

Um Estudo sobre Emuladores de Aplicações para a Televisão Digital Interativa

Carlos A. Piccioni

piccioni@das.ufsc.br

Carlos Montez

montez@das.ufsc.br

LCMI - Depto de Automação e Sistemas - Univ. Fed. de Santa Catarina

Caixa Postal 476 - 88040-900 - Florianópolis - SC – Brasil

Resumo

O surgimento da televisão digital, além de uma melhora significativa na qualidade do vídeo e áudio, trouxe consigo a possibilidade de interação do telespectador com o conteúdo transmitido através da Televisão Digital Interativa. Especificações como a API JavaTV da Sun Microsystems surgiram com o intuito de fornecer suporte aos desenvolvedores desse ramo. Essa API introduziu o conceito de Xlet, uma nova abstração para as televisões digitais, equivalente ao de uma applet Java para um browser na Internet. Como consequência, algumas ferramentas foram criadas para testes de Xlets em PCs, tais como a implementação de referência da API JavaTV criada pela Sun, e outros emuladores independentes, como o XleTView. Este artigo faz um levantamento dos principais emuladores disponíveis, comparando suas principais características.

Palavras Chave: Televisão Interativa, JavaTV, Xlet.

1. Introdução

As redes televisivas tradicionais, em seu princípio, utilizavam-se de ferramentas para a captura, armazenamento, edição e difusão de seu conteúdo de vídeo e áudio de forma analógica. Com o passar do tempo e a revolução da tecnologia digital, as produtoras passaram a adotar a tecnologia digital na manipulação de seu conteúdo televisivo. Ultimamente, a tendência é a difusão analógica ser substituída pela digital, seja ela via satélite, via cabo, ou via radiodifusão.

Essa nova tecnologia que chega ao mercado – intitulada Televisão Digital (DTV, *Digital Television*) [1] – promete revolucionar a forma como assistimos a televisão atualmente. Dentre as suas grandes vantagens, talvez a principal seja a de que será possível transformar o telespectador de um mero ‘espectador’ em um ser que possa interagir com a programação. Assim, nasce um novo campo no mundo televisivo a ser explorado: o da Televisão Interativa.

Nesse novo segmento de aplicação, são várias as tecnologias emergentes buscando explorar todo o potencial da televisão digital a fim propiciar a melhor experiência interativa para o telespectador. Vários padrões já foram estabelecidos, sendo que a maioria faz uso da especificação *JavaTV* da *Sun Microsystems* e do conceito de *Xlets* estabelecido pela mesma.

Esse artigo, dividido em 7 partes, tem por objetivo apresentar algumas ferramentas que podem ser usadas para criar e testar aplicativos nesse conceito, além de dar uma visão geral sobre o contexto atual da Televisão Interativa. Na seqüência a esta introdução é contextualizado o que é uma Televisão Digital Interativa e alguns serviços que essa nova mídia irá possibilitar. Na seção 3 é caracterizado o que são set-top boxes, receptores usados para processar conteúdo digital interativo. A seção 4 apresenta a API JavaTV e sua principal abstração: as Xlets. Na seção 5 são apresentadas as principais ferramentas de emulação de ambiente de execução de Xlets. Na seqüência, na seção 6 é feita uma comparação entre as principais características desses emuladores. Finalmente, na seção 7 são apresentadas as conclusões deste artigo.

2. O que é Televisão Digital Interativa

Uma *difusão* pode ser concebida como o envio de um sinal contendo o conteúdo televisivo, de um ponto, geralmente de uma entidade denominada rede, para vários outros pontos, os receptores, onde se encontram os usuários finais, os telespectadores [2]. Uma rede é a uma companhia produtora ou gerenciadora de conteúdo e/ou serviços e responsável por sua difusão. Um receptor de televisão

analógico atual tem a função básica de sintonizar determinado canal e repassar seu conteúdo áudio visual ao telespectador. Com a chegada da televisão digital, as possibilidades nesse campo tornaram-se muito mais amplas, pois o receptor de televisão passou a receber conteúdo digital, o qual pode processar.

