

O perfil EDTV, *Enhanced Digital TV*, é o principal perfil da versão 3.0 de NCL. Um documento, ou aplicação, nesse perfil é um XML que descreve o relacionamento entre objetos de mídia (textos, imagens, vídeos, etc.) no espaço e no tempo. A sintaxe completa do EDTV é definida através de uma série de esquemas XML, disponíveis em <http://www.ncl.org.br> — cf. Apêndice A para um resumo do conteúdo desses esquemas. Já a semântica associada a cada construção do perfil pode ser encontrada em [2, 1]. Neste capítulo, apresentamos uma visão geral da sintaxe e da semântica dos principais elementos que compõem um documento no perfil NCL EDTV, ou simplesmente, documento EDTV. Este capítulo está organizado da seguinte forma. A Seção 2.1 define a estrutura básica de um documento EDTV. Como veremos, esta estrutura é formada por dois grandes grupos de elementos: cabeçalho e corpo. Ainda nessa seção, são definidas as convenções notacionais utilizadas pelas próximas seções. A Seção 2.2 descreve em detalhes os elementos que constituem o corpo do documento. Finalmente, a Seção 2.3 apresenta os elementos que compõem o cabeçalho do documento. De modo geral, ordem de apresentação das seções e dos elementos dentro de cada seção foi escolhida de forma a minimizar o número de referências a conceitos ainda não definidos.

2.1

Estrutura Básica

Um documento EDTV é uma aplicação XML e, portanto, consiste de elementos aninhados com subelementos ordenados. Cada um desses elementos possui associados um nome (ou *tag*) e uma lista possivelmente vazia de atributos do tipo chave-valor. Para o propósito de representação do documento, podemos considerar um XML como sendo uma árvore ordenada em que cada nó possui um rótulo [8]. Em XML, os elementos também podem conter *strings* arbitrárias, porém esse tipo de construção não aparece no EDTV.

Todo documento EDTV possui um nó raiz `<ncl>`, que contém o cabeçalho `<head>` e o corpo `<body>` do documento. O cabeçalho contém as declarações globais do documento, e o corpo especifica os objetos e relacionamentos que o compõem. O atributo obrigatório *id* do elemento `<ncl>` especifica o identificador do documento. O elemento `<ncl>` possui ainda um atributo op-

cional¹ *xmlns* cujo valor é a URI (*Uniform Resource Identifier*) do esquema correspondente. As três *tags* `<ncl>`, `<head>` e `<body>` são obrigatórias — i.e. devem estar presentes em todo documento EDTV. A estrutura básica de um documento EDTV é apresentada na Figura 2.1.

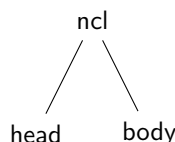


Figura 2.1 Estrutura básica de um documento EDTV.

No diagrama anterior, os nós da árvore representam os elementos da linguagem, enquanto as arestas e a disposição vertical dos nós indicam o sentido do relacionamento de inclusão — i.e. os elementos superiores incluem (ou contém) os inferiores. A cardinalidade do relacionamento é indicada através de subscritos do tipo “1...*n*.” Nós que não possuem subscrito tem cardinalidade igual a um. Por exemplo, na Figura 2.1 o elemento `<ncl>` contém exatamente um elemento `<head>` e um elemento `<body>`. Em geral, a ordem em que os elementos aparecem na árvore é irrelevante. As únicas exceções são os elementos `<bind>`, filho do elemento `<causalConnector>`, e `<bindRule>`, filho dos elementos `<switch>` e `<descriptorSwitch>`.

2.2 Elementos do Corpo

O corpo do documento EDTV, delimitado pela elemento `<body>`, define o conteúdo, a estrutura, e a dinâmica da apresentação associada ao documento. O conteúdo — i.e. aquilo que é apresentado — é definido através dos nós de mídia e âncoras. A estrutura do documento é definida através dos nós de composição. A dinâmica da apresentação, incluindo o suporte à interatividade, é definida através dos elos e portas. De modo geral, podemos dividir os elementos que aparecem no corpo de um documento EDTV em três grupos: nós de conteúdo, nós de composição e nós de ligação (ou elos). A seguir, apresentamos os elementos que compõem cada um desses grupos.

¹De agora em diante vamos omitir o adjetivo “opcional”. A menos que indicado de outra forma, atributos definidos sem o adjetivo “obrigatório” devem ser considerados opcionais.

