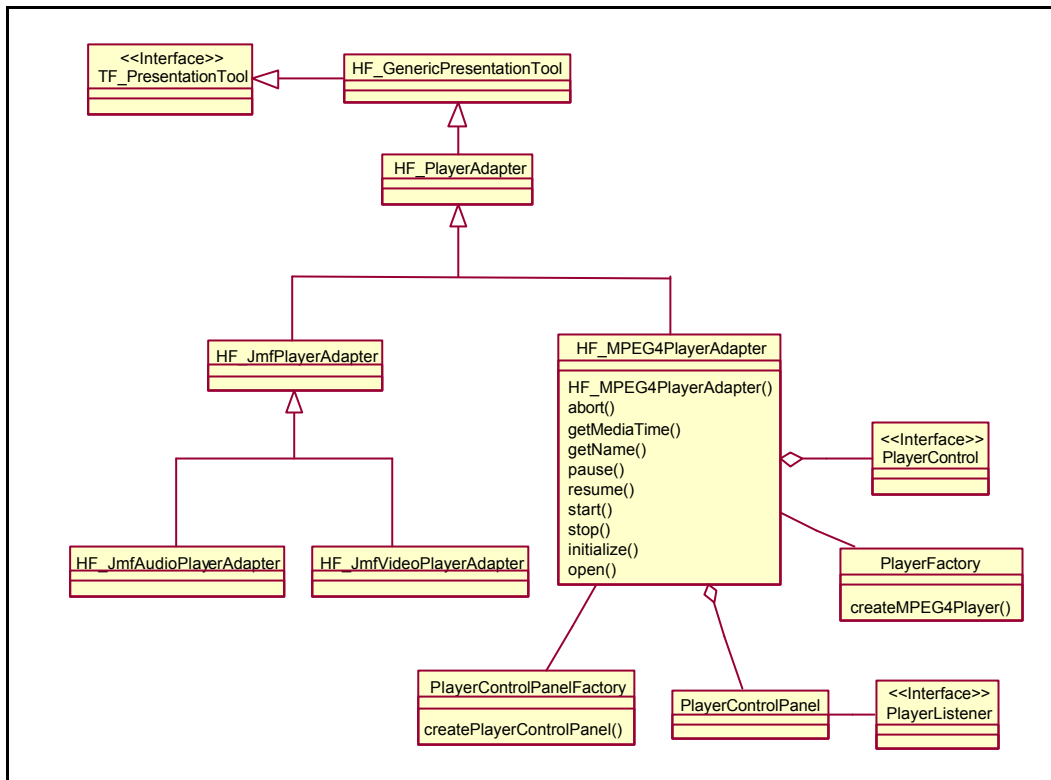


Figura 4.3 – Diagrama de classes da ferramenta de exibição MPEG-4

Durante a execução da apresentação no formatador, quando um objeto de representação específico deve ser exibido, a classe relativa à ferramenta de exibição desse objeto que implementa a interface *TF_PresentationTool* é instanciada e o seu método *initialize* executado, com os parâmetros necessários (Rodrigues et al., 2001). Quando esse objeto é um componente MPEG-4, o método *initialize* da classe *HF_MPEG4PlayerAdapter* é acionado, recebendo como parâmetro o objeto de execução, cuja estrutura possui quatro atributos principais: identificador único, objeto de dados, descritor e lista de eventos.

No método *initialize* da classe *HF_MPEG4PlayerAdapter*, as APIs para criação da ferramenta são utilizadas. Basicamente, essas APIs permitem a definição de duas regiões principais, que compõem a ferramenta. A primeira região corresponde ao local onde o componente MPEG-4 é executado, enquanto a segunda define onde o controle da apresentação dessa ferramenta pode ser disponibilizado. Esse controle da apresentação permite ao usuário interagir com o componente exibido, através de uma interface com botões relativos às ações de iniciar, parar, pausar, reiniciar etc. Além de oferecer opções de interação, esse controle fornece também informações, como o tempo transcorrido a partir do início da apresentação.

No método *initialize*, a ferramenta de exibição propriamente dita é instanciada, através da chamada ao método *createMPEG4Player* da classe *PlayerFactory*. Esse método retorna uma instância do exibidor, através da implementação da interface *PlayerControl*. Todos os acessos às funções dessa ferramenta são realizados através dos métodos dessa interface. Essas funções resumem-se ao controle dos seus estados e das suas propriedades. As propriedades definem aspectos da apresentação, como o controle do volume e o tempo transcorrido a partir do início da apresentação. Os estados, definidos internamente à ferramenta, são apresentados na Figura 4.4.