Um sinal digital correspondente a um canal televisivo, com a mesma qualidade de um canal analógico após ser modulado em uma onda portadora analógica, ocupa uma faixa de frequência menor que seu equivalente analógico. Dessa forma, a largura de banda no qual um canal analógico é atualmente difundido pode conter vários canais equivalentes no formato digital, ou um canal com uma qualidade de imagem e som extremamente superior.

Na televisão digital, devido ao fato da informação difundida estar no formato digital, se o receptor possuir poder de processamento, é possível também difundir os mais diversos tipos de informação além do tradicional vídeo e áudio, como por exemplo, aplicativos, dados e legendas.

Com as possibilidades citadas até então, podemos definir um tipo de interação: a *interação local*. Nela, por exemplo, o espectador pode decidir entre vários ângulos de um evento esportivo, por exemplo, ou entre várias legendas e/ou dublagens de um determinado filme, visto que várias seqüências de áudio, vídeo e dados podem ser difundidos em um mesmo espaço onde atualmente é ocupado por apenas um canal analógico.

Também é possível a difusão de aplicativos, como jogos, que serão executados na televisão digital e com os quais o telespectador pode interagir. Outro exemplo de interatividade local é o Guia de Programação Eletrônica, onde o telespectador pode navegar entre as mais diversas grades de programação e selecionar o serviço (ou canal) desejado.

O conceito de interatividade pode tornar-se muito mais amplo quando o receptor possuir o chamado canal de retorno [4]. Com ele, ao invés de apenas receber sinais de uma emissora, o telespectador pode enviar dados de volta para mesma, tornando possível assim a criação de uma imensa gama de novos serviços, como por exemplo:

- Acesso a Internet;
- Acesso a bancos (t-banking);
- Comércio eletrônico (t-comercing);
- Governo eletrônico (t-government)
- Vídeo sob-demanda (VOD - *Vídeo On-Demand*);
- Jogos interativos;
- Bate-Papo (*chat*);
- Serviços de mensagens instantâneas;
- Correio eletrônico (e-mail).

Quanto maior a capacidade do canal de retorno, maior são as possibilidades de uma verdadeira experiência interativa por parte do telespectador.

3. Set-Top Boxes

Atualmente, os preços de televisores digitais são proibitivos para a maioria dos telespectadores. Dessa forma, a fim de atender a maioria desses, e ainda aproveitar as televisões analógicas, são comercializados receptores denominados *set-top boxes* e são responsáveis por todo o processamento do sinal digital, pela interação com o telespectador e pela entrega de conteúdo áudio visual para o mesmo através de uma televisão analógica comum.

Existe uma grande variedade de set-top boxes, cada qual com seu conjunto de funcionalidades. Esses equipamentos podem ser divididos primeiramente em três categorias de acordo com o tipo de conexão com o meio de difusão televisivo [1]:

- *Enhanced Broadcast Receivers*: Além da tradicional exibição de um canal televisivo, são capazes de lidar com gráficos, imagens e textos transmitidos junto com a difusão do sinal, os quais podem ser controlados por aplicações também transmitidas com o sinal. Não possui um canal de retorno, assim suporta apenas a interatividade local.

- *Interactive Broadcast Receivers*: Além das capacidades do *Enhanced Broadcast Receiver*, possui um canal de retorno com o servidor, possibilitando assim serviços interativos.
- *Multi-Network Receivers*: Além das capacidades do *Interactive Broadcast Receiver*, possui acesso a mais de uma rede de difusão e um canal de retorno. Assim, possibilita a utilização de outros serviços de comunicação, como por exemplo, telefonia local e Internet.

As principais funcionalidades de um *set-top box* são [3]:

- Sintonização, demodularização, demultiplexação e decodificação do sinal recebido;
- Verificação dos direitos de acesso e níveis de segurança;
- Saída de vídeo com qualidade de cinema ao televisor;
- Saída de som com qualidade Estéreo, *Surround* ou *Dolby Digital*;
- Processar e verificar os serviços de Televisão Interativa.

