

4

Representação de Aplicações Hipermídia Declarativas – Aplicações NCL

Este capítulo trata da tradução das especificações do autor para o HTG, a partir de uma sintaxe de autoria específica, a sintaxe da linguagem NCL. O controle do sincronismo temporal das aplicações NCL é necessário, tanto para garantir a qualidade das apresentações para TV quanto para uma melhor utilização dos recursos existentes nos sistemas de TV digital interativa.

Voltada para suporte à autoria declarativa, NCL oferece várias abstrações com o objetivo de facilitar essa tarefa (Neto, 2009; Muchaluat-Saade, 2002; Soares, 2010b). Essas facilidades, contudo, podem tornar difícil o controle da apresentação, uma vez que os instantes temporais para a ocorrência dos eventos existentes nas aplicações precisam ser calculados a partir de estruturas com alto nível de abstração. Através do uso do HTG, as especificações temporais das aplicações NCL podem ser simplificadas, sob o ponto de vista do controle da apresentação, como será discutido ao longo deste capítulo.

4.1. Módulos, Perfis e Aplicações NCL

NCL 3.0 segue uma abordagem modular, onde cada módulo agrupa elementos e atributos XML (W3C, 2008a) semanticamente relacionados. Esses módulos, por sua vez, podem ser agrupados em diferentes perfis de linguagem, construídos para atender a uma demanda específica. Este capítulo explora aplicações do perfil avançado para TV (perfil EDTV) (Soares, 2006a), padronizado para a especificação de aplicações declarativas no ISDB-T_B e para IPTV segundo o ITU-T.¹⁰

O perfil avançado contém todas as funcionalidades da NCL para a especificação de aplicações nos STVDI, que podem ser, em grande parte,

¹⁰A definição completa dos módulos de NCL 3.0 e dos seus principais perfis é realizada utilizando XML Schema (XMLSchema, 2004a; XMLSchema, 2004b; XMLSchema, 2004c). Essas definições estão disponíveis no site web da NCL (www.ncl.org.br/NCL3.0).

descritas através de uma aplicação proposta como exemplo. Nessa aplicação, um filme, uma animação sobre um jogador de futebol, é recebida por difusão nos receptores de TV. Também por difusão, diversos outros objetos de mídias são recebidos e devem ser exibidos, sincronamente, de acordo com a especificação da aplicação em NCL que é, na verdade, mais um dos conteúdos recebidos por difusão.

Durante toda a apresentação da animação (filme), uma música de fundo é executada. Após 12 segundos do início dessa animação, uma foto real do jogador (imagem), ator principal retratado na animação, é apresentada na tela. Dependendo do tamanho da tela, informação que em NCL pode ser obtida através de uma variável, a duração da apresentação dessa imagem irá variar. Serão 20 segundos de apresentação se essa tela tiver um tamanho mínimo e 10 segundos caso contrário, evitando que detalhes importantes da animação permaneçam encobertos.

Em outro momento da apresentação da animação, uma publicidade sobre equipamentos esportivos, formada pela imagem de uma chuteira, pode ser apresentada, mas apenas se ações interativas forem permitidas. Novamente, essa possibilidade é definida a partir de uma outra variável da aplicação.

Caso a imagem da chuteira seja apresentada, esse evento será iniciado aos 45 segundos do início da animação e será encerrado quando essa duração atingir 51 segundos. Essa apresentação indica aos usuários a possibilidade da ocorrência de outros eventos, disparados em função da ação interativa. Para dar maior destaque a essa possibilidade, a posição da imagem da chuteira irá variar na tela durante o seu período de exibição.

Nessa aplicação, o usuário pode selecionar a imagem da chuteira pressionando o botão vermelho do controle remoto (mesma cor da chuteira) e, como resultado, sua apresentação será encerrada, o vídeo da animação redimensionado (através da manipulação de outras variáveis da aplicação), uma propaganda e um formulário serão exibidos na tela, todos sobre uma imagem de fundo. Para ilustrar esse comportamento, a Figura 4.1 mostra a aplicação em dois momentos: primeiro, quando a imagem da chuteira é apresentada e, depois, caso essa imagem seja selecionada.



Figura 4.1 - Apresentação de uma aplicação NCL com interatividade.

A propaganda, apresentada como resultado do evento de seleção da imagem da chuteira, é formada por um vídeo e por um formulário. Esse formulário pode ser uma aplicação XHTML em português ou outra aplicação em inglês, dependendo, novamente, do valor de outra variável que contém o idioma do usuário. O final da apresentação do vídeo da propaganda irá provocar o término da apresentação do formulário e da imagem de fundo, além de redimensionar a animação ao seu tamanho original.

A aplicação descrita nos parágrafos anteriores contém várias especificações associadas ao controle do sincronismo e à realização de adaptações. A Tabela 4.1 resume essas especificações, com pelo menos um exemplo da aplicação.

Características	Exemplo (na aplicação NCL)
Sincronismo no tempo (sem interação)	Apresentação de uma imagem (foto do jogador) é iniciada, quando atingido os 12 segundos do início da apresentação do vídeo da animação.
Adaptação da apresentação	Apresentação de uma imagem (foto do jogador) pode durar 10 ou 20 segundos, dependendo das características do ambiente (tamanho da tela).

Sincronismo no tempo (com condição)	Apresentação de uma imagem (chuteira) é iniciada, quando a animação atinge 45 segundos a partir do seu início; e finalizada no máximo aos 51 segundos da duração da animação. Essa imagem somente será apresentada caso a interatividade seja permitida (outra característica do ambiente de apresentação).
Animação	Durante a apresentação de uma imagem (chuteira) sua posição na tela irá variar.
Sincronismo no tempo (com interação)	Seleção de uma imagem (chuteira) encerra a apresentação desse objeto e inicia outros eventos.
Adaptação do conteúdo	Uma aplicação (formulário) será apresentada, após escolha entre duas alternativas. Essa escolha depende das características do ambiente de apresentação (idioma do usuário).

Tabela 4.1 - Principais características relacionadas ao sincronismo e à adaptação encontradas na aplicação NCL exemplo.

Além das funcionalidades da linguagem NCL listadas na Tabela 4.1, outras funcionalidades, como a estruturação da aplicação pelo uso de contextos e a possibilidade de reuso das suas entidades (ABNT, 2009), devem ser consideradas na construção de um HTG. Todas essas características da linguagem serão discutidas nas próximas seções. A especificação completa da aplicação descrita nesta seção é apresentada no final deste capítulo, na Seção 4.6.

Nas próximas seções, a construção do HTG do exemplo citado é realizada paulatinamente.

4.2. Objetos de Mídia e Interfaces

Na NCL, os tipos básicos de objetos de mídia são definidos no módulo *Media*. Para a especificação dos objetos, esse módulo define o elemento *<media>*, contendo um conjunto de atributos para, entre outras informações (ABNT, 2009), identificar cada objeto (atributo *id*) e o seu conteúdo (atributo *src*).

Também na NCL, sobre cada objeto de mídia podem ser especificadas âncoras de conteúdo, correspondentes a uma porção das unidades de informação que compõem o conteúdo de um objeto. Essas âncoras são definidas no módulo *MediaContentAnchor*, através do elemento `<area>`. Diferentes propriedades também podem ser especificadas sobre um objeto de mídia. A especificação de cada propriedade é realizada através do elemento `<property>`, definido no módulo *PropertyAnchor*.

A Figura 4.2 apresenta os principais objetos de mídia da aplicação NCL proposta como exemplo. Esses objetos são: imagem de fundo (“background” – linha 35), animação (“animation” – linhas 36 a 39), música de fundo (“music” – linha 40), foto do jogador (“photo” – linha 41), imagem da chuteira (“icon” – linhas 46 a 48), vídeo da propaganda (“shoes” – linha 49) e as propriedades globais da aplicação (“globalVar” – linhas 50 a 52).

```

...
35. <media id="background" src="background.png" descriptor="backgroundDesc"/>
36. <media id="animation" src="animGar.mp4" descriptor="screenDesc">
37.   <area id="segPhoto" begin="12s"/>
38.   <area id="segIcon" begin="45s" end="51s"/>
39. </media>
40. <media id="music" src="choro.mp3" descriptor="audioDesc"/>
41. <media id="photo" src="photo.png" descriptor="photoDesc">
...
46. <media id="icon" src="icon.png" descriptor="iconDesc">
47.   <property name="left"/>
48. </media>
49. <media id="shoes" src="shoes.mp4" descriptor="shoesDesc"/>
50. <media id="globalVar" type="application/x-ginga-settings">
51.   <property name="channel.interactivity" value="true"/>
52. </media>
...

```

Figura 4.2 - Objetos de mídia e suas interfaces na aplicação NCL exemplo.