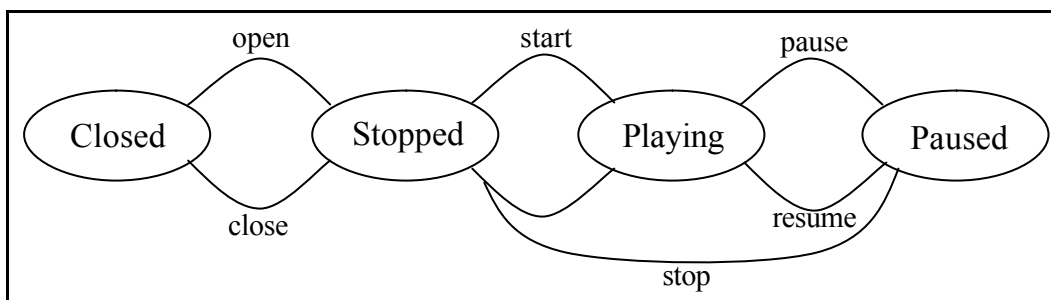


Figura 4.4 – Máquina de estados da ferramenta de exibição MPEG-4

Quando a ferramenta de exibição é criada, ela se encontra no estado *Closed*. O método *open*, definido na interface *PlayerControl*, recebe como parâmetro a URL de um documento MPEG-4 e carrega esse documento no exibidor. O método *open* somente é executado com sucesso se a ferramenta se encontra no estado *Closed*, alterando o estado da ferramenta para *Stopped*. Nesse estado, o método *start* pode ser acionado para iniciar a apresentação do componente MPEG-4 e, conseqüentemente, para alterar o estado da ferramenta para *Playing*. A execução de um componente MPEG-4 pode ser interrompida, através da chamada ao método *stop*. A execução também pode sofrer uma pausa, através da chamada ao método *pause*. Caso o método *stop* seja acionado, a ferramenta volta para o estado *Stopped*, por outro lado, sendo o método *pause* acionado, o estado da ferramenta torna-se *Paused*, estado no qual a ferramenta pode voltar a exibir o componente, do ponto interrompido, passando para o estado *Playing*, através do método *resume*, ou finalizar a apresentação, através do método *stop*, voltando para o estado *Stopped*. Finalmente, no estado *Stopped*, o documento MPEG-4 pode ser fechado, fazendo com que a ferramenta altere o seu estado para *Closed*.

Ao término natural da apresentação de um componente, a ferramenta encontra-se no estado *Playing*. Quando isso ocorre, ela pode voltar naturalmente para o estado *Stopped*, condição definida na ferramenta como

END_ACTION_STOP. Outra possibilidade consiste na ferramenta permanecer no estado *Playing* indefinidamente, até que o método *stop* seja acionado. Essa condição é definida na ferramenta como *END_ACTION_CONTINUE*. Uma terceira possibilidade consiste em voltar, quando o componente termina, ao estado *Stop* e, imediatamente, retornar para o estado *Playing*, condição definida como *END_ACTION_REPEAT*. A escolha do modo de término da apresentação pode ser realizada na ferramenta e, na última opção apresentada, o número de repetições pode ser previamente configurado.

Além dos métodos citados, a máquina de estados da ferramenta é complementada pelos métodos *playURL* e *stopURL*. No primeiro, a ferramenta, independentemente do estado original, passa para o estado *Playing*. Esse método recebe como parâmetro a URL de um documento MPEG-4. No segundo, a ferramenta passa para o estado *Stopped* independente do estado original em que se encontrava.

Voltando ao método *initialize*, o controle da ferramenta de apresentação pode ser instanciado, através da chamada ao método *createPlayerControlPanel* da classe *PlayerControlPanelFactory*. Esse método retorna uma instância para a classe *PlayerControlPanel*, que consiste em um pequeno painel contendo botões de controle para iniciar, pausar, interromper, controlar o volume etc. Os métodos associados aos botões desse painel são controlados diretamente pela interface *PlayerListener*. Dessa forma, mudanças no estado da ferramenta são refletidas diretamente no painel sem a necessidade de adicionar novos métodos. Como exemplo, se a ferramenta encontra-se no estado *Paused*, o botão relativo à ação de iniciar, encontra-se desabilitado.

Ao término do método *initialize*, o conteúdo do componente MPEG-4 é carregado na ferramenta através da chamada ao método *open*. A função básica desse método é acionar o método *open*, definido na interface *PlayerControl* da ferramenta. Vários outros métodos definidos na classe *HF_MPEG4PlayerAdapter* têm como função acionar chamadas aos métodos da máquina de estados da ferramenta, entre esses, estão os seguintes métodos: *abort (stop)*, *pause (pause)*, *resume (resume)*, *start (start)* e *stop (stop)*. Além de alterar os estados da ferramenta de exibição, os métodos definidos em *HF_MPEG4PlayerAdapter* devem informar ao formatador as transições nas máquinas de estados dos eventos NCM durante a apresentação dos objetos.