No contexto da televisão digital, para poderem ser difundidas, tanto as informações áudio visuais como os dados são codificados no formato padrão MPEG-2 e multiplexados em seqüências de transporte MPEG-2. Uma rede pode ser responsável pela transmissão de uma ou várias seqüências de transporte. Após isso, a seqüência de transporte é convertida em um sinal analógico através de um processo conhecido como modulação, para ser então amplificado e difundido.

O *set-top box* funciona como um receptor e sua primeira tarefa é sintonizar o sinal difundido. Após isso, o sinal é demodularizado, resultando assim na seqüência de transporte MPEG-2, para ser posteriormente demultiplexado em seqüências MPEG-2 elementares, as quais serão decodificadas. Após esse processo, o *set-top box* possuirá todas as informações necessárias em formatos convenientes para que o mesmo possa manipulá-las e entregá-las aos telespectador.

Hardware do set-top box

Como já foi apresentado, diferente do que ocorre com uma televisão analógica, que apenas sintoniza determinado canal e o exibe em sua tela, o *set-top box* possui poder de processamento. A unidade central de processamento, CPU, é o cérebro do receptor e responsável por esse poder. Dentre as funções da CPU podemos citar [3]:

- Execução de programas residentes no *set-top box*;
- Execução de programas oriundos de seqüências de dados MPEG2;
- Processamento de aplicações interativas;
- Monitorização e administração das interrupções de hardware;
- Gerenciamento da memória.

Assim como ocorre em um PC, o receptor possui também um processador gráfico, responsável pela exibição na televisão (convencional ou digital) do vídeo e também por processar elementos gráficos gerados por aplicações, como por exemplo, gráficos em 3D de jogos.

Com relação à memória, um *set-top box* possui tanto o tipo RAM (*Random Access Memory*), necessária para o armazenamento temporário de aplicações em execução e de dados, quanto memória ROM (*Read Only Memory*), geralmente dos tipos Flash ou EEPROM, para guardar de forma permanente programas e configurações do aparelho.

Os receptores podem possuir também unidades de armazenamento de grande capacidade, utilizando interfaces comuns em PCs como a interface IDE ou a SCSI, além dos discos do tipo ZIP ou JAZZ. Dessa forma, esses aparelhos tornam-se capazes de arquivar várias aplicações difundidas e, até mesmo, seqüências áudio visuais, funcionando de uma forma similar a um vídeo-cassete digital.

O canal de retorno pode utilizar as mais diversas tecnologias disponíveis, como por exemplo, linha telefônica, ADSL, cabo, etc, através do uso de *modems* específicos, para fazer a comunicação no sentido inverso da difusão, do telespectador para o operador da rede.

Os *set-top boxes* podem fazer uso de várias outras interfaces para a comunicação com o mundo externo. Podemos citar dentre elas as seguintes [3]:

- IEEE-1284 (Interface Paralela) – Utilizada para conexão a impressoras;

- RS-232 (Interface Serial) – Utilizada para conectar impressoras, PCs e modems;
- USB (Universal Serial Bus) – Utilizada para conectar teclados, joysticks, mouses e vários outros equipamentos ao set-top box;
- IEEE-1394 (Firewire) – Conexão de alta velocidade para filmadoras e outros dispositivos áudios-visuais;
- 10 Base-T (Ethernet) – Conexão em rede local com PCs e impressoras;
- PCMCIA tipo II – Vários usos como, por exemplo, estender a capacidade de memória do set-top-box, deciframento de acesso condicional para serviços de várias redes diferentes, aumentar a capacidade de armazenamento e adicionar novos sintonizadores ao set-top box;
- SmartCards – Permite acesso a diversos serviços de várias redes, além do uso em comércio eletrônico;
- Controles remotos e teclados sem fio.

Software do set-top box

Sobre todo o conjunto de hardware, descrito até então, existe uma pilha de software, dividida em várias camadas, e que faz uso de todas as funcionalidades do hardware descritas até então para interagir com o telespectador e entregá-lo o conteúdo televisivo interativo.