2.2.1

Nós de conteúdo

Em geral, os nós de conteúdo representam algum conteúdo multimídia externo — e.g. arquivo de imagem, áudio, etc. — que é referenciado pelo documento. Nós de conteúdo são definidos através de elementos `<media>`. De fato, a *tag* `<media>` define o que chamamos de objeto de mídia. Um *objeto de mídia* é um conteúdo associado a um conjunto de propriedades, elemento `<property>`, e âncoras, elemento `<area>`. Todo objeto de mídia possui um identificador único definido através do atributo obrigatório *id*. O conteúdo do objeto é especificado através do atributo *src*. A Figura 2.2 apresenta a estrutura de um objeto de mídia EDTV.

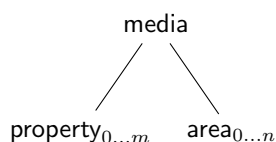


Figura 2.2 Estrutura de um objeto de mídia EDTV.

Os demais atributos do elemento `<media>` são: *type*, que define o tipo do conteúdo associado; *refer* e *instance* que indicam se o objeto é, na verdade, uma referência para algum outro objeto; e *descriptor*, que permite que diversos objetos compartilhem uma única declaração de valores iniciais de propriedades (cf. Seção 2.3.3).

Cada objeto de mídia possui um conjunto de propriedades pré-definidas que, em geral, determina a aparência do objeto durante a apresentação do documento. Toda propriedade possui um nome, atributo obrigatório *name*, e um valor, atributo *value*. As propriedades pré-definidas possuem um valor *default*. Portanto, ao escrever

```

<media id="sample1" src="image.png">
  <property name="transparency" value="50%"/>
</media>
  
```

estamos declarando um objeto de mídia com identificador “sample1” cujo conteúdo é uma imagem que será apresentada com a metade da transparência da imagem original. Se omitirmos o elemento `<property>` no trecho de código anterior, a imagem será apresentada com a transparência *default* (zero). Esse tipo de declaração corresponde, portanto, à inicialização da propriedade com algum valor possivelmente diferente do *default*. A Tabela 2.1 apresenta algumas propriedades pré-definidas do EDTV e respectivos valores iniciais.

Além das propriedades pré-definidas, que variam de acordo com o tipo do objeto, o EDTV permite a declaração de propriedades adicionais, cuja se-

Tabela 2.1 Algumas propriedades pré-definidas do EDTV.

Nome	Valor Inicial	Descrição
top	0	posicionamento a partir do topo
bottom	0	posicionamento a partir da base
left	0	posicionamento a partir da esquerda
right	0	posicionamento a partir da direita
width	0	largura do objeto
height	0	altura do objeto
transparency	0%	transparência do objeto
visible	true	indica se o objeto está visível ou não

mântica é dada pelo usuário. Nesse caso, a propriedade funciona como uma variável geral, que pode ser utilizada pelo autor do documento para guardar *strings* arbitrárias. É o atributo obrigatório *name* que determina se uma dada declaração refere-se a uma propriedade pré-definida ou não.

Objetos de mídia podem conter âncoras, elemento `<area>`, em que cada âncora representa um segmento do conteúdo do objeto. Por *default*, todo objeto de mídia define uma âncora implícita, chamada de âncora *lambda* (λ), que denota o conteúdo do objeto como um todo. Toda âncora possui um atributo obrigatório *id* cujo valor é o identificador da âncora. Os demais atributos — *coords*, *begin*, *end*, *text*, *position*, *first*, *last* e *label* — determinam o tipo da âncora em questão. Por exemplo, no trecho de código

```
<media id="sample2" src="audio.mp3">
  <area id="solo" begin="30s" end="45s"/>
</media>
```

o elemento `<area>` com *id* “solo” define uma âncora temporal, i.e. um intervalo de tempo, no conteúdo do objeto de mídia “sample2”. Quando a âncora é apresentada, apenas o intervalo especificado é reproduzido. O perfil EDTV também define âncoras espaciais que permitem selecionar trechos de textos, ou regiões de imagens ou vídeos.

