

Proposta de adequação do padrão HTML+TIME ao modelo de referência de sincronização multimídia

Fernando Luiz de Oliveira, Fabiano Fagundes

Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas (CEULP)
Palmas, TO, Brasil

nandoluiz@ulbra-to.br, fagundes@ulbra-to.br

***Resumo.** A necessidade de integrar, em uma mesma aplicação, diversos formatos de mídias, tornou o HTML insuficiente no tratamento de sistemas multimídias que, para serem corretamente apresentados, precisam que as partes que os compõem sejam tanto apresentadas como sincronizadas entre si. Para suprir estes requisitos foram desenvolvidos padrões de sincronização, tais como o HTML+TIME e o SMIL, definidos de acordo com um ou mais métodos de especificação do modelo de referência de sincronização multimídia. Este artigo apresenta um estudo sobre o padrão de sincronização HTML+TIME propondo elementos que os tornem totalmente compatíveis com os métodos de especificação de sincronização nos quais são baseados.*

1 Introdução

A Internet trouxe várias mudanças na forma como os sistemas de computação são tratados. Sistemas que antes se preocupariam apenas com o tratamento de um ou dois tipos de dados (texto e imagem, por exemplo), necessitam agora prover mecanismos para armazenamento, comunicação e apresentação de diversos tipos de dados (mídias) embutidos em uma mesma aplicação.

Um sistema capaz de tratar mais de uma mídia, que suporte o processamento integrado, contendo pelo menos uma mídia contínua (áudio, vídeo ou animação) e que possibilite a manutenção do grau de integração entre as mídias que o compõe, pode ser definido como um sistema multimídia [Blakowski & Steinmetz 1996]. Estes sistemas são estabelecidos de acordo com especificações de sincronização, que são responsáveis por estabelecer as formas de integração entre mídias, preocupando-se em oferecer os mecanismos para que uma aplicação seja apresentada com uma qualidade aceitável.

Para possibilitar que os dados fossem apresentados, sem considerar questões de sincronização, foram criadas linguagens de marcação, tais como o HTML [W3C 1999]. Apesar das linguagens de marcação possibilitarem que mídias sejam exibidas, elas não são suficientes para a correta apresentação de um sistema multimídia. Isto porque estes sistemas, além de permitir que os objetos de uma aplicação sejam visualizados, necessitam que tais objetos sejam sincronizados, característica que as linguagens mencionadas não possuem. Para oferecer estes recursos foram criados, baseados em XML [W3C 2000], os padrões de sincronização HTML+TIME e SMIL – *Synchronized Multimedia Integration Language*. Estes padrões são definidos de acordo com DTDs (*Document Type Definition*), e nestas DTDs são especificados os elementos e atributos responsáveis por prover a sincronização entre os objetos.

Nas próximas seções serão tratadas questões necessárias para promover a sincronização entre os objetos de uma apresentação, tais como o modelo de referência de sincronização multimídia e as especificações de sincronização, bem como o padrão HTML+TIME, sendo que este último é o objeto principal de investigação neste artigo. Na sequência será apresentado o resultado da verificação da real adequação do padrão HTML+TIME aos modelos de referência de sincronização multimídia. Como contribuição deste artigo será apresentada uma proposta de elementos que complementaríamos o padrão para sua completa adequação ao modelo de referência.

2 Multimídia: Sincronização e Especificação

A necessidade de compor uma aplicação multimídia com áudio, vídeo, imagens e textos, torna de extrema necessidade que regras para a apresentação, armazenamento e comunicação sejam definidas. Estas regras podem especificar tanto as formas de interações entre estas mídias ou entre partes de uma mesma mídia, como também a forma de tratamento caso haja perda de informação.