A camada inferior da pilha de software é formada por um conjunto de *devices drivers* necessários para que um sistema operacional de tempo real executando na camada adjacente possa controlar o hardware. Esse sistema operacional possui peculiaridades com relação aos usados em PCs. A primeira é que o mesmo dispõe de recursos mais escassos que em um micro-computador, como uma memória menor e processadores mais lentos, ou seja, precisa estar apto para ser executado nesse ambiente. A segunda, é que precisa ser extremamente robusto, pois nenhum telespectador espera que sua televisão “congele” ou desligue sozinha abruptamente.

Acima da camada do sistema operacional, encontra-se o *middleware* [5], que é a camada responsável por fornecer uma interface de programação de aplicação (API - *Application Programming Interface*) aos desenvolvedores de aplicações. Essa interface é independente do hardware e das tecnologias de comunicação subjacentes, facilitando assim a vida do programador.

Vários padrões de *middleware* já foram estabelecidos, como o padrão europeu MHP (*Multimedia Home Platform*) da DVB (*Digital Video Broadcasting*), o americano DASE (*DTV Application Software Environment*) da ATSC (*Advanced Television Systems Committee*), e o padrão japonês ARIB da ISDB.

A última camada é a camada de aplicação, onde residem os aplicativos produzidos para fazerem parte ou interagir com os diversos serviços, entregando assim ao telespectador o conteúdo digital em sua televisão. A Figura 1 ilustra o conjunto de camadas típica em um *set-top box*.



Figura 1: As diversas camadas típicas de um set-top box.

4. A tecnologia JavaTV e as Xlets

O *JavaTV* é uma interface de programação de aplicação, ou API (*Application Programming Interface*), e é uma extensão da linguagem Java da *Sun* que tem por objetivo facilitar o trabalho dos desenvolvedores de aplicações de Televisão Interativa para set-top boxes. A API *JavaTV* torna possível a criação de aplicações interativas independente da tecnologia utilizada nos protocolos de transmissão, do sistema operacional dos set-top boxes e da camada de hardware das redes de difusão [4].

O ambiente de software em um *set-top box* com a tecnologia *JavaTV* consiste basicamente de um ambiente de aplicações Java, a API *JavaTV* e outras APIs, e suas aplicações, além de um sistema operacional de tempo real (RTOS – *Real Time Operating System*).

A Figura 2 compara a arquitetura de um *set-top box* genérico com um utilizando a tecnologia *JavaTV*. A camada de aplicação utiliza a API *JavaTV* e os pacotes Java da camada imediatamente abaixo, a *Java Technology Layer*. Todas as aplicações em Java são executadas sobre a máquina virtual Java (VM – *Virtual Machine*). Essa arquitetura é que torna possível ao programador abstrair-se dos detalhes de hardware e protocolos de comunicação dos set-top boxes.

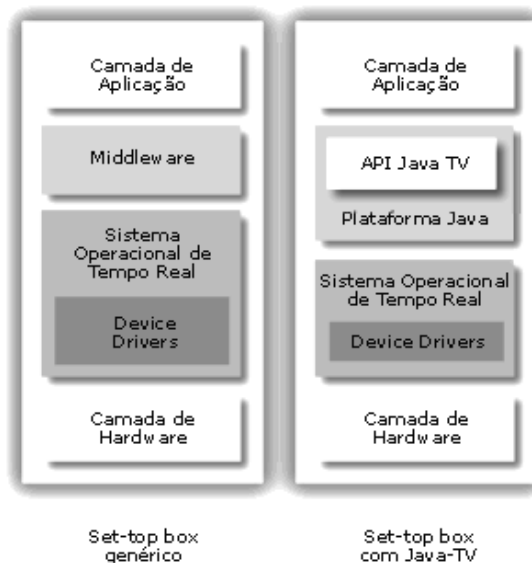


Figura 2: Arquitetura de um set-top box genérico e um utilizando a tecnologia Java.

