

## 6

# Utilização da Ferramenta de Ajuste em Sistemas Hipermídia

Este capítulo primeiramente descreve as necessidades de adaptabilidade de sistemas hipermídia, para depois apresentar como a ferramenta de ajuste pode ser utilizada nesses sistemas e, enfim, como foi realizada a integração da ferramenta com sistema HyperProp.

### 6.1.

#### Necessidades de Adaptabilidade de Sistemas Hipermídia

Segundo Rodrigues (Rodrigues, 2003), mecanismos de adaptação em sistemas hipermídia são transformações necessárias para tornar os hiperdocumentos consistentes (especialmente e temporalmente) ou adequá-los a um contexto de exibição (plataformas, usuários etc.). Adaptações podem ser necessárias por causa da flexibilidade na descrição dos documentos; pela própria navegação do usuário, gerando eventos que não puderam ser previstos antes de iniciar a exibição do documento; por atrasos, também impossíveis de serem previstos no sistema operacional e na rede; ou por quaisquer outras informações provenientes do contexto de exibição (recursos de hardware disponíveis na máquina do cliente, preferências e perfil do usuário etc.).

Um sistema hipermídia pode ser subdividido em diferentes ambientes e todos eles podem ou não suportar mecanismos de adaptação. O ambiente de autoria é onde os autores criam os hiperdocumentos. A autoria é realizada utilizando algum modo de especificação, como, por exemplo, uma linguagem declarativa, para descrever os objetos de mídia do documento, suas propriedades e relacionamentos. Além disso, o autor deve especificar características de exibição para cada objeto de mídia, entre as quais, pode-se destacar, sua duração esperada de apresentação. Tais características devem poder ser especificadas de modo

flexível, utilizando, por exemplo, diferentes alternativas de exibição, como explicado mais adiante.

Adaptações, como ajuste elástico, podem ser realizadas nesse ambiente, normalmente em tempo de compilação, com objetivo de manter a consistência temporal de um documento, desde sua especificação, e evitar que seja feito em tempo de exibição no cliente. Por exemplo, em um sistema de TV pode-se desejar adaptar o documento no transmissor para evitar processá-lo em todos os clientes.

O ambiente de armazenamento oferece mecanismos para gravar, recuperar e adaptar os documentos. É possível separar o armazenamento em duas classes, que apresentam requisitos distintos e podem operar de maneira independente: o armazenamento da descrição dos documentos hipermídia e o armazenamento do conteúdo dos objetos de mídia (arquivos de texto, imagem, áudio, vídeo etc.). Dentre as funções do ambiente de armazenamento, vale detalhar aqui a possibilidade de adaptação. O principal objetivo da adaptação localizada no ambiente de armazenamento é que o mesmo forneça conteúdos com configurações que melhor atendam ao contexto em que eles venham a ser exibidos, buscando ao mesmo tempo otimizar a utilização dos recursos que estejam situados no servidor (ambiente de armazenamento), no caminho até o cliente hipermídia (provedor de serviços de comunicação) e no próprio cliente (ambiente de execução).

O último ambiente é o de execução, também conhecido como ambiente de apresentação. Esse ambiente normalmente encontra-se situado junto ao cliente hipermídia e deve conter os mecanismos para interagir com os usuários e controlar a exibição das mídias de tal forma que as intenções especificadas pelos autores sejam respeitadas. Entre os elementos que compõem o ambiente de execução podem ser destacados: as ferramentas de exibição e o formatador hipermídia.

As ferramentas de exibição correspondem aos módulos, de software ou hardware, responsáveis por lidar com os dispositivos de entrada e saída e controlar a exibição do conteúdo dos objetos, cuidando, por exemplo, de questões de descompressão, tratamento de interações do usuário, controle da mudança de velocidade de exibição para o caso de objetos de mídia contínua etc.