Sincronização multimídia, numa visão geral, engloba relações de espaço, de tempo e de composição [Vazirglannis & Mourlas 1995]. O relacionamento espacial dos objetos é genericamente expresso pela localização dos mesmos em um conjunto de coordenadas no espaço. O relacionamento de composição indica as conexões entre objetos inter-relacionados e o efeito que alterações em um objeto têm sobre o outro. Por fim, as relações de tempo são aquelas que definem as dependências temporais entre os objetos de mídia. É sob o enfoque das relações temporais que o termo sincronização multimídia será utilizado neste artigo.

2.1 Modelo de referência

A representação de relações temporais entre objetos de mídia está baseada em modelos de tempo. Estes modelos tornam possíveis a identificação e a especificação de relações temporais entre diferentes mídias, e em particular as relevantes ao processo de sincronização multimídia [Fagundes 2002].

Em [Blakowski & Steinmetz 1996] é apresentado um modelo de referência baseado em quatro camadas em que cada camada, através de uma interface, oferece serviços acerca de sincronização diretamente para a aplicação ou para a camada superior. Esta divisão em camadas favorece o bom entendimento do modelo por causa da divisão do nível de abstração usada entre uma camada e outra.

As quatro camadas que compõem este modelo serão apresentadas através da Figura 1, porém neste artigo será tratada apenas a camada de especificação, já que esta é a responsável pela definição das especificações de sincronização multimídia.

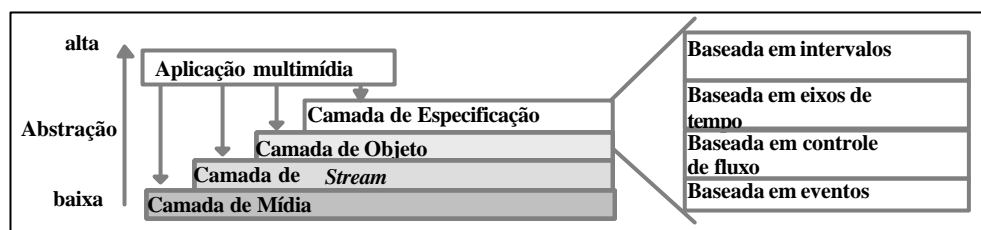


Figura 1. Modelo de referência para sincronização ([Blakowski & Steinmetz 1996]-modificado)

2.2.1 Especificação baseada em intervalos

Na especificação baseada em intervalo, o período gasto para a apresentação de cada um dos objetos que constituem uma aplicação multimídia, é identificado como um intervalo. Existem treze diferentes formas para sincronizar a apresentação de dois intervalos (objetos) distintos, sendo que, pelas características inversas apresentadas por seis destas relações, pode-se resumir estas treze relações em apenas sete [Allen 1983]. Um exemplo de uma relação inversa à outra, seria a relação B After A, inversa a relação A Before B. Na Figura 2 será apresentada esta relação.

A Before B

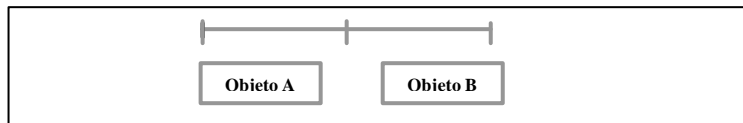


Figura 2. A Before B

Esta relação define que o objeto identifica por “B” somente poderá ser apresentado após a apresentação de um objeto identificado por ”A”. Esta relação caracteriza-se por ter um intervalo entre a apresentação dos objetos. De forma idêntica são trabalhadas as outras seis relações: A Overlaps B, A Starts B, A Meets B, A Equals B, A During B e A Finishes B.

Operações com um parâmetro de atraso				
before(? ₁)	beforeendof(? ₁)	cobegin(? ₁)	coend(? ₁)	
Operações com dois parâmetros de atraso				
while (? ₁ , ? ₂)	delayed (? ₁ , ? ₂)	startin (? ₁ , ? ₂)	endin (? ₁ , ? ₂)	cross (? ₁ , ? ₂)
Operações com três parâmetros de atraso				
overlaps (? ₁ , ? ₂ , ? ₃)				

Figura 3. Operações do modelo avançado de sincronização baseado em intervalos [Buford 1994]

Esta especificação, por lidar com relações entre dois intervalos, condiciona a sincronização da apresentação de um intervalo a outro, trazendo o benefício de não necessitar prever a duração da apresentação do primeiro intervalo para programar a apresentação do segundo, como foi apresentado na relação A Before B (Figura 2).