Propriedades e âncoras são o que chamamos de *interfaces*. Toda interface define eventos que podem ser relacionados através de elos (ou *links*), cf. Seção 2.2.3. Observe que o objeto de mídia em si não define uma interface. Se quisermos denotar o conteúdo do objeto como um todo devemos usar a âncora λ . No entanto, quando a interface do objeto não é especificada, e.g. durante a definição de um elo ou porta, assume-se que se trata da âncora λ .

2.2.2

Nós de composição

Os nós de composição permitem agrupar, sob uma mesma entidade, objetos de mídia e elos (ou regras) correspondentes. O principal nó de composição definido pelo EDTV é o contexto, elemento `<context>`. Contextos podem conter objetos de mídias, elos e outros contextos que por sua vez podem conter outros objetos de mídia, elos e contextos. A única restrição é que um contexto não pode conter, recursivamente, algum contexto que o contenha. Mais precisamente, a estrutura definida por um contexto e seus filhos é um grafo dirigido acíclico (ou DAG, *Directed Acyclic Graph*). O elemento `<context>` possui apenas dois atributos: *id*, atributo obrigatório que define o identificador do contexto; e *refer*, que indica se o contexto é, na verdade, uma referência para algum outro objeto.

Além dos objetos de mídia, elos e outros contextos, um contexto pode conter portas, propriedades e *switches* (o outro tipo de nó de composição definido pelo EDTV). A Figura 2.3 apresenta a estrutura de um contexto EDTV. O elemento `<body>` possui a mesma estrutura do contexto, a única diferença é que o seu atributo *id* é opcional.

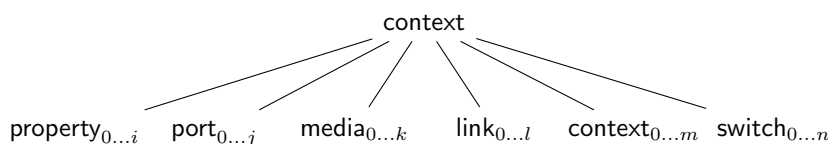


Figura 2.3 Estrutura de um contexto EDTV.

O contexto possui dois tipos de interface: propriedades e portas. As propriedades, elemento `<property>`, funcionam como variáveis locais do contexto. O comportamento dessas propriedades é análogo ao das propriedades sem-semântica dos objetos de mídia. Já as portas, elemento `<port>`, permitem que interfaces definidas dentro do contexto — e.g. mídias, outros contextos, etc. — sejam referenciadas por elos e portas de fora do contexto. Mais precisamente, por elos e portas definidos no contexto imediatamente acima na hierarquia. Dessa forma, dizemos que uma porta “mapeia” ou “torna visível”, através do seu atributo obrigatório *component*, um objeto definido dentro contexto. Por exemplo, no trecho de código

```

<context id="outter">
  <context id="inner">
    <port id="p1" component="sample3"/>
    <media id="sample3"/>
    <media id="sample4"/>
  
```

```

    </context>
    ...
  </context>

```

os elos e portas definidos no contexto “outter” não podem referenciar diretamente os objetos de mídia “sample3” e “sample4”. Porém, esses mesmos elos e portas podem referenciar a porta “p1”, que torna visível a interface λ do objeto de mídia “sample3”. Além do *component*, o outro atributo obrigatório do elemento `<port>` é o *id*, que especifica o identificador da porta. O elemento `<port>` possui ainda um atributo *interface*, usado nos casos em que é preciso mapear uma interface específica de algum objeto, e.g. uma propriedade de um objeto de mídia.

Além de mapear interface, as portas também definem a semântica da apresentação do contexto. Apresentar um contexto significa apresentar todas as interfaces mapeadas por suas portas. Dessa forma, ao definir uma porta estamos: (i) tornando a interface visível; e, ao mesmo tempo, (ii) associando a apresentação da interface à apresentação do contexto. Observe que é impossível apresentar um contexto que não possui portas, visto que o estado da apresentação do contexto depende exclusivamente do estado da apresentação de seus componentes. Se esses componentes não puderem ser acessados, i.e. se o estado da sua apresentação não puder ser modificado, então não há como interferir no estado da apresentação do contexto. Na Seção 2.2.3, os mecanismos que controlam a apresentação de interfaces são discutidos em detalhes.

