

Apresentações Hiperídia Apoiadas por uma Ferramenta de Animação Facial*

**Rogério F. Rodrigues, Paula S. Lucena Rodrigues¹, Bruno Feijó¹,
Luiz Velho², Luiz Fernando G. Soares¹**

¹ DI – PUC-Rio
Rua Marquês de São Vicente, 225,
Rio de Janeiro, RJ - 22453-900.

² IMPA – Inst. de Mat. Pura e Aplicada
Estrada Dona Castorina, 110
Rio de Janeiro, RJ - 22460-320.

rogerio@telemidia.puc-rio.br, pslr@telemidia.puc-rio.br,
bruno@inf.puc-rio.br, lvelho@impa.br, lfgs@inf.puc-rio.br

Resumo. *Este artigo descreve a integração de um formatador para controle de apresentações hiperídia com uma ferramenta de animação que oferece sincronização de fala e expressões faciais. O sistema resultante estabelece uma base que pode ser explorada no desenvolvimento de classes importantes de aplicações Web e de TV interativa, tais como ensino a distância, jogos, entretenimento, jornalismo, teleconferência etc.*

Abstract. *This paper describes the integration of a formatter for controlling hypermedia presentations with an animation tool, which offers synchronization of speech and facial expressions. The integrated system provides an infrastructure that can be explored in the development of important classes of Web and Interactive TV applications, such as distance learning, games, entertainment, news, teleconference etc.*

* Este trabalho foi realizado com apoio do Fundo Setorial para o Desenvolvimento Tecnológico das Telecomunicações (FUNTTEL), através do contrato 0594/02; e do CNPq

1. Introdução

A inclusão social (digital) e a integração regional pelas tecnologias da informação buscam, em uma direção, possibilitar que todo cidadão tenha acesso às informações. No entanto, esse é apenas um viés da questão. Em outra direção, é importante garantir a todos o direito de geração e disseminação de informação (conteúdo digital), permitindo ao cidadão que sua produção cultural seja registrada e divulgada, tanto localmente quanto para o público em geral. Seja qual for o viés, é desejável que o uso e domínio das tecnologias da informação promovam a formação e inserção profissional do cidadão no mercado de trabalho, associadas ao domínio dos equipamentos, programas de computador e das inúmeras atividades que envolvem o acesso e a capacidade de gerar e apresentar informações.

A realização de ações de estímulo, valorização e difusão da cultura gestada, vivida e produzida em regiões marginalizadas brasileiras exige políticas governamentais conseqüentes, que contemplem também o domínio tecnológico de ferramentas que as propiciem. Um exemplo de tais políticas começa a ser esboçado nos primeiros passos rumo ao sistema brasileiro de TV digital.

Nesse cenário, sistemas hipermídia (ou multimídia interativos) constituir-se-ão em uma das ferramentas mais importantes a serem dominadas. Sistemas de autoria hipermídia são o suporte para a geração de informação, não se restringindo apenas à concepção dos conteúdos em si, mas incluindo também a ideação de como eles devem ser apresentados. Sistemas de exibição hipermídia (foco deste artigo) são os responsáveis pela apresentação especificada. Todos esses sistemas têm por base um modelo conceitual embutido em alguma linguagem de especificação de documentos.

Sistemas hipermídia têm se tornado cada vez mais comuns no dia-a-dia de um grande número de pessoas, sendo a WWW a principal responsável pela popularização desse paradigma de estruturação das informações. Um componente de grande importância em um sistema hipermídia é o formatador de documentos, também chamado de *browser*, *player* ou *engine*. As principais funções de um formatador são: controlar a exibição dos diferentes fragmentos de informação (*objetos de mídia*) e manter as relações de sincronização temporal e espacial especificadas pelo autor do documento [Rodr03].

Um aspecto relevante na apresentação de documentos hipermídia são os diferentes tipos de conteúdo que podem estar presentes. Quanto maior a diversidade de tipos de mídia e de formatos, maior o potencial de expressividade para os autores. O suporte aos diferentes formatos está relacionado com as ferramentas de exibição (decodificadores, descompressores, renderizadores etc.) existentes na plataforma de execução dos documentos. É interessante que haja um desacoplamento das funções de gerência da apresentação (manutenção do sincronismo, monitoramento da apresentação, adaptação das exibições, negociação de QoS para exibição dos objetos etc.), da tarefa de exibição, propriamente dita, dos conteúdos dos objetos. A idéia é seguir uma arquitetura onde as ferramentas são, na realidade, módulos (*plug-ins*) que estendem os recursos do sistema de exibição hipermídia, sendo acopladas ao formatador. Para isso, uma interface aberta deve ser especificada a fim de permitir não apenas a incorporação de novas ferramentas de exibição, mas também possibilitar que os conteúdos dos objetos exibidos nas diferentes ferramentas possam estar verdadeiramente relacionados [RoRS01].

Seguindo a idéia apresentada no parágrafo anterior, este artigo discorre sobre o trabalho de integração de um formatador hipermídia, o formatador do sistema HyperProp [Rodr03], com uma ferramenta de exibição do tipo *talking head*, a ferramenta Expressive Talking Heads (ETHs) [Luce02, LuFV03], capaz de sintetizar textos em áudio e enunciarlos por meio de uma face com emoção e sincronismo dos movimentos labiais.

A importância do ETHs para apresentação de conteúdos textuais pode ser evidenciada em vários exemplos, desde a apresentação de conteúdos textuais para pessoas portadoras de deficiências visuais, onde apenas a funcionalidade de síntese de áudio seria explorada, até em aplicações onde avatares assumiriam papel preponderante, como em teleconferência, aplicações em educação a distância, telemedicina, jogos eletrônicos, telejornalismo, comércio eletrônico etc., onde toda a potencialidade de síntese de áudio e de face com emoção seria explorada. A integração de tal ferramenta de exibição a um formatador hipermídia vai possibilitar a sincronização temporal e espacial do avatar falante com outros conteúdos (objetos de mídia).