O Sistema Operacional de Tempo Real (RTOS – *Real Time Operational System*) fornece o suporte em nível de sistema necessário para a camada Java (deve ser apto a executar a máquina virtual), além de controlar o receptor através de uma coleção de *device drivers*. O *PersonalJava* é o ambiente de aplicação geralmente utilizado em *set-top boxes*, e a API *JavaTV* foi desenvolvida para fazer uso desse ambiente.

Dentre as principais funcionalidades da API *JavaTV*, podemos citar:

- *Serviços de Acesso e Informação de Serviço*: A API *JavaTV* trata tanto uma programação de televisão interativa e uma programação convencional como um serviço. As APIs de Informação de Serviço (SI, *Service Information*) fornecem suporte para a obtenção da informações sobre os diversos serviços disponíveis, através do acesso ao banco de dados de Informação de Serviço. Tal processo é utilizado para a escolha de determinado serviço, através das APIs de Seleção de Serviços do *JavaTV*.
- *Controle do Canal de Difusão*: O *JavaTV* utiliza o JMF (*Java Media Framework*) para controlar e exibir o conteúdo áudio-visual de um *set-top box*.
- *Acesso aos Dados de Difusão*: Um serviço é modelado como um conjunto de seqüências de dados. Todos as seqüências de dados necessárias a uma aplicação podem estar presentes ou

O próximo tópico apresenta algumas ferramentas úteis para os desenvolvedores de conteúdo de televisão interativa criarem suas próprias Xlets e as testarem em um PC convencional.

5. Ferramentas

Um ambiente de televisão interativa é bem diferente de um ambiente de um PC. O primeiro possui recursos de memória e processamento mais escassos que o segundo. Além disso, um *set-top box* recebe o conteúdo através da difusão de um sinal que contém uma seqüência de transporte MPEG-2, enquanto um PC pode utilizar as mais diversas mídias armazenadas localmente ou através da Internet. Além disso, o *set-top box* possui vários dispositivos de hardware específicos que não são encontrados normalmente em um microcomputador, tais como decodificadores MPEG e demodularizadores. Apesar dessas diferenças, é possível desenvolver e testar aplicações para televisão interativa em um PC. Para isso, existem algumas ferramentas disponíveis que tornam possível a execução e depuração de Xlets nesse ambiente. A seguir, serão apresentadas algumas opções ao desenvolvedor.

A implementação de referência JavaTV.

A Sun, logo após apresentar sua especificação da API *JavaTV*, desenvolveu uma implementação de referência (*JavaTV RI*) da mesma para o Windows NT. Assim, desenvolvedores de aplicações interativas podem testar as potencialidades dessa API e também os desenvolvedores de *middlewares* de *set-top boxes* podem verificar o exemplo de implementação da mesma. A Figura 4 compara a arquitetura da API sendo executada em um receptor digital com a mesma sendo executada sobre o Windows NT.

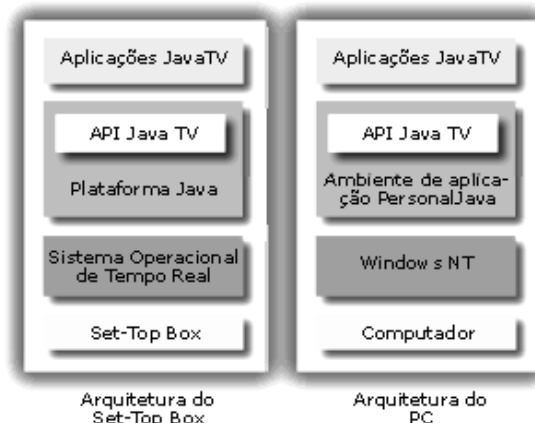


Figura 4: A arquitetura da API JavaTV executada em um PC e em um set-top box.

A *JavaTV RI* pode ser encontrada a partir do respectivo endereço na Internet da Sun [4], e segundo suas especificações, foi desenvolvida para fazer uso do ambiente de emulação do *PersonalJava* no Windows [6]. O *PersonalJava* é a antiga especificação Java da Sun para dispositivos com menor capacidade de processamento e memória como é o caso dos *set-top boxes*.