2.2.2 Especificação baseada em eixos

Neste modelo, os eventos de apresentação, como o início e o fim de uma apresentação, são mapeados para um eixo (linha de tempo) compartilhado por todos os objetos pertencentes a uma apresentação multimídia. Este eixo corresponde a uma abstração do tempo real [Steinmetz & Nahrstedt 1995], e ao contrário do modelo baseado em intervalos, caso um objeto não seja devidamente apresentado (por atraso ou removido), não comprometerá a apresentação dos demais objetos subsequentes. Isto porque, a sincronização de um objeto não está centrada no objeto que foi apresentado ou no objeto que ainda será. As sincronizações dos objetos estão centradas na definição de seus próprios parâmetros, tais como início, duração e fim de sua apresentação.

2.2.3 Especificação baseada em controle de fluxo

Neste tipo de especificação, o fluxo da apresentação de caminhos concorrente é sincronizado em pontos pré-definidos. As especificações baseadas em Redes de Petri são um exemplo de aplicação deste modelo.

Para ilustrar, será apresentado através da Figura 4, um exemplo de especificação baseada em Redes de Petri. Este exemplo exhibe uma aplicação multimídia composta por diversos objetos sendo esta especificação utilizada para sincronizá-los.

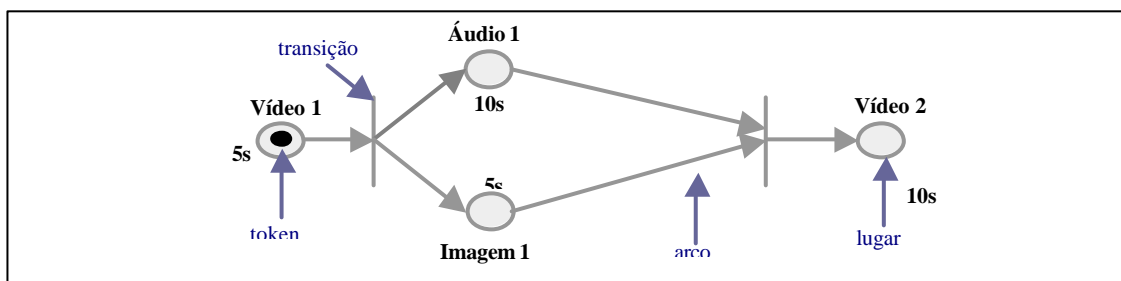


Figura 4. Exemplo de uma especificação baseada em controle de fluxo

Conforme definido na Figura 4, o primeiro vídeo possui o *token* e por isto logo que a aplicação for iniciada começará a ser apresentado. Após cinco segundos, o primeiro vídeo é encerrado, e passa o *token* para a primeira transição. Esta transição repassa para os dois objetos seguintes. O áudio e a imagem recebem a permissão para serem iniciados. A imagem por ter um período de visualização menor, será finalizada e passa o seu *token* para a segunda transição. No entanto esta transição somente poderá repassá-lo quando receber a segunda permissão, vinda da finalização do áudio. Ao receber os dois *tokens* a segunda transição repassa-o e o segundo vídeo será exibido. Finalizando a apresentação deste vídeo a aplicação será fechada.