Este artigo encontra-se organizado da seguinte forma. De maneira resumida, as Seções 2 e 3 descrevem o formatador HyperProp e a ferramenta Expressive Talking Heads, respectivamente. A Seção 4 comenta os aspectos relacionados com a integração desses dois sistemas. A Seção 5 oferece um exemplo de documento hipermídia que faz uso do sistema integrado. Por fim, a Seção 6 comenta trabalhos relacionados e a Seção 7 apresenta as conclusões e trabalhos futuros.

2. Formatador HyperProp

O formatador HyperProp é o componente responsável pelo controle da apresentação dos documentos, utilizando-se de um modelo de execução baseado em eventos e máquinas de estados dos eventos. A partir desse modelo, o formatador é capaz de construir um plano de exibição para guiá-lo na manutenção do sincronismo da apresentação [Rodr03].

Para evitar que as apresentações sejam especificadas com base no modelo de execução do formatador, que é um modelo orientado ao controle e, portanto, oferece um nível mais baixo de abstração, o formatador HyperProp dispõe de alguns módulos conversores. Existem atualmente três desses módulos implementados, permitindo assim que os documentos sejam descritos na linguagem NCL (*Nested Context Language*) [MuSS03], na linguagem SMIL (*Synchronized Multimedia Integration Language*) [W3C01], ou utilizando diretamente objetos do modelo NCM (*Nested Context Model*) [Rodr03]. Como exemplo, a Seção 5 apresenta um documento descrito em NCL.

Para incorporação das ferramentas de exibição, o formatador HyperProp especifica uma API de integração, estabelecendo métodos que as ferramentas devem implementar e o modo como elas devem notificar o formatador da ocorrência de eventos na apresentação. Os métodos implementados pelas ferramentas são chamados para comandar a exibição dos conteúdos (métodos para preparar, iniciar, parar, pausar etc.). O formatador utiliza o padrão de projeto *Observer* [Gamm95] para ser notificado das transições nas máquinas de estados dos eventos e controlar a execução do documento como um todo, deixando o controle dessas máquinas sob a responsabilidade de cada ferramenta. Como exemplo, quando uma ferramenta *mpeg1* termina de exibir o conteúdo de um vídeo, ela deve gerar a transição na máquina de estados do evento de

apresentação do objeto, que retrata o fim da sua exibição. Os exibidores que não implementam todos os métodos necessários, ou não sabem como exercer o controle sobre as máquinas de estados dos eventos do formatador, devem ser incorporados ao sistema de exibição hipermídia através de um adaptador (*Adapter* [Gamm95]), conforme ilustrado na Figura 1. Essa foi a abordagem utilizada na integração da ferramenta ETHs com o formatador HyperProp. Além do ETHs, o formatador possui integração similar com exibidores JMF (*Java Media Framework*) e Quicktime.¹

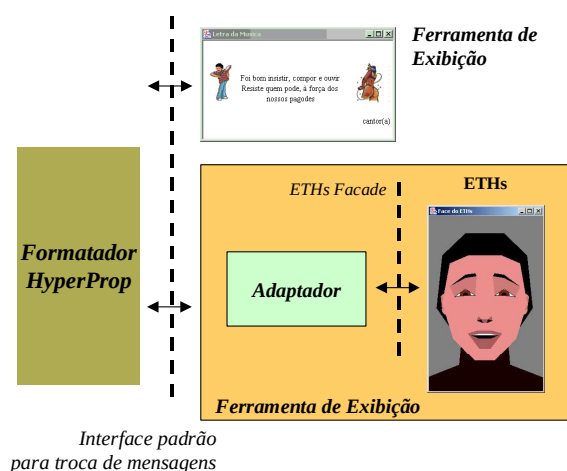


Figura 1 – Integração das ferramentas de exibição ao formatador HyperProp.

O formatador HyperProp encontra-se implementado em Java e possui uma versão disponível para *download*.² Os detalhes do seu modelo de execução e da API de integração podem ser encontrados em [RoRS01, Rodr03].

3. Ferramenta Expressive Talking Heads

O Expressive Talking Heads [Luce02, LuFV03] é um sistema de animação facial interativo, capaz de receber como entrada um texto com marcações descrevendo, por exemplo, a emoção do personagem, o gênero da voz, o idioma da fala, o posicionamento da cabeça e dos olhos, e gerar como saída um áudio (síntese do texto) e a face de um personagem virtual pronunciando as palavras. Durante a animação, a ferramenta reflete na face não somente os movimentos dos lábios enunciando a fala, mas também as expressões faciais solicitadas pelas marcações embutidas no texto. A Figura 2 oferece uma visão geral da organização e do funcionamento da ferramenta ETHs. A figura também ilustra um exemplo da linguagem de marcação de entrada. As marcações são definidas como simples extensões da linguagem HTML, através de novos atributos para o elemento *div*.

¹ Existem, também implementadas no sistema, ferramentas para os tipos básicos de texto, incluindo HTML, e de imagem estática, sendo que essas ferramentas não necessitam de adaptadores.