Como exemplo de aplicação da ferramenta, considere um componente MPEG-4, composto apenas por um objeto de vídeo. No documento NCL são definidos vários relacionamentos de sincronização desse componente MPEG-4 com objetos NCL de áudio, texto e imagem. A Figura 4.5 apresenta a visão temporal da apresentação do documento. Na Figura 4.5 pode ser observado que, conforme o componente MPEG-4 é exibido, vários objetos de áudio e de texto deverão ser exibidos, de forma síncrona. Durante toda a exibição do componente MPEG-4, uma imagem, relativa a um logotipo deve ser apresentada.

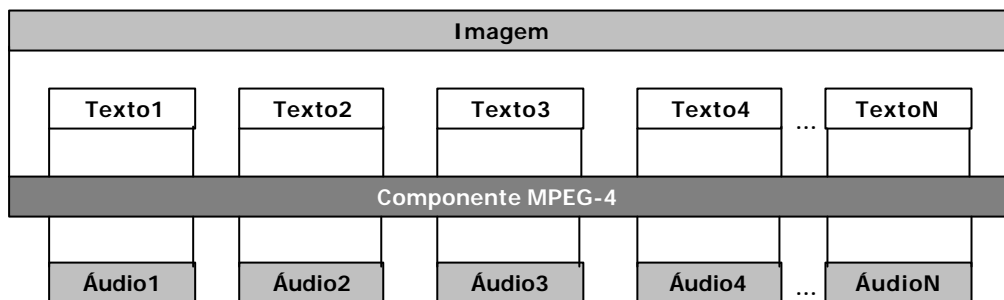


Figura 4.5 – Visão temporal de um documento NCL contendo um componente MPEG-4

A Figura 4.6 apresenta um instantâneo da exibição do documento NCL correspondente à visão temporal da Figura 4.5. O componente MPEG-4 é exibido através da ferramenta de exibição desenvolvida, onde se pode ver, além da sua região de exibição, uma barra de controle da apresentação. Através dessa interface é possível, por exemplo, pausar a apresentação do componente. Nesse caso, toda a apresentação permanece aguardando o reinício da exibição desse componente, preservando o sincronismo temporal estabelecido.



Figura 4.6 – Apresentação do documento NCL com um componente MPEG-4

4.3. Composições MPEG-4

Componentes MPEG-4 são formados por uma estrutura complexa, que pode ser comparada a uma composição NCL. Essa estrutura contém objetos internos, que podem estar relacionados entre si, ou com objetos definidos em cenas audiovisuais distintas, pertencentes a outros componentes. As propriedades e os relacionamentos definidos internamente a esses componentes são especificados em um formato próprio, que define a sua semântica de apresentação. Tradicionalmente, essa sincronização segue o paradigma *timeline*, no entanto, usando esse paradigma, com pequenas extensões, alguns relacionamentos são preservados.

Os relacionamentos sempre preservados são aqueles associados a eventos de interatividade, cuja ocorrência não pode ser prevista durante uma apresentação. Para definir esses eventos, rotas (*routes*) e sensores (*sensors*) (ISO/IEC, 2001), propostos originalmente em VRML (ISO/IEC, 1997), são adotados pelo MPEG-4.

No MPEG-4, conforme apresentado no Capítulo 2, uma cena audiovisual é construída como um grafo dirigido acíclico. Os nós desse grafo podem ser classificados em: nós de grupo (*grouping nodes*), com a função de agrupar outros nós; nós interpoladores (*interpolator nodes*), para o controle das animações; nós sensores (*sensor nodes*), que são utilizados em eventos interativos; e, finalmente, nós filhos (*children nodes*), que efetivamente representam os objetos (Bouilhaguet et al., 2000).

Independente do tipo do nó, suas propriedades são definidas nos seus campos. Existem vários tipos de campos, que se encontram especificados na referência (ISO/IEC, 2001). Do ponto de vista da definição dos eventos de interatividade, esses campos podem ser classificados nos seguintes tipos: campo (*field*), destinado aos campos cujos valores são estabelecidos apenas uma vez; evento de entrada (*eventIn*), que somente estão habilitados a receber eventos; evento de saída (*eventOut*), que somente podem emitir eventos; e campo exposto (*exposedField*), onde os campos podem emitir e receber eventos e ainda ter valores iniciais, de forma similar aos campos do tipo *field* (Bouilhaguet et al., 2000; Concolato & Feuvre, 2003).

Nós classificados como sensores geram eventos, que podem ser disparados pela interação do usuário com um nó específico ou, eventualmente, por modificações na cena (ISO/IEC, 2001). Os eventos gerados nos sensores alteram os valores dos seus campos do tipo evento de saída (*eventOut*). As alterações nesses campos, por sua vez, podem ser atribuídas aos campos do tipo evento de entrada (*eventIn*) e campo exposto (*exposedField*) de um nó qualquer da cena audiovisual, modificando o valor desse campo e, conseqüentemente, as propriedades do nó escolhido.