Formatador hipermídia é o nome dado ao elemento que reúne as funções para o controle da apresentação previamente especificada pelo autor. A Figura 68

ilustra a comunicação de um formatador hipermedia com uma ferramenta de exibição através de uma interface padronizada de comunicação entre eles (API FF), conforme proposto em Rodrigues (Rodrigues, 2003). A ferramenta de exibição recebe eventos de apresentação do formatador (como *play*, *stop* e *resume*) e a mídia a ser exibida, que pode vir como um arquivo ou fluxo de dados. A ferramenta de exibição é responsável por exibir a apresentação hipermedia para o usuário e repassar para o formatador eventos resultantes da interação do usuário com a apresentação e eventos resultantes da apresentação de um determinado trecho do conteúdo. Os exibidores de conteúdo que não obedecem à API FF podem interagir com o formatador através de um adaptador, que mapeia as funções da API FF para as funções do exibidor de conteúdo.

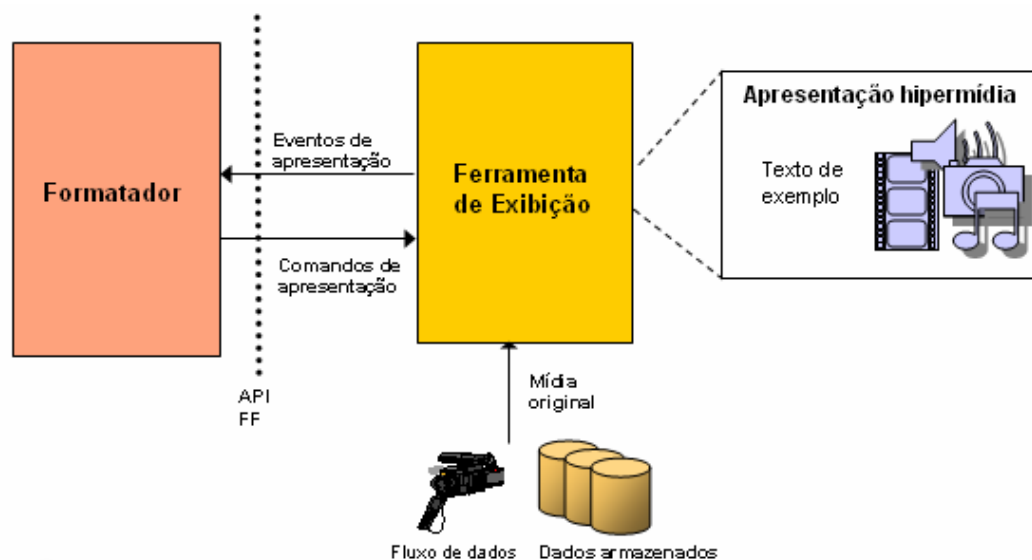


Figura 68 - Integração de um formatador hipermedia com uma ferramenta de exibição.

Para orquestrar a apresentação, o formatador precisa construir um *plano de execução* que contém as informações de sincronismo entre todas as entidades do documento, incluindo aquelas relações que não podem ser previstas, como interação com usuário (Bachelet et al., 2004). A fim de manter consistente uma apresentação, muitas vezes o formatador precisará ajustar o hiperdocumento (estrutura ou a forma de apresentação), ou requisitar que as ferramentas de exibição façam o ajuste do conteúdo. A forma mais comum de adaptar objetos de mídia é possibilitar a escolha de partes do documento baseando-se em diferentes alternativas de conteúdo. Com isso, os objetos do documento passam a poder ter não apenas os seus conteúdos, mas também os seus aspectos de exibição adaptáveis.

Uma adaptação pode ser realizada em tempo de compilação, durante a construção do plano de execução. Nessa situação, as técnicas de adaptação podem considerar as preferências do usuário e as características da plataforma de exibição. A vantagem dessas técnicas é a diminuição do número de adaptações uma vez iniciada uma apresentação.

Infelizmente, as adaptações em tempo de compilação não são suficientes para garantir uma boa apresentação. Existem situações em que é necessário utilizar adaptações em tempo de exibição, que modificam o já criado plano de execução, em função de algum evento imprevisível. As técnicas de adaptação em tempo de exibição podem considerar as preferências do usuário e as características da plataforma de exibição (como as técnicas de tempo de compilação) e outras características mais dinâmicas, como interação do usuário.