² <http://www.telemidia.puc-rio.br/products/formatter/>

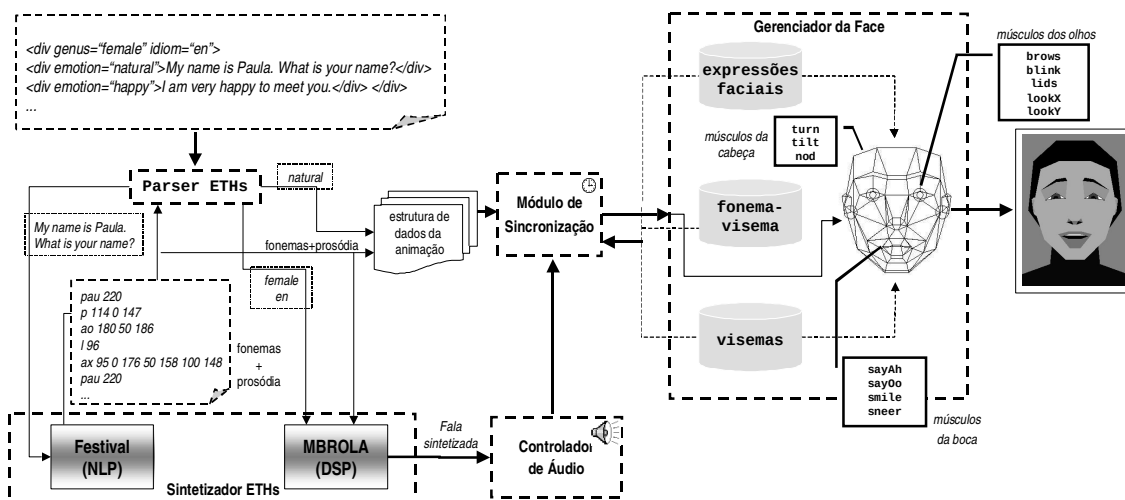


Figura 2 - Visão geral da ferramenta Expressive Talking Heads.

O texto contendo as marcações é passado a um componente denominado *parser* ETHs, que separa o conteúdo da fala propriamente dita das marcações de controle.³ O fluxo de texto (conteúdo da fala) é então enviado ao sintetizador ETHs.

O sintetizador ETHs é composto pelos sintetizadores Festival [WaTC99] e MBROLA [Duto97], que trabalham de maneira integrada. Basicamente, o Festival atua como unidade NLP (*Natural Language Processing*) e o MBROLA como unidade DSP (*Digital Signal Processing*). A saída do Festival, que consiste na descrição fonética da fala (lista de fonemas, cada um contendo sua respectiva duração e entonação), é entrada para o sintetizador MBROLA, que gera como saída o áudio digitalizado (atualmente formato *wav*). A integração desses dois sintetizadores permite que o *parser* ETHs, ou mesmo as aplicações usuárias da ferramenta, interfiram no processo de síntese. Por exemplo, através da manipulação das entonações e durações dos fonemas é possível introduzir um realismo maior no áudio gerado. Essa junção também permite parametrizar o idioma e o gênero da voz, uma vez que o MBROLA oferece uma base de vozes. As informações de emoção extraídas pelo *parser* ETHs são armazenadas em uma estrutura de dados que é completada com a descrição fonética retornada pelo Festival.

O módulo de gerenciamento da face associa a ferramenta ao subsistema ResponsiveFace [Perl97], de onde foi herdada a face no ETHs. Essa face é modelada através de uma malha poligonal tridimensional. Com o agrupamento dos vértices da malha, são formados os músculos faciais, utilizados para efetuar a animação. Valores são aplicados a cada músculo para informá-lo o quanto contrair ou relaxar. O ETHs acrescenta vida ao modelo poligonal do ResponsiveFace através da adição da fala e da definição dos visemas (representação visual dos fonemas – principalmente a posição dos lábios). O módulo de gerenciamento da face define uma base de 16 visemas e 8 expressões faciais de emoção, guardando o valor de contração/relaxamento dos músculos para cada um desses estados. O módulo também define uma tabela que estabelece a relação entre os fonemas e os visemas correspondentes [Luce02].

³ Se o texto não possuir marcações definidas, os parâmetros *default* do sistema serão utilizados (idioma: inglês americano, voz: feminina, emoção: natural). As opções de valores para os parâmetros estão descritas em [Luce02].

Por fim, o ETHs contém um módulo que é encarregado da sincronização fina entre a fala e os movimentos dos músculos faciais (módulo de sincronização ou módulo de *lip-sync*). A operação desse módulo depende da saída gerada pelo módulo de síntese e das informações contidas nas bases do módulo de gerenciamento da face. Em paralelo com a reprodução do arquivo de áudio (sinal da fala), o módulo de sincronização consulta, em intervalos regulares, o instante efetivo em que o áudio se encontra. Pelas durações fonéticas, o controlador de *lip-sync* descobre o fonema sendo pronunciado. Utilizando a duração do fonema corrente juntamente com a do fonema subsequente, o controlador de *lip-sync* determina qual é o percentual de contribuição para cada um deles. O controlador então consulta a tabela de mapeamentos do gerenciador de face para encontrar os visemas correspondentes aos dois fonemas. De posse desses dados, o controlador solicita que o gerenciador da face aplique as transformações nos músculos proporcionalmente à contribuição de cada visema/fonema (interpolação simples com base nas durações). A razão para utilizar difones (par de fonemas subsequentes) advém da importância de capturar os aspectos de co-articulação da fala, pois a posição dos lábios para um mesmo fonema pode mudar em função do seu contexto.

Analogamente, o módulo de sincronização identifica o estado de ânimo corrente e solicita ao módulo de gerenciamento da face que também aplique as transformações necessárias nos músculos da face para refletir a emoção desejada. O módulo de sincronização também possui componentes para controle dos movimentos da cabeça e dos olhos, procurando acrescentar uma aleatoriedade na expressão, a fim de aproximar a saída de um resultado mais natural.

O ETHs encontra-se implementado em Java e foi elaborado para servir de apoio ao desenvolvimento de aplicações que desejem fazer uso de síntese de voz e animação facial. Para facilitar a utilização da ferramenta, os seus serviços são disponibilizados através de uma fachada (padrão de projeto *Facade* [Gamm95]). Dessa forma, a ferramenta esconde do usuário programador a organização interna em módulos descrita nesta seção. Detalhes a respeito dos métodos oferecidos na fachada da ferramenta podem ser encontrados em [LuFV03].

4. Integração do Expressive Talking Heads com o Formatador HyperProp

A introdução de uma ferramenta sintetizadora de voz em um sistema hipermídia traz benefícios para os documentos que venham a ser desenvolvidos. Por exemplo, como já mencionado, usuários com deficiência visual podem ter uma alternativa para navegação em certos conteúdos.

Acrescentar à ferramenta uma face capaz de enunciar falas com recursos de expressão e emoção aumenta ainda mais a potencialidade do sistema. É possível que documentos elaborados para crianças tornem-se mais atrativos fazendo uso de personagens virtuais/caricaturais, ao invés de utilizarem vídeos convencionais. A substituição de vídeos por faces como a do Expressive Talking Heads também pode gerar economia de banda passante junto ao sistema de comunicação.