Para associar os campos definidos nos sensores aos campos dos demais nós, uma rota deve ser estabelecida entre eles. Nessas rotas, o escopo do sensor é limitado a sua cena audiovisual, isto é, rotas propagam eventos entre os elementos pertencentes a uma mesma cena. Como exemplo, a Figura 4.7 apresenta a representação gráfica da transmissão de eventos no formato binário do MPEG-4.

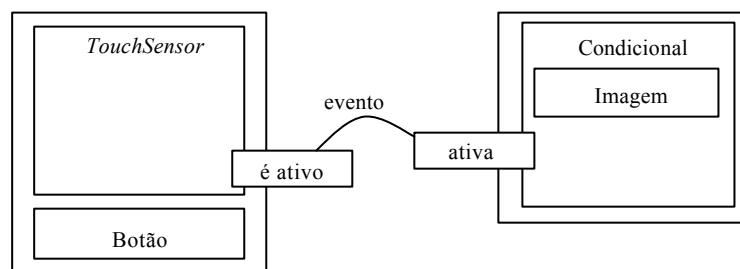


Figura 4.7 – Eventos em BIFS

Na Figura 4.7, o sensor (*TouchSensor*) é especificado internamente a um nó de grupo que contém também um nó filho, cujo conteúdo corresponde ao objeto sintético denominado “Botão”. Um nó do tipo sensor possui vários campos, entre eles, destaca-se o campo associado ao evento do clique do mouse sobre os objetos definidos internamente ao seu grupo, denominado “é ativo” (*isActive*), do tipo evento de saída (*eventOut*). Ainda na Figura 4.7, esse campo do sensor encontra-se vinculado, através de uma rota, ao campo que define se um nó condicional encontra-se ativo ou não, denominado “ative” (*activate*), do tipo evento de entrada (*eventIn*). Dentro desse nó condicional, é definido um outro nó, contendo um objeto de mídia imagem. Caso o evento de interatividade do usuário, definido pela ação do clique do mouse sobre o objeto sintético ocorra, o campo “é ativo” do sensor recebe o valor verdadeiro e, em virtude dessa mudança, através da rota, o campo “ative” do nó condicional é modificado, tendo o seu conteúdo substituído pelo valor do campo “é ativo”. Considerando que originalmente, o

valor do campo “ative” é definido como falso, quando o seu valor tornar-se verdadeiro, a imagem, definida internamente ao nó condicional, será apresentada.

A Tabela 4.1 apresenta alguns dos principais sensores definidos no MPEG-4. A lista completa desses sensores pode ser encontrada em (ISO/IEC, 2001). Todos os sensores apresentados possuem campos que representam alterações relacionadas aos eventos do mouse sobre os nós de uma cena audiovisual. Esses campos podem ter valores de diversos tipos, como no exemplo apresentado na Figura 4.7, onde o campo “é ativo” do sensor é do tipo booleano (*SFBool*).

Nós BIFS	Descrição
TouchSensor	Detecta o movimento do mouse e as ações dos botões.
DiscSensor	Detecta o movimento do mouse e o traduz em eventos de rotação, com origem no sistema de coordenadas cartesianas.
PlaneSensor2D	Detecta o movimento do mouse e o traduz em eventos de translação, com origem no sistema de coordenadas cartesianas.
ProximitySensor2D	Detecta a presença do mouse em uma região específica e gera eventos de posicionamento, tempo e estado (se está dentro ou fora da região).
Anchor	Detecta as ações dos botões do mouse.

Tabela 4.1 – Principais sensores BIFS

Sensores e rotas definidos em componentes MPEG-4 são elementos atípicos, não encontrados em outros padrões de codificação. Para que esses elementos possam ser representados em documentos multimídia/hipermídia especificados em NCL, é desejável representar os componentes MPEG-4 como composições NCL. Sensores devem ser mapeados em âncoras, associadas aos eventos. Caso se queira utilizar os eventos para definir relacionamentos com objetos externos à cena, as âncoras, correspondentes aos sensores, devem ser mapeadas em portas (interfaces) da composição. A Figura 4.8 apresenta uma composição, onde um elo é especificado associando um nó interno do componente MPEG-4 a um nó de mídia (*A*), especificado no documento multimídia/hipermídia representado pela composição mais externa.