## **6.2. Utilização da Ferramenta de Ajuste em Sistema Hipermídia**

O ajuste elástico atua na apresentação dos objetos, aumentando ou diminuindo suas durações em relação aos valores originais (ou ideais), com o objetivo de garantir a consistência temporal do documento, ou seja, respeitar as restrições temporais entre objetos de mídia impostas pelo autor<sup>13</sup>. Quando o formato da mídia a ser apresentado é textual, ou corresponde a imagens estáticas, o ajuste da duração de exibição é trivial. No entanto, se um objeto de mídia contínua precisa ter sua duração corrigida, o ideal é que esse ajuste ocorra sem que seja prejudicada a qualidade da apresentação.

A Figura 69 ilustra a inclusão da ferramenta de ajuste, definida no Capítulo 4, dentro de uma ferramenta de exibição de um sistema hipermídia. Nesse cenário, a ferramenta de ajuste é formada por três aplicações diferentes: o adaptador, a ferramenta de ajuste elástico e o exibidor de conteúdo. O formatador interage com o adaptador de modo análogo ao previamente ilustrado na Figura 68. O adaptador é uma aplicação que faz uso da ferramenta de ajuste e do exibidor de conteúdo para fornecer uma interface de uma ferramenta de exibição com capacidade de realizar ajuste elástico. A ferramenta de ajuste não foi modificada e

---

<sup>13</sup> É importante observar que os conceitos em que se baseia o ajuste elástico temporal podem ser transpostos para o arranjo espacial dos objetos, resultando em um ajuste elástico das dimensões dos objetos cujos conteúdos são mídias visíveis.

é acessada através das APIs definidas na Seção 4.1 e o exibidor de conteúdo não depende da ferramenta de ajuste elástico nem do adaptador.

O adaptador recebe do formatador comandos de apresentação, que incluem a mídia a ser exibida, executa algumas operações (como cálculo do fator de ajuste necessário) e solicita serviços à ferramenta de ajuste elástico (como também pode se comunicar diretamente com o exibidor de conteúdo). A ferramenta de ajuste, por sua vez, passa para o exibidor de conteúdo comandos de apresentação (como *start* e *stop*) e o conteúdo processado da mídia a ser exibida. Enfim, o exibidor de conteúdo apresenta a mídia e retorna eventos de apresentação para a ferramenta de ajuste (como, por exemplo, a informação que a apresentação terminou) e para o adaptador, que os repassa para o formatador. Por fim, a ferramenta de ajuste pode levantar aqueles eventos de apresentação definidos nas APIs de ajuste (Seção 4.1).

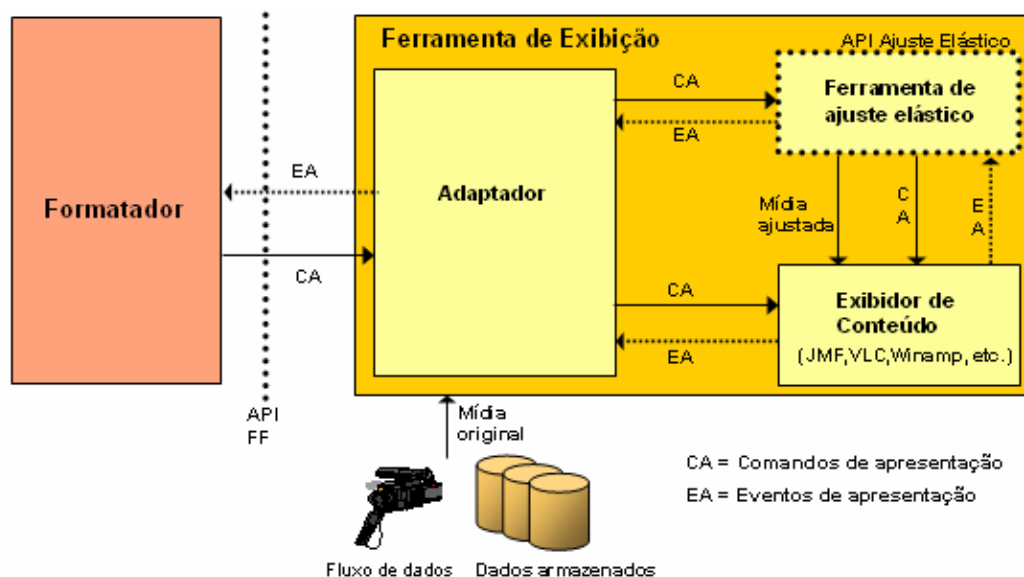


Figura 69 - Integração de uma ferramenta de exibição com ajuste elástico com um formatador hipermídia.

