NCL: Uma Linguagem Declarativa para Especificação de Documentos Hipermídia na Web

Meire Juliana Antonacci
mjuliana@telemidia.puc-rio.br

Débora C. Muchaluat-Saade
debora@telemidia.puc-rio.br

Rogério Ferreira Rodrigues
rogerio@telemidia.puc-rio.br

Luiz Fernando Gomes Soares
lfgs@inf.puc-rio.br

Laboratório TeleMídia – Departamento de Informática – PUC-Rio
R. Marquês de São Vicente, 225 – Gávea – 22453-900, Rio de Janeiro, RJ, Brasil

Resumo

Este artigo apresenta uma linguagem declarativa para autoria hipermídia, chamada NCL – Nested Context Language, que permite especificar, de forma estruturada, documentos complexos contendo relacionamentos de sincronização temporal e espacial entre seus componentes. NCL satisfaz alguns requisitos desejáveis em linguagens de autoria hipermídia que foram identificados através de uma análise de diversas linguagens existentes. Dentre esses requisitos, pode-se destacar a facilidade de reuso de conteúdo, da estrutura e das características de apresentação de componentes de documentos hipermídia. Além disso, NCL segue os padrões de intercâmbio e interoperabilidade da Web, pois foi especificada utilizando XML, uma recomendação do W3C.

Abstract

This paper presents a declarative language for hypermedia authoring, called NCL – Nested Context Language, which permits specifying, in a structured way, complex documents containing temporal and spatial synchronization relationships among their components. NCL satisfies some desirable requirements for hypermedia authoring languages that were identified through the analysis of several existent languages. Among those requirements, we can highlight the facility for reusing content, structure and presentation characteristics of hypermedia document components. Furthermore, NCL follows the Web interchange and interoperability standards, as it was specified using XML, a W3C recommendation.

1. Introdução

Em virtude da grande difusão da Web e da necessidade de incorporação de informação multimídia aos documentos publicados, a demanda por linguagens de autoria hipermídia que tenham maior poder de expressão têm sido cada vez mais intensa. Nesse contexto, considerando a necessidade de facilitar o intercâmbio de documentos e a interoperabilidade entre as aplicações, o papel das linguagens declarativas é fundamental.

A linguagem HTML, apesar de ter realizado bem o seu papel em difundir a utilização da Web devido a sua simplicidade, não é adequada à definição de documentos hipermídia com relacionamentos de sincronização temporal e espacial. Com isso, outras propostas de linguagens para especificar documentos hipermídia que satisfaçam essas novas necessidades têm surgido, seguindo diferentes abordagens.

Algumas dessas propostas partiram da própria linguagem HTML e definiram extensões, objetivando solucionar algumas de suas limitações, mas mantendo sua simplicidade, como por exemplo, o HTML+TIME [TIMEH98] e a proposta de [RoDu98]. No entanto, apesar de incluírem a capacidade de definir relacionamentos de sincronização temporal, essas propostas herdam diversas limitações do HTML como, por exemplo, o fato dos elos estarem embutidos no conteúdo dos documentos, impossibilitando, entre outras coisas, o reuso do conteúdo sem a herança obrigatória das relações.

Outras propostas, por exemplo HyTime [HyTime97], basearam-se no padrão SGML [SGML86], cuja complexidade de processamento tornaram difícil seu uso. Assim, verificou-se a necessidade de uma proposta mais robusta que o HTML, porém especificada em alguma meta-linguagem de tratamento mais simples que o SGML. O surgimento do padrão XML [XML98], adotado como recomendação pelo W3C[1], vem preencher essa lacuna. Seu objetivo principal é representar, genérica e estruturadamente, documentos que possam ser intercambiados e processados na Web, sanando a dificuldade de se obter interoperabilidade entre sistemas hipermídia heterogêneos.

Atualmente, já existem algumas linguagens baseadas no padrão XML que possibilitam a especificação de documentos hipermídia, dentre as quais a mais difundida é a linguagem SMIL [SMIL98], também adotada como recomendação pelo W3C. Todavia, tanto o SMIL como as propostas de extensão do HTML mencionadas, por tentarem manter a característica de simplicidade, resolvem algumas das limitações, mas permanecem com outras, como discutido em [RRMS99].

A razão principal do poder limitado dessas linguagens reside no fato delas se basearem em modelos hipermídia com pouco poder de expressão. Modelos conceituais com maior poder de expressão provêem mais recursos para autoria, permitindo a criação de documentos mais complexos. Esse é o caso do modelo conceitual hipermídia NCM (*Nested Context Model*) [SoCR95, Soar00].

Entre outras funcionalidades, o NCM permite a estruturação lógica do documento através do uso de composições aninhadas, oferece a possibilidade de definição de relacionamentos causais e de restrição entre os seus componentes, permite especificar o comportamento temporal e espacial do documento de forma flexível e oferece suporte à autoria cooperativa.

O objetivo principal deste artigo é apresentar uma nova linguagem declarativa para autoria de documentos hipermídia, denominada NCL (*Nested Context Language*) [Anto00, AnSo00a]. Baseada no modelo NCM e especificada através de uma DTD (*Document Type Definition*) XML, a linguagem NCL visa suprir a necessidade de uma linguagem padronizada para descrição de documentos hipermídia na Web, sem as limitações identificadas em linguagens como o SMIL. NCL satisfaz requisitos desejáveis em linguagens de autoria hipermídia que foram identificados através de uma análise de diversas propostas existentes. Dentre esses requisitos, pode-se destacar a facilidade de reuso de conteúdo, da estrutura e das características de apresentação dos componentes de um documento.

NCL foi concebida de forma que documentos SMIL e HTML possam ser facilmente convertidos em documentos NCL, o que de fato é realizado em formatadores (exibidores de documentos) NCL. Nesse caso, novas facilidades de autoria estarão disponíveis para aumento da expressividade dos documentos originais. Inversamente, documentos NCL simples, contendo apenas as facilidades presentes na linguagem SMIL, podem ser convertidos e apresentados em formatadores SMIL.

Este artigo está organizado da seguinte forma. A Seção 2 discute algumas características identificadas como desejáveis em linguagens de autoria de documentos hipermídia. Na Seção 3 é descrita a linguagem NCL, ressaltando suas características principais. A Seção 4 compara a linguagem NCL com alguns trabalhos relacionados, utilizando como base de comparação os requisitos apresentados na Seção 2. Finalmente, a Seção 5 é reservada às conclusões e trabalhos futuros.

[1] http://www.w3c.org

2. Características Desejáveis em Linguagens para Especificação de Documentos Hipermídia

Após realizar uma análise de várias linguagens de autoria hipermídia, algumas delas discutidas na Seção 4, foram levantados requisitos importantes que toda linguagem deveria satisfazer. Esta seção discute cada uma dessas características desejáveis.