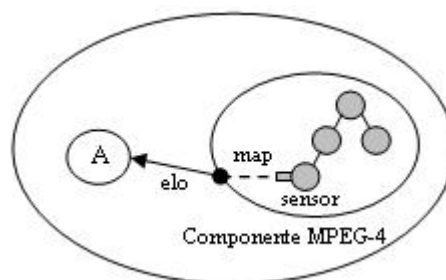


Figura 4.8 – Elo estabelecido entre uma composição MPEG-4 e um objeto

Na Figura 4.8, o relacionamento entre os nós é mapeado através de uma interface definida sobre o componente MPEG-4. Nesse componente os eventos originados pelos sensores definem mudanças nos valores dos seus campos (*eventOut* e *exposedField*). No mapeamento desses eventos para as interfaces (portas da composição), alterações nos valores dos campos pertencentes aos sensores devem ser externadas, indicando sua ocorrência. No sentido inverso, alterações nos eventos ligados às portas da composição devem ser mapeadas para os campos dos sensores.

Nos elos NCL, *binds* devem associar as interfaces dos componentes MPEG-4 a papéis do tipo ação ou propriedade, especificados nos conectores hipermídia. Nesses conectores os valores dessas interfaces definem condições para a execução de uma ação, como, por exemplo, para executar um papel do tipo ação. No sentido inverso, conectores, a partir de alguma condição, podem definir papéis para atribuir valores a essas interfaces. A Figura 4.9 apresenta um relacionamento entre um nó interno ao componente MPEG-4 e um nó definido no documento multimídia/hipermídia.

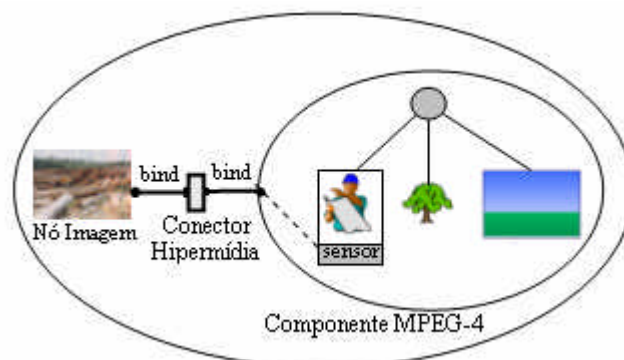


Figura 4.9 – Apresentação NCL contendo uma composição MPEG-4

No componente MPEG-4 da Figura 4.9, um nó sensor é definido internamente ao nó de grupo que contém um objeto de vídeo. No documento NCL (representado pela composição mais externa da figura), um conector hipermídia é definido, com especialização semântica causal, contendo dois papéis. Nesse documento, um elo é definido, estabelecendo um relacionamento entre o componente MPEG-4 e um nó representando uma mídia imagem. A relação desse elo é definida pelo conector, onde um dos seus papéis é associado, através de um *bind*, ao nó de imagem. O outro papel desse conector encontra-se associado, através de outro *bind*, à interface do componente MPEG-4.

Quando o evento relativo ao clique do mouse é acionado sobre o objeto de vídeo do componente MPEG-4, o campo “é ativo” (*isActive*) do sensor torna-se verdadeiro. Nesse momento, o valor desse campo deve ser mapeado para a interface correspondente do componente, que reporta os seus eventos internos. Essa interface encontra-se associada ao conector hipermídia, onde uma condição é estabelecida ao valor desse papel a fim de executar um papel do tipo ação, associado ao nó de imagem. A condição deve ser verdadeira quando o valor da interface também o for, realizando a apresentação da imagem.

A fim de apresentar os componentes MPEG-4 como composições NCL, a ferramenta de exibição associada a esses componentes deve mapear as ocorrências entre os seus eventos internos e suas interfaces. A Seção 4.2 discute como uma ferramenta de exibição MPEG-4 foi adicionada ao conjunto de ferramentas do sistema HyperProp. A partir dessa integração, para apresentar os componentes MPEG-4 como composições, são necessárias modificações na ferramenta, implementando o mapeamento entre os eventos e os pontos de interface dos componentes.

Como a ferramenta de exibição apresentada na Seção 4.2 não pode ser adaptada, pelo fato da sua licença de uso não permitir modificações no seu código, para implementar modificações no exibidor MPEG-4 foi necessário adotar um exibidor licenciado como código aberto, o que restringiu as opções disponíveis à ferramenta de exibição publicada pelo padrão (ISO/IEC, 2000b) e ao OSMO4 (*Osmose*) (GPAC, 2002). É importante ressaltar que, nas opções existentes, as implementações estão disponíveis apenas na linguagem C/C++.

O OSMO4 é um exibidor MPEG-4 desenvolvido pelo GPAC que segue o padrão de referência do MPEG-4. Os softwares desenvolvidos por esse grupo têm como objetivo serem alternativas de implementação menores, mais compreensíveis e flexíveis em relação aos softwares originais desse padrão (GPAC, 2002), motivos pelos quais o OSMO4 foi escolhido como a ferramenta de exibição a ser adaptada. A versão dessa ferramenta, adotada nesta dissertação, é denominada 2.0 e foi publicada em setembro de 2004. Nessa versão, o código fonte é fornecido na linguagem C, com algumas classes C++. Esse código encontra-se organizado sobre um projeto para o compilador Visual C++ 6.0¹⁶.