O outro tipo de nó de composição definido pelo EDTV é o *switch*, elemento `<switch>`. Em termos de estrutura, podemos dizer que o *switch* é similar ao contexto. Ambos possuem os mesmos atributos e compartilham a mesma restrição de composicionalidade, i.e. o fato de ser um DAG. A grande diferença entre ambos está na semântica. No lugar de portas e elos, o *switch* utiliza regras para controlar a apresentação dos seus componentes. As regras são expressões Booleanas definidas no cabeçalho do documento (cf. Seção 2.3.6). Essas expressões fazem referência à variáveis globais mantidas num objeto de mídia especial chamado *settings* (cf. Seção 2.2.4). A Figura 2.4 apresenta a estrutura de um *switch* EDTV.

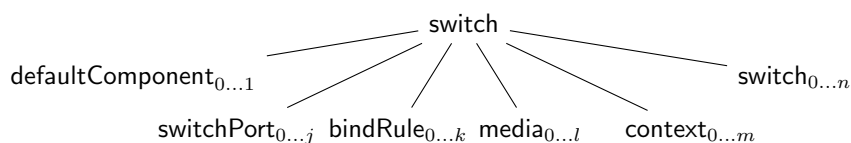


Figura 2.4 Estrutura de um *switch* EDTV.

O elemento `<bindRule>` associa uma regra, definida no cabeçalho, uma

interface — e.g. objeto de mídia, contexto, etc. — definida no corpo do `<switch>`. Os atributos obrigatórios *rule* e *constituent* do elemento `<bindRule>` contém, respectivamente, o identificador da regra e o identificador do objeto associado. Quando o *switch* é apresentado, as regras são avaliadas em sequência, uma a uma. Assim que uma delas é avaliada como verdadeira, o processo de avaliação é interrompido e o componente associado a regra é apresentado. Por exemplo, no trecho de código

```
<switch id="switch0">
  <bindRule rule="r1" constituent="sample5"/>
  <bindRule rule="r2" constituent="sample6"/>
  <media id="sample5"/>
  <media id="sample6"/>
</switch>
```

assim que o *switch* “switch0” é apresentado, a regra “r1” é avaliada. Se “r1” for avaliada como verdadeira, a interface λ do objeto de mídia “sample5” é apresentada. Caso contrário, a regra “r2” é avaliada. Se “r2” for avaliada como verdadeira, a interface λ do objeto de mídia “sample6” é apresentada, senão nada é apresentado. Observe que ordem de especificação das regras determina o comportamento do *switch*. De fato, esse é o único caso em que a ordem dos elementos influencia no resultado da apresentação. O perfil EDTV define ainda o elemento `<defaultComponent>` que pode ser usado para definir um componente *default*, apresentado caso nenhuma regra seja avaliada como verdadeira.

Assim como o contexto, o *switch* define portas, elemento `<switchPort>`, que tornam visíveis para elementos externos as interfaces definidas no corpo do *switch*. A diferença entre o `<switchPort>` e o `<port>` do contexto é que as interfaces expostas pelo `<switchPort>` estão atreladas às regras do *switch*. Ou seja, uma interface mapeada por um *switchPort* é apresentada apenas quando a regra associada é avaliada como verdadeira.

2.2.3

Nós de ligação (elos)

Os elos (ou *links*) são o principal mecanismo para criação de apresentações dinâmicas e interativas no NCL EDTV. Através deles, é possível especificar, de forma declarativa, um conjunto de condições e ações associadas, que são disparadas sempre que as essas condições forem satisfeitas. Esse tipo de relacionamento entre condições e ações é chamado de relacionamento de causa e efeito.

No EDTV, ações e condições são, na verdade, eventos associados a interfaces (i.e. âncoras ou propriedades) dos objetos que compõem o documento. Cada âncora define exatamente dois eventos²: um *evento de apresentação*, que indica (ou controla) o andamento da exibição do objeto; e um *evento de seleção*, que indica o andamento da seleção do objeto. Além disso, cada propriedade define um *evento de atribuição*, que indica (ou controla) o andamento da atribuição de um valor a uma propriedade. Todo evento define uma máquina de estados similar a apresentada na Figura 2.5. Na figura, os vértices representam os possíveis estados do evento: *occurring*, o evento está ocorrendo; *paused*, o evento está pausado; e *sleeping* o evento está dormindo ou (parado). E as arestas representam as transições entre os estados da máquina. O símbolo ‘|’ no rótulo da aresta indica um conjunto de alternativas.