2.2.4 Especificação baseada em eventos

Em [Rodrigues 1997] um evento é definido através da apresentação de um conjunto marcado de unidades de informação de um nó (evento de apresentação) ou pela sua seleção (evento de seleção) ou pela alteração de um atributo de um nó (evento de

atribuição). Um evento pode estar em um dos estados: *sleeping*, *preparing*, *prepared*, *occurring*, e *paused*, e cada evento tem um atributo associado no nó onde ele é definido, chamado, *ocurred*, que tem a função de contabilizar quantas vezes um evento transita dos estados *occurring* para *prepared* durante a apresentação do documento (Rodrigues 97).

As especificações apresentadas fornecem as regras e as formas de interação entre os diversos objetos de uma apresentação multimídia. Os padrões de sincronização, portanto, são implementados tendo como base uma ou mais destas especificações. A seguir será apresentado o padrão HTML+TIME, abordando as especificações usadas em sua implementação, características e sua adequação ao modelo avançado de sincronização.

3 HTML+TIME

O padrão HTML+TIME (*Timed Interactive Multimedia Extensions*) foi desenvolvido por um grupo de especialistas, das empresas Microsoft, Compaq e Macromedia, pertencentes do comitê de desenvolvimento do SMIL, padrão de sincronização multimídia, recomendado pela W3C, órgão regulamentador dos padrões técnicos para *Web* [W3C 1998]. O HTML+TIME objetiva integrar áudio, vídeo, imagem e/ou texto em uma mesma aplicação, definindo como, quando e por quanto tempo cada uma destas mídias deve ficar ativa [Microsoft 2002]. Com a utilização desta tecnologia pode-se criar *sites* com conteúdo dinâmico melhorando a forma como ele é apresentado. Isto porque, em uma mesma aplicação, pode-se associar uma imagem a um texto definindo efeitos e regras de sincronização as suas apresentações.

O HTML+TIME é definido de acordo com uma DTD que descreve os elementos, atributos, propriedades, eventos e métodos que podem ser utilizados, bem como o resultado produzido pela utilização de um destes itens. No HTML+TIME atributos e propriedades se diferenciam pela forma de sua utilização. Os atributos são usados juntamente com as *tags* HTML, enquanto que as propriedades são usadas através de *scripts*. Por padrão, o HTML+TIME adota letras maiúsculas para representar os atributos e letras minúsculas para escrever as propriedades.

Atributo BEGIN | Propriedade begin

Tabela 1. Sintaxe do atributo begin

	Sintaxe
Atributo:	< Element t:BEGIN = "valorTempo" >
Propriedade	obj.begin = "valorTempo"
Utilizado por:	Todos os elementos

Descrição: Esta propriedade/atributo determina o momento a partir do qual um objeto de uma apresentação será apresentado.