Atualmente, existe um esforço por parte da Sun de substituir o *PersonalJava* pelo perfil *Personal Profile* da configuração *Connected Device Configuration* (CDC) do segmento J2ME – *Java 2 MicroEdition*. Porém, devido ao fato de que padrões como o MHP adotaram em sua especificação o uso do *PersonalJava*, a migração para o CDC pode ser lenta.

A *JavaTV RI* é dividida em três segmentos: o código público da API, a camada portátil e a camada de emulação [8]. A camada portátil – por ser totalmente programada em Java – é independente de plataforma. Ela é responsável pela implementação do gerenciador de aplicação (gerenciar o ciclo de vida de uma aplicação), da seleção de serviços e do acesso aos dados de difusão.

A camada de emulação consiste de classes que são específicas para determinada plataforma e não podem ser reutilizadas em outras plataformas. É responsável por armazenar e gerar eventos de informação de serviços e apresentar recursos de mídia através do *Java Media Framework* (JMF).

Uma seqüência de transporte que seria sintonizada por um *set-top box* trazendo consigo toda as informações do serviços que a mesma contém, além do conteúdo áudio visual, pode ser emulada de duas formas ao se utilizar a implementação de referência: (i) através de uma programação da base de dados das informações de serviços, utilizando uma classe em Java; ou (ii) através de um arquivo XML. Em ambos os casos, as informações de serviços contém caminhos de arquivos multimídia relacionados com os mesmos armazenados localmente no PC. Dessa forma, pode-se emular uma hipotética recepção de vários serviços (ou canais).

Uma implementação de referência do JMF, chamada de *JMF Lite*, é responsável pela exibição da mídia no PC. A mesma possui parte de código nativo que faz uso da API *DirectShow* do Windows NT. Porém, é capaz apenas de exibir vídeos no formato MPEG-1. Caso deseje-se a exibição de vídeos no formato MPEG-2, deve-se substituir o *JMF Lite* por outra versão do JMF (preferencialmente a dois ou superior) e anexar um *plug-in* que possibilite a execução desse formato de mídia.

Utilizando o *Java 2 Standard Development Kit*, o desenvolvedor pode criar e compilar Xlets para testar com a implementação de referência do *JavaTV*. Para executar as mesmas, deve-se utilizar o ambiente de emulação *PersonalJava*, ou o próprio *Java 2 Standard Edition*, utilizando preferencialmente as classes de compatibilidade do *PersonalJava* que a *Sun* também disponibiliza [6].

A implementação de referencia da *JavaTV* fornece algumas Xlets de exemplos para o desenvolvedor, além de uma classe utilizada para carregar as Xlets, visto que as mesmas não possuem o método *main* e não podem ser executadas via linha de comando (assim como as *applets*).

XleTView

Outro emulador que pode ser usado para executar Xlets em um PC é o *XleTView* [9]. Possui o código aberto sob a licença GPL, e além de uma implementação de referência da API *JavaTV*, traz consigo implementações de outras APIs especificadas no padrão MHP, como a Havi (*Home Audio-Video interoperability*) [10], DAVIC (*Digital Audio-Video Council*) [11] e implementações especificadas pela própria DVB [12], além das bibliotecas do *PersonalJava* que o mesmo padrão faz uso.

Como é programado totalmente em Java, pode ser executado tanto em uma plataforma Linux ou Windows, bastando para isso utilizar o *Java 2 Standard Development Kit* para compilar Xlets e executar o *XleTView*. Esse emulador utiliza o JMF 2.1.1, porém com várias deficiências, como a incapacidade de exibir vídeo MPEG relacionado ou controlado por uma Xlet.

Outros Emuladores

Um outro emulador disponível é uma implementação parcial da especificação MHP criada pela *Espial* [13][14]. Além de uma implementação da API *JavaTV*, apresenta também uma implementação da API Havi. Também totalmente programado em Java, e pode ser utilizado em qualquer plataforma.