Além das vantagens comentadas no parágrafo anterior, a inserção de uma ferramenta *talking head* em um sistema hipermídia permite criar relações entre a fala pronunciada pela face e outros objetos de mídia, expandindo as aplicações que podem fazer uso do sistema. Como exemplo, um jornal virtual pode ser construído utilizando a

face como apresentadora dos noticiários, estando a fala da face relacionada aos textos, imagens, vídeos e outros tipos de mídia que complementem as notícias.

4.1. Modificações na Ferramenta ETHs

Como, a princípio, o Expressive Talking Heads não permitia que uma aplicação usuária fosse sinalizada quando da ocorrência de trechos internos da fala (eventos de apresentação), característica desejável para integração da ferramenta a um formatador hipermídia, foram feitas algumas modificações para tornar esse recurso disponível.

Em primeiro lugar, o *parser* ETHs (Seção 3) foi alterado para tratar um novo atributo do elemento *div* (linguagem de marcação de entrada): o atributo *anchor*. Essa marcação tem como função especificar trechos do conteúdo que definirão eventos de apresentação (exibição do trecho) durante a fala. A Figura 3 exemplifica o uso da marcação para a especificação de uma âncora no texto.

```
...
Texto que se encontra fora da âncora.
<div anchor="idAnchor">Este texto é conteúdo de uma âncora.</div>
...
```

Figura 3 – Especificação de âncoras para o *parser* ETHs.

Além da modificação da linguagem, a estrutura de dados da animação foi alterada para contemplar essa nova informação, passando a armazenar, além da lista de fonemas e das informações de emoção, uma lista das âncoras, com cada âncora identificando os fonemas de início e fim correspondentes.

Para construção dessa estrutura, o *parser* ETHs envia ao Festival o conteúdo de cada elemento da linguagem de marcação separadamente (Seção 3). No exemplo da Figura 3, primeiro o *parser* envia o texto “*Texto que se encontra fora da âncora*” e recebe como retorno a estrutura fonética correspondente. Isso permite ao *parser* identificar o fonema que corresponde ao início desse trecho da fala e o fonema que representa o seu final. Em seguida, o *parser* envia ao Festival o texto “*Este texto é conteúdo de uma âncora*”, recebendo como retorno a estrutura fonética da nova sentença. O *parser* continua o processo até concluir o tratamento do texto de entrada. Somente então, o *parser* concatena as várias estruturas fonéticas retornadas e envia uma única requisição para que o MBROLA gere a voz sintetizada. Essa estratégia evita segmentar o som e gerar falhas na enunciação da fala.

Com a extensão da linguagem e do módulo de síntese, o passo seguinte foi permitir que um componente da aplicação usuária pudesse se registrar como observador da fachada ETHs (classe *ETHsFacade*), e ser sinalizado toda vez que o texto correspondendo à âncora começasse a (ou terminasse de) ser apresentado pela ferramenta. Para viabilizar esse requisito, foi criada uma interface observadora *ETHsAnchorListener*, que deve ser implementada por aquelas classes de objetos que tiverem interesse em ser notificadas da ocorrência desses eventos. A classe *ETHsFacade* foi então estendida com dois novos métodos: um (*addAnchorListener*) que permite que um objeto registre-se como observador de âncoras, e outro (*removeAnchorListener*) que desfaz um registro previamente solicitado.

Por fim, o módulo de sincronização foi ligeiramente alterado para, durante o controle dos movimentos faciais, também monitorar o início e fim das âncoras, e gerar o pedido de notificação para os objetos registrados como observadores junto à ferramenta.

4.2. Adaptador ETHs

Conforme comentado na Seção 2, para integrar a ferramenta ETHs com o formatador HyperProp, foi desenvolvido um componente adaptador (classe *HF_ETHsAdapter*), responsável por realizar o mapeamento entre a API fornecida pela fachada ETHs e a API especificada pelo formatador HyperProp (Figura 1). O adaptador ETHs foi definido como uma subclasse da classe abstrata de ferramentas de exibição especificada pelo formatador HyperProp [RoRS01] e também como uma aplicação usuária da ferramenta ETHs. Para isso, o adaptador contém uma instância da fachada ETHs. Para ser capaz de atuar como um observador das âncoras sinalizadas pela fachada ETHs (início/fim das âncoras contidas na fala e início/fim da fala como um todo), o adaptador ETHs implementa a interface *ETHsAnchorListener*. A Figura 4 ilustra o diagrama de classes da modelagem. Vários métodos e atributos, assim como os argumentos de cada método, foram ocultados para não sobrecarregar a figura.

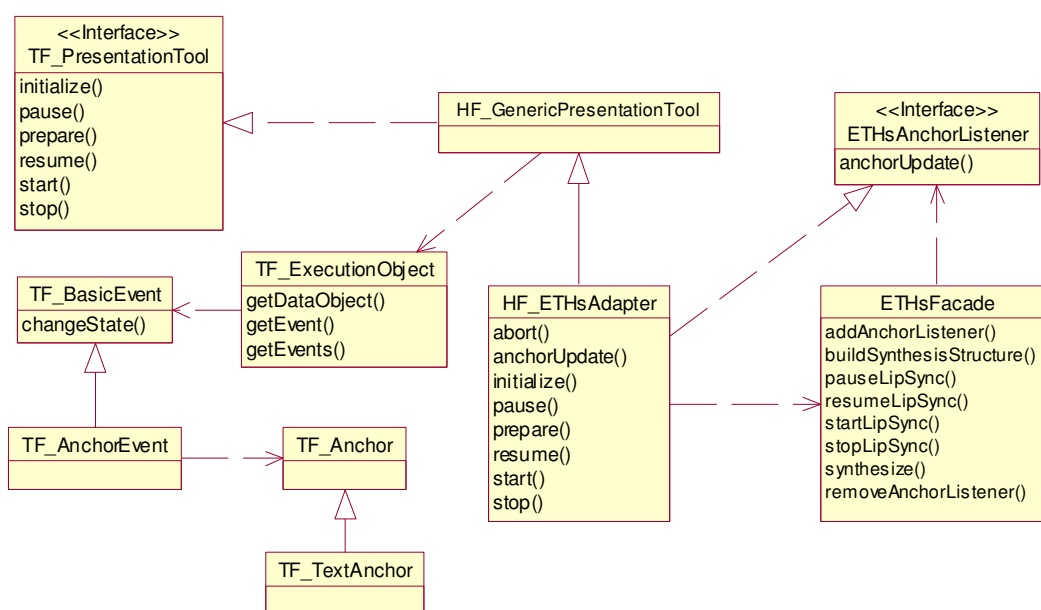


Figura 4 – Diagrama de classes da implementação do adaptador ETHs.