Exemplo de Utilização

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<html>
<XML:NAMESPACE PREFIX="t"/>
<STYLE>
.time {behavior:url(#default#time);}
</STYLE>
<script>
function iniciar(objID)
{
    objID.begin = "0"
}
</script>
</head>
<body bgcolor="#CECCB9">
<h3 ID="testarAtributo" CLASS="time" t:BEGIN="1">Apresentação de um texto usando o
atributo "t:BEGIN" </h3>
<h4 ID="testarProp" CLASS="time" t:BEGIN="indefinite">Apresentação de um texto
através da propriedade "begin" </h4>
<input type="button" name="btProp" value="Usar a propriedade" onClick="iniciar(testarProp);">
</body></html>
```

Figura 5. Utilização do atributo | propriedade BEGIN

Este exemplo (Figura 5) apresenta dois textos na tela de forma sincronizada, onde um deles (ID = "testarAtributo") será apresentado um segundo após a página ser iniciada (t:BEGIN = "1"). Já o outro texto (ID = "testarProp") ficará a espera de uma interação com o usuário. No momento em que o usuário clicar no botão, o evento "onClick" do botão será disparado e a função "iniciar" será chamada definindo que o objeto, no caso o segundo texto, seja imediatamente apresentado (objID.begin = "0"). Neste exemplo, caso o usuário não clicasse no botão, o segundo texto nunca seria visualizado. Isto porque, através da utilização do atributo t:BEGIN = "indefinite" definiu-se que o objeto associado não teria um tempo certo para sua apresentação, ficando a cargo da propriedade objID.begin = "0", ativada quando o evento "onClick" fosse chamado, definir a exibição deste objeto.

4 Adequação ao Modelo Avançado de Sincronização

Uma característica importante e que deve ser ressaltada ao se trabalhar com uma linguagem de sincronização é o método de especificação que esta linguagem utiliza. Isto porque cada uma das especificações citadas na seção 2.2 define um conjunto de características que podem auxiliar no bom entendimento da linguagem. No caso do HTML+TIME o modelo seguido é formado pelas melhores características tanto da especificação baseada em eixos como da especificação baseadas em relações de intervalos [Microsoft 2002].

A afirmação de que o HTML+TIME é formado pelas melhores características das especificações citadas acima leva a crer que todas as relações binárias do modelo baseado em intervalos poderão ser facilmente especificadas através desta linguagem.

Esta seção possui a finalidade de apresentar os resultados da verificação da adequação dos padrões estudados a estas operações. Para isso, serão apresentados, na

seqüência, alguns exemplos de utilização de elementos do HTML+TIME para realizar a especificação de algumas relações propostas no modelo avançado de sincronização.

O HTML+TIME, na forma como está definido, suporta apenas duas das operações com um parâmetro de atraso (before e cobegin). A figura abaixo apresenta a especificação da relação before. Esta operação caracteriza-se pelo tempo de atraso entre o fim da apresentação de um objeto e começo de outro (Figura 6).

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<html>

<head>
<title> Exemplo de Utilização </title>
<XML:NAMESPACE PREFIX="t"/>
<STYLE>
        .time {behavior:url(#default#time);}
</STYLE>
<body bgcolor="#CECCB9">
  <par>
    
    
  </par>
</body>
</html>

```

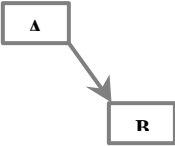


Figura 6. before(?₁)

O padrão HTML+TIME oferece o atributo/propriedade t:BEGINAFTER, que especifica que um objeto somente poderá ser visualizado após um outro. No caso deste exemplo (Figura 6), a segunda imagem (ID="bolaVerde.gif") será apresentada dois segundos após o término da apresentação da primeira imagem (ID="bolaAzul.gif"), possuindo como parâmetro de atraso $?_1 = 2$. Observa-se que, neste caso, os elementos definidos para o HTML+TIME atendem perfeitamente a especificação da relação before. O mesmo ocorre com a relação cobegin, possível de ser especificada através dos elementos já existentes.

Este padrão não suporta nenhuma das operações (do modelo avançado de sincronização) que utilizam dois ou três parâmetros de atraso. Isto porque, para poder implementá-las, necessitaria de alguns recursos que não estão disponíveis. A próxima seção apresenta e descreve a utilização de novos atributos que, acrescentados ao padrão, o tornaria habilitado para a especificação de todas as relações binárias do modelo avançado de sincronização.

4.1 Elementos para especificação de sincronização

Com o acréscimo de dois elementos, que seriam o t:ENDWITH e o t:ENDAFTER, o HTML+TIME estaria apto a permitir a especificação completa das relações do modelo avançado de sincronização. Estes elementos seriam definidos como apresenta a Tabela 2.

Tabela 2. Sintaxe dos atributos t:ENDWITH e t:ENDAFTER

	Sintaxe		Sintaxe
Atributo:	< Element t:ENDWITH = "id" >	Atributo:	< Element t:ENDAFTER = "id" >
Propriedade	obj.endWith = "id"	Propriedade	obj.endAfter = "id"
Utilizado por:	Todos os elementos	Utilizado por:	Todos os elementos

A utilização destes elementos será melhor exemplificada através da especificação de algumas relações binárias que não poderiam ser realizadas anteriormente, sem a existência destes.

