

# 1 Introdução

A evolução das técnicas de codificação digital, aliada aos esquemas eficientes de modulação para transmissões digitais, tornou possível o advento da TV digital. Atualmente, os sistemas de TV digital podem ser definidos, de forma resumida, como um conjunto de especificações que determinam as técnicas de codificação digital para transmitir o conteúdo de áudio, vídeo e dados, das emissoras (ou provedores de conteúdo) aos terminais de acesso dos telespectadores, e um conjunto de facilidades que dão suporte ao desenvolvimento de aplicações interativas.

No contexto de TV digital, terminal de acesso é o dispositivo responsável por tratar corretamente o sinal digital recebido, decodificando e exibindo de forma consistente a programação de TV, as aplicações e outros serviços avançados. Um terminal de acesso de TV digital, para oferecer um custo acessível, possui forte limitação na quantidade de recursos computacionais e hardware especializado para a exibição de conteúdo televisivo. Um terminal de acesso típico geralmente possui um processador de baixo custo para executar as tarefas das aplicações e do sistema operacional (possivelmente Linux ou Windows CE (O'Driscoll, 2000)) e pouca memória para uso das aplicações e do sistema operacional. Obviamente, podem existir terminais mais sofisticados com capacidade de processamento mais alta para suportar aplicações mais complexas, disco rígido com capacidade de armazenamento para gravar os programas transmitidos, ou mesmo armazenar aplicações oferecidas pelos provedores de conteúdo. Entretanto, melhoras como essas aumentam sensivelmente o custo do terminal de acesso.

O hardware de um terminal de acesso pode variar de acordo com o sistema de TV digital utilizado e seus requisitos de desempenho. Como exemplo, um terminal de acesso do sistema de TV digital europeu terrestre deve possuir hardware para decodificar transmissões de acordo com o padrão de modulação europeu COFDM (*Coded Orthogonal Frequency Division Multiplexing*), enquanto que um terminal do sistema americano terrestre deve possuir hardware

para decodificar transmissões de acordo com o padrão de modulação americano VSB (*Vestigial Side Band*) (Schwalb, 2004). Considerando que a maioria dos padrões de TV adotam codificação MPEG, em geral, os terminais de acesso possuem uma decodificadora MPEG para recuperar os sinais de áudio e vídeo principais da programação sintonizada.

Ainda nos terminais de acesso, é comum existir uma camada de abstração, denominada middleware, responsável por esconder das aplicações a complexidade dos mecanismos definidos pelos padrões, protocolos de comunicação e até mesmo sistema operacional do equipamento. Simplificadamente, as implementações de middleware devem oferecer as bibliotecas necessárias às aplicações através de uma API (*Application Program Interface*) bem definida.

A seguir, são apresentadas a motivação para realização deste trabalho, os objetivos a serem atingidos, bem como a organização deste documento.

### **1.1. Motivação**

O primeiro padrão aberto utilizado como middleware de TV digital interativa foi definido pelo grupo MHEG (*Multimedia and Hypermedia Experts Group*) e publicado em 1997 pela ISO (*International Standards Organization*). O padrão oferece um paradigma declarativo para o desenvolvimento de aplicações hipermídia.

As primeiras especificações do grupo MHEG, denominadas MHEG-1 (ISO, 1997a), definem uma forma declarativa para a representação de objetos hipermídia por meio da notação base ASN.1 (*Abstract Syntax Notation One*) (ISO, 2002). MHEG-1 define um formato de intercâmbio para possibilitar que uma mesma representação dos objetos hipermídia seja apresentada em máquinas diferentes e ainda permite a associação de códigos procedurais aos objetos hipermídia. Para garantir a portabilidade entre diferentes plataformas dessa parte procedural embutida nos objetos, o grupo MHEG publicou as especificações

MHEG-3<sup>1</sup> (ISO, 1997b), que definem uma máquina virtual MHEG e uma representação em *bytecode* MHEG (Morris & Chaigneau, 2005).

Atualmente, com o objetivo de deixar as aplicações presentes nos terminais de acesso independentes de plataforma, a maioria das implementações de middleware lança mão do mesmo recurso utilizado pelo grupo MHEG: máquinas virtuais como ambiente de execução para as aplicações. Ou seja, uma máquina virtual do middleware cria um ambiente de execução comum às suas aplicações. Nesse cenário, as aplicações, após desenvolvidas em código-fonte, são compiladas para *bytecode* de sua máquina virtual não sendo, portanto, necessário manter diferentes versões das aplicações para cada plataforma de terminal de acesso contemplada.