Estruturação lógica do documento

Um requisito fundamental em uma linguagem de autoria é permitir a especificação dos documentos de forma estruturada, ou seja, oferecer uma forma de agrupar os componentes em blocos e utilizar esses blocos para construir a especificação. Normalmente, esse recurso é fornecido através da definição de composições que agrupam elementos em um documento, onde esses elementos podem inclusive ser outras composições.

Muitas vezes a autoria é facilitada utilizando composições que tenham alguma semântica de apresentação embutida, como por exemplo, composições que agrupam elementos que serão apresentados em paralelo durante a exibição do documento. No entanto, também é desejável oferecer composições que não têm semântica de apresentação associada, mas que forneçam unicamente uma forma de estruturar o documento logicamente, da mesma maneira que um livro está estruturado em capítulos, os capítulos em seções e assim por diante.

Representação de objetos de mídia

Um outro requisito importante está associado à representação dos objetos de mídia. É necessário que as linguagens de autoria ofereçam suporte pelo menos aos tipos básicos de mídia, tais como texto, áudio, imagem e vídeo, pois diferenciá-los facilita a definição de âncoras específicas a cada tipo.

Uma outra característica desejável é permitir a especificação de um objeto em separado do seu conteúdo[2], o que torna possível definir a estrutura de um documento independente do conteúdo das mídias utilizadas. Isso permite que um mesmo conteúdo seja referenciado por objetos diferentes, evitando sua replicação, caso seja reutilizado em documentos diferentes ou em partes diferentes de um mesmo documento.

Características de apresentação separadas do componente

Uma característica bastante desejável em uma linguagem de autoria de documentos hipermídia é poder especificar as características de apresentação dos componentes[3] de um documento em um objeto separado. Isso permite especificar apresentações diferentes para um mesmo componente ou então associar um único conjunto de características de apresentação a componentes diferentes.

Uma outra característica desejável é permitir que não apenas o conteúdo de um objeto de mídia possa ser exibido durante uma apresentação, mas também o conteúdo de uma composição, isto é, sua estrutura de componentes e relacionamentos. Para isso, a linguagem de autoria precisa oferecer mecanismos que permitam ao autor definir se a apresentação de uma composição se refere à exibição de seus componentes constituintes ou de sua estrutura.

Um requisito também interessante é a possibilidade de definir alternativas de apresentação para um mesmo componente a serem escolhidas em tempo de execução. A escolha pode ser

[2] Um objeto de mídia tem outros atributos além do conteúdo, tais como título, autor, descrição, etc.
[3] Entende-se por componente, qualquer objeto de mídia ou composição contido em um documento.

realizada dependendo da navegação feita para atingir aquele componente ou dependendo de parâmetros de qualidade de serviço especificados para uma determinada plataforma de apresentação.

Reuso

Um requisito fundamental, e talvez um dos mais importantes, em uma linguagem de autoria declarativa é permitir a reutilização de componentes de um documento em outros documentos ou em outras partes do mesmo documento, pois facilita o processo de autoria, uma vez que evita a redefinição de informações. Porém, mais desejável ainda é que o reuso não seja feito através de cópias, evitando que alterações sejam efetuadas em múltiplos locais e com isso tornando o processo de manutenção mais simples.

Além da reutilização de componentes, é desejável permitir também a reutilização de suas características de apresentação, caso as mesmas sejam definidas separadamente.

Definição de elos no escopo de uma composição

Uma caraterística interessante em uma linguagem hipermídia é permitir que os elos associados a um determinado componente sejam válidos no escopo de uma composição. Esse recurso, unido à característica de reuso, permite que um mesmo componente tenha diferentes elos associados, dependendo da composição onde está inserido. Por exemplo, na Figura 1, o componente "A" dentro da composição "Comp1" tem dois elos associados, enquanto que o mesmo componente dentro da composição "Comp2" tem apenas um outro elo associado.

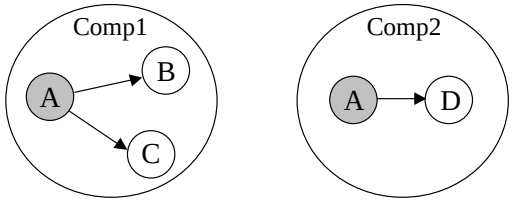

Figura 1 – Exemplo de reuso do componente "A" com diferentes elos associados (componentes são representados por círculos e elos por setas)

Relacionamentos complexos entre componentes

A característica marcante de um documento hipermídia é a possibilidade de criar relacionamentos espaço-temporais conectando partes diferentes de um documento ou mesmo documentos distintos. Para isso, é fundamental que a linguagem de autoria tenha um bom poder de expressão para a definição desses relacionamentos.

É desejável também que as linguagens ofereçam uma granularidade fina, permitindo que uma relação ancore em pontos internos a um objeto de mídia e não somente no início ou no fim do objeto. Isso torna possível, por exemplo, iniciar a apresentação de um objeto a partir de um ponto qualquer em seu conteúdo.

Não basta fornecer meios para especificar apenas o relacionamento tradicional hipermídia que exibe um componente quando uma âncora é selecionada. Em documentos mais complexos, às vezes é necessário definir relacionamentos que envolvam mais de dois componentes e que possuam a semântica de causalidade ou de restrição.

Em relacionamentos com semântica de causalidade, alguma ação é disparada quando uma condição específica for satisfeita. Um exemplo desse tipo é "se o componente *A* estiver sendo

exibido (condição) e o componente *B* terminou sua exibição (condição), exiba o componente *C* (ação)".

Em outros casos, relacionamentos representam restrições entre os componentes. Por exemplo, uma restrição que especifica que dois componentes devem terminar juntos significa que o instante de término da exibição do conteúdo de cada componente deve coincidir. Note que, nesse caso, não há nenhuma causalidade envolvida, pois a ocorrência da apresentação de um componente sem a ocorrência da apresentação do outro também satisfaz a restrição, que apenas especifica que, se e somente se os dois forem apresentados, seus términos devem coincidir. Um outro exemplo de restrição seria definir uma relação de alinhamento espacial entre as áreas de exibição de dois componentes, especificando, por exemplo, que os topos das janelas de exibição de determinados componentes devem estar alinhados, quando os dois estiverem sendo exibidos. Nesse caso, se uma das janelas for movida, a outra deve acompanhar o movimento respeitando a relação da restrição.

Não importa se de causalidade ou restrição, relacionamentos podem ser especificados através de composições ou através de elos. Como exemplo do uso de composições, tem-se as composições paralelas, já mencionadas, que especificam que todos os seus componentes devem ser exibidos em paralelo.