Na construção do HTG, para cada objeto devem ser construídos dois vértices, correspondentes ao início e ao fim do seu evento de apresentação.¹¹ Esses vértices devem estar unidos por arestas, caso o objeto tenha uma duração associada (implícita ou explícita). A duração implícita é obtida pelo próprio conteúdo da mídia, enquanto a explícita é definida na especificação da aplicação. Tão logo essa duração seja conhecida, ela deve ser especificada como condição de

¹¹ Na conversão de uma aplicação NCL, toda transição de fim de um evento deve ser representada pela ação “stop”, uma vez que essa linguagem não define formas diferentes para o término de um evento (“natural end” e “stop”) (Soares, 2009a).

uma aresta. Uma parte do HTG que representa as transições dos eventos de apresentação para os objetos e as interfaces descritas na Figura 4.2 é apresentado na Figura 4.3.¹²

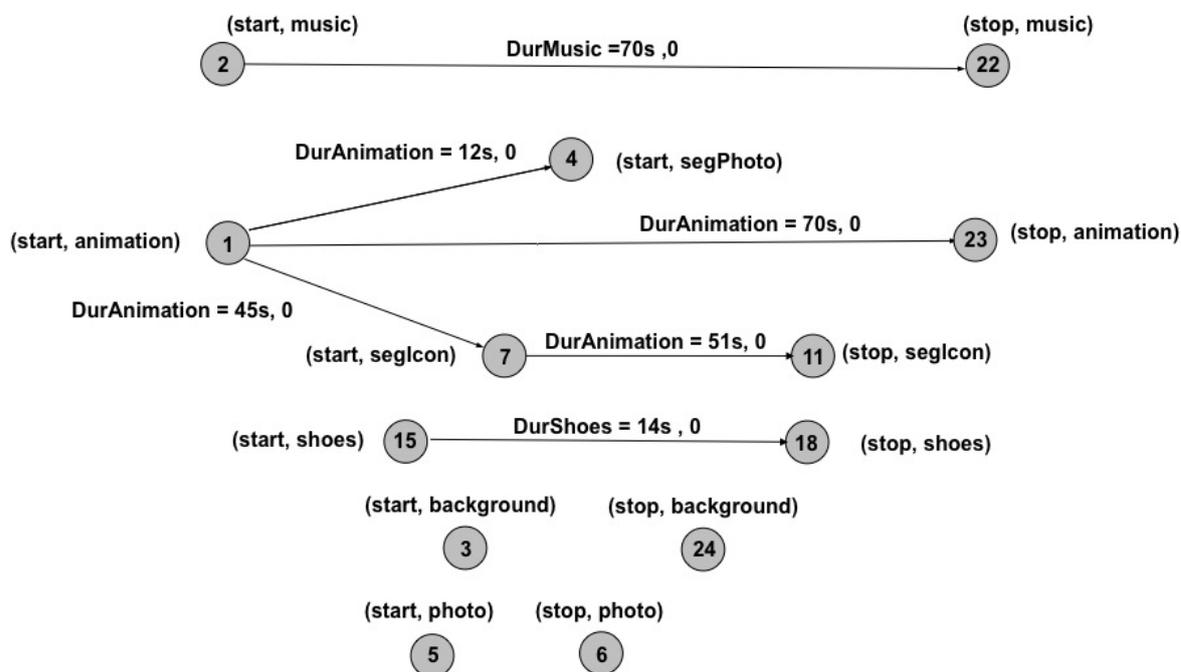


Figura 4.3 - Parte do HTG que representa os objetos e suas interfaces descritos na Figura 4.2.

As âncoras de conteúdo de uma aplicação NCL, que têm uma duração explícita, também devem ser representadas, cada uma, por dois vértices no HTG. Esses vértices correspondem ao início e ao fim do evento de apresentação sobre cada âncora. Esses pares de vértices também devem estar unidos, cada par, por uma aresta que tem como condição a duração desse evento sobre a âncora. Esse é o caso da aresta entre os vértices 7 e 11 da Figura 4.3 (“segIcon”).

Os relacionamentos de um objeto de mídia com as suas âncoras de conteúdo também devem estar especificados no HTG. Esse é o caso da aresta entre os vértices 1 e 4 e também da aresta entre os vértices 1 e 7 na Figura 4.3. Essas arestas têm como condição uma porção temporal da duração do conteúdo do objeto de mídia “animation”.

Os demais vértices da Figura 4.3 representam transições de início e de fim do evento de apresentação que podem ocorrer sobre os demais objetos de mídia especificados na Figura 4.2 (imagem de fundo, foto do jogador, imagem da

¹² Para facilitar o entendimento das figuras que representam parte de um HTG, a descrição do evento de apresentação será sempre suprimida neste capítulo. Isso acontece nas triplas correspondentes aos vértices e também nas variáveis que representam a duração desse evento.

chuteira). As ocorrências dessas transições são especificadas em outras partes da aplicação NCL, como é o caso, por exemplo, da apresentação do objeto correspondente à foto do jogador, que possui uma duração explícita, discutida na próxima seção.

4.3. Adaptação da Apresentação e Adaptação do Conteúdo

O autor pode definir, através da NCL, formas alternativas para a apresentação de um mesmo conteúdo (adaptação da apresentação). Também é possível ao autor definir conteúdos alternativos para apresentação (adaptação do conteúdo). Tanto a forma de apresentação quando o conteúdo a ser apresentado são escolhidos em tempo de apresentação da aplicação.

Nas aplicações NCL, a escolha entre conteúdos alternativos ou formas de apresentação alternativas é realizada através da avaliação de regras que, quando satisfeitas, selecionam uma opção. Para a especificação dessas regras, o módulo *TestRule* define o elemento *<ruleBase>*, que agrupa um conjunto de regras.

As informações espaço-temporais necessárias para a iniciação da apresentação dos componentes em uma aplicação NCL são especificadas através de descritores (ABNT, 2009). O elemento *<descriptor>*, definido no módulo *Presentation Specification*, é utilizado nessa especificação. Assim, a adaptação da apresentação é realizada através da escolha de um dos possíveis descritores para um conteúdo. Cada conjunto de descritores alternativos são agrupados através do elemento *<descriptorSwitch>*, também definido no módulo *Presentation Specification*.

Na aplicação NCL escolhida como exemplo, a duração da apresentação de uma imagem, correspondente à foto do jogador de futebol, é especificada pelo autor da aplicação de forma explícita. Na verdade, mais de uma duração é especificada, através de um conjunto de descritores alternativos. A Figura 4.4 apresenta esse conjunto de descritores (linhas 18 a 23) e o teste utilizado para a escolha de um desses descritores (linha 5).

```

...
03.   <ruleBase>
...
05.   <rule id="screenSize" var="system.screenGraphicSize" value="800,600"
           comparator="lt"/>
06.   </ruleBase>
...
18.   <descriptorSwitch id="photoDesc">
19.     <bindRule constituent="withoutImgDur" rule="screenSize"/>
20.     <defaultDescriptor descriptor="withImgDur"/>
21.     <descriptor id="withoutImgDur" region="frameReg" explicitDur="10s"/>
22.     <descriptor id="withImgDur" region="frameReg" explicitDur="20s"/>
23.   </descriptorSwitch>
...

```

Figura 4.4 - Adaptação da apresentação na aplicação NCL exemplo.

Na construção do HTG, cada duração explícita deve ser representada como uma condição associada a uma aresta. Na especificação da Figura 4.4, caso a resolução gráfica do dispositivo de apresentação seja menor que 800x600 *pixels*, a duração da apresentação da foto do jogador será de 10 segundos. Caso contrário, essa duração será de 20 segundos (linha 22).

A Figura 4.5 apresenta a parte do HTG correspondente à adaptação da apresentação especificada na Figura 4.4. Nessa parte do grafo, cada uma das duas possíveis durações e o teste correspondente para a escolha de uma dessas durações são representados através de uma condição composta. Cada uma dessas condições, por sua vez, é associada a uma aresta construída entre as transições de início e de fim do evento de apresentação do objeto correspondente à foto do jogador.

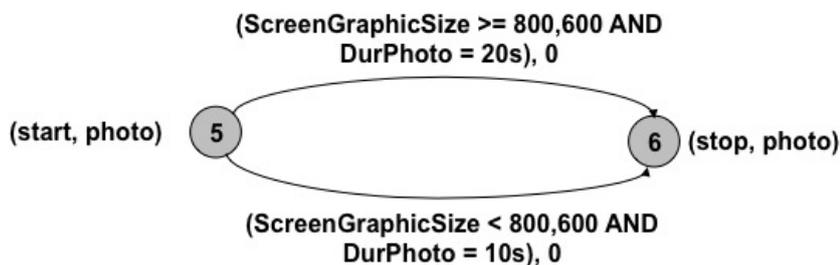


Figura 4.5 - Parte do HTG que representa a adaptação da apresentação descrita na Figura 4.4.

Em relação à adaptação do conteúdo, um conjunto de possíveis conteúdos para apresentação devem estar agrupados nas aplicações NCL através do elemento

`<switch>`, definido no módulo *ContentControl*.¹³ Na aplicação NCL definida como exemplo, um *switch* agrupa dois objetos de mídia, mais especificamente, dois formulários XHTML, um em português e outro em inglês. A Figura 4.6 apresenta esses objetos (linhas 60 e 61), especificados como componentes de um elemento `<switch>` (linhas 53 a 62). Dependendo do idioma escolhido no ambiente de apresentação, cujo valor é armazenado na variável “*system.language*”, um desses objetos de mídia será escolhido (linha 4).

```

...
03.     <ruleBase>
04.         <rule id="en" var="system.language" value="en" comparator="eq"/>
...
06.     </ruleBase>
...
53.     <switch id="form">
54.         <switchPort id="spForm">
55.             <mapping component="enForm"/>
56.             <mapping component="ptForm"/>
57.         </switchPort>
58.         <bindRule constituent="enForm" rule="en"/>
59.         <defaultComponent component="ptForm"/>
60.         <media id="ptForm" src="ptForm.htm" descriptor="formDesc"/>
61.         <media id="enForm" src="enForm.htm" descriptor="formDesc"/>
62.     </switch>
...