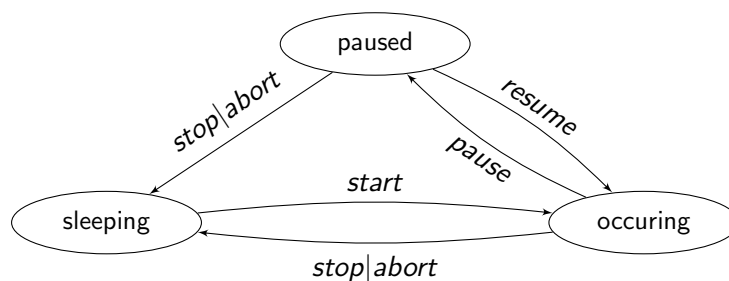


Figura 2.5 Máquina de estados de evento.

Condições e ações nada mais são do que as transições entre os estados. Uma condição é uma transição aguardada, e uma ação é uma transição induzida na máquina de alguma interface. Dessa forma, os elos conectam transições de máquinas de estado de eventos. No EDTV, para construir um elo entre duas interfaces o autor precisa especificar quatro parâmetros: (i) a interface observada, (ii) a interface a ser acionada, (iii) a transição aguardada, e (iv) a transição a ser induzida. De fato, os dois últimos parâmetros são especificados no conector, elemento <connector>, e não no elo, elemento <link>. O conector permite criar combinações entre transições aguardadas e induzidas sem especificar a quais interfaces essas transições pertencem. Dessa forma, todo elo referencia algum conector. A principal motivação por trás da separação entre <link> e <conector> é permitir que um mesmo conector possa ser reusado por diversos elos.

O motivo da separação entre a especificação das interfaces e a especificação das transições é que, com isso, é possível reusar um mesmo conector em elos distintos. (Conectores são discutidos em detalhe na Seção 2.3.5.)

²As interfaces associadas à nós de composição possuem um evento adicional, chamado de *evento de composição*, que indica (ou controla) o andamento da exibição da estrutura da composição.

A Figura 2.6 apresenta a estrutura de um elo EDTV. Todo elo define o atributo obrigatório *xconnector* que contém o identificador do conector associado. Um elo é basicamente, um conjunto de elementos `<bind>`. Cada `<bind>` associa uma interface, atributo obrigatório *component*, a um papel definido no conector, atributo obrigatório *role*. O papel determina o tipo da transição (condição ou ação) e a transição propriamente dita — e.g. *start*, *pause*, etc. Os elementos `<linkParam>` e `<bindParam>` permitem que o autor especifique no elo parâmetros a serem passados para o conector.

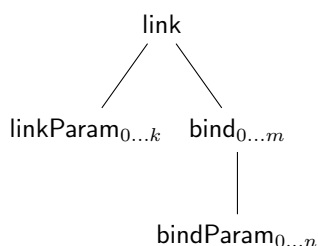


Figura 2.6 Estrutura de um elo EDTV.

2.2.4 Nó settings

O nó *settings* é um objeto de mídia especial que contém as variáveis globais do documento. Só pode haver um *settings* por documento. Apesar das variáveis do *settings* caráter global, o nó em si funciona como uma `<media>` qualquer. A Tabela 2.2 apresenta algumas variáveis de ambiente pré-definidas que podem ser exportadas através do nó settings.

Tabela 2.2 Atributos opcionais do elemento `<transition>`.

Nome	Descrição
system.language	linguagem de áudio
system.caption	linguagem de caption
system.screenSize	tamanho da tela
system.CPU	desempenho da CPU em MIPS
system.memory	memória em <i>megabytes</i>
system.operatingSystem	tipo de sistema operacional
system.luaVersion	versão da máquina Lua

2.3

Elementos do Cabeçalho

Os elementos do cabeçalho contém as declarações globais do documento. No EDTV, esses elementos são agrupados em *bases* de acordo com o seu tipo. Ao todo são definidas seis bases de elementos: base de documentos importados, base de regiões, base de descritores, base de transições, base de conectores e base de regras. As seções seguintes apresentam a sintaxe e semântica dos elementos que compõem cada uma dessas bases.

2.3.1

Base de documentos importados

A base de documentos importados, delimitada pelo elemento `<importedDocumentBase>`, contém as declarações de importação, i.e. referencias à documentos externos, do documento. A Figura 2.7 apresenta a estrutura de uma base de documentos importados do EDTV.