Uma outra característica desejável em linguagens de autoria é permitir a definição de elos entre componentes contidos em composições diferentes e não restringir que elos sejam definidos apenas entre componentes contidos diretamente em uma mesma composição. Isso possibilita uma flexibilidade maior para modificar o comportamento de componentes internos a uma composição, sem precisar alterar a estrutura de composições do documento.

Flexibilidade na especificação temporal

Outro requisito desejável é a possibilidade de especificar o comportamento temporal de componentes do documento de uma forma flexível. Isso implica em permitir a definição de funções de custo de elasticidade que servirão como métricas para que o formatador possa escolher a melhor qualidade de apresentação possível. Os domínios dessas funções definem intervalos temporais que são associados a duração de objetos de mídia, a segmentos de tempo, etc. A flexibilidade irá permitir manter a consistência temporal do documento, isto é, atender a todos os requisitos impostos pelo autor, mesmo quando não for possível utilizar os valores ideais (de menor custo) dentro dos intervalos. A flexibilidade pode também permitir que sejam feitos ajustes para corrigir a apresentação quando ocorrerem eventos imprevisíveis, como por exemplo, atrasos no sistema de comunicação, atrasos no sistema operacional ou mesmo ações do usuário.

Adaptação à plataforma de exibição

Uma outra característica desejável é a possibilidade de adaptação de um documento à plataforma em que ele estiver sendo exibido. Essa adaptação pode ser feita, por exemplo, selecionando componentes de acordo com parâmetros de descrição da plataforma. Esses parâmetros podem especificar informações do perfil do usuário, tal como o idioma nativo, ou então parâmetros de qualidade de serviço, tal como a banda passante disponível no sistema de comunicação.

3. A Linguagem NCL

Um documento NCL é sempre formado por um elemento raiz, denominado *ncl*, que contém dois outros elementos, o *head* e o *doc-body*[4]. O *head* representa o cabeçalho do documento, contendo informações relativas à exibição dos componentes, como por exemplo, o layout espacial da apresentação. O elemento *doc-body* representa o corpo do documento, contendo a definição dos componentes e seus relacionamentos. A Figura 2 ilustra a estrutura básica de um documento especificado em NCL.

```
<ncl>
  <head>
     <!--   ...definição do layout espacial
            ...definição dos descritores -->
  </head>
  <doc-body>
     <!--   ...especificação dos objetos de mídia
            ...especificação das composições
            ...especificação dos relacionamentos -->
  </doc-body>
</ncl>
```

Figura 2 – Estrutura de um documento NCL

O layout espacial da apresentação (elemento *layout*) contém a definição de janelas e regiões onde os componentes do documento serão exibidos. Ao contrário de outras linguagens, para cada dispositivo de exibição, pode ser definida uma janela, representada pelo elemento *root-layout*. Também podem ser definidas regiões (elemento *region*), associadas a essas janelas, que definem a área de exibição de algum componente. A linguagem permite que essas regiões sejam definidas independente das janelas, possibilitando o reuso de regiões em janelas diferentes.

Cada região, além dos atributos de identificação (*id*) e título (*title*), pode especificar os atributos *width*, *height*, *left* e *top*. Esses atributos identificam a largura e altura da região e a coordenada (x,y) onde seu vértice superior esquerdo será posicionado na janela de exibição associada, identificada por um outro atributo, chamado *root-layout*. Os valores dos atributos que especificam a área espacial podem ser valores absolutos, definidos em número de pixels, ou valores percentuais relativos ao tamanho da janela de exibição.

A Figura 3 ilustra a definição de dois elementos *root-layout*. O primeiro deles define uma janela de apresentação com largura igual a 640 e altura igual a 400 pixels (atributos *width* e *height*). O atributo *device* associa uma janela a um dispositivo de exibição, no caso o monitor do computador. O nome desse dispositivo deve ser reconhecido pelo formatador do documento. O segundo elemento *root-layout* define uma janela para um outro dispositivo, identificado por "tv".

Na mesma figura, são definidas três regiões espaciais. A região "norte", por exemplo, define uma área, posicionada na coordenada (0,0), com largura igual a 100% e altura igual a 20% relativas à janela de exibição, que corresponde à janela definida pelo *root-layout* "monitor".

[4] Como convenção, os nomes dos elementos e atributos em NCL serão apresentados em itálico. Procurou-se na definição dos nomes utilizar uma nomenclatura parecida com o padrão SMIL. Além do fato disso ajudar na concepção e conversão de documentos entre os dois modelos, ajuda também na comparação das linguagens, podendo-se identificar mais facilmente em que elementos SMIL a especificação NCL acrescenta funcionalidades.

```
<ncl id="/.telemidia:apresentacao">
  <head>
    <layout>
      <root-layout id="monitor" width="640" height="400" device="monitor"/>
      <root-layout id="tv" width="860" height="600" device="tv"/>
      <region id="norte" left="0" top="0" width="100%" height="20%" root-layout="monitor"/>
      <region id="leste" left="20%" top="20%" width="80%" height="60%"/>
      <region id="oeste" left="0" top="20%" width="20%" height="60%"/>
    </layout>
    <descriptors-set>
      <descriptor id="d1" region="oeste" root-layout="monitor"/>
      <descriptor id="d2" region="norte"/>
      <descriptor id="d3" region="leste" root-layout="monitor"/>
      <descriptor id="d4" repetitions="5" dur="9 10 11 3 3" root-layout="monitor"/>
      <descriptor id="d5" repetitions="10" dur="10" root-layout="tv"/>
      <descriptor id="d6" region="norte" root-layout="tv"/>
      <descriptor id="d7" left="0" top="20%" width="100%" height="80%" root-layout="tv"/>
    </descriptors-set>
  </head>
  <doc-body>
    <!-- ... -->
  </doc-body>
</ncl>
```

Figura 3 – Exemplo de definição do layout espacial da apresentação e das características de exibição dos componentes de um documento NCL

Nas linguagens hipermídia, usualmente, parte das informações relativas à apresentação de um componente são definidas no mesmo elemento que representa esse componente. No entanto, na NCL essas informações são inteiramente definidas em um elemento separado, denominado descritor (*descriptor*). Cada componente do documento pode estar associado a um ou mais descritores, o que permite especificar apresentações diferentes para um mesmo componente.