A ferramenta de exibição no formatador HyperProp, quando iniciada (método *initialize*), recebe o objeto de mídia (classe *HF_ExecutionObject*) que deve ter a sua exibição controlada. Além do conteúdo em uma dada mídia (*objeto de dados*), o objeto contém uma lista dos eventos que devem ter as suas máquinas de estados controladas pelo adaptador. Esses eventos possuem âncoras associadas que especificam regiões do conteúdo. Ao contrário do conteúdo ETHs que possui as âncoras embutidas, o modelo de execução do formatador HyperProp permite que as âncoras sejam definidas externamente ao conteúdo dos objetos. O intuito dessa separação é fazer com que os relacionamentos entre objetos possam ser especificados utilizando quaisquer subconjuntos de seus conteúdos, e possam ser descritos fora desses conteúdos.

Ao receber a especificação das âncoras dessa forma, cabe ao adaptador ETHs embuti-las dinamicamente no conteúdo a ser passado para a fachada ETHs. A Figura 5 ilustra esse procedimento, que é desempenhado quando o adaptador ETHs recebe a solicitação para preparar a exibição (método *prepare*).

Com o texto produzido, o adaptador requisita à fachada ETHs que seja realizado o primeiro passo da síntese, ainda na fase de preparação (chamada ao método *buildSynthesisStructure* na fachada ETHs). Internamente à ferramenta ETHs, essa solicitação gera uma chamada ao *parser* ETHs e, posteriormente, uma solicitação ao sintetizador Festival (Figura 2).

Tanto o Festival como o MBROLA podem executar em uma máquina servidora, não necessitando que estejam instalados na máquina cliente do usuário que assiste ao documento (máquina em que o formatador, o adaptador e a ferramenta ETHs executam). Uma solicitação utilizando uma conexão TCP é feita pelo *parser* ao Festival-MBROLA, requisitando que o processamento da linguagem natural seja realizado. Ao final do processamento, o adaptador ETHs recebe a estrutura de dados contendo a lista de fonemas, duração de cada fonema, entonação, emoção, assim como o mapeamento entre as âncoras e os fonemas da fala (quais fonemas correspondem ao início e ao fim de cada uma das âncoras). O adaptador ETHs solicita então que o segundo passo da síntese seja executado, a fim de gerar o arquivo de áudio contendo a fala (chamada ao método *synthesize* oferecido pela fachada ETHs⁴). O adaptador ETHs recebe como retorno da função a URL do arquivo contendo o áudio digitalizado da fala, que é criado na máquina onde executam os sintetizadores Festival-MBROLA. Na implementação atual, essa máquina possui um servidor Web instalado, permitindo que o arquivo do áudio sintetizado seja buscado utilizando o protocolo HTTP. Uma possível extensão é a utilização de um protocolo de *streaming*, como o RTP, para busca desse conteúdo. Ao final do processo de síntese, o adaptador ETHs solicita a preparação do áudio sintetizado, ou seja, requisita que as primeiras amostras do conteúdo sejam trazidas para o *buffer* do controlador de áudio. Com isso, a fase de preparação da exibição é encerrada.

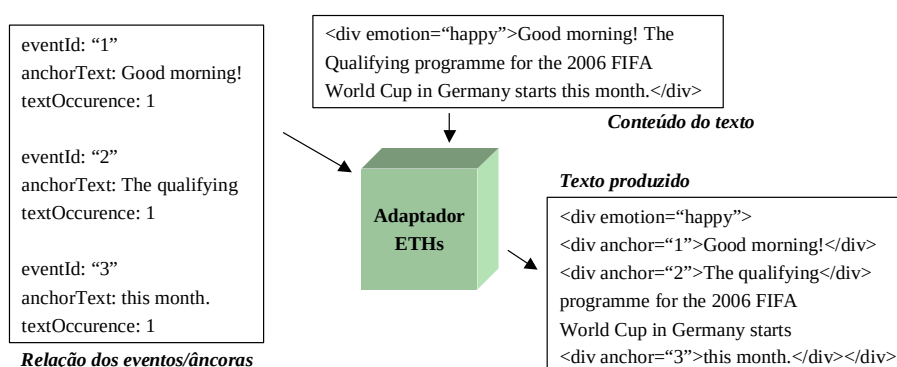


Figura 5 – Inserção dinâmica das âncoras textuais na fala a ser sintetizada.

⁴ Na versão anterior da fachada ETHs [LuFV03], os métodos *buildSynthesisStructure* e *synthesize* eram denominados *runNLP* e *runDSP*, respectivamente.

Note que todo o processo, quando as ferramentas Festival e MBROLA não executam na máquina cliente, pode causar um grande atraso, resultante tanto da execução da síntese na estação servidora, quanto da comunicação na rede. Esse atraso pode impedir a sincronização correta da fala sintetizada com outros objetos do documento hipermídia. No entanto, o formatador HyperProp possui um módulo que procura antecipar-se às ações de apresentação [Rodr03]. Esse módulo constrói um plano de pré-busca, que faz com que o formatador solicite a preparação da exibição (chamada ao método *prepare* do adaptador) antecipadamente, a fim de que o início da apresentação aconteça quase que imediatamente após a chamada ao método *start* do adaptador ETHs. Também foi implementada uma versão simplificada de adaptador ETHs que trabalha com áudios e estruturas fonéticas pré-sintetizadas, eliminando todo o processo de síntese no momento da execução do documento.