Entretanto, o padrão MHEG-1 não foi bem sucedido no mercado. Seu mau resultado foi atribuído à complexidade dos conceitos definidos em suas especificações, bem como ao mercado não estar ainda preparado para o grande número de recursos oferecidos pelo padrão. Para resolver esses problemas, o grupo MHEG criou um perfil simplificado do padrão MHEG-1, publicando em abril de 1997 as especificações MHEG-5<sup>2,3</sup> (ISO, 1997c).

Paralelamente, a tendência tecnológica Java ganhava força, ameaçando o MHEG e fazendo com que o grupo de padronização especificasse uma API Java para o padrão em 1998: o MHEG-6 (ISO, 1998a). O padrão MHEG-6 permite que as aplicações MHEG-5 sejam associadas a aplicações Java por meio de uma API (*iso.mheg5*), que mapeia a hierarquia de classes MHEG-5 em classes Java. Assim, MHEG-6 permite às aplicações Java acessarem as variáveis do domínio MHEG e obter resultados após computação realizada nesse domínio. Além disso, as aplicações Java podem manipular os objetos MHEG-5 de forma a controlar as apresentações desses objetos. Além de oferecer uma API Java para o MHEG-5, o

---

<sup>1</sup> O padrão MHEG-2 especificaria o SGML como linguagem-base para a codificação das representações definidas pelo MHEG-1, porém foi cancelado por falta de recursos.

<sup>2</sup> O padrão MHEG-4 define o procedimento oficial ISO para registro de identificadores usados nas outras partes do MHEG. Portanto, devem ser registrados no MHEG-4 os novos objetos relacionados ao MHEG que surgirem, modificações de valores na numeração existente, tabelas com eventos de entrada, atributos de conteúdo explicitamente definidos, como tabela de cores ou fontes, entre outros.

MHEG-6 possui uma especificação de máquina virtual Java, para garantir sua portabilidade.

Apesar de nunca ter sido desenvolvido um middleware conforme o padrão MHEG-6, suas especificações serviram de base para um padrão de middleware para TV digital interativa, publicado em 1998 e desenvolvido pelo consórcio de empresas DAVIC (*Digital Audio Visual Council*) [DAVIC99]. Esse consórcio representou diversos setores da indústria audiovisual e foi iniciado em 1994, mas extinto após cinco anos de atividade, conforme previsto em seu estatuto. No padrão DAVIC, novas APIs Java foram adicionadas ao MHEG-6, fazendo com que a linguagem Java tivesse mais acesso aos recursos do terminal de acesso como, por exemplo, controle da apresentação de conteúdo audiovisual por meio do arcabouço JMF (*Java Media Framework*) (Sun, 1999), e gerenciamento dos recursos do terminal de acesso.

A linguagem Java destacou-se durante o desenvolvimento dos padrões de TV digital europeu DVB (*Digital Video Broadcasting*) (DVB, 2004). O fato de algumas empresas que faziam parte do consórcio DAVIC, e tiveram experiências com as APIs Java, estarem envolvidas no desenvolvimento dos padrões DVB, contribuiu substancialmente para que, em 1999, surgisse a primeira implementação em Java sobre um padrão aberto de middleware. Sua implementação seguiu as especificações do middleware europeu MHP (*Multimedia Home Platform*) (ETSI, 2003) e significou a mudança do paradigma de programação declarativo para o paradigma orientado a objetos (ou de middleware declarativo para procedural, como é a denominação comum no mundo de TV digital).

O desenvolvimento de um middleware puramente procedural não supriu, entretanto, a necessidade de um middleware declarativo. Por oferecerem um nível de abstração mais elevado, as linguagens declarativas não exigem tanto conhecimento de programação quanto as linguagens procedurais.

Os middlewares declarativos são mais adequados para aplicações cujo foco casa com o objetivo específico para o qual a linguagem declarativa foi

---

<sup>3</sup> O padrão MHEG-5 foi o primeiro padrão aberto utilizado como middleware de TV digital interativa e ainda é utilizado pela principal rede de TV digital do Reino Unido.

desenvolvida; ao contrário do uso de middlewares procedurais que são de propósitos mais gerais e ideais para aplicações que precisam de uma linguagem com maior poder de expressão. Aos poucos os padrões de TV digital foram reincorporando as funções do middleware declarativo ao middleware procedural Java. O foco da linguagem declarativa deixou de ter a rica expressividade da linguagem MHEG, passando a ter a simplicidade excessiva da linguagem HTML (ISO, 2000b).

Atualmente, a maioria das implementações de middleware (ATSC, 2005); (DVB, 2004) oferece simultaneamente ambientes distintos para as aplicações, ou seja, um ambiente para controle das aplicações procedurais e um ambiente para controle das aplicações declarativas. Entretanto, isso é possível apenas em terminais que possuam poder de processamento suficiente para executar, ao mesmo tempo, duas máquinas virtuais de características bem distintas como, por exemplo, uma máquina virtual Java e uma máquina virtual MHEG. Na realidade, a presença de cada máquina virtual acarreta em aumento de custo do terminal de acesso, considerando não só o maior poder de processamento necessário para executá-la como os pagamentos sobre royalties e propriedades intelectuais.