Como primeiro exemplo, visualizado através da Figura 7, tem-se a especificação da relação beforeendof(?₁) que utiliza o elemento t:ENDWITH. Esta operação caracteriza-se pelo tempo de atraso entre o início da exibição de um objeto e o final da apresentação de um outro (Figura 7).

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN" >
<html>

<head>
<title> Exemplo de Utilização </title>
<XML:NAMESPACE PREFIX="t"/>
<STYLE>
        .time {behavior:url(#default#time);}
</STYLE>
<body bgcolor="#CECCB9">
  <par>
    
    
  </par>
</body></html>

```

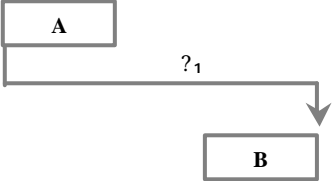


Figura 7. beforeendof(?₁)

Como já foi ressaltado anteriormente, o HTML+TIME não oferece recursos para que esta operação seja atendida. No entanto, uma alternativa para que ela fosse implementada neste padrão seria a criação de um novo atributo: t:ENDWITH, inverso ao já existente: t:BEGINWITH.

Este novo atributo teria o mesmo princípio do já definido, porém, ao invés de definir que um objeto iniciasse “n” segundos após o início de um outro, especificaria que este objeto terminasse “n” segundos após o início da mídia referenciada. No caso do exemplo acima (Figura 7), a segunda imagem será finalizada cinco segundos (t:END="5") após o início da primeira imagem (t:ENDWITH="img1"), possuindo como parâmetro de atraso: ?₁ = 5.

O segundo exemplo, que apresenta a especificação da relação overlaps(?₁, ?₂, ?₃), permite verificar a utilização do atributo t:ENDAFTER que, em conjunto com o atributo descrito anteriormente, t:ENDWITH, possibilita ao padrão HTML+TIME atender a todas as relações binárias. Este exemplo foi escolhido justamente por ser o mais complexo e que utiliza estes dois atributos em uma mesma especificação.

O segundo atributo a ser criado, t:ENDAFTER, permitiria que um determinado objeto fosse finalizado apenas quando se tivesse passado “n” segundos após a finalização de um outro.

A operação overlaps utiliza três parâmetros de atraso onde: o primeiro (?₁) define que o começo de um objeto “B” seja de “n” segundos após o início de um outro (“A”); o segundo (?₂) especifica que o fim de “A” seja “n” segundos após o início da apresentação de “B”; e o terceiro (?₃) estabelece que o término de “B” seja de “n” após a finalização de “A” (Figura 8).

<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN"> <html> <head> <title> Exemplo de Utilização </title> <XML:NAMESPACE PREFIX="t"/> <STYLE> .time {behavior:url(#default#time);} </STYLE> <body bgcolor="#CECCB9"> <par></pre>	
<pre> </pre>	
<pre></par></body></html></pre>	

Figura 8. overlaps(?₁)

O exemplo apresentado utiliza as modificações sugeridas nos atributos para implementar esta operação. Em ambos os casos, a primeira imagem será finalizada quatro segundos após o início da segunda e esta será iniciada dois segundos após a primeira, sendo finalizada cinco segundos após o término da primeira.

Todas as demais treze relações são possíveis de serem especificadas com o acréscimo destes dois atributos. A alteração na forma como os navegadores trabalham o HTML+TIME permitiria que, sem grandes dificuldades, fossem visualizadas aplicações multimídia corretamente sincronizadas a partir de especificações baseadas em intervalos.

5 Considerações Finais

As novas necessidades dos sistemas de computação, em questões de armazenamento, comunicação e apresentação de diversos formatos de dados, tornaram as tecnologias existentes insuficientes para que, sistemas multimídias, fossem corretamente sincronizados e apresentados. Esta insuficiência possibilitou diversos estudos acerca do tema, o que viabilizou a criação de linguagens e padrões que comportassem estas necessidades.