```

Figura 4.6 - Adaptação do conteúdo na aplicação NCL exemplo.

No HTG que representa a aplicação NCL, para cada elemento filho de um *switch*, devem ser construídos vértices representando as transições dos eventos que podem ocorrer sobre esse *switch*. Essas possíveis transições são definidas através dos relacionamentos com a participação do *switch* (relacionamentos serão vistos na Seção 4.5). A Figura 4.7 apresenta uma parte do HTG, correspondente a representação do *switch* especificado na Figura 4.6.

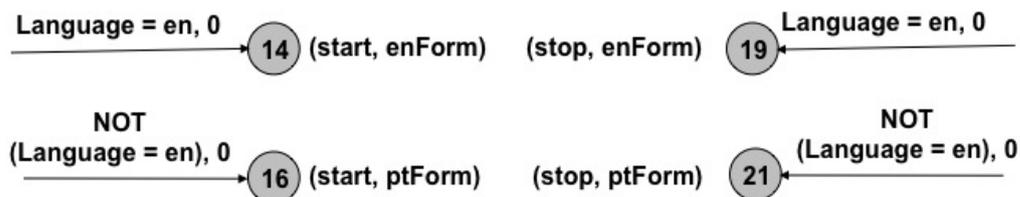


Figura 4.7 - Parte do HTG que representa a adaptação do conteúdo descrita na Figura 4.6.

¹³ Um elemento `<switch>` agrupa um conjunto de nós alternativos, que podem ser objetos de mídia, outros elementos `<switch>` ou ainda contextos, que serão vistos ainda neste capítulo.

Na Figura 4.7, os vértices representam as transições de início (vértices 14 e 16) e de fim (vértices 19 e 21) da apresentação de cada um dos objetos de mídia existentes no *switch* (“enForm” e “ptForm”). Quando um relacionamento tem como consequência o disparo de uma transição de evento sobre um *switch*, uma aresta no HTG deve ser construída para cada um dos elementos filhos desse *switch*. A condição associada a cada uma dessas arestas deve ser a mesma definida nas regras para a escolha de um dos componentes do *switch*. Esse é o caso das arestas com destino nos vértices 14 e 19 (início e fim da apresentação de “ptForm”) que têm como condição que o idioma do sistema de apresentação seja inglês (“Language = en”). Esse também é o caso das arestas com destino nos vértices 16 e 21 (início e fim da apresentação de “enForm”), que têm como condição que esse idioma não seja inglês (“NOT Language = en”).

Quando um relacionamento na NCL tem como causa o disparo de uma transição de evento sobre um *switch*, uma aresta também deve ser construída para cada um dos elementos filhos desse *switch*. Essas arestas, no entanto, não têm como condição regras para a escolha de um dos elementos filhos do *switch*. Apenas as condições existentes no próprio relacionamento devem ser condições dessas arestas (relacionamentos serão vistos na Seção 4.5).

Além de objetos de mídia, um *switch* pode conter outros elementos <*switch*> e também contextos como filhos. No caso do elemento filho também ser um *switch*, na construção do HTG, as regras dos elementos <*switch*> filhos devem ser adicionadas às regras do elemento pai, formando regras compostas. No caso dos contextos, os elementos filhos desse contexto devem ser mapeados, para a construção dos vértices e arestas. A representação dos contextos da NCL no HTG será discutida na Seção 4.6.

4.4. Reúso

Uma das principais facilidades oferecidas pela NCL é a possibilidade de reusar os seus elementos (Neto, 2009), em particular, os objetos de mídia e os contextos. O reúso deve ser especificado através do atributo *refer*, definido no módulo *Reuse*. O valor desse atributo deve ser o identificador (valor do atributo *id*) do elemento que será reusado.

O reuso em um contexto (elemento `<context>`) faz com que uma cópia do contexto referenciado (valor do atributo `refer`), incluindo todos os seus elementos filhos, seja realizada.

O reuso em um objeto de mídia (`<media>`) também gera uma cópia do objeto referenciado (indicado no atributo `refer`). No entanto, diferente do reuso de contextos, o objeto que realiza a referência pode definir novos elementos filhos (`<area>` e `<property>`). Todos os atributos de um elemento `<media>`, que faz referência a outro objeto, devem ser ignorados e substituídos pelos atributos do elemento referenciado. Exceto o atributo `id` e, em alguns casos, o atributo `descriptor`, dependendo do valor do atributo `instance`, presente no elemento que faz a referência. Esse atributo pode ter os valores “`instSame`”, “`new`” ou “`gradSame`”. Cada um desses valores define uma variação na forma com que os objetos podem ser reusados, gerando arestas e vértices diferentes no HTG, como será descrito a seguir.

Quando o atributo `instance` do objeto que faz a referência tem o valor “`instSame`”, o objeto que referencia é o mesmo objeto referenciado, desde a sua instanciação, incluindo todas as suas propriedades e âncoras de conteúdo. Esse é o caso do objeto “`reusedAnimation`”, apresentado na Figura 4.8, que reusa as especificações do objeto “`animation`”.

```
...
43.     <media id="reusedAnimation" refer="animation" instance="instSame">
44.         <property name="bounds"/>
45.     </media>
...
```

Figura 4.8 - Reuso na aplicação NCL exemplo.

A especificação da Figura 4.8 foi retirada de parte da aplicação NCL utilizada como exemplo ao longo deste capítulo. Nessa aplicação, um contexto é utilizado para agrupar semanticamente os objetos e os relacionamentos associados a uma publicidade (a propaganda da chuteira). Nesse contexto, existem relacionamentos com a participação do objeto que representa a animação, tanto para iniciar a apresentação da imagem da chuteira, quanto para redimensionar a animação, caso essa imagem seja selecionada. Assim, o objeto “`animation`” é reusado no contexto para facilitar a especificação dos relacionamentos.

Na construção do HTG referente ao objeto “`reusedAnimation`”, são utilizados os mesmos vértices que representam as transições do evento de

apresentação do objeto “animation” e de suas âncoras de conteúdo. A parte do HTG que representa a especificação da Figura 4.8 é a mesma apresentada na Figura 4.3, para o objeto “animation” e suas âncoras de conteúdo (vértices 1, 4, 7, 11 e 13). Assim, quando o reuso “*instSame*” é utilizado, nenhum vértice ou aresta precisa ser adicionado ao HTG, exceto se novas âncoras de conteúdo forem especificadas.

O atributo *instance* de um objeto também pode ter o valor “*new*”. Nesse caso, uma cópia do objeto referenciado é realizada, isto é, um objeto totalmente independente, cópia de outro, é construído. Somente nesse caso o atributo *descriptor* do objeto que faz a referência pode ser redefinido. O valor “*new*” é o padrão (*default*) para o atributo *instance*, caso esse atributo não seja especificado.

Quando o reuso na aplicação NCL envolve uma nova instância de um objeto, uma cópia dos vértices e das arestas associadas ao objeto referenciado deve ser realizada no HTG que representa a aplicação. Assim, um novo objeto é representado no grafo, e novos vértices e arestas são construídos para representar as transições dos eventos sobre esse novo objeto.

O outro valor possível para o atributo *instance* é “*gradSame*”. Nesse outro caso, o objeto que faz a referência e o objeto referenciado são o mesmo, porém, as incorporações no objeto que faz a referência serão realizadas à medida que suas propriedades e âncoras de conteúdo tiverem suas apresentações iniciadas.

No reuso “*gradSame*”, as transições de início para o evento de apresentação dos objetos (que faz a referência e o referenciado) podem ocorrer em instantes temporais distintos. Nesse caso, no HTG que representa uma aplicação NCL, devem ser construídos vértices correspondentes ao início e ao fim do evento de apresentação para um objeto que utiliza o reuso “*gradSame*”.

Para exemplificar a representação do HTG relativo a aplicações NCL com objetos que utilizam o reuso “*gradSame*”, uma aplicação contendo três vídeos (“vídeoA”, “vídeoB” e “vídeoC”) é proposta. A apresentação dessa aplicação é iniciada através da apresentação do objeto “vídeoA”. O início da apresentação desse objeto inicia também a apresentação do objeto “vídeoB”. O objeto “vídeoC”, que reusa o objeto “vídeoB” e que especifica o atributo *instance* com valor “*gradSame*”, somente é iniciado caso ocorra a seleção do objeto “vídeoA”. A Figura 4.9 apresenta o HTG dessa aplicação.

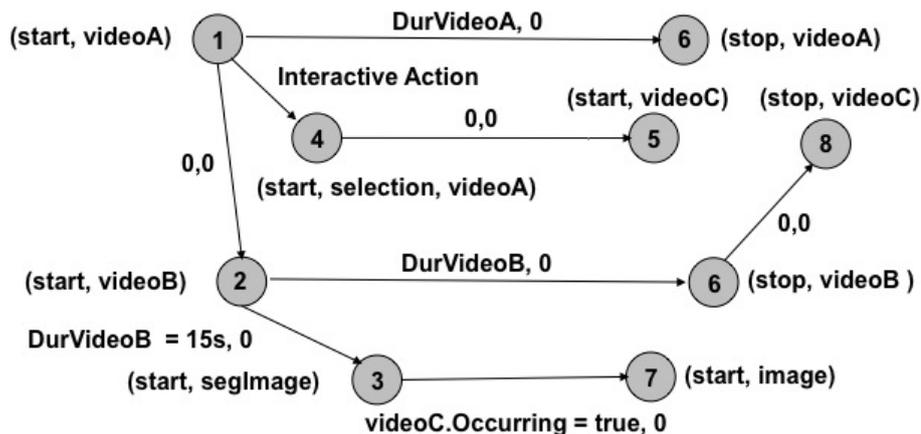


Figura 4.9 - HTG para o exemplo de reuso “gradSame”.