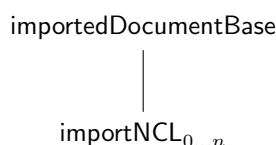


Figura 2.7 Estrutura de uma base de documentos importados EDTV.

Cada elemento `<importNCL>` contido na base aponta para algum documento externo e define um rótulo associado. Os atributos obrigatórios *documentURI* e *alias* do `<importNCL>` definem, respectivamente, a URI do documento referenciado e o rótulo associado. Na importação, as bases do documento importado são fundidas com as bases do documento importador. Ou seja, as bases de regiões, descritores, regras, etc. são incorporadas às bases correspondentes no documento importador (pai). Além disso, as mídias do documento importado tornam-se disponíveis para reúso através do atributo *refer*. No documento pai, o *id* dos elementos importados é o resultado da concatenação entre o rótulo definido no atributo *alias* do `<importNCL>` e o *id* original do elemento. O elemento `<importBase>`, filho das demais bases, pode ser usado para importar bases específicas ao invés de importar todas de uma vez.

2.3.2 Regiões

O elemento `<region>` define os parâmetros que determinam a dimensão e o posicionamento do objeto de mídia na tela. Além do *id*, a região possui os seguintes atributos: *left*, *right*, *top*, *bottom*, *height*, *width* e *zIndex*. Os quatro primeiros determinam o posicionamento do objeto, *height* e *width* determinam a sua dimensão, e o último, *zIndex*, especifica a precedência de sobreposição de uma região em relação as outras regiões. Regiões podem conter outras regiões. Nesse caso, os atributos do elemento filho (com exceção do *zIndex*) podem ser especificados de forma relativa aos atributos correspondentes na região pai. As regiões de um documento estão contidas nas bases de regiões do documento. Observe que pode haver mais de uma base de regiões por documento. A Figura 2.8 apresenta a estrutura de uma base de regiões EDTV.

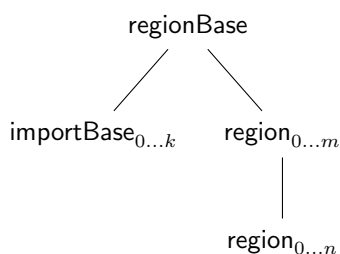


Figura 2.8 Estrutura de uma base de regiões EDTV.

2.3.3 Descritores

O elemento `<descriptor>` (cf. Seção 2.3.3) define os valores iniciais para as propriedades audiovisuais dos objetos de mídia associados. Esses valores podem ser especificadas através de atributos do elemento `<descriptor>` ou como parâmetros desse elemento, i.e. elemento `<descriptorParam>`. Os descritores do documento estão contidos na base de descritores de documento. Todo documento possui no máximo uma base de descritores. A Figura 2.9 apresenta a estrutura de uma base de descritores do EDTV.

O elemento `<descriptorSwitch>` permite associar a um objeto de mídia um conjunto de descritores alternativos. Cada descritor do conjunto possui uma regra associada. Quando o objeto de mídia é apresentado, as regras do `<descriptorSwitch>` associado ao objeto são avaliadas em sequência, uma a uma. Assim que uma delas é avaliada como verdadeira, o processo de avaliação é interrompido e o descritor correspondente é utilizado para inicializar as propriedades do objeto de mídia.

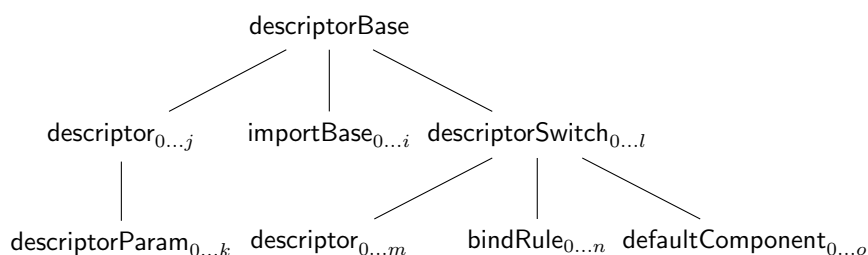


Figura 2.9 Estrutura de uma base de descritores EDTV.

2.3.4 Transições

As transições, elemento <transition>, especificam feitos visuais que podem ser associados ao início ou ao fim da apresentação de uma interface. Uma transição possui os atributos obrigatórios *id* e *type* que definem, respectivamente, o identificador e o tipo da transição. Além desses atributos, cada transição pode definir uma série de atributos opcionais listados na Tabela 2.3.