Uma vez preparada a exibição de um objeto de áudio sintetizado, o adaptador ETHs aguarda pela solicitação do início da apresentação propriamente dita (chamada ao seu método *start*), quando então repassa o pedido para a fachada ETHs (chamada ao método *startLipSync*), que por sua vez dispara o módulo de sincronização. O adaptador ETHs passa a observar as ocorrências de início e fim de apresentação das âncoras na fala e, toda vez que é notificado (chamada ao seu método *anchorUpdate*), ele realiza a transição na máquina de estados do evento correspondente [RoRS01]. O início e o fim da exibição da fala como um todo também são monitorados pelo adaptador ETHs e sinalizados ao formatador (existe no formatador uma máquina de estados para controlar a exibição dos objetos como um todo, uma vez que essa exibição é também um evento).

5. Exemplo de Documento Utilizando a Plataforma HyperProp-ETHs

Na Seção 2, comentou-se que o formatador HyperProp pode receber a especificação do documento em linguagens de autoria mais próximas do nível de abstração do autor. A Figura 6 oferece um exemplo de documento hipermídia especificado através da linguagem NCL [MuSS03].

O documento retrata um trecho de um noticiário esportivo. A linguagem NCL separa os objetos de mídia das características de apresentação desses objetos. Sendo assim, no cabeçalho do documento (linhas 7-20), é definido o *layout* espacial da apresentação (análogo ao *layout* da linguagem SMIL) e uma base de descritores. No exemplo, existe um descritor (*faceDescriptor*) que, além de especificar a região espacial onde o objeto será exibido, especifica também o uso da ferramenta de exibição “HF_ETHsAdapter”, através do atributo *player*.

No corpo do documento (linhas 21-56) existem quatro objetos de mídia especificados: um texto contendo a notícia (*newsText* – linhas 23-29), dois fragmentos desse texto que são utilizados como legenda da notícia (*subt01* e *subt02* – linhas 30-33) e um vídeo (*goal* – linhas 34-35). Cada um desses objetos especifica uma URI para o seu conteúdo e um determinado descritor para ser utilizado na exibição de seu conteúdo. Para o objeto *newsText* é definido o descritor *faceDescriptor*, fazendo com que o texto HTML seja associado ao adaptador ETHs. Além do descritor e da URI do conteúdo, o objeto *newsText* também especifica quatro âncoras, cada uma identificando um fragmento de texto em seu conteúdo (linhas 25-28).