Outra opção é o emulador *mhp4free* [15]. Criado para ser executado em Linux, é totalmente em alemão e possui também uma implementação da API *JavaTV* além das outras utilizadas pelo padrão MHP.

6. Comparação entre as plataformas de emulação

A *Sun* foi umas das pioneiras ao lançar uma especificação para a televisão digital, a API *JavaTV*, além de uma implementação de referência para a mesma. A implementação de referência da API *JavaTV* pode ser um caminho para quem deseja desenvolver conteúdo para televisão interativa utilizando um PC como ambiente de desenvolvimento. Porém, para quem deseja apenas escrever Xlets, ela apresenta algumas desvantagens. A primeira é a dificuldade de instalação e a necessidade de componentes extras Java, como o JAXP (*Java API for XML Parsing*), que não acompanham e

implementação de referência. A configuração, apesar de abrangente, é também confusa a primeira vista.

Considerando esse ponto de vista – facilidade de configuração – o *XleTView* leva extrema vantagem. É de fácil configuração e dedicado apenas a executar Xlets. Porém, o *XleTView* não emula as informações de serviços como a *JavaTV* API RI, e não possibilita o controle e exibição de vídeo pela Xlet. O desenvolvedor desse emulador, na versão atual do mesmo (0.3.4) [9], sugere que uma seqüência de vídeo ou áudio seja simulada por uma figura ao fundo da tela, enquanto o mesmo ainda não possui o devido suporte para as mídias, prometido em novas versões do emulador.

A implementação de referência da *Sun* possibilita a exibição de conteúdo áudio visual controlado por uma Xlet, porém, com a limitação de ser em uma janela diferente da qual a Xlet está exibindo seus elementos gráficos. Ou seja, não é possível, nessa implementação, a exibição de vídeo com gráficos sobrepostos no PC.

Uma das grandes vantagens do *XleTView* sobre a solução da *Sun* é que a mesma busca suprir todo o conjunto de APIs especificadas pelo padrão MHP, enquanto a implementação de referência fornecida pela *Sun* é puramente uma implementação da API *JavaTV*. Assim, o *XleTView* aumenta o leque de possibilidades ao desenvolvedor de Xlets. Isso faz desse emulador um dos mais citados e recomendados em diversos fóruns na Internet [16].

A solução da *Sun* possui parte de código nativo para a plataforma Windows NT, para a criação de um *player* que seja capaz de executar arquivos MPEG-1. Assim, por ser de código aberto também, serve de exemplo aos desenvolvedores de *middlewares* para *set-top boxes* de como implementar essas funcionalidades em suas soluções. Ou seja, a implementação de referência *JavaTV*, apesar de ser um pouco deficiente em alguns recursos para a televisão interativa, serve de referência aos produtores de software de *set-top boxes* no que se diz respeito a criação do próprio *middleware*.

Aos outras ferramentas citadas no decorrer do texto como a alemã *mhp4free* e a implementação de referência proprietária da *Espial* também são soluções interessantes aos desenvolvedores. Um dos pontos negativos do *mhp4free* é não possuir suporte à plataforma Windows, além de ser todo em alemão, enquanto a solução da *Espial* possui implementada apenas parte da especificação MHP, porém possui grande parte da API *JavaTV*.

7. Conclusão

No ano de 2001, segundo a pesquisa PNAD/IBGE, 89% dos domicílios possuíam televisão no Brasil, enquanto apenas 51% possuíam telefone [17]. Esse número mostra a importância que a televisão irá ter nos próximos anos como veículo de disseminação de informação. A possibilidade de interação, oferecida nas televisões digitais, exacerba essa importância, possibilitando o oferecimento de serviços através da Televisão Digital Interativa.

Com o avanço das tecnologias digitais, nos meios de difusão de conteúdo televisivo, várias portas se abrem para os desenvolvedores nesse ramo, principalmente no segmento que se refere à televisão interativa. A *Sun* foi umas das pioneiras na área a lançar uma especificação para a televisão digital, a API *