O baixo custo do terminal de acesso é fator crucial para o sucesso da TV digital aberta, principalmente nos países em desenvolvimento. Para que o baixo custo comprometa o mínimo possível dos recursos dos terminais de acesso, é interessante que eles estejam isentos de custos adicionais como, por exemplo, software, propriedade intelectual e royalties. Um dos principais pontos para tornar isso possível concentra-se na escolha do middleware, que faz uso de mecanismos definidos por protocolos de comunicação e pelo sistema operacional e suas bibliotecas, para suporte às aplicações. No entanto, tal middleware deverá ter um foco mais geral, de forma a atender a maioria das aplicações a serem desenvolvidas.

Os middlewares declarativos existentes como, por exemplo, DVB-HTML (ETSI, 2003) e DASE declarativo (ATSC, 2003), possuem foco na interatividade, um tipo de sincronismo de mídias que facilita o desenvolvimento das aplicações com interação do telespectador. Assim, para esses middlewares, qualquer outro tipo de sincronismo de mídias, que não seja a interatividade, deve ser descrito de forma procedural.

A necessidade de sincronismo no contexto da TV digital, sem a interação do usuário telespectador, está presente mesmo em sua aplicação mais primária: a exibição temporalmente sincronizada do fluxo de vídeo e áudio principal de um programa. As aplicações para TV digital, na maioria das vezes, devem lidar com a sincronização, espacial e temporal, de objetos de diferentes tipos de mídia, além dos objetos de vídeo e áudio que compõem o fluxo principal. Assim, o sincronismo de mídias deve ser o foco da linguagem declarativa, que deve tratar a interatividade como um caso particular do sincronismo.

Uma abordagem possível para contemplar linguagens que vislumbram o sincronismo de mídias, mas que não resolve o problema da limitação de recursos da plataforma, foi desenvolvida pelo *Telecommunications Software and Multimedia Laboratory* da *Helsinki University of Technology*. Esse laboratório realizou recentemente a adaptação de um formatador SMIL (*Synchronised Multimedia Integrated Language*) (Bulterman, 2004), uma linguagem declarativa com foco em sincronismo de mídias, para o contexto da TV digital (Lamadon et al, 2003). Além de questões de interface com o usuário, métodos de interatividade para acessar o conteúdo multimídia como, por exemplo, a utilização do canal de retorno, foram considerados. O formatador SMIL para TV digital foi desenvolvido com a tecnologia Java para executar sobre o middleware MHP, consistindo em um Xlet (ETSI, 2003) que necessita de uma máquina virtual DVB-Java (ETSI, 2003), ou DVB-J, para controlar seu ciclo de vida (i.e., carregar, iniciar, parar e terminar) (Lamadon et al, 2003). Essa mesma abordagem foi também seguida pelas primeiras implementações do formatador do sistema HyperProp, adaptado para o ambiente de TV Digital.

Desenvolvido no Laboratório TeleMídia, o Formatador HyperProp (Rodrigues, 2003) consiste de uma ferramenta capaz de interpretar documentos especificados em NCL (*Nested Context Language*) (Muchalut-Saade, 2003), uma das principais linguagens declarativas existentes com foco no sincronismo de mídias, baseada no modelo conceitual NCM (*Nested Context Model*) (Soares et al, 2003). A partir da especificação do documento NCL recebida, o Formatador HyperProp constrói um plano de apresentação que irá conter as características de apresentação de cada objeto de mídia, a programação das tarefas a serem escalonadas e as informações dos relacionamentos de sincronização entre os objetos de mídia. A sincronização inter-mídia direciona a sincronização intra-

objetos, que é realizada com o auxílio de mecanismos de reserva de recursos e também de um mecanismo de pré-busca, escalonado através de um plano de pré-busca construído a partir do plano de apresentação. Baseado nos eventos gerados pelos exibidores, nos eventos gerados pelo usuário e no plano de apresentação, o escalonador de apresentação controla a execução sincronizada de todo o documento, realizando ajustes quando esses se fazem necessários.

No formatador HyperProp, existe uma independência entre o controlador da apresentação e as ferramentas de exibição (módulos responsáveis pelo controle da exibição de cada objeto em si). Para isso, um modelo de interface, genérico e adaptável, para integração de ferramentas de exibição, é implementado, permitindo tratar igualmente todos os objetos sendo exibidos, independente da aplicação exibidora que os controla. O *framework* de integração das ferramentas permite que novas ferramentas de exibição venham a ser incorporadas, e outras tantas existentes possam ser adaptadas, funcionando de maneira integrada entre si e com o formatador hipermídia (Rodrigues et al, 2001). A versão atual do Formatador HyperProp é implementada em Java e, naturalmente, utiliza JMF para exibir conteúdos de mídia contínua.