Três tipos de adaptadores de exibidor de conteúdo foram definidos. O *FlexMediaFileAdapter* utiliza a ferramenta de ajuste para gerar um arquivo temporário com a mídia processada e o envia para um exibidor de conteúdo. Como consequência, esse adaptador é mais adequado para executar ajustes em tempo de compilação.

Para permitir realizar ajuste em tempo de exibição, dois outros adaptadores foram implementados: *FlexMediaRTPAdapter* e *FlexMediaDataSourceAdapter*. Os dois adaptadores lêem os dados de mídia de um fluxo de dados, realizam o

ajuste elástico e o enviam como um novo fluxo ao exibidor de conteúdo. A diferença entre esses adaptadores é que o *FlexMediaRTPAdapter* transmite e recebe os dados da mídia utilizando RTP enquanto o *FlexMediaDataSourceAdapter* os envia utilizando um objeto Java. O Apêndice B ilustra uma aplicação desenvolvida para enviar dados de mídia, recebê-los e solicitar ajuste em tempo de exibição. Essa ferramenta pode ser utilizada por aplicações clientes da ferramenta de ajuste.

### 6.3. Integração da Ferramenta de Ajuste com o Sistema HyperProp

O formatador HyperProp inclui várias funcionalidades para orquestrar apresentações hipermídia, dentre elas destacam-se a adaptação de documentos em tempo de compilação e de execução, estratégias de antecipação das ações de exibição (mecanismos de pré-busca), mecanismos de integração com exibidores de mídia e de integração com a infra-estrutura de exibição (em especial, sistema operacional e rede).

O adaptador de exibidor de conteúdo é responsável por traduzir os comandos recebidos do formatador pela API da ferramenta de exibição do HyperProp (Rodrigues, 2003) (API FF, ver Figura 69) para as APIs da ferramenta de ajuste elástico (ver Seção 4.1). A Tabela 6, abaixo, ilustra o mapeamento dos métodos entre a API FF e a API de ajuste elástico em tempo de compilação desenvolvida nesta dissertação.

API FF	API de compilação de ajuste elástico
prepare	config, addTimescaleListener, getTimescaleInstant, start
start	(mecanismo de comunicação com exibidor de conteúdo)
stop	(mecanismo de comunicação com exibidor de conteúdo)
getAttributeValue	getOutputTools, getReport

Tabela 6 - Mapeamento dos métodos da API FF com os da API de ajuste elástico em tempo de compilação.

O ajuste em tempo de compilação deve ocorrer durante a preparação do adaptador. Por isso, o método `prepare` da API FF deverá chamar os métodos `config` e `start` da ferramenta de ajuste. Na preparação, o adaptador é informado da duração implícita da mídia a ser exibida e da sua duração esperada. A duração implícita do objeto pode já estar especificada ou então ser calculada (por exemplo, através da chamada `getDuration` oferecida pelo *player* JMF). A duração

esperada foi calculada pelo algoritmo de otimização do formatador HyperProp. O adaptador calcula o fator de ajuste a ser aplicado através da divisão da duração esperada pela implícita. No caso de não ser possível determinar a duração implícita da mídia *a priori*, por exemplo, para fluxos de dados ao vivo, o fator de ajuste deve ser explicitamente informado ao adaptador. Para tal, pode-se acrescentar um parâmetro ao conjunto de parâmetros de exibição da mídia passados pelo formatador e acrescentar a lógica de tratar esse novo parâmetro no adaptador<sup>14</sup>.

Caso seja desejado que o ajuste aconteça de modo assíncrono (isto é, a ferramenta de ajuste deva executar em uma nova *thread*), o adaptador deverá se registrar como observador do evento de finalização do processamento de ajuste, através do método `addTimescaleListener`, pois a sua preparação só pode terminar depois que o processamento de ajuste já estiver terminado.

Ainda durante a preparação do adaptador, ao final do processamento de ajuste, o adaptador pode desejar descobrir novos valores para determinados instantes de tempo, o que é possível através do método `getTimescaleInstant`. Esse método é particularmente importante para manter a coerência da apresentação hipermídia. A justificativa para isso é que o modelo conceitual do formatador HyperProp prevê que um autor crie âncoras temporais internas especificadas através de intervalos cujas extremidades são tempos relativos ao início da apresentação do objeto inteiro. Logo, uma vez ajustada a duração de uma mídia, os intervalos das âncoras também precisam ser corrigidos.