¹⁶ <http://msdn.microsoft.com/visualc/default.asp>

Na implementação do OSO4, as classes *cMainFrame* e *WinGPAC* definem a sua interface gráfica principal. Nelas, as funções para manipulação dos documentos MPEG-4 são implementadas e disponibilizadas através de componentes gráficos. Essas classes implementam a região onde os componentes MPEG-4 são apresentados, bem como as interfaces relacionadas à apresentação, como a barra de informações da aplicação, a barra de ferramentas e o menu de opções, onde as ações de iniciar, parar, pausar, reiniciar, entre outras, aplicadas aos componentes MPEG-4 exibidos, são oferecidas.

Com exceção das funções presentes nas interfaces gráficas, que são especificadas através de classes, a maior parte das funcionalidades dessa ferramenta são especificadas através de bibliotecas C. A Tabela 4.2 apresenta os principais arquivos da implementação dessa ferramenta, selecionados nesta dissertação para a manipulação de cenas audiovisuais MPEG-4, em relação aos sensores e rotas. É importante mencionar que os arquivos listados na Tabela 4.2 foram sintetizados a partir da análise direta do código fonte da ferramenta, pois não há documentação disponível sobre o seu funcionamento (GPAC, 2002).

Arquivo	Descrição
<i>m4_esm_dev.h</i>	Define a estrutura da região para apresentação de um componente MPEG-4 e a estrutura da cena audiovisual apresentada nessa região.
<i>m4_scenegrph dev.h.</i>	Define detalhes da estrutura audiovisual como rotas.
<i>route.c</i>	Implementa os procedimentos para manipulação das rotas.
<i>m4_nodes.h</i>	Define a estrutura dos nós pertencentes a uma cena audiovisual.
<i>render2d.c</i>	Implementa os procedimentos para manipulação da região onde componentes MPEG-4 são apresentados.
<i>sensor_stacks.c</i>	Implementa os procedimentos para manipulação dos sensores.
<i>M4Term.c</i>	Implementa os procedimentos para acesso através do terminal audiovisual. Nesse arquivo devem ser implementados os procedimentos a serem acessados a partir da interface gráfica.

Tabela 4.2 – Arquivos principais da ferramenta de exibição

Na Tabela 4.2, a biblioteca *m4_esm_dev.h* especifica as informações sobre a região da apresentação de um componente MPEG-4, através de uma estrutura denominada *_m4_client*. Nessa estrutura, o conteúdo da cena audiovisual apresentada é armazenado através de um ponteiro para o elemento raiz da cena, denominado *root_scene*.

No exibidor OSO4, a própria cena audiovisual também é uma estrutura, denominada *_inline_scene*, especificada também na biblioteca *m4_esm_dev.h*. Essa estrutura armazena as informações que uma cena audiovisual deve conter, como os descritores de objeto para o conteúdo dos seus nós e o seu tempo para apresentação. As informações sobre rotas e sensores, definidos na cena

audiovisual, são armazenadas através de um ponteiro, denominado *graph*, para a estrutura *_tagSceneGraph*.

Na biblioteca *m4_scenegraph_dev.h* é definida a estrutura denominada *_tagSceneGraph*, que armazena as informações sobre os sensores. No entanto, as informações sobre os sensores não existem isoladamente, e sim, dentro das rotas especificadas em uma cena audiovisual MPEG-4. Evidentemente, essas informações podem ser obtida através das rotas, porém sensores que não estejam associados a nenhuma rota não serão identificados.

Dentro da estrutura *_tagSceneGraph* é especificada uma estrutura de fila, correspondente às rotas estabelecidas em uma cena audiovisual. Cada rota, por sua vez, é definida por uma estrutura, denominada *struct_route*, que também é especificada na biblioteca *m4_scenegraph_dev.h*. A Figura 4.10 apresenta parte dessa estrutura.

```
typedef struct _route
{
    SFNode *FromNode;
    u32 FromFieldIndex;
    FieldInfo FromField;

    SFNode *ToNode;
    u32 ToFieldIndex;
    FieldInfo ToField;

    const char *fromFieldName;
    ...
} Route;
```

Figura 4.10 – Estrutura das rotas na ferramenta de exibição

Na Figura 4.10 as três variáveis iniciais da estrutura *Route* definem o nó de origem da rota, através da variável **FromNode*, e o campo, através das variáveis *FromFieldIndex* e **FromField*. As três variáveis seguintes nessa estrutura, definem o nó de destino da rota, através da variável **ToNode*, e o campo, através das variáveis *ToFieldIndex* e **ToField*. Ainda nessa estrutura, a variável **fromFieldName* define o nome do campo de origem do sensor associado à rota, isto é, o nome do campo que atua como evento de origem da rota.