Os padrões de sincronização oferecem recursos para que um sistema multimídia seja corretamente apresentado, de forma a manter uma qualidade na execução desta tarefa. O padrão HTML+TIME trabalha com dois métodos de sincronização: baseado em eixos tempo e baseado em intervalos. Com relação a especificação de sincronização baseado em eixos de tempo, o HTML+TIME oferece um conjunto de recursos, para que os seus requisitos sejam corretamente atendidos. No entanto, isto não é válido para a especificação baseada em intervalos. Ao trabalhar com esta especificação, este padrão não oferece suporte a todas as

operações do modelo avançado de sincronização. Neste caso, o HTML+TIME implementa apenas duas (`before(?1)` e `cobegin(?1)`).

Considerando a utilidade deste padrão, e a possibilidade do mesmo se adequar às características de outros padrões de sincronização como o SMIL, leva a necessidade de se trabalhar seus elementos e atributos para torná-lo completo em sua proposta de especificação. O estudo aqui realizado verificou que o acréscimo de dois atributos aos elementos de mídia é o suficiente para que o padrão HTML+TIME permita especificar todas as relações binárias do modelo avançado de sincronização multimídia.

Assim, para que este padrão suporte as operações deste modelo, foi proposta a criação de dois atributos: `t:ENDWITH` e `t:ENDAFTER`. A definição destes elementos, dada a sua simplicidade, permite que se dê continuidade a este trabalho através da implementação de um navegador que apresente corretamente todos os elementos definidos no HTML+TIME e acrescente a apresentação também de acordo com os atributos aqui especificados.

Referências

- [Allen 1983] Allen, J. F. “Maintaining knowledge about temporal intervals”. Communications of the ACM, 1983.
- [Blakowski & Steinmetz 1996] Blakowski, G., Steinmetz, R. “A media synchronization survey: reference model, specification and case studies”. IEEE JSAC, v. 4, n.1, pp. 5-35, jan. 1996.
- [Buford 1994] Buford, J. F. K. et al. “Multimedia Systems”. ACM Press, 1994.
- [Fagundes 2002] Fagundes, Fabiano. “Especificação de uma meta-linguagem para sincronização multimídia”, Dissertação de Mestrado, UFSC: Florianópolis, 2002.
- [Microsoft2002] Microsoft. “Introduction to HTML+TIME”, <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/behaviors/time.asp>, 2002.
- [Rodrigues 1997] Rodrigues, Rogério Ferreira. “Formatação Temporal e Espacial no Sistema Hyperprop”, Dissertação (Mestrado em Informática) 1997. Departamento de Informática da Universidade PUC-Rio. PUC-Rio, Rio de Janeiro. 116p.
- [Steinmetz & Nahrstedt 1995] Steinmetz, R., Nahrstedt, K. “Resource Management in Networked Multimedia Systems”. IEEE Computer, Maio, 1995.
- [Vazirglannis & Mourlas 1995] Vazirglannis, M., Mourlas, C. “Integrated Multimedia Object and Application Modelling Based on Events and Scenarios”. Proceedings of the Int. Workshop on Multi-Media Database Management Systems, pp. 48-55, Agosto 1995.
- [Wahl & Rothermel 1994] Wahl, T., Rothermel, K. “Representing Time in Multimedia-Systems”. In *Proc. IEEE International Conference on Multimedia Computing and Systems*, Boston, Maio 1994
- [W3C 1999] W3C. “HTML 4.01 Specification”, <http://www.w3.org/TR/html4/>, 1999.
- [W3C 2000] W3C. “Extensible Markup Language (XML) 1.0 (Second Edition)”, <http://www.w3.org/TR/2000/REC-xml-20001006>, 2000.
- [W3C 1998] W3C. “Especificação de Synchronized Multimedia Integration Language 1.0 (SMIL)”, <http://www.utad.pt/~leonelm/w3ctranslations/smil/>, 1998.