Dentre os atributos dos descritores, podem ser salientados:

- *dur*, *begin*, *end* – diferente das outras linguagens nas quais os valores desses atributos geralmente especificam tempos fixos, NCL, oferece flexibilidade para especificação desses tempos. Esses atributos são especificados através de funções de custo de elasticidade relativas à duração, início e término da apresentação do componente. Por exemplo, suponha um vídeo de duração igual a 20s, porém com uma tolerância de mais ou menos 10%, implicando que o mesmo pudesse ser acelerado ou atrasado em até 2s sem que isso comprometesse sua inteligibilidade. Supondo ainda que a perda de qualidade da apresentação fosse proporcional ao desvio da duração ideal, poderia ser definida uma função de custo linear para especificação da duração do vídeo, determinando um valor mínimo (18s), ideal (20s) e máximo (22s) para exibição, e dois coeficientes (c_1 e c_2) que indicassem, respectivamente, a proporção de perda de qualidade (aumento do custo) caso o vídeo fosse encurtado ou prolongado. Dessa forma, o custo de exibir o vídeo com a duração ideal seria 0, com a duração mínima seria $2c_1$ e com a duração máxima seria $2c_2$.
- *repetitions* – especifica o número de vezes que o componente deverá ter sua exibição repetida, em seqüência.
- *root-layout* – referencia um elemento *root-layout*, já definido no elemento *layout*.
- *region* – especifica uma área para exibição do componente ou faz referência a uma região já definida no elemento *layout*.

Voltando ao exemplo da Figura 3, dentro do elemento *descriptors-set*, foram definidos sete descritores que serão associados aos componentes do documento para definir características das suas apresentações. Alguns deles ("d1", "d3" e "d6") definem, através dos atributos *root-layout* e *region*, uma janela e uma área para exibição do componente dentro dessa janela. Os valores desses atributos correspondem aos valores dos atributos *id* dos elementos *root-layout*

e *region* associados. Um outro descritor (“d2”) define apenas uma região, pois nela já foi especificada uma janela a ser associada. Os descritores “d4” e “d5” definem uma janela de apresentação, o número de vezes que o componente deve ser apresentado (*repetitions*) e a função de custo de elasticidade relativa à duração do componente (*dur*). Para o descritor “d4”, foi especificada uma função de custo linear com os valores mínimo (9s), ideal (10s) e máximo (11s) para a duração da exibição, e os coeficientes que indicam o aumento do custo quando a duração afasta-se do valor ideal. Para o descritor “d5”, foi especificada uma função de custo com o valor ótimo em 10s. O descritor “d7” também referencia uma janela de apresentação e especifica uma área para exibição do componente dentro dessa janela. Porém, nesse descritor, essa área foi definida através dos atributos *left*, *top*, *width* e *height*, que possuem a mesma semântica dos atributos homônimos especificados no elemento *region*. Se no elemento descritor, o autor referenciar uma região e, além disso, definir os atributos *left*, *top*, *width*, *height* ou *root-layout*, os valores dos atributos homônimos do elemento *region* serão sobrepostos pelos valores definidos no descritor. Esse é o caso do atributo *root-layout* definido no descritor “d6” do exemplo ilustrado na Figura 3.

Em NCL, os componentes de um documento podem ser simples (objetos de mídia) ou compostos (composições).

Existem elementos específicos que possibilitam a definição de objetos de mídia de tipos diferentes, tais como texto (*text*), imagem (*img*), vídeo (*video*) e áudio (*audio*). Esses elementos não contêm o conteúdo dos dados propriamente dito, e sim uma referência (URI) para esse conteúdo. Essa referência é especificada através do atributo *src* dos elementos, conforme ilustrado na Figura 4, onde foram especificados três elementos do tipo texto, uma imagem e um áudio. Em cada um desses elementos, foram especificados, além do atributo de identificação (*id*), que será definido posteriormente, o atributo *uid*, contendo a URI do componente, o atributo *src*, contendo a URI do conteúdo da mídia e o atributo *descriptor-list*. Esse último possui a especificação dos descritores que poderão ser associados ao elemento para definir sua apresentação. Seu valor pode ser formado por uma lista de valores correspondentes aos atributos *id* de descritores definidos no cabeçalho do documento. Essa lista possibilita que um mesmo elemento possa ser apresentado de maneiras diferentes, de acordo com parâmetros de qualidade de serviço. No exemplo da Figura 4, o elemento *audio* pode ser associado aos descritores “d4” ou “d5” definidos na Figura 3.

A linguagem permite ainda definir âncoras temporais e espaciais específicas para cada tipo de objeto de mídia. Por exemplo, uma âncora de um texto (*text-anchor*) determina uma seqüência de caracteres (atributo *text*) que se encontra em uma posição especificada pelo número de caracteres a partir do início do texto (atributo *position*). Na Figura 4, o texto “introducao_menu” possui duas âncoras. Também pode ser delimitada uma âncora em uma mídia do tipo áudio, a partir de duas amostras (início e fim) especificadas em um arquivo de áudio. Em uma mídia do tipo imagem, pode ser criada uma âncora espacial (*img-anchor*) delimitada por um retângulo, especificado a partir de seu vértice superior esquerdo, altura e largura. Na Figura 4, a imagem “introducao_imagem” possui duas âncoras. As âncoras são bastante úteis na especificação de relacionamentos entre os componentes, descritos a seguir.

```
<ncl id="/.telemidia:apresentacao">
  <head>
     <!-- ... -->
  </head>
  <doc-body>
     <!-- ... -->
       <text id="titulo_monitor" uid="./telemidia:introducao.titulo_mn"
        src="./telemidia:introducao.titulo_mn.txt" descriptor-list="d2"/>
       <text id="titulo_tv" uid="./telemidia:introducao.titulo_tv"
        src="./telemidia:introducao.titulo_tv.txt" descriptor-list="d6"/>
       <text id="introducao_menu" uid="./telemidia:introducao.menu"
        src="./telemidia:introducao.menu.txt" descriptor-list="d1">
         <text-anchor id="menu_projetos" text="projetos" position="10"/>
         <text-anchor id="menu_equipe" text="equipe" position="22"/>
       </text>
       <img id="introducao_imagem" uid="./telemidia:introducao.imagem"
        src="./telemidia:introducao.imagem.gif" descriptor-list="d3">
         <img-anchor id="img_projetos" left="0" top="10" width="30" height="25"/>
         <img-anchor id="img_equipe" left="0" top="35" width="30" height="25"/>
       </img>
       <audio id="introducao_audio" uid="./telemidia:introducao.audio"
        src="./telemidia:introducao.audio.aiff" descriptor-list="d4 d5"/>
     <!-- ... -->
  </doc-body>
</ncl>
```

Figura 4 – Exemplo de definição de objetos de mídia e âncoras específicas em NCL

Em NCL, uma composição pode conter objetos de mídia e composições[5], assim como os relacionamentos entre eles. Elas podem ser de quatro tipos: composição de estruturação (*context*), composição paralela (*par*), composição seqüencial (*seq*) ou composição alternativa (*switch*). Composições de estruturação permitem a definição da estrutura lógica do documento, sem nenhuma semântica de apresentação embutida. Composições paralelas e seqüenciais definem a forma de apresentação dos seus componentes. Numa composição paralela, os componentes são exibidos simultaneamente, enquanto em uma composição seqüencial, os componentes são apresentados em série. A composição alternativa provê meios para definir comportamentos alternativos para a apresentação de um documento. Nela podem ser definidos componentes dentre os quais apenas um será selecionado para ser exibido. Essa seleção é feita em tempo de execução, baseada em testes sobre parâmetros da plataforma, especificados nos atributos dos componentes e no formatador.