Continuando a descrição do exemplo apresentado na Figura 4.9, uma imagem (“image”) deve ser apresentada em um instante temporal específico da apresentação do objeto “videoC”. Esse instante é definido através de uma âncora de conteúdo (“segImage”), originalmente especificada no objeto “videoB” e herdada pelo objeto “videoC”.

Como pode ser observado na Figura 4.9, no reuso “gradSame”, as mesmas âncoras de conteúdo do objeto referenciado são utilizadas nos relacionamentos com a participação das âncoras herdadas pelo objeto que faz a referência. Porém, para representar a semântica desse reuso, uma variável, referente à ocorrência (*occurring*) da apresentação do objeto que faz a referência, é necessária como condição nos relacionamentos com origem nessas âncoras. Nesse exemplo, a apresentação do objeto “image” será iniciada apenas se o objeto “videoC” estiver sendo apresentado quando o instante temporal previsto para o início da apresentação da âncora “segImage” for alcançado (15 segundos da duração de “videoB”).

O controle da ocorrência da apresentação dos objetos somente precisa ser realizado quando a transição de início da apresentação de um objeto que especifica o reuso “gradSame” e as transições das suas âncoras de conteúdo estão em cadeias temporais diferentes. Esse é o caso do HTG da Figura 4.9, onde os vértices 4 e 5 são parte de uma cadeia secundária. Por outro lado, quando todas as transições de eventos associadas a um mesmo objeto fazem parte da mesma cadeia, essas transições são previsíveis em relação a um mesmo ponto de origem, o início da cadeia temporal. Nesse outro caso, relacionamentos que dependem da

ocorrência de um objeto podem ser avaliados antes da apresentação, simplificando o controle da apresentação em relação à especificação da NCL.

4.5. Relacionamentos

Nas aplicações NCL, os relacionamentos são especificados através do elemento `<link>`, definido no módulo *Linking*. Esse elemento agrupa elementos `<bind>`, também definidos no módulo *Linking*, que especificam os participantes de uma relação. É o conector, definido no módulo *ConnectorBase*, que especifica a semântica da relação de um elo. Por exemplo, em uma relação causal, o conector especifica condições que devem ser satisfeitas para que ações possam ser disparadas.

Conectores podem ser especificados na própria aplicação NCL, através do elemento `<causalConnector>`. Cada elemento `<causalConnector>` é especificado como um filho de um elemento `<connectorBase>`, que agrupa (define uma base de) conectores. Uma outra opção consiste em importar os conectores, através do elemento `<importBase>`. A Figura 4.10 contém uma parte da aplicação NCL proposta como exemplo, onde uma base de conectores é especificada. Nessa base são importados conectores especificados em outro documento (`causalConnBase.ncl`).

```
...
29.     <connectorBase>
30.         <importBase documentURI="../causalConnBase.ncl" alias="conEx"/>
31.     </connectorBase>
...
```

Figura 4.10 - Base de conectores utilizados na aplicação NCL exemplo.

Neste ponto, é importante mencionar que todas as relações exploradas nesta tese são especificadas por conectores causais, uma vez que elas são as únicas permitidas no perfil da linguagem para TV digital. Além disso, as condições dessas relações podem simples ou compostas, isto é, podem ser formadas por mais de uma condição, mas têm somente uma única condição que deve ser satisfeita em um instante de tempo infinitesimal.

Nos conectores, condições e ações definem um papel (atributo *role*), que deve ser único na especificação do conector. São esses papéis que serão assumidos pelos participantes de um relacionamento, associados através dos

<*binds*> de um elo. Para facilitar a especificação das aplicações, dois conjuntos de papéis são pré-definidos na NCL. Esses papéis associam transições das máquinas de estado dos eventos, que podem representar tanto condições (primeiro conjunto, descrito na Tabela 4.2) quanto ações (segundo conjunto, descrito na Tabela 4.3 em um relacionamento).

Papel (role)	Ação que executa a transição	Evento
<i>onBegin</i>	<i>start</i>	<i>presentation</i>
<i>onEnd</i>	<i>stop</i>	<i>presentation</i>
<i>onAbort</i>	<i>abort</i>	<i>presentation</i>
<i>onPause</i>	<i>pause</i>	<i>presentation</i>
<i>onResume</i>	<i>resume</i>	<i>presentation</i>
<i>onSelection</i>	<i>start</i>	<i>selection</i>
<i>onBeginAttribution</i>	<i>start</i>	<i>attribution</i>
<i>onEndAttribution</i>	<i>start</i>	<i>attribution</i>

Tabela 4.2 - Valores reservados para papéis utilizados como condição.

Papel (role)	Ação que executa a transição	Evento
<i>start</i>	<i>start</i>	<i>presentation</i>
<i>stop</i>	<i>stop</i>	<i>presentation</i>
<i>abort</i>	<i>abort</i>	<i>presentation</i>
<i>pause</i>	<i>pause</i>	<i>presentation</i>
<i>resume</i>	<i>resume</i>	<i>presentation</i>
<i>set</i>	<i>start</i>	<i>attribution</i>

Tabela 4.3 - Valores reservados para papéis utilizados como ação.

Voltando ao exemplo inicial sobre a animação introduzido na Seção 4.1, como vários elos são especificados na aplicação NCL, a construção de partes do HTG correspondentes aos relacionamentos será realizada paulatinamente. A Figura 4.11 apresenta alguns dos elos especificados nessa aplicação NCL. Nessa figura estão os elos utilizados na especificação do sincronismo temporal da aplicação NCL utilizada como exemplo que têm uma condição e uma ação simples.

```

...
94.   <link id="lMusic" xconnector="conEx#onBeginStart">
95.     <bind role="onBegin" component="animation"/>
96.     <bind role="start" component="music"/>
97.   </link>
98.   <link id="lBackground" xconnector="conEx#onBeginStart">
99.     <bind role="onBegin" component="animation"/>
100.    <bind role="start" component="background"/>
101.  </link>
102.  <link id="lPhoto" xconnector="conEx#onBeginStart">
103.    <bind role="onBegin" component="animation" interface="segPhoto"/>
104.    <bind role="start" component="photo"/>
105.  </link>
106.  <link id="lEnd" xconnector="conEx#onEndStop">
107.    <bind role="onEnd" component="animation"/>
108.    <bind role="stop" component="background"/>
109.  </link>
110.  <link id="lEndIcon" xconnector="conEx#onEndStop">
111.    <bind role="onEnd" component="animation" interface="segIcon"/>
112.    <bind role="stop" component="icon"/>
113.  </link>
...

```

Figura 4.11 - Relacionamentos associados ao sincronismo no tempo (sem interação) na aplicação NCL exemplo.

Na Figura 4.11, três elos têm como condição que o evento de apresentação sobre um objeto seja iniciado (“*onBegin*”) para que a apresentação de um outro objeto também seja iniciada (“*start*”). Outros dois elos têm como condição que o evento de apresentação sobre um objeto seja encerrado (“*onEnd*”) para que a apresentação, de um outro objeto, também seja encerrada (“*stop*”).

Na construção do HTG, elos da aplicação NCL que possuem uma condição e uma ação simples são representados por uma única aresta, unindo os vértices correspondentes à condição e à ação desse relacionamento. A Figura 4.12 apresenta as arestas construídas em parte do HTG para os elos especificados na Figura 4.11. Os vértices e arestas correspondentes a esses elos são destacados em vermelho nessa figura.

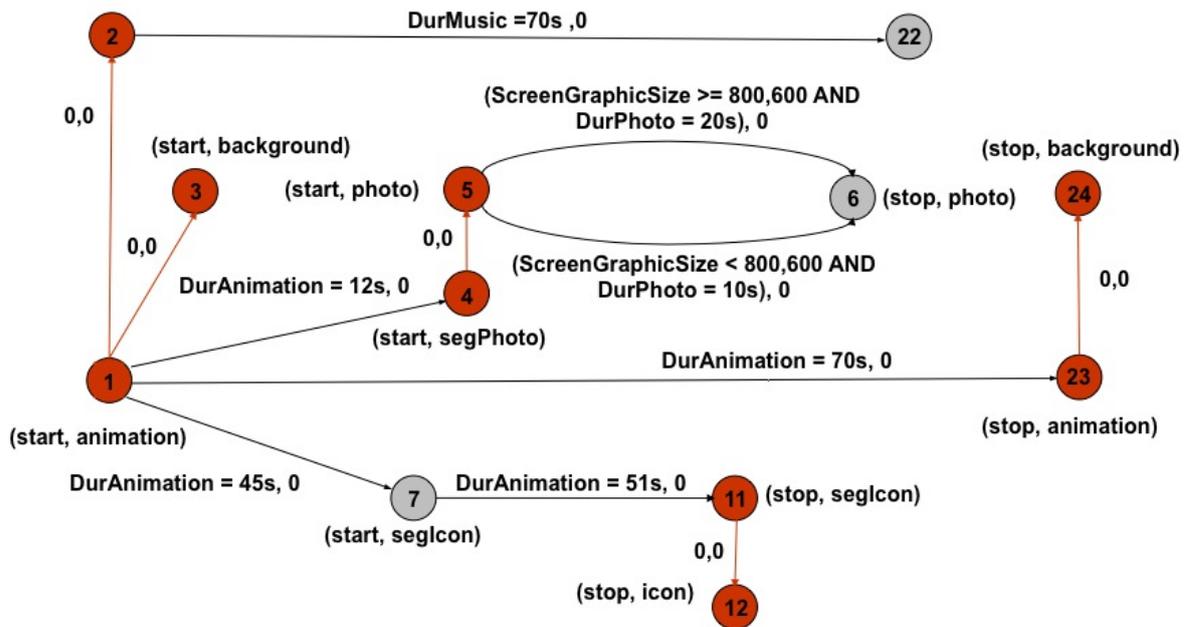


Figura 4.12 - Parte do HTG que representa o sincronismo no tempo especificado na Figura 4.11.