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <ncl id="sportNews"
03 xmlns="http://www.telemidia.puc-rio.br/specs/xml/NCL"
04 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
05 xsi:schemaLocation="http://www.telemidia.puc-rio.br/specs/xml/NCL
06 http://www.telemidia.puc-rio.br/specs/xml/NCL.xsd">
07 <head>
08 <layout>
09 <topLayout id="hpropWindow" title="HyperProp News" width="800" height="600">
10 <region id="faceRegion" width="250" height="400"/>
11 <region id="subtRegion" top="400" width="800" height="200"/>
12 <region id="videoRegion" top="80" left="365" width="320" height="240"/>
13 </topLayout>
14 </layout>
15 <descriptorBase>
16 <descriptor id="faceDescriptor" region="faceRegion" player="HF_ETHsAdapter"/>
17 <descriptor id="subtitleDescriptor" region="subtRegion"/>
18 <descriptor id="videoDescriptor" region="videoRegion"/>
19 </descriptorBase>
20 </head>
21 <body>
22 <port id="entryPoint" component="newsText"/>
23 <text id="newsText" descriptor="faceDescriptor"
24 src="file:../mediaContent/examples/semish/sports/news.html">
25 <area id="anchor01-beg" text="The 2001" occurrence="1"/>
26 <area id="anchor01-end" text="decision game." occurrence="1"/>
27 <area id="anchor02-beg" text="The decisive" occurrence="1"/>
28 <area id="anchor02-end" text="red-black team." occurrence="1"/>
29 </text>
30 <text id="subt01" descriptor="subtitleDescriptor"
31 src="file:../mediaContent/examples/semish/sports/newsSubt01.html"/>
32 <text id="subt02" descriptor="subtitleDescriptor"
33 src="file:../mediaContent/examples/semish/sports/newsSubt02.html"/>
34 <video id="goal" descriptor="videoDescriptor"
35 src="file:../mediaContent/examples/semish/sports/goal.mpg"/>
36 <linkBase>
37 <link id="link1" xconnector="file:../hpropFiles/data/connectors/starts.xml">
38 <bind component="newsText" port="anchor01-beg" role="on_x_presentation_begin"/>
39 <bind component="subt01" role="start_y"/>
40 </link>
41 <link id="link2" xconnector="file:../hpropFiles/data/connectors/finishes.xml">
42 <bind component="newsText" port="anchor01-end" role="on_x_presentation_end"/>
43 <bind component="subt01" role="stop_y"/>
44 </link>
45 <link id="link3" xconnector="file:../hpropFiles/data/connectors/startsMP.xml">
46 <bind component="newsText" port="anchor02-beg" role="on_x_presentation_begin"/>
47 <bind component="subt02" role="start_y"/>
48 <bind component="goal" role="start_y"/>
49 </link>
50 <link id="link4" xconnector="file:../hpropFiles/data/connectors/finishesMP.xml">
51 <bind component="newsText" port="anchor02-end" role="on_x_presentation_end"/>
52 <bind component="subt02" role="stop_y"/>
53 <bind component="goal" role="stop_y"/>
54 </link>
55 </linkBase>
56 </body>
57 </ncl>

```

Figura 6 – Exemplo de documento NCL, um dos formatos de especificação de apresentações multimídia no formatador HyperProp

Como pode ser observado, além desses objetos, o corpo do documento possui também uma base de elos (linhas 36-55). Na linguagem NCL, os elos são utilizados para definir os relacionamentos de sincronização entre os nós (ou possivelmente as âncoras dos nós). No exemplo, existem quatro elos especificados. O primeiro (linhas 37-40) deles sincroniza o início da exibição da primeira âncora do texto com o início da exibição da legenda. O segundo elo (linhas 41-44) especifica que a legenda deve deixar

de ser exibida ao final da exibição da segunda âncora do objeto *newsText*. O elo *link3* (linhas 45-49) é um elo multiponto, que inicia a exibição da segunda legenda e a apresentação do vídeo quando a terceira âncora da notícia tem sua exibição iniciada. Por fim, o quarto elo (linhas 50-54) especifica que o final da exibição do fragmento de texto correspondendo à quarta âncora da notícia deve causar o encerramento da exibição tanto da segunda legenda como do vídeo. Cabe comentar que a especificação da relação é feita em uma entidade à parte, denominada conector hipermídia [MuSS03].

Quando o documento da Figura 6 é entregue ao formatador HyperProp, o conversor NCL é acionado. Esse conversor percorre o documento sequencialmente, gerando uma estrutura de dados que obedece ao modelo de execução do formatador [Rodr03]. Sempre que o conversor identifica um objeto de mídia (por exemplo *newsText*), ele procura pelo seu descritor para obter as informações de como exibi-lo. O formatador também utiliza a URI para determinar o local onde o conteúdo do objeto se encontra. Se o objeto possuir âncoras, o conversor constrói os eventos correspondentes. Mesmo que não haja âncoras especificadas no objeto de mídia, o conversor cria pelo menos o evento que representa a apresentação do conteúdo como um todo. O resultado desse processamento é a criação dos objetos de execução no modelo de execução do formatador (classe *HF_ExecutionObject* na Figura 4). Quando o conversor trata um objeto composto (nó de composição como o corpo do documento), uma tarefa importante é a análise das bases de elos, pois os elos são as entidades que permitem descobrir os relacionamentos existentes entre os eventos.

Com a estrutura de execução construída, é iniciado o controle da apresentação. A porta da composição mais externa do documento (o elemento *body*) é utilizada para identificar o ponto de entrada da apresentação, isto é, o primeiro objeto a ser exibido (linha 22). A partir desse objeto, no exemplo *newsText*, e dos relacionamentos de sincronização, o formatador constrói o plano da exibição [Rodr03]. Antes de começar a apresentação propriamente dita do documento, o formatador inicia, se habilitado, o seu módulo de pré-busca (Seção 4).

Seja por intermédio do módulo de pré-busca, seja por intermédio do controlador principal da apresentação, assumindo o documento de exemplo da Figura 6, haverá um momento em que uma ferramenta de exibição precisará ser instanciada para controlar a exibição do objeto de mídia *newsText* (na realidade, o objeto de execução que contém esse objeto de mídia). Uma vez que o objeto de mídia utiliza um descritor que especifica *HF_ETHsAdapter* como ferramenta de exibição, o formatador procurará pela classe homônima em um diretório pré-configurado de execução e instanciará um objeto dessa classe. Uma vez criada a instância da ferramenta, no caso uma instância do adaptador ETHs, o formatador lhe entrega o objeto de execução com suas informações (URI do conteúdo, descritor do objeto, região espacial de exibição, eventos a serem monitorados etc.).

A Figura 7 ilustra a execução do documento de exemplo no formatador HyperProp. A figura retrata o momento em que a segunda parte da notícia está sendo dada, paralelamente às exibições da segunda legenda e do vídeo. O grafo exibido na lateral direita oferece uma visão da estrutura do documento sendo apresentado. O objetivo dessa visão é guiar o usuário, principalmente, em documentos que possuem muitos nós e elos.



Figura 7 – Execução de um documento no formatador HyperProp utilizando a ferramenta ETHs sincronizada com outros objetos.

Apesar do texto sendo noticiado (falado pela apresentadora virtual) encontrar-se em inglês, as legendas na figura são exibidas em português porque foram predefinidas pelo autor em nós de mídia separados (*subt01* e *subt02*), que como pode ser observado, apontam para URIs diferentes daquela apontada pelo texto da notícia (*newsText*).