Os métodos `start` e `stop` não são mapeados para métodos da API FF, mas sim para a classe `IThirdPartPlayer` de integração com exibidor de conteúdo apresentada no Capítulo 5. O método `getAttributeValue` da API FF pode ser usado para recuperar qualquer atributo do adaptador e, por isso, pode ser mapeado para todos os métodos `get` das APIs de ajuste de seu interesse. São eles: `getOutputTools` e `getReport`.

De maneira similar à tabela anterior, a Tabela 7 ilustra o mapeamento dos métodos entre a API FF e a API de ajuste elástico em tempo de execução.

---

<sup>14</sup> Vale ressaltar que, no estágio atual de desenvolvimento, o algoritmo de otimização do formatador HyperProp que calcula as durações esperadas para todas as mídias do documento não

API FF	API ajuste elástico
Prepare	config, addTimescaleInstantListener
Start	start
Stop	stop
setAttributeValue	setFactor
getAttributeValue	getOutputTools, getReport

Tabela 7 - Mapeamento dos métodos da API FF com os da API de ajuste elástico em tempo de execução.

Nesse caso, o ajuste é configurado durante a preparação do adaptador e, conseqüentemente, o método `prepare` da API FF deve chamar o método `config` da ferramenta. Ainda durante a preparação, o adaptador pode chamar o método `addTimescaleInstantListener` para cadastrar-se como observador do evento de descoberta dos novos valores de determinados instantes de tempo (indicados pelo método `config`). Ademais, a execução do ajuste é realizada paralelamente a apresentação da mídia, o que implica que os métodos `start` e `stop` podem ser diretamente mapeados (além de também precisar se mapeado para métodos da classe `IThirdPartPlayer` de integração com exibidor de conteúdo).

O método `setAttributeValue` da API FF pode ser usado para modificar qualquer atributo do adaptador, em particular, o fator de ajuste utilizado, efetuando-se, assim, o mapeamento com o método `setFactor`. Essa característica é particularmente importante caso o formatador perceba que eventos imprevisíveis modificaram os cálculos previamente efetuados na elaboração do plano de execução. Por fim, o mapeamento do método `getAttributeValue` é análogo ao descrito para a API de compilação.

Somente o exibidor de conteúdo JMF foi testado na comunicação com o formatador. As classes utilizadas foram `JMFFilePlayer` e `JMFDataSourcePlayer` gerando respectivamente os adaptadores *FlexMediaFileAdapter* e *FlexMediaDataSourceAdapter*. Considerando que esses adaptadores desenvolvidos podem ser executados de modo integrado (ver Capítulo 5), é possível monitorar os eventos de apresentação e notificá-los ao formatador. O adaptador é responsável por traduzir o modelo de eventos do JMF no modelo de eventos do formatador e vice-versa. Desse modo, o formatador pode sincronizar a apresentação de outras mídias (por exemplo, um navegador HTML, um

---

está preparado para receber objetos com durações indeterminadas.



visualizador de JPEG etc) a partir dos eventos de apresentação recebidos pelo adaptador do JMF. Além disso, também é possível integrar espacialmente a exibição do conteúdo ajustado com outras mídias. Por exemplo, uma janela do sistema operacional pode conter diversos objetos diferentes (texto de legenda, vídeo, imagens etc), escondendo do usuário a existência de vários exibidores diferentes.

Para exibir formatos não suportados pelo JMF, as ferramentas VLC e Winamp podem ser facilmente utilizadas numa comunicação não-integrada. Atualmente, o VLC pode ser utilizado em um *FlexMediaFileAdapter* e *FlexMediaRTPAdapter* e o Winamp em um *FlexMediaFileAdapter*. Conforme já discutido no Capítulo 5, o desenvolvimento de uma comunicação via programação com o VLC pode permitir uma comunicação integrada com suporte a vários formatos em um adaptador *FlexMediaDataSourceAdapter*.

O Apêndice C ilustra um exemplo de hiperdocumento que precisa de ajuste elástico para ser corretamente apresentado pelo formatador HyperProp.