A Figura 5 ilustra uma composição de estruturação "telemidia" que contém duas composições paralelas e uma outra composição de estruturação. A composição paralela "equipe", por sua vez, contém uma composição seqüencial aninhada.

Todos os elementos que representam objetos de mídia ou composições podem ser reusados em um mesmo documento ou em documentos diferentes. Para isso, eles devem especificar dois atributos obrigatórios:

- *id* – esse atributo é útil para identificar o componente dentro de um documento.
- *uid* – esse atributo deve identificar a URI correspondente ao componente.

Quando o componente estiver sendo reusado, o autor deve especificar somente seus atributos *id* e *uid*. Além disso, o nome do elemento utilizado para indicar reuso é *nomeElemento-ref*, como por exemplo, para reutilizar um componente tipo texto, deve-se utilizar o elemento *text-ref*. Uma observação importante é que nenhum atributo do componente reusado pode ser alterado em outro lugar diferente de onde ele foi especificado.

[5] A única restrição é que uma composição não pode estar recursivamente contida nela mesma.

```
<ncl id="/.telemidia:apresentacao">
  <head>
    <!-- ... -->
  </head>
  <doc-body>
    <context id="telemidia" uid="./telemidia:apresentacao">
      <par id="introducao" uid="./telemidia:introducao">
        <!-- ... -->
      </par>
      <context id="projetos" uid="./telemidia:projetos">
        <text id="projetos_descricao" uid="./telemidia:projetos.descricao"
         src="./telemidia:projetos.descricao.txt" descriptor-list="d1 d3">
        <audio id="projetos_audio" uid="./telemidia:projetos.audio"
         src="./telemidia:projetos.audio.aiff" descriptor-list="d4 d5"/>
      </context>
      <par id="equipe" uid="./telemidia:equipe" descriptor-list="d3">
        <text id="equipe_descricao" uid="./telemidia:equipe.descricao"
         src="./telemidia:equipe.descricao.txt" descriptor-list="d3">
        <img id="equipe_imagem" uid="./telemidia:equipe.imagem"
         src="./telemidia:equipe.imagem.gif" descriptor-list=d1">
        <seq id="equipe_seq" uid="./telemidia:equipe.seq">
          <video id="equipe_video1" uid="./telemidia:equipe.video1"
           src="./telemidia:equipe.video1.mov" descriptor-list="d7">
          <video id="equipe_video2" uid="./telemidia:equipe.video2"
           src="./telemidia:equipe.video2.mov" descriptor-list="d7">
        </seq>
      </par>
      <!-- ... -->
    </context>
  </doc-body>
</ncl>
```

Figura 5 – Exemplo de definição de composições em um documento NCL

A linguagem NCL permite a definição de elos que representam relacionamentos complexos entre dois ou mais componentes do documento. Um elo é composto por um conjunto de pontos terminais de origem, um conjunto de pontos terminais de destino e um ponto de encontro. O conjunto de pontos terminais de origem e destino definem eventos em componentes que constituem as origens e os destinos do elo. Cada ponto terminal pode especificar os atributos *id*, *node-list*, *anchor*, *event*, *attr-name* e *descriptor-list*. O atributo *id* tem o significado usual. O atributo *node-list* especifica uma lista de valores de atributos *id* correspondentes à seqüência de componentes aninhados que deve ser percorrida até encontrar o componente do documento que atua como origem ou destino do elo (último valor de *id* especificado na lista). O atributo *anchor* especifica o valor do atributo *id* de uma âncora contida no último componente identificado por *node-list*. O atributo *event* define um evento associado à âncora. Seus possíveis valores são `presentation`, `selection`, ou `attribution`, representando, os eventos definidos pelo modelo NCM [Soar00]. Segundo o modelo, um evento pode ser a apresentação ou a seleção de uma determinada âncora ou ainda a atribuição de um valor a um dos atributos de um componente. Se o evento for de atribuição, *attr-name* deve identificar o nome do atributo correspondente. O atributo *descriptor-list* especifica uma lista de descritores que poderão ser associados ao último componente identificado por *node-list*, permitindo especificar como será a apresentação desse componente quando for feita uma navegação através desse elo. Para ilustrar a definição de elos em NCL, a Figura 6 especifica um elo (*link*) denominado "elo1", definido na composição "telemidia", relacionando componentes recursivamente contidos nessa composição. Esse elo possui dois pontos terminais de origem (*source-ep*), um relativo à seleção da âncora "menu_projetos" do componente "introducao_menu" e outro relativo à seleção da âncora "img_projetos" do componente "introducao_imagem", e dois pontos terminais de destino (*target-ep*) relativos à apresentação de todo o conteúdo dos componentes "projetos_descricao" e "projetos_audio".

O ponto de encontro de um elo (*meeting-point*) define um conjunto de condições (*condition*) e um conjunto de ações (*action*). Condições dizem respeito aos pontos terminais de origem e ações são operações que devem ser executadas nos pontos terminais de destino. Essa definição de ponto de encontro representa um relacionamento causal entre os pontos terminais do elo. Caso o elo represente um relacionamento de restrição, apenas as condições devem ser definidas.

O elemento *condition* representa condições que avaliam valores booleanos, podendo ser simples (*simple-condition*) ou compostas (*compound-condition*). Toda condição composta (*compound-condition*) é formada por uma ou mais condições simples e uma expressão lógica relacionando as condições (*expression_condition*), através dos operadores AND, OR e NOT. Uma condição composta é satisfeita se sua expressão for verdadeira.