Conforme discutido, além das preocupações com royalties e propriedade intelectual, implementações Java têm seus problemas ligados ao desempenho e recursos de máquina demandados. Mesmo hoje em dia as máquinas virtuais Java impõem perdas de desempenho. Para resolver essas e outras questões relacionadas ao contexto de TV digital, a implementação de um middleware declarativo independente do uso de tecnologia Java fez-se necessária.

## **1.2. Objetivos**

Este trabalho tem como objetivo principal propor um middleware declarativo para sistemas de TV digital interativa. Entre outros requisitos ligados ao custo, esse middleware declarativo deve estar restrito às tecnologias que não acrescentam dispêndio através de propriedades intelectuais, patentes ou royalties. O objetivo principal deste trabalho condiz com os requisitos do Sistema Brasileiro de TV Digital (SBTVD).

Para promover a implementação do middleware proposto, a arquitetura modular do Formatador HyperProp, que serviu como base da implementação, foi

estendida e reestruturada para um perfil simples, direcionado à TV digital, e reimplementada na linguagem C++. Todos os exibidores de mídia (ferramentas responsáveis por controlar a exibição de cada objeto que compõe um programa de TV digital) foram também desenvolvidos, de forma a suportar os requisitos dos terminais de acesso (i.e. bibliotecas de código aberto com baixos requisitos de processamento). Além disso, outras questões relacionadas ao contexto da TV digital foram consideradas:

- Ao contrário da interação entre o usuário e os aplicativos para PC, usualmente realizada através do uso de um mouse, o middleware declarativo deve oferecer às aplicações uma interface com o usuário que possibilite a interação através de um dispositivo de controle remoto simples;
- No desenvolvimento de software para PC, a portabilidade não é uma questão difícil de se resolver devido ao uso ilimitado de máquinas virtuais. No entanto, para garantir portabilidade, o middleware declarativo deverá ser desenvolvido conforme o padrão ISO para a linguagem C++ (ISO, 2003), dessa forma, qualquer plataforma provida de script gnu g++ deve ser capaz de compilar, no mínimo, o núcleo do middleware declarativo;
- Ao contrário de usuários de computadores, os telespectadores não pensam na possibilidade de uma aplicação fechar por falta de recursos. O middleware declarativo deve possuir um gerenciamento de recursos de forma a ter o mínimo de consumo possível;
- Para a composição do middleware declarativo, ao novo formatador NCL devem ser integradas funções para recepção de objetos de mídia provenientes de fluxos MPEG-2 Sistemas (ISO, 2000a);
- Ainda com relação ao fluxo MPEG-2 Sistemas, deve ser dado suporte ao transporte de dados segundo a proposta DSM-CC (*Digital Storage Media – Command and Control*) (ISO, 1998b), e suporte a eventos de sincronização, cujo formato de dados deve fazer parte da especificação da API do middleware;
- O middleware declarativo deve possibilitar que edições das aplicações NCL sejam realizadas simultaneamente a suas apresentações. As modificações das aplicações declarativas,

especificadas no provedor de conteúdo por um ambiente de autoria, devem ser coerentemente atualizadas nos terminais de acesso, preservando todos os relacionamentos, incluindo aqueles que definem a estruturação lógica de um documento hipermídia (Moreno et al, 2005c).

### **1.3. Organização do Documento**

O restante deste documento encontra-se organizado da seguinte forma. O Capítulo 2 apresenta discussões sobre os principais middlewares declarativos de TV digital existentes atualmente, realizando ainda uma análise comparativa. No Capítulo 3 são apresentadas as tecnologias relacionadas a esta dissertação. Uma atenção especial é dada às especificações do padrão MPEG-2, por ser o padrão para o transporte de conteúdo audiovisual mais utilizado nos sistemas de TV digital, bem como ao modelo de apresentação de interfaces gráficas adotado na maioria dos sistemas de TV digital, por sua importância no âmbito das aplicações para TV digital. O Capítulo 4, por sua vez, discorre sobre a arquitetura do middleware declarativo proposto. Para isso, são detalhadas as funcionalidades de cada um dos módulos pertencentes a essa arquitetura, assim como a forma de comunicação entre eles. O Capítulo 5 apresenta a descrição da implementação da arquitetura proposta no Capítulo 4. Por fim, o Capítulo 6 encerra a dissertação, descrevendo as conclusões obtidas a partir de todo trabalho realizado e os possíveis trabalhos futuros.