A ativação das rotas é realizada através do procedimento *Node_OnEventOutSTR*, implementado no arquivo *route.c*. Esse procedimento recebe como parâmetros ponteiros para o nó do tipo sensor e para o seu campo, informações correspondentes às variáveis **FromNode* e **FromField*, respectivamente, da Figura 4.10. A partir desses parâmetros as rotas associadas ao sensor e ao seu campo são ativadas.

Na realidade, antes da ativação das rotas, o mapeamento das interfaces dos componentes MPEG-4 para os sensores deve ser realizado. A ativação das rotas sem a realização prévia de alterações nos valores dos campos pertencentes aos sensores não altera a exibição do componente MPEG-4. Para realizar esse mapeamento, a interface das composições NCL, representando componentes MPEG-4, deve ser especificada contendo as informações sobre qual é o tipo do sensor a ela associado, qual é o nó da cena audiovisual correspondente ao sensor associado e, por fim qual é o evento do sensor (campo) que a interface representa.

Note que essas informações são necessárias à configuração dos sensores MPEG-4, pois nos componentes MPEG-4 podem existir vários tipos de sensores, que possuem comportamentos distintos. Esses sensores, por sua vez, podem estar associados a praticamente todos os nós de uma cena audiovisual. Por fim, um mesmo sensor pode definir vários eventos, através dos seus campos, como, por exemplo, o evento de clique do mouse, posicionamento do mouse, pressionamento do teclado etc.

A biblioteca *m4_nodes.h* define uma estrutura de dados para cada tipo de sensor que pode ser especificado no MPEG-4. A Figura 4.11 apresenta a estrutura relativa ao sensor denominado *B_TouchSensor*. Na figura, os campos desse sensor estão classificados como evento de saída (*eventOut*) ou campo exposto (*exposedField*).

```

Typedef struct _tagTouchSensor
{
    SFBool enabled;                /*exposedField*/
    SFVec3f hitNormal_changed;     /*eventOut*/
    SFVec3f hitPoint_changed;     /*eventOut*/
    SFVec2f hitTexCoord_changed;  /*eventOut*/
    SFBool isActive;             /*eventOut*/
    SFBool isOver;               /*eventOut*/
    SFTime touchTime;           /*eventOut*/
} B_TouchSensor;

```

Figura 4.11 – Estrutura do *TouchSensor* na ferramenta de exibição

Para acionar eventos em um componente MPEG-4, foi implementado um procedimento, denominado *M4T_NCLActiveNode*, dentro do arquivo *M4Term.c*. Esse procedimento é executado quando um evento deve ser acionado a partir de um relacionamento especificado em um documento NCL. A Figura 4.12 apresenta parte desse procedimento, que recebe como parâmetros: o nó relativo ao sensor (**node*), o nome do campo desse sensor (**eventName*), seu tipo (*tag*) e o valor a ser atribuído ao seu campo. Os três primeiros parâmetros são definidos pela

interface do componente relativo ao sensor, e o último, pelo papel do conector associado a essa interface. Após a execução desse procedimento, basta executar o procedimento *Node_OnEventOutSTR*, anteriormente citado, a fim de acionar as rotas associadas ao sensor modificado.

```
void M4T_NCLActiveNode(SFNode *node, const char *eventName, u32 tag, u32 value)
{
    if (tag == TAG_TouchSensor) {
        B_TouchSensor *ts = (B_TouchSensor *)node;

        if (!strcmp(eventName, "isActive")) ts->isActive =value;
        if (!strcmp(eventName, "isOver"))  ts->isOver =value;
    }
    ...
}
```

Figura 4.12 – Procedimento para acionar os sensores

Para demonstrar as interfaces que podem ser definidas em um componente MPEG-4, a ferramenta de exibição OSMO4 foi estendida com o objetivo de apresentar os sensores associados às rotas de uma cena audiovisual. Conhecer os sensores e, conseqüentemente, os eventos internos definidos em um componente MPEG-4 é necessário para que, durante a especificação do documento multimídia/hipermídia, o autor saiba quais são os nós internos ao componente MPEG-4 que contêm sensores associados.

A Figura 4.13 apresenta a exibição de um componente MPEG-4 nessa ferramenta, onde foi desenvolvido um menu que permite ao usuário acessar as possíveis interfaces do componente. Na figura, o componente MPEG-4 exibido contém cinco sensores, definindo relacionamentos com semântica de referência, que são acionados pelos eventos de clique do mouse sobre as legendas exibidas.