No "elo1", ilustrado na Figura 6, a expressão da condição composta (*expression-condition*) foi definida usando o operador OR, e será verdadeira se um dos dois eventos de seleção especificados pelos pontos terminais de origem do elo acontecer.

```
<ncl id="/.telemidia:apresentacao">
  <head>
    <!-- ... -->
  </head>
  <doc-body>
   <context id="telemidia" uid="./telemidia:apresentacao">
     <!-- ... -->
     <link id="elo1">
       <source-ep id="sep_menu_projetos" node-list="introducao introducao_menu"
        anchor="menu_projetos" event="selection" descriptor-list="d1"/>
       <source-ep id="sep_img_projetos" node-list="introducao introducao_imagem"
        anchor="img_projetos" event="selection" descriptor-list="d3"/>
       <target-ep id="tep_projetos_descricao" node-list="projetos projetos_descricao"
        event="presentation" descriptor-list="d3"/>
       <target-ep id="tep_projetos_audio" node-list="projetos projetos_audio"
         event="presentation" descriptor-list="d5"/>
       <meeting-point>
         <condition>
           <compound-condition>
             <simple-condition id="sc_menu_projetos" previous-SEP="sep_menu_projetos"/>
             <simple-condition id="sc_img_projetos" previous-SEP="sep_img_projetos"/>
             <expression-condition exp="sc_menu_projetos OR sc_img_projetos"/>
           </compound-condition>
         </condition>
         <action>
           <compound-action>
             <simple-action id="sa_projetos_descricao" TEP="tep_projetos_descricao" action-name="start"/>
             <simple-action id="sa_projetos_audio" TEP="tep_projetos_audio" action-name="start"/>
             <expression-action exp="PAR(sa_projetos_descricao, sa_projetos_audio)"/>
           </compound-action>
         </action>
       </meeting-point>
     </link>
    <!-- ... -->
   </context>
  </doc-body>
</ncl>
```

Figura 6 – Exemplo de definição de um elo multiponto em NCL

O elemento *action* representa as ações que devem ser executadas nos pontos terminais de destino de um elo causal. Elas podem ser simples (*simple-action*) ou compostas (*compound-action*).

As ações simples podem ser aplicadas a pontos terminais que definem eventos de apresentação (ações `prepare`, `start`, `stop`, `pause`, `resume`, `abort`) ou a pontos terminais que definem eventos de atribuição (ações `relative-assign`, `absolute-assign`, `enable`, `disable`, `activate`). Esses valores são definidos através do atributo *action-name*

do elemento *simple-action*. A semântica de cada um desses valores pode ser encontrada em [Soar00].

Toda ação composta (*compound-action*) é formada por uma ou mais ações simples e uma expressão relacionando as ações (*expression_action*) através dos operadores PAR e SEQ. Esses operadores definem a ordem de execução de cada ação simples, indicando se elas serão iniciadas em paralelo ou respeitando alguma ordem predefinida.

No "elo1", ilustrado na Figura 6, a expressão da ação composta (*expression-action*) foi definida usando o operador PAR, indicando que a apresentação do texto e do áudio definidos como destinos do elo devem ser iniciados em paralelo, usando os descritores especificados nos respectivos pontos terminais.

Essa seção apresentou um resumo das características principais da linguagem NCL, para maiores detalhes sobre sua especificação, deve-se consultar [AnSo00a].

4. Trabalhos Relacionados

Esta seção apresenta uma comparação entre a linguagem NCL e algumas linguagens declarativas para especificação de documentos hipermídia. Essas linguagens foram escolhidas por representarem exemplos mais significativos, primeiro, daquelas que tiveram como ponto de partida a linguagem HTML e propuseram extensões a ela, tal como HTML+TIME [TIMEH98]; segundo, de linguagens que são especificadas em XML, mas que se baseiam em um modelo hipermídia próprio, tal como a linguagem declarativa do sistema Madeus [JLRST98] e a linguagem SMIL [SMIL98]. A comparação é feita baseando-se nas características desejáveis em linguagens para especificação de documentos hipermídia, discutidas na Seção 2.

Estruturação lógica do documento

Em HTML+TIME e SMIL, podem ser definidas composições seqüenciais e paralelas, nas quais todos os elementos contidos são exibidos, respectivamente, em série ou em paralelo, o que facilita muito o processo de autoria. Porém, nenhuma das duas linguagens oferece composições sem semântica de apresentação embutida.

Madeus[6] permite a definição de componentes compostos, que podem ser formados por outros componentes simples e compostos, propiciando uma organização hierárquica do documento. Entretanto, a especificação de relacionamentos de sincronização temporal e espacial só pode ser feita entre componentes diretamente contidos em uma mesma composição, o que obriga o autor a fazer a estruturação do documento de acordo com suas características de apresentação. Com isso, apesar das composições de Madeus terem uma semântica mais geral do que as de SMIL e HTML+TIME, elas também possuem semântica de apresentação associada.

A linguagem NCL, além de permitir a especificação de composições seqüenciais e paralelas, permite a especificação de composições para estruturação lógica do documento. Essas últimas não possuem semântica de apresentação associada. Além disso, diferente de Madeus, NCL não restringe que a definição de elos seja feita somente entre componentes diretos de uma mesma composição.

[6] Neste texto, entende-se por Madeus a linguagem declarativa do sistema de autoria e apresentação hipermídia chamado Madeus [JLRST98].

Representação de objetos de mídia

Tanto a linguagem NCL, como as linguagens SMIL, HTML+TIME e Madeus oferecem o suporte mínimo à definição de objetos de mídia. SMIL, HTML+TIME e Madeus permitem especificar elementos que representam, além dos tipos básicos, outros tipos de mídia mais específicos.

NCL, SMIL e HTML+TIME especificam como definir âncoras para cada tipo de objeto de mídia, no entanto, Madeus não o faz.

Em todas as linguagens em questão, o conteúdo de um objeto de mídia é definido em separado do documento hipermídia, sendo identificado através de uma referência descrita na forma de URI.

Características de apresentação separadas do componente

A linguagem HTML+TIME herda do HTML sua forma de especificar as características espaciais de apresentação dos componentes. Essas características são especificadas quando um componente é inserido em um documento HTML, ou seja, junto à definição do componente, ou então de forma independente através do uso de folhas de estilo (*style sheets*) [CSS298].

SMIL e NCL possuem uma linguagem para disposição espacial de elementos em seus documentos bastante similar. Podem ser especificadas regiões de apresentação onde os objetos visuais do documento serão exibidos. Também é possível especificar uma janela de apresentação do sistema, na qual as regiões deverão estar contidas. Ao contrário do SMIL, na NCL essa janela de apresentação não precisa ser única. Pode-se especificar uma janela diferente para cada dispositivo de saída utilizado para a exibição do documento.

HTML+TIME e SMIL permitem a definição de características espaciais de apresentação de um componente em um objeto separado, entretanto, as características temporais de apresentação, tais como início, duração e fim da exibição, estão embutidas em sua definição. Em Madeus, todas as características de apresentação de um componente, tanto espaciais como temporais, estão sempre embutidas em sua definição.