Na Figura 4.12, as arestas entre os vértices 1 e 2, 1 e 3, 4 e 5, 11 e 12, 23 e 24, representam, respectivamente, a semântica dos elos “lMusic”, “lBackground”, “lPhoto”, “lEndIcon” e “lEnd”.

Na mesma aplicação NCL encontram-se também relacionamentos com condições e ações compostas. Uma condição composta, como mencionado, pode ter outras condições, simples ou compostas, incluindo itens a serem avaliados. Uma ação composta pode ser formada por outras ações, simples ou compostas, que devem ser executadas em paralelo ou sequencialmente, uma após a outra (não é necessário aguardar o término de uma ação para o disparo da próxima).

O elo “lIcon”, definido na aplicação NCL e especificado na Figura 4.13, tem como condições o início da apresentação de uma âncora (“segIcon”) do objeto “reusedAnimation” (linha 64) e também que a interatividade esteja habilitada (linhas 65 a 67). Atendidas essas condições, a apresentação da imagem da chuteira deve ser iniciada (linha 68) e também iniciada a atribuição de valores propriedade “left” dessa imagem (posição espacial) (linhas 69 a 73). Os valores atribuídos a essa variável variam no tempo, em um intervalo de 5 segundos (animação).

```

...
63.     <link id="lIcon" xconnector="conEx#onBeginTestStartSetDelay">
64.         <bind role="onBegin" component="reusedAnimation"
                                                interface="segIcon"/>
65.         <bind role="testVar" component="globalVar"
                                                interface="channel.interactivity">
66.             <bindParam name="var" value="true"/>
67.         </bind>
68.         <bind role="start" component="icon"/>
69.         <bind role="set" component="icon" interface="left">
70.             <bindParam name="var" value="87.5%"/>
71.             <bindParam name="delay" value="1s"/>
72.             <bindParam name="duration" value="5s"/>
73.         </bind>
74.     </link>
...
    
```

Figura 4.13 - Relacionamento associado ao sincronismo no tempo (com condição) na aplicação NCL exemplo.

Para representar condições compostas, uma única aresta é construída no HTG. O vértice de origem dessa aresta é uma dessas condições. As demais condições são associadas à aresta como condição (do HTG) para a ocorrência do seu vértice de destino. Mais de uma aresta pode ser necessária para representar uma ação composta. O vértice que representa cada ação simples de uma ação composta é o destino de uma aresta construída. A Figura 4.14 representa os vértices e arestas (em vermelho) construídos para representar a semântica do elo da Figura 4.13 (“lIcon”).

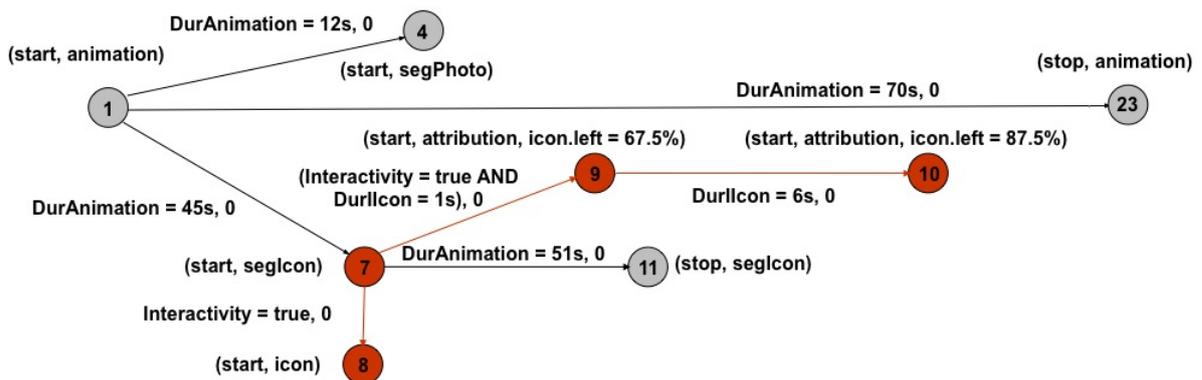


Figura 4.14 - Parte do HTG que representa o sincronismo no tempo (com condição) especificado na Figura 4.13.

Na Figura 4.14, as condições do elo “*Icon*” são representadas pelo vértice 7, correspondente à transição de início do evento de apresentação de “*segIcon*”,¹⁴ e pela condição associada às arestas (“*interactivity = true*”). Como esse elo especifica duas ações, é criada uma aresta para cada uma dessas ações, com origem no vértice 7 (condição). Uma dessas arestas tem como destino o vértice 8, que representa o início da apresentação do objeto “*icon*” (imagem da chuteira). O destino da outra aresta é o vértice 9, que representa o início da atribuição da propriedade “*left*” do objeto “*icon*”.

No elo “*Icon*”, um atraso deve ser obedecido antes do início da atribuição da propriedade “*left*”. Esse atraso é representado no HTG como mais uma condição da aresta entre os vértices 7 e 9. O valor final da atribuição da propriedade “*left*” é representado pelo vértice 10. A aresta que une os vértices 9 e 10 tem como condição uma duração (5 segundos) que deve ser obedecida para que o valor final da atribuição da propriedade “*left*” seja alcançado.

Assim como os objetos, os elos da NCL também podem ter uma duração. Essa duração é uma variável identificada pelo nome do elo. O seu valor é iniciado quando o vértice que corresponde à condição do elo é alcançado. Como será discutido na Seção 4.6, o valor dessa variável pode permanecer inalterado, caso o contexto onde o elo foi definido seja pausado.

Nas aplicações NCL, a condição de um relacionamento também pode ser uma ação interativa. Esse é o caso do elo especificado na Figura 4.15, que tem como condição o início da seleção da imagem da chuteira. Para iniciar essa seleção, uma ação interativa deve acontecer, através do pressionamento do botão vermelho do controle remoto pelo usuário (“*keyCode = RED*”). Uma vez atendida essa condição, um conjunto de transições serão disparadas: o início da apresentação do vídeo da propaganda (linha 79), o início da apresentação do elemento <*switch*> (linha 80), o início da atribuição da animação, através do objeto “*reusedAnimation*” (linhas 81 a 83) e a interrupção da apresentação da imagem da chuteira (linha 84).

¹⁴ Como pode ser observado na Figura 4.14, a condição do elo é um vértice que representa um evento sobre uma âncora de conteúdo do objeto “*animation*”. Essa associação deve-se ao reuso do objeto “*animation*”, especificado no objeto “*reusedAnimation*” (Seção 4.4).

```

...
75.     <link id="lBeginShoes" xconnector="conEx#onKeySelectionStopSetStart">
76.         <bind role="onSelection" component="icon">
77.             <bindParam name="keyCode" value="RED"/>
78.         </bind>
79.         <bind role="start" component="shoes"/>
80.         <bind role="start" component="form" interface="spForm"/>
81.         <bind role="set" component="reusedAnimation" interface="bounds">
82.             <bindParam name="varSet" value="5%,6.67%,45%,45%"/>
83.         </bind>
84.         <bind role="stop" component="icon"/>
85.     </link>
...
    
```

Figura 4.15 - Relacionamento associado ao sincronismo no tempo (com interaão) na aplicaão NCL exemplo.

A Figura 4.16 representa os vrtices e arestas construdas para representar o elo da Figura 4.15 (“lBeginShoes”). Nessa figura, o vrtice 13 representa a transião de incio da seleão da imagem da chuteira. As arestas com origem nesse vrtice tem como destino vrtices que representam cada uma das aões do elo “lBeginShoes”. Essas aões so o fim da apresentaão da imagem da chuteira (vrtice 12), o incio da apresentaão do vdeo da propaganda (vrtice 15), o incio da atribuião do filme da animaão e o incio da apresentaão do elemento <switch>. Como mencionado na Seão 4.3, transiões de um evento sobre um switch so realizadas sobre os seus elementos filhos, no caso, o incio da apresentaão do formulrio em ingls (vrtice 14) e do formulrio em portugus (vrtice 16).