6. Trabalhos Relacionados

Originalmente proposto pela Netscape [Nets98], o suporte à tecnologia de *plug-ins* é comum em navegadores (*browsers*) WWW. Uma API permite que aplicações externas sejam incorporadas ao ambiente de navegação da Web, principalmente para que formatos não reconhecidos pelo navegador sejam tratados em um ambiente integrado. Porém, na API para integração de *plug-ins* WWW não é oferecido suporte para definição de relacionamentos entre objetos apresentados em diferentes *plug-ins*, e nem mesmo entre *plug-ins* e páginas HTML. Na realidade, alguns mecanismos, como LiveConnect e ActiveX, permitem a integração entre páginas HTML e *plug-ins*, ou até mesmo com outras tecnologias como *scripts* e *applets*. No entanto, esses mecanismos dependem da implementação do navegador e são baseados em códigos de programação não triviais, que devem ser embutidos nos próprios objetos de mídia. Uma outra limitação dessa API é o fato do *plug-in* ter seu ciclo de vida amarrado ao ciclo de vida da página HTML que o contém.

A referência [BoSc99] propõe extensões à API de *plug-ins* com o objetivo de permitir a criação de elos entre objetos e sub-regiões desses objetos, independente da ferramenta (*plug-in*) que os exibe. No entanto, na extensão descrita, os elos que o sistema possibilita criar são sempre elos de interatividade, não havendo suporte para a definição de relacionamentos de sincronização temporal, como permite a API do formatador HyperProp.

Na literatura, é possível encontrar diversos trabalhos que tratam da questão da animação facial, sincronização labial, tratamento de emoções etc. VideoRewrite [BrCS97], Facade [Dipa01] e MikeTalk [EzPo99] são alguns exemplos. Apesar dessas

ferramentas produzirem um resultado nos movimentos labiais bastante satisfatório, as ferramentas VideoRewrite e MikeTalk não são interativas, isto é, elas exigem um tempo elevado de preparação da animação e geram um vídeo pré-compilado contendo o resultado, o que dificulta a edição dos dados para ajuste das âncoras, por exemplo. Apesar de bastante interativo, o Facade não menciona a existência de uma API aberta para a integração com outras aplicações. A referência [GaMa01] descreve a implementação de uma ferramenta de suporte a aplicações *talking head* na *web*, utilizando MPEG-4 como base para a animação das faces. Apesar de oferecer o suporte, o artigo apenas comenta algumas das possíveis aplicações *web* que poderiam fazer uso das animações faciais, deixando as suas implementações como trabalho futuro.

7. Conclusões

Este artigo descreve o trabalho de integração de uma ferramenta *talking head*, o ETHs, com o formatador de documentos hipermídia do sistema HyperProp. Além de detalhar os aspectos da integração, o artigo ilustra o uso do sistema integrado com um exemplo de documento especificado na linguagem NCL.

Encontra-se em andamento o trabalho de migração do formatador HyperProp para que execute em *set-top boxes* de TV interativa. Nesse sentido, a ferramenta ETHs pode ser extremamente útil para criação de documentos que visam entretenimento, difusão cultural, comércio eletrônico e ensino, via o futuro sistema brasileiro de TV digital (SBTVD).

Infelizmente, não existe ainda disponível para o sintetizador Festival uma base de fonemas para o idioma português, mas é desejável que a ferramenta ETHs seja estendida para permitir que também esse idioma possa ser utilizado pelas aplicações. Um outro trabalho futuro relacionado ao Expressive Talking Heads é a implementação de um novo *parser* que interprete uma descrição utilizando a linguagem SSML (*Speech Synthesis Markup Language*), cuja recomendação encontra-se em fase final de elaboração [W3C03].

Por fim, encontra-se também em andamento a exploração da ferramenta para adaptação de apresentações no formatador HyperProp. O ETHs pode ser utilizado para atender restrições dos usuários, por exemplo substituindo texto por voz, ou satisfazendo requisitos de banda passante, substituindo vídeo pela animação da face. Um outro ponto que vem sendo abordado é a implementação de ajustes na duração da fala a fim de respeitar restrições temporais impostas pelo autor da apresentação [Rodr03].

Referências

- [BoSc99] Bouvin N.O., Schade R. "Integrating Temporal Media and Open Hypermedia on the World Wide Web". *Eighth Int. World Wide Web Conference*, 1999, pp. 375-387.
- [BrCS97] Bregler C., Covell M., Slaney M. "Video Rewrite: Driving visual speech with audio". *SIGGRAPH'97*, Los Angeles, EUA, 1997.
- [Dipa01] DiPaola S. "Facade – The Stanford Facial Animation System". Technical Report, Stanford University, EUA, 2001. Disponível em <http://dipaola.org/stanford/facade/>.

- [Duto97] Dutoit T. et al. “A Short Introduction to Text-to-Speech Synthesis”, *Relatório Técnico*, TTS Research Team, TCTS Lab, Faculté Polytechnique de Mons, Bélgica, 1997.
- [EzPo99] Ezzat T., Poggio T. “Visual Speech Synthesis by Morphing Visemes”, *Technical Report 1658*, Center for Biological & Computational Learning and the Artificial Intelligence Lab, Massachusetts Institute of Technology, EUA, 1999.
- [GaMa01] Gachery S. Magnenat-Thalmann N. “Designing MPEG-4 Facial Animation Tables for Web Applications”, *Multimedia Modeling Conference*, Amsterdam, Holanda, 2001.
- [Gamm95] Gamma E. et al. “Design Patterns: Elements of Reusable Object-Oriented Software”. Addison Wesley, 1995.
- [Luce02] Lucena P.S. “Expressive Talking Heads: Um Estudo de Fala e Expressão Facial em Personagens Virtuais”. *Dissertação de Mestrado*, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2002.
- [LuFV03] Lucena-Rodrigues P.S., Feijó B., Velho L. “Expressive Talking Heads: uma Ferramenta com Fala e Expressão Facial Sincronizadas para o Desenvolvimento de Aplicações Interativas”. *IX Simpósio Brasileiro em Sistemas Multimídia e Web*, Salvador, Brasil, novembro de 2003.
- [MuSS03] Muchaluat-Saade D.C., Silva H.V.O., Soares L.F.G. “Linguagem NCL versão 2.0 para Autoria Declarativa de Documentos Hipermídia”. *IX Simpósio Brasileiro em Sistemas Multimídia e Web*, Salvador, Brasil, novembro de 2003.
- [Nets98] Netscape Communications Corporation. “Plug-in Guide”. 1998. Disponível em <http://developer.netscape.com/docs/manuals/communicator/plugin/index.htm>.
- [Perl97] Perlin K. “Responsive Face”, *Media Research Lab*, New York University, EUA, 1997. Disponível em <http://mrl.nyu.edu/~perlin/demox/Face.html>.
- [Rodr03] Rodrigues R.F. “Formatação e Controle de Apresentações Hipermídia com Mecanismos de Adaptação Temporal”, *Tese de Doutorado*, Departamento de Informática – PUC-Rio, Rio de Janeiro, Brasil, 2003.
- [RoRS01] Rodrigues R.F., Rodrigues L.M., Soares L.F.G. “A Framework for Event-Driven Hypermedia Presentation Systems”, *Multimedia Modeling Conference - MMM’2001*, Amsterdam, Holanda, novembro de 2001. pp. 169-185.
- [W3C01] World-Wide Web Consortium. “Synchronized Multimedia Integration Language (SMIL 2.0) Specification”. *W3C Recommendation*, agosto de 2001.
- [W3C03] World-Wide Web Consortium. “Speech Synthesis Markup Language Version 1.0”. *W3C Candidate Recommendation*, dezembro de 2003.
- [WaTC99] Watt A., Taylor P., Caley R. “The Festival Speech Synthesis System: System Documentation”, *Relatório Técnico*, University of Edinburgh, Escócia, 1999.