Tabela 2.3 Atributos opcionais do elemento <transition>.

Nome	Descrição
subtype	subtipo da transição
dur	duração da transição
startProgress	valor inicial do progresso da transição
endProgress	valor final do progresso da transição
direction	direção da transição (“forward” ou “backward”)
fadeColor	cor final ou inicial do efeito <i>fade</i>
horRepeat	número de repetições no eixo horizontal
vertRepeat	número de repetições no eixo vertical
borderWidth	largura da borda em <i>pixels</i>
borderColor	cor da borda

As transições de um documento EDTV estão contidas na base de transições do documento, declarada no cabeçalho. Todo documento possui no máximo uma base de transições. A Figura 2.10 apresenta a estrutura da base de transições de um documento EDTV.

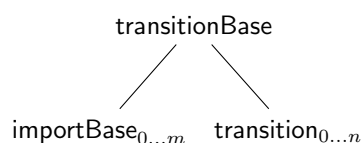


Figura 2.10 Estrutura de uma base de transições EDTV.

2.3.5

Conectores

O elemento `<connectorBase>` delimita a base de conectores do documento. Todo conector, elemento `<connector>`, define uma relação do tipo causa-efeito que pode ser usada por um ou mais elos. Nesse tipo de relação, um conjunto de condições deve ser satisfeito para que ações associadas sejam executadas. Uma condição simples, elemento `<simpleCondition>`, representa a “espera” de uma transição na máquina de estado de evento de alguma interface. Essas condições são usadas para construir condições compostas, elemento `<compoundCondition>`, que também pode conter testes de propriedade, elemento `<assessmentStatement>`. Ações simples, elemento `<simpleAction>`, representam o disparo de uma transição na máquina de estado de evento de alguma interface. Como na condição, as ações simples podem ser compostas através do elemento `<compoundAction>`. Quando o conjunto de eventos (e testes) esperados pela condição ocorre, a ação associada é executada, ou seja, o conjunto de transições especificado pela ação é disparado.

Os conectores do documento estão contidos na base de conectores. Todo documento EDTV possui no máximo uma base de conectores. A Figura 2.11 apresenta a estrutura de uma base de conectores EDTV.

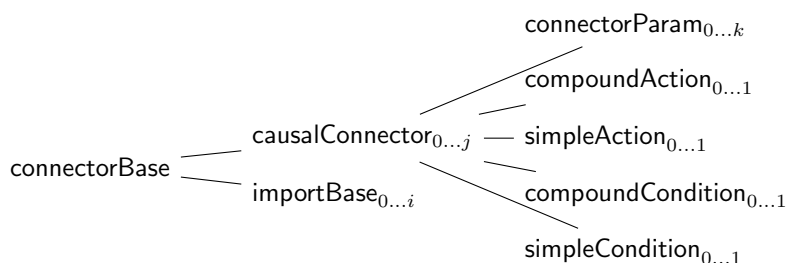


Figura 2.11 Estrutura de uma base de conectores EDTV.

2.3.6

Regras

A base de regras do documento é definida pelo elemento `<ruleBase>`. As regras simples, elemento `<rule>` definem testes Booleanos que referenciam as propriedades declaradas no objeto de mídia *settings* (cf. Seção 2.2.4). Regras simples podem ser reunidas em regras compostas através do elemento `<compositeRule>`. As regras definidas na base de regras podem ser referenciadas por elementos `<switch>` ou `<descriptorSwitch>`. A Figura 2.12 apresenta a estrutura da base de regras de um documento EDTV. Observe que pode haver no máximo uma base de regras por documento.

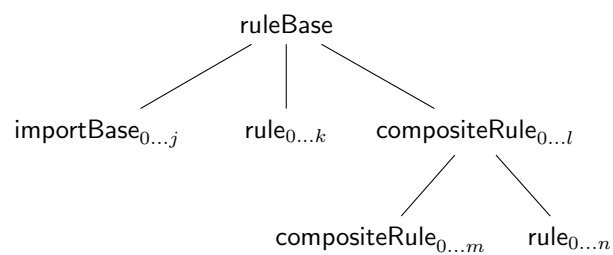


Figura 2.12 Estrutura de uma base de regras EDTV.