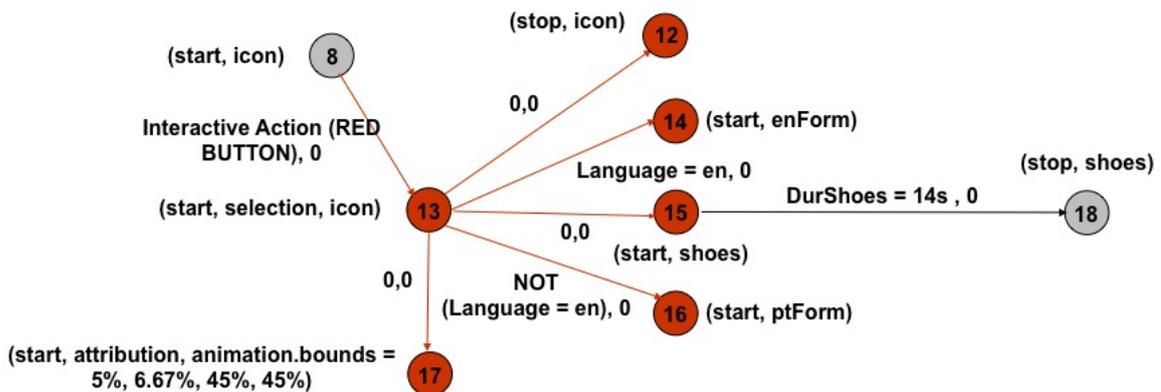


Figura 4.16 - Parte do HTG que representa o sincronismo no tempo (com interaão) especificado na Figura 4.15.

A Figura 4.17 apresenta o ltimo elo da aplicaão NCL exemplo. Esse elo tem como condião o fim da apresentaão do vdeo da propaganda (linha 87).

Como consequência dessa condição, duas ações devem ser executadas: o início da atribuição do filme da animação (linhas 88 a 90), restaurando as dimensões originais desse objeto e o fim da apresentação do elemento *<switch>* (linha 91).

```

...
86.   <link id="lEndShoes" xconnector="conEx#onEndSetStop">
87.     <bind role="onEnd" component="shoes"/>
88.     <bind role="set" component="reusedAnimation" interface="bounds">
89.       <bindParam name="varSet" value="0,0,222.22%,222.22%"/>
90.     </bind>
91.     <bind role="stop" component="form"/>
92.   </link>
...

```

Figura 4.17 - Relacionamento associado a um evento de atribuição na aplicação NCL exemplo.

A Figura 4.18 apresenta os vértices e arestas construídas para representar o elo da Figura 4.17. O vértice 18 representa a transição de evento que é a condição desse elo (fim da apresentação do vídeo da propaganda). Esse vértice é a origem de três arestas. Uma das arestas, entre os vértices 18 e 20, associa o final do evento de apresentação do vídeo da propaganda ao início da atribuição do objeto “*animation*”. As outras duas arestas, entre os vértices 18 e 19 e entre os vértices 18 e 21, associam o final do evento de apresentação do vídeo da propaganda ao final desse evento para cada um dos objetos filhos do *switch* (“*enForm*” e “*ptForm*”).

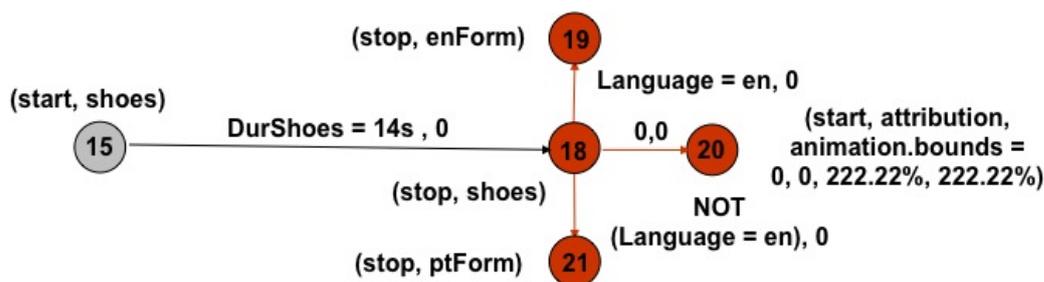


Figura 4.18 - Parte do HTG que representa o evento de atribuição especificado na Figura 4.17.

4.6. Contextos

NCL é uma linguagem estruturada, que permite agrupar objetos de mídia e seus relacionamentos (elos) em diferentes contextos, definidos no módulo

Context. Cada contexto pode ter um conjunto de portas (elemento *<port>*), definidas no módulo *CompositeNodeInterface*. Uma porta especifica um mapeamento para um dos componentes de um contexto ou para uma outra interface de um desses componentes.¹⁵

Contextos não têm semântica temporal implícita, eles representam a estrutura lógica de uma aplicação. Dessa forma, contextos não interferem no sincronismo temporal de uma aplicação NCL, exceto nos casos em que os relacionamentos são especificados diretamente sobre um contexto, quer seja como condição ou como ação de uma relação causal.¹⁶

A Tabela 4.4 resume o comportamento dos componentes de um contexto, quando uma ação, relativa ao evento de apresentação, é executada diretamente sobre esse contexto, sem que nenhuma das suas portas seja mencionada.

Ações	Comportamento
<i>start</i>	Aplicada a todos os componentes mapeados pelas portas existentes no contexto.
<i>stop</i>	Aplicada a todos os componentes existentes no contexto. Caso o contexto contenha elos sendo avaliados ou com a avaliação pausada, essas avaliações serão suspensas.
<i>abort</i>	Aplicada a todos os componentes existentes no contexto. Caso o contexto contenha elos sendo avaliados ou com a avaliação pausada, essas avaliações serão suspensas.
<i>pause</i>	Aplicada a todos os componentes existentes no contexto. Caso o contexto contenha elos sendo avaliados, essas avaliações serão pausadas.

¹⁵ Um contexto também pode ter propriedades (elemento *<property>*), definidas no módulo *PropertyAnchor*. O HTG não diferencia propriedades pertencentes a um objeto de mídia ou a um contexto. Assim, a representação no HTG de ambos os casos é a mesma.

¹⁶ Esse caso é diferente da especificação de relacionamentos diretamente sobre os componentes de um contexto, acessados através de suas portas. Nesse outro caso, o HTG é construído considerando apenas os elementos filhos do contexto.

<i>resume</i>	Aplicada a todos os componentes existentes no contexto. Retoma a apresentação dos elementos anteriormente pausados pela ação de pause no contexto (os que estavam pausados antes da ação de pause no contexto permanecem pausados) e a avaliação dos elos existentes em um contexto que foram pausados por uma ação de pause aplicada ao contexto.
---------------	--

Tabela 4.4 - Comportamento dos componentes de um contexto de acordo com as ações realizadas sobre esse contexto.

Para ilustrar a construção de um HTG para cada possível ação executada diretamente sobre um contexto, a Figura 4.19 apresenta, como exemplo, a visão estrutural de um contexto que agrupa três objetos de mídia (A, B e C). Nesse contexto são especificadas duas portas, que definem, cada uma, um mapeamento para os objetos “A” e “B”.

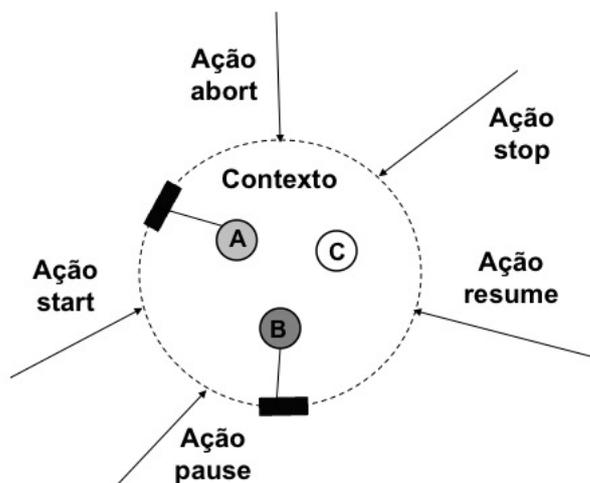


Figura 4.19 - Visão estrutural de um contexto com ações.

Um vértice no HTG, para cada componente de um contexto, deve ser construído para representar as ações do tipo “*stop*” ou “*abort*” executadas diretamente sobre esse contexto. No caso das ações do tipo “*start*”, devem ser construídos vértices apenas para os componentes mapeados através das suas portas. A Figura 4.20 apresenta os vértices e as arestas de parte de um HTG que representa a visão estrutural proposta na Figura 4.19, para as ações “*start*”, “*stop*” e “*abort*”.

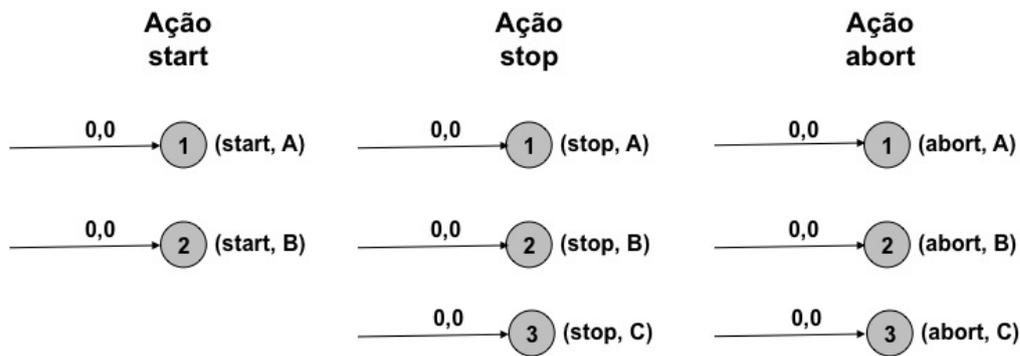


Figura 4.20 - Parte de um HTG que representa as ações de “start”, “stop” e “abort” sobre o contexto da Figura 4.19.