A linguagem NCL é a única que permite a especificação de todas as características de apresentação, tanto espaciais como temporais, separadas da definição do componente, através do elemento *descriptor*. NCL também permite que diferentes descritores sejam associados a um componente e ainda que um mesmo descritor seja associado a componentes distintos. A associação de um descritor a um componente pode ser especificada em qualquer composição onde o componente esteja inserido, ou em qualquer elo que tenha o componente como ponto terminal. Isso permite uma flexibilidade maior na especificação da apresentação, já que um mesmo objeto de mídia pode ser apresentado de formas diferentes, dependendo da parte do documento onde esteja inserido e dependendo até da forma de navegação para esse objeto.

Reuso

Permitir a definição do conteúdo separado do objeto de mídia, possibilita o reuso de conteúdo tanto em um mesmo documento, como em documentos diferentes, sendo essa a forma básica de reuso provida pelas linguagens aqui comparadas.

SMIL também permite o reuso de características espaciais de apresentação de componentes através das regiões. Entretanto, os componentes de um documento SMIL são reutilizados

somente através de cópias, o que dificulta a manutenção. O mesmo acontece com a linguagem HTML+TIME.

As linguagens Madeus e NCL provêem, além do reuso de conteúdo, a facilidade de reutilização de estruturas. Em Madeus, essa facilidade está relacionada à definição de composições, que podem ser reutilizadas na definição de outros documentos ou dentro de um mesmo documento. No entanto, [JLRST98] não deixa claro se o reuso é feito através de cópias da definição da composição ou simplesmente através de referências, como na linguagem NCL.

A linguagem NCL é mais abrangente que as demais linguagens aqui comparadas com relação ao requisito de reuso. Em NCL, é possível reusar tanto objetos de mídia quanto qualquer tipo de composição em um mesmo documento ou em documentos diferentes. Esse reuso é realizado através da especificação do atributo *uid* que possui uma referência para a URI do componente. Além disso, dentro de um mesmo documento, podem ser reusadas as regiões espaciais que definem áreas de exibição em alguma janela e ainda os descritores, que possuem informações relacionadas à apresentação dos componentes.

Definição de elos no escopo de uma composição

Em HTML+TIME e SMIL, o relacionamento temporal entre objetos é feito através das composições *par* e *seq* e o único tipo de elo oferecido é o elo tradicional hipermídia, que não é definido no escopo de uma composição.

Em Madeus, os relacionamentos temporais e espaciais entre componentes são definidos no escopo de uma composição, entretanto, os elos tradicionais hipermídia podem ser definidos entre dois componentes quaisquer. Sendo assim, a linguagem oferece a característica de reusar um mesmo componente com diferentes elos associados.

A linguagem NCL também limita o escopo de um elo a uma determinada composição, que é a composição (*context*, *par* ou *seq*) onde o elo foi definido. No entanto, NCL permite que elos sejam definidos entre componentes recursivamente contidos em uma composição. Isso fornece uma flexibilidade maior, pois além de permitir relacionamentos distintos dependendo da composição em que um componente estiver diretamente inserido, também permite relacionamentos distintos dependendo da hierarquia de composições onde um componente estiver recursivamente inserido [Soar00].

Relacionamentos complexos entre componentes

Em HTML+TIME e SMIL, as especificações temporais dos documentos são bastante semelhantes, devido ao fato da linguagem HTML+TIME ter sido definida baseando-se nos conceitos temporais da linguagem SMIL. A definição dos relacionamentos de sincronização temporal é realizada através de composições paralelas e seqüenciais ou através de elos hipermídia tradicionais. Esses elos associam uma única âncora de origem a uma única âncora de destino e são sempre disparados por uma seleção do usuário. As âncoras de origem e de destino podem representar um elemento inteiro ou identificar regiões espaciais ou temporais de objetos de mídia.

Madeus não fornece nenhum tipo de composição para a definição de sincronismo, mas permite especificar a sincronização temporal através de relações que determinam restrições ou relacionamentos causais entre componentes do documento. A autoria é bastante facilitada pois a linguagem oferece um conjunto de operadores de alto nível representando vários tipos de relacionamentos. No entanto, apesar da simplicidade, nem todos os tipos de relacionamentos

desejáveis em um documento hipermídia podem ser especificados. Por exemplo, em um documento Madeus não é possível especificar relacionamentos causais que possuem mais de uma condição baseadas em eventos de tipos diferentes (apresentação, seleção ou atribuição). Além disso, em Madeus, relacionamentos só podem ser definidos entre elementos diretamente contidos em uma mesma composição, o que restringe a forma como um documento deve estar estruturado.

Similar às linguagens HTML+TIME e SMIL, NCL permite que relacionamentos temporais entre componentes sejam definidos através de composições (*par* e *seq*). E ainda, similar a Madeus, NCL também permite representar relacionamentos causais ou restrições temporais entre vários componentes através de elos. Os elos em NCL associam âncoras que podem identificar componentes inteiros ou regiões espaciais e temporais em cada mídia. Esses componentes devem estar recursivamente contidos na composição onde o elo será definido, no entanto, podem estar diretamente contidos em composições distintas.

Todas as linguagens citadas permitem a definição de relações temporais entre componentes de uma forma relativa, isto é, o posicionamento temporal de um componente em relação ao outro e não apenas de modo absoluto no tempo. Em Madeus, também é permitido definir relações espaciais entre componentes, que os posicionam de forma relativa, facilitando bastante a autoria.

Flexibilidade na especificação temporal

SMIL e HTML+TIME não oferecem meios de especificar flexibilidade para a duração dos componentes. Madeus permite especificar a duração de um objeto de mídia através de um intervalo contendo um limite mínimo e um limite máximo para sua apresentação. No entanto, não existe o conceito de custo associado com a qualidade de apresentação na escolha de um valor dentro do intervalo. Com isso, a especificação de um documento Madeus não oferece parâmetros que guiem o formatador na escolha da melhor forma de realização do ajuste.

A linguagem NCL permite que um autor especifique a duração dos componentes de um documento, sejam eles objetos de mídia ou composições, através de funções de custo. Essas funções informam os valores mínimo, máximo e ideal para a escolha do tempo de exibição do componente, bem como o custo envolvido no ajuste desse tempo quando ele for diferente do ideal, oferecendo com isso a métrica necessária para escolha de uma solução ótima. Além disso, os tempos de espera envolvidos nos relacionamentos entre componentes (por exemplo, atributos *begin* e *end* das composições *par* e *seq*) também podem ser descritos através de funções de custo.

Adaptação à plataforma de exibição

Em NCL, SMIL e HTML+TIME podem ser definidas composições alternativas (*switch*), nos quais apenas um elemento constituinte é selecionado para ser exibido, de acordo com características da plataforma de exibição. Madeus não oferece esse recurso.