Figura 4.13 – Apresentação de um componente MPEG-4

A Figura 4.14 apresenta os sensores e as rotas existentes no componente MPEG-4 exibido pela ferramenta. Através de uma outra janela, as informações sobre o tipo do sensor, seu campo e o nó a ele relacionado, são apresentadas. Nessa janela, que representa as interfaces que podem ser estabelecidas em um documento multimídia/hipermídia, os eventos podem ser ativados através da sua seleção. Na atual versão, somente os campos do tipo booleano podem ser ativados, isto é, quando o evento é selecionado, o valor booleano do campo passa para verdadeiro. Para incluir campos de outros tipos, basta que seja definida uma interface para a passagem dos parâmetros relativos aos tipos dos campos.

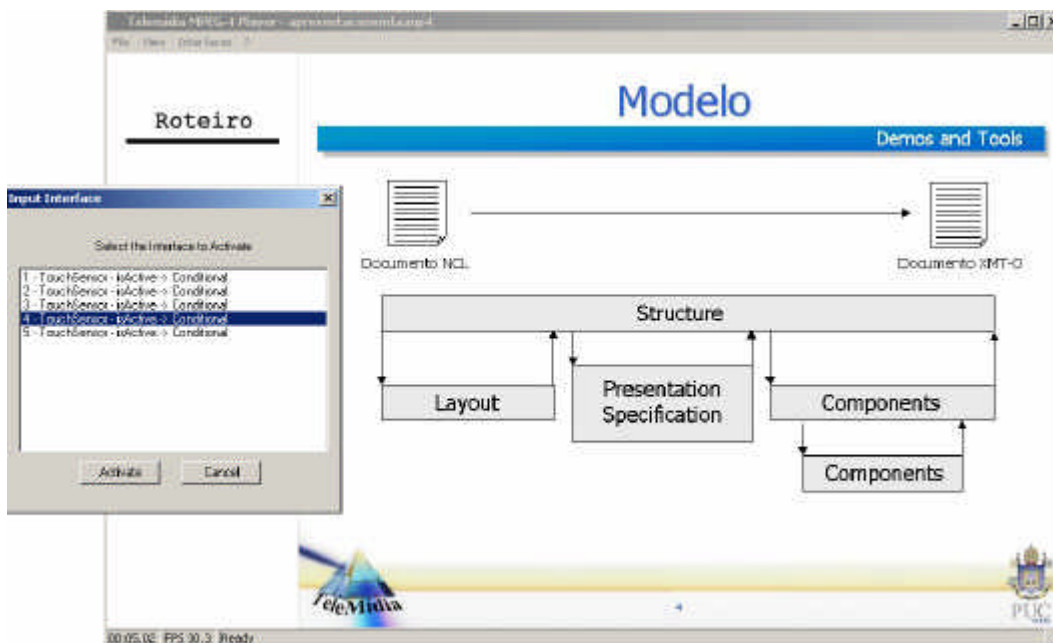


Figura 4.14 – Sensores definidos em um componente MPEG-4

Complementarmente, para mapear os eventos ocorridos em um componente MPEG-4 nas interfaces (portas) de composições NCL, representando esses componentes, outras adaptações na ferramenta devem ser realizadas. Essas adaptações têm como objetivo capturar as mudanças ocorridas nos campos dos sensores, provocadas por eventos de interatividade.

No arquivo *render2d.c*, o procedimento *R2D_ExecuteEvent* monitora todos os eventos que ocorrem na região de apresentação da ferramenta de exibição. Quando os eventos são detectados, procedimentos específicos definidos no arquivo *sensor_stacks.c* são executados, de acordo com o sensor envolvido. Nesses procedimentos, os campos dos sensores, relativos ao evento ocorrido, são ativados. Como exemplo, caso o procedimento *R2D_ExecuteEvent* observe a ação do clique do mouse sobre um nó da cena audiovisual, onde um sensor do tipo

TouchSensor é definido, o procedimento *OnTouchSensor*, definido no arquivo *sensor_stacks.c*, é acionado. Nesse caso, dentro desse procedimento, o campo “é Ativo” do sensor recebe o valor booleano verdadeiro.

Logo após o procedimento, relativo ao sensor ser executado, o procedimento *Node_OnEventOutSTR* é acionado, para executar as rotas associadas ao sensor. Nesse momento, métodos relativos ao mapeamento do sensor para as interfaces da composição NCL, representando o componente MPEG-4, devem ser executados. Esses métodos devem ser definidos na integração da ferramenta ao sistema HyperProp, de forma similar à integração apresentada na Seção 4.2. No entanto, essa integração, operacionalmente, é limitada pela restrição de comunicação entre as linguagens da ferramenta e da interface definida para comunicação com o formatador. Futuramente, pretende-se estender a ferramenta, através da definição de uma biblioteca de ligação dinâmica, que possa ser integrada com facilidade ao formatador do sistema HyperProp.