Ações do tipo “*pause*” para a apresentação de um contexto devem ser representadas no HTG por dois vértices, para cada componente desse contexto. Um desses vértices corresponde à transição de pausa (“*pause*”) da apresentação de cada componente do contexto. O outro vértice corresponde à transição de início do evento de atribuição de uma variável (*paused*). Essa variável contém a informação de que um componente teve sua apresentação pausada por uma ação executada sobre o seu contexto. Essa informação é importante, uma vez que, ações do tipo “*resume*” sobre um contexto, discutidas a seguir, só terão efeito sobre os componentes desse contexto se a apresentação desses componentes foi pausada por uma ação do tipo “*pause*”, previamente executada sobre esse mesmo contexto.

A Figura 4.21 apresenta os vértices de parte de um HTG construídos para representar uma ação do tipo “*pause*” executada sobre o contexto da visão estrutural da Figura 4.19. Para que as transições representadas pelos vértices da Figura 4.21 sejam executadas, é necessário que o objeto de mídia correspondente ao vértice esteja ocorrendo (*occurring*). Essa condição é representada nas arestas que têm como destino os vértices construídos.

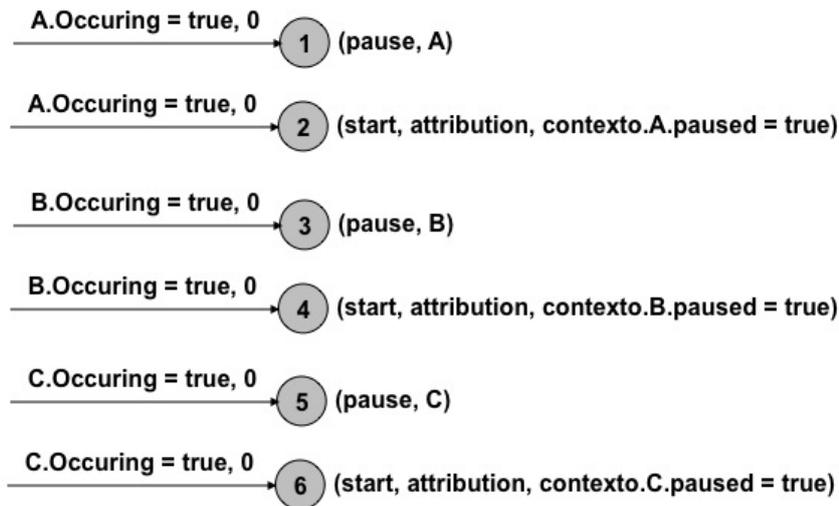
Ação pause

Figura 4.21 - Parte de um HTG que representa a ação de "pause" sobre o contexto da Figura 4.19.

Um vértice no HTG, para cada componente de um contexto, deve ser construído para representar as ações do tipo "resume" na apresentação desse contexto. Para que as transições representadas em cada vértice construído sejam executadas, é necessário que a apresentação do componente tenha sido anteriormente pausada por uma ação do tipo "pause", executada sobre o seu contexto. Essa condição é associada à aresta que tem como destino cada vértice construído, conforme apresentado em parte do HTG da Figura 4.22.

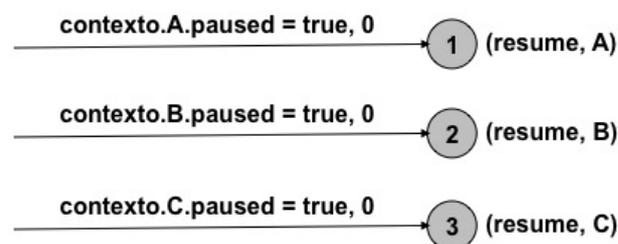
Ação resume

Figura 4.22 - Parte de um HTG que representa a ação de "resume" sobre o contexto da Figura 4.19.

Na aplicação NCL, o estado do evento de apresentação de um contexto pode ser especificado como condição em um relacionamento. A Tabela 4.5 resume os possíveis estados da apresentação de um contexto. Esses estados dependem dos eventos de apresentação dos componentes (elementos filhos) de um contexto.

Estados	Estados dos eventos de apresentação dos elementos filhos
<i>occurring</i>	Se ao menos um componente do contexto tiver o seu evento de apresentação no estado <i>occurring</i>
<i>sleeping</i>	Se todos os componentes do contexto tiverem os seus eventos de apresentação no estado <i>sleeping</i> .
<i>paused</i>	Se todos os componentes do contexto tiverem os seus eventos de apresentação em um estado diferente de <i>occurring</i> e pelo menos um desses componentes tiver o seu evento de apresentação no estado <i>paused</i> .

Tabela 4.5 - Estados da apresentação de um contexto.

Para ilustrar a construção de um HTG para cada possível condição associada aos estados de um contexto, a Figura 4.23 apresenta, como exemplo, a visão estrutural do mesmo contexto da Figura 4.19. A diferença da Figura 4.23 são as palavras reservadas (“*onBegin*”, “*onPause*”, “*onEnd*”) utilizadas para destacar as possíveis condições que podem estar associadas a um contexto.

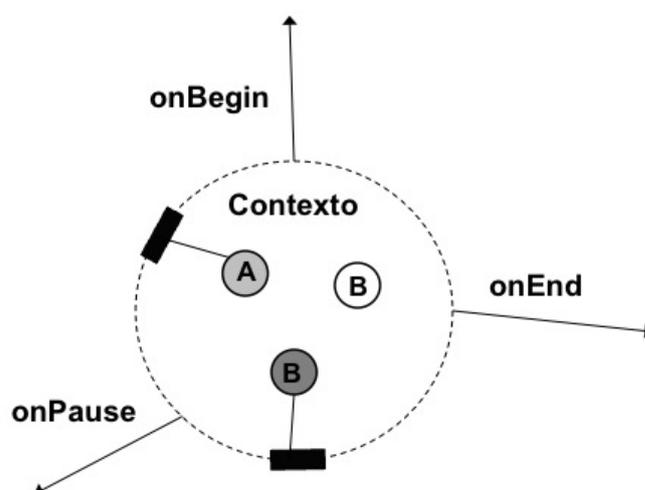


Figura 4.23 - Visão estrutural de um contexto com condições associadas.

Uma aresta no HTG, para cada componente de um contexto, deve ser construída para representar as condições do tipo “*onBegin*”, “*onEnd*” e “*onPause*” especificadas sobre esse contexto. No caso das condições do tipo “*onBegin*” nenhuma condição associada às arestas é necessária. Essas arestas têm como origem o vértice que representa o início da apresentação de cada componente (elemento filho) desse contexto. Para que essa condição seja atendida, basta que qualquer uma das apresentações de um dos componentes do contexto seja

iniciada. A Figura 4.24 apresenta os vértices e as arestas de parte de um HTG que representa a visão estrutural proposta na Figura 4.23.

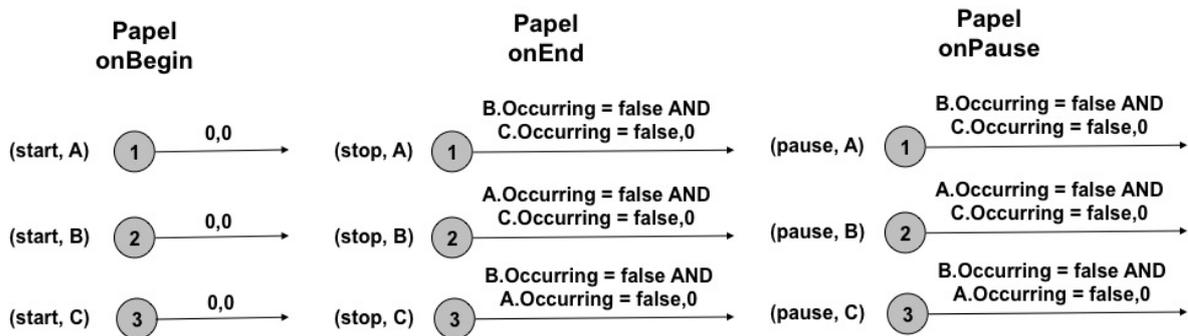


Figura 4.24 - Parte do HTG que representa as condições “onBegin”, “onPause” e “onEnd” sobre o contexto da Figura 4.23.

Para representar as demais condições que podem estar associadas a um contexto em uma aplicação NCL, as arestas do HTG devem ter como condição que a apresentação dos demais componentes do contexto não estejam ocorrendo. Esse é o caso quando o contexto participa em um relacionamento associado a um papel “onEnd”. O mesmo acontece quando o contexto participa em um relacionamento associado a um papel “onPause”. A diferença entre esses casos é que, no primeiro, as arestas têm origem na transição de fim do evento de apresentação de cada componente do contexto e, no segundo, o vértice origem de cada aresta corresponde à transição de pausa do evento de apresentação de cada componente do contexto.

As variáveis utilizadas para representar a semântica dos contextos de uma aplicação NCL através do HTG (*occurring*, *paused*) somente são necessárias no grafo quando os seus valores podem ser alterados em cadeias temporais diferentes daquela onde os valores dessas variáveis são avaliados. Essa simplificação para o controle da apresentação em relação à especificação da NCL é a mesma utilizada na representação do reuso “*gradSame*”, abordada na Seção 4.4.