5. Conclusões e Trabalhos Futuros

O objetivo deste trabalho foi apresentar a linguagem NCL para a especificação declarativa de documentos hipermídia. Seguindo as tendências de padronização, a linguagem foi especificada através de uma DTD XML, tornando-se também um formato adequado para intercâmbio de documentos em um ambiente distribuído. NCL foi elaborada buscando satisfazer os principais requisitos identificados como desejáveis em linguagens de autoria hipermídia. Destacam-se assim:

- estruturação através do uso de composições;
- separação da descrição do comportamento de apresentação da especificação dos componentes;
- reuso de conteúdo, estrutura e características de apresentação;
- flexibilidade na especificação temporal e possibilidade de adaptação à plataforma de exibição;
- possibilidade de especificar relacionamentos multiponto, expressando causalidade ou restrição entre componentes.

A linguagem NCL vem sendo utilizada não só para autoria declarativa, mas também para intercâmbio dos documentos hipermídia no sistema HyperProp. O HyperProp é um sistema implementado em Java, para tratamento de documentos hipermídia baseados no modelo NCM, que vem sendo desenvolvido no Laboratório TeleMídia da PUC-Rio. Para integração da linguagem ao sistema foram implementados conversores capazes de transformar uma descrição NCL em uma representação Java dos objetos NCM, e vice-versa. O conversor NCL→HyperProp permitiu que o ambiente de execução do sistema HyperProp fosse utilizado para apresentação de documentos especificados em NCL, ao passo que o conversor HyperProp→NCL propiciou o uso das ferramentas gráficas do sistema HyperProp como alternativa para autoria de documentos NCL.

A definição da linguagem NCL possibilitou a especificação de uma outra linguagem, denominada X-SMIL [AnSo00b], cuja proposta foi estender a linguagem SMIL, contornando algumas de suas limitações anteriormente identificadas [RRMS99], através da incorporação de facilidades da linguagem NCL. Pretende-se, como trabalho futuro, relacionar a linguagem X-SMIL com a nova proposta de padrão que estende a linguagem SMIL, denominada SMIL-Boston [SMILB99].

Uma característica importante para o sucesso de uma linguagem de autoria declarativa é a sua capacidade de oferecer poder de expressão mantendo simplicidade na especificação. Embora, sempre que possível, a simplicidade tenha sido um alvo na definição da NCL, a prioridade deste trabalho foi alcançar um máximo de flexibilidade e poder de expressão. Nesse sentido, pretende-se como trabalho futuro identificar pontos da linguagem onde possam ser acrescentados elementos redundantes que facilitem a autoria declarativa. Um ponto de possível melhora seria incluir mecanismos de alto nível para a especificação de relacionamentos entre componentes, tais como as relações temporais oferecidas por Madeus. Um outro recurso importante, também existente em Madeus, e que poderia ser incorporado na linguagem NCL, é a possibilidade de definição de restrições espaciais entre componentes.

Finalmente, um outro estudo em andamento visa tornar a DTD NCL extensível, permitindo, por exemplo, que novos tipos de objetos de mídia, novos modos de expressar relacionamentos e novas formas de definir funções de custo sejam incorporadas sem causar mudanças na especificação de documentos já existentes.

Referências

[AnSo00a] Antonacci, M. J.; Soares, L. F. G. “Especificação NCL”. *Relatório Técnico do Laboratório TeleMídia, Departamento de Informática da PUC-Rio,* Rio de Janeiro, 2000.

[AnSo00b] Antonacci, M. J.; Soares, L. F. G. “Especificação X-SMIL”. *Relatório Técnico do Laboratório TeleMídia, Departamento de Informática da PUC-Rio,* Rio de Janeiro, 2000.

[Anto00] Antonacci, M. J.; "NCL: Uma Linguagem Declarativa para Especificação de Documentos Hipermídia com Sincronização Temporal e Espacial". *Dissertação de Mestrado, Departamento de Informática da PUC-Rio*, Rio de Janeiro, Abril 2000.

[CSS298] "Cascading Style Sheets, level 2 (CSS2) Specification". *W3C Recommendation*, Maio 1998. Disponível em: http://www.w3.org/TR/REC-CSS2

[HyTime97] "Hypermedia/Time-Based Structuring Language (HyTime)". *ISO/IEC 10744*, Agosto 1997. Disponível em: http://www.ornl.gov/sgml/wg8/document/n1920/html/n1920.html

[JLRST98] Jourdan, M.; Layaïda, N.; Roisin, C.; Sabry-Ismail, L.; Tardif, L. "Madeus, an Authoring Environment for Interactive Multimedia Documents", *Proceedings of the ACM Multimedia Conference 98*, pp. 267-272, Inglaterra, Setembro 1998.

[RoDu98] Rousseau, F.; Duda. A. "Synchronized Multimedia for the WWW". *Proceedings of the 7th. International World-Wide Web Conference*, Brisbane, Abril 1998.

[RRMS99] Rodrigues, L.M.; Rodrigues, R.F.; Muchaluat-Saade, D.C.; Soares, L.F.G. "Improving SMIL Documents with NCM Facilities", *Proceedings of the Multimedia Modeling Conference'99*, Canada, Outubro 1999.

[SGML86] "Standard Generalized Markup Language (SGML)". *ISO 8879*; Outubro 1986.

[SMIL98] "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification". *W3C Recommendation*, Junho 1998. Disponível em: http://www.w3.org/TR/REC-smil

[SMILB99] "Synchronized Multimedia Integration Language (SMIL) Boston Specification" — *W3C Working Draft*. Novembro 1999. Disponível em: http://www.w3.org/TR/1999/WD-smil-boston-19991115/smilboston.pdf.

[Soar00] Soares, L.F.G. "Modelo de Contextos Aninhados Versão 2.3". *Relatório Técnico do Laboratório TeleMídia, Departamento de Informática da PUC-Rio*, Rio de Janeiro, 2000.

[SoCR95] Soares, L.F.G.; Casanova, M.A.; Rodriguez, N.L.R. "Nested Composite Nodes and Version Control in an Open Hypermedia System". *International Journal on Information Systems; Special Issue on Multimedia Information Systems*, 20(6). Elsevier Science Ltd, Inglaterra. 1995. pp. 501-519.

[TIMEH98] "Timed Interactive Multimedia Extensions for HTML (HTML+TIME)". *W3C Note*, Setembro 1998. Disponível em: http://www.w3.org/TR/NOTE-HTMLplusTIME

[XML98] "Extensible Markup Language (XML) 1.0". *W3C Recommendation*, Fevereiro 1998. Disponível em: http://www.w3.org/TR/REC-xml