A Figura 4.25 apresenta a especificação da aplicação NCL descrita neste capítulo. O HTG construído para essa aplicação ao longo deste capítulo pode ser visualizado na Figura 4.26.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Exemplo de Aplicação NCL -->
01. <ncl id="htg" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
02.   <head>
03.     <ruleBase>
04.       <rule id="en" var="system.language" value="en" comparator="eq"/>
05.       <rule id="screenSize" var="system.screenGraphicSize" value="800,600"
                                comparator="lt"/>
06.     </ruleBase>
07.     <regionBase>
08.       <region id="backgroundReg" width="100%" height="100%" zIndex="1"/>
09.       <region id="screenReg" width="100%" height="100%" zIndex="2">
10.         <region id="frameReg" left="5%" top="6.7%" width="18.5%"
                                height="18.5%" zIndex="3"/>
11.         <region id="iconReg" left="67.5%" top="11.7%" width="8.45%"
                                height="6.7%" zIndex="3"/>
12.         <region id="shoesReg" left="15%" top="60%" width="25%" height="25%"
                                zIndex="3"/>
13.         <region id="formReg" left="56.25%" top="8.33%" width="38.75%"
                                height="71.7%" zIndex="3"/>
14.       </regionBase>
15.       <descriptorBase>
16.         <descriptor id="backgroundDesc" region="backgroundReg"/>
17.         <descriptor id="screenDesc" region="screenReg"/>
18.         <descriptorSwitch id="photoDesc">
19.           <bindRule constituent="withoutImgDur" rule="screenSize"/>
20.           <defaultDescriptor descriptor="withImgDur"/>
21.           <descriptor id="withoutImgDur" region="frameReg"
                                explicitDur="10s"/>
22.           <descriptor id="withImgDur" region="frameReg" explicitDur="20s"/>
23.         </descriptorSwitch>
24.         <descriptor id="audioDesc"/>
25.         <descriptor id="iconDesc" region="iconReg"/>
26.         <descriptor id="shoesDesc" region="shoesReg"/>
27.         <descriptor id="formDesc" region="formReg"/>
28.       </descriptorBase>
29.       <connectorBase>
30.         <importBase documentURI="../causalConnBase.ncl" alias="conEx"/>
31.       </connectorBase>
32.     </head>
33.     <body>
34.       <port id="entry" component="animation"/>
35.       <media id="background" src="background.png"
                                descriptor="backgroundDesc"/>
36.       <media id="animation" src="animGar.mp4" descriptor="screenDesc">
37.         <area id="segPhoto" begin="12s"/>
38.         <area id="segIcon" begin="45s" end="51s"/>
39.       </media>
40.       <media id="music" src="choro.mp3" descriptor="audioDesc"/>
41.       <media id="photo" src="photo.png" descriptor="photoDesc">
42.       <context id="advert">

```

```

43.     <media id="reusedAnimation" refer="animation" instance="instSame">
44.         <property name="bounds"/>
45.     </media>
46.     <media id="icon" src="icon.png" descriptor="iconDesc">
47.         <property name="left"/>
48.     </media>
49.     <media id="shoes" src="shoes.mp4" descriptor="shoesDesc"/>
50.     <media id="globalVar" type="application/x-ginga-settings">
51.         <property name="channel.interactivity" value="true"/>
52.     </media>
53.     <switch id="form">
54.         <switchPort id="spForm">
55.             <mapping component="enForm"/>
56.             <mapping component="ptForm"/>
57.         </switchPort>
58.         <bindRule constituent="enForm" rule="en"/>
59.         <defaultComponent component="ptForm"/>
60.         <media id="ptForm" src="ptForm.htm" descriptor="formDesc"/>
61.         <media id="enForm" src="enForm.htm" descriptor="formDesc"/>
62.     </switch>
63.     <link id="lIcon" xconnector="conEx#onBeginTestStartSetDelay">
64.         <bind role="onBegin" component="reusedAnimation"
65.             interface="segIcon"/>
66.         <bind role="testVar" component="globalVar"
67.             interface="channel.interactivity">
68.             <bindParam name="var" value="true"/>
69.         </bindParam>
70.         <bind role="start" component="icon"/>
71.         <bind role="set" component="icon" interface="left">
72.             <bindParam name="var" value="87.5%"/>
73.             <bindParam name="delay" value="1s"/>
74.             <bindParam name="duration" value="5s"/>
75.         </bindParam>
76.     </link>
77.     <link id="lBeginShoes" xconnector="conEx#onKeySelectionStopSetStart">
78.         <bind role="onSelection" component="icon">
79.             <bindParam name="keyCode" value="RED"/>
80.         </bindParam>
81.         <bind role="start" component="shoes"/>
82.         <bind role="start" component="form" interface="spForm"/>
83.         <bind role="set" component="reusedAnimation" interface="bounds">
84.             <bindParam name="varSet" value="5%,6.67%,45%,45%"/>
85.         </bindParam>
86.         <bind role="stop" component="icon"/>
87.     </link>
88.     <link id="lEndShoes" xconnector="conEx#onEndSetStop">
89.         <bind role="onEnd" component="shoes"/>
90.         <bind role="set" component="reusedAnimation" interface="bounds">
91.             <bindParam name="varSet" value="0,0,222.22%,222.22%"/>
92.         </bindParam>
93.         <bind role="stop" component="form"/>

```

```

92.     </link>
93. </context>
94. <link id="lMusic" xconnector="conEx#onBeginStart">
95.   <bind role="onBegin" component="animation" />
96.   <bind role="start" component="music" />
97. </link>
98. <link id="lBackground" xconnector="conEx#onBeginStart">
99.   <bind role="onBegin" component="animation" />
100.  <bind role="start" component="background" />
101. </link>
102. <link id="lPhoto" xconnector="conEx#onBeginStart">
103.   <bind role="onBegin" component="animation" interface="segPhoto" />
104.   <bind role="start" component="photo" />
105. </link>
106. <link id="lEnd" xconnector="conEx#onEndStop">
107.   <bind role="onEnd" component="animation" />
108.   <bind role="stop" component="background" />
109. </link>
110. <link id="lEndIcon" xconnector="conEx#onEndStop">
111.   <bind role="onEnd" component="animation" interface="segIcon" />
112.   <bind role="stop" component="icon" />
113. </link>
114. </body>
115. </ncl>
    
```

Figura 4.25 - Aplicação NCL exemplo.

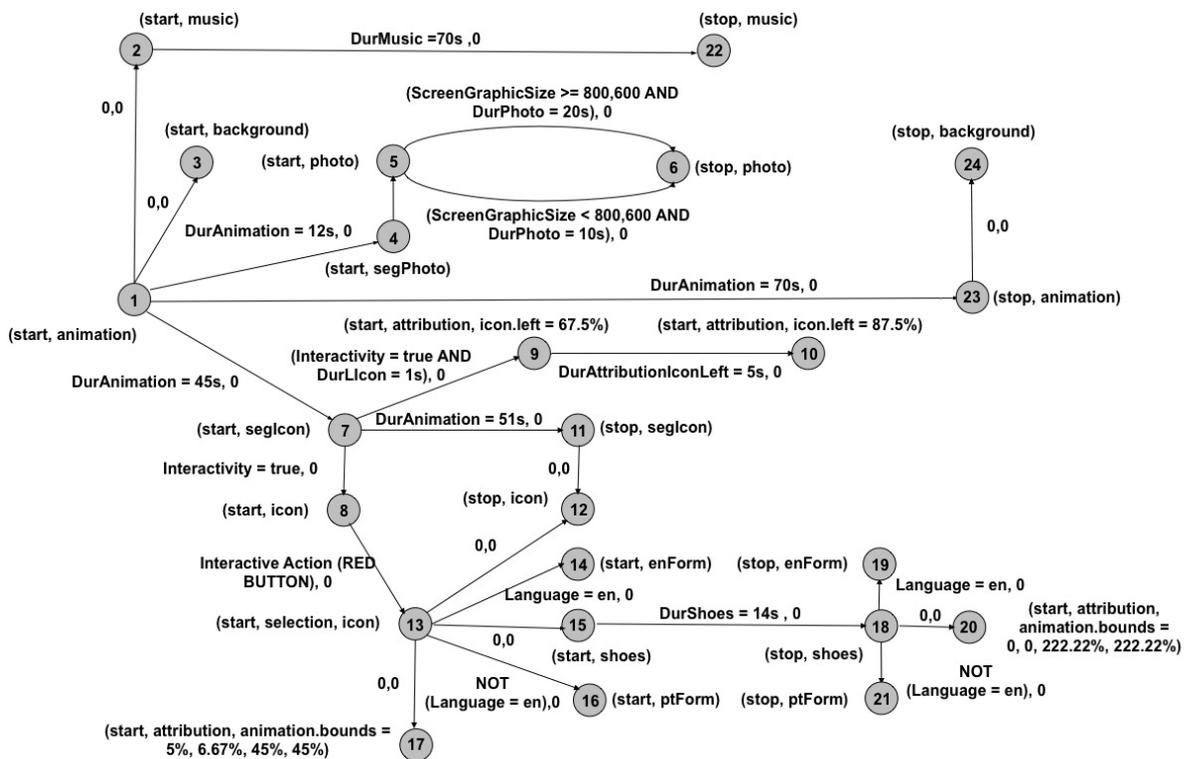


Figura 4.26 - HTG correspondente à aplicação NCL da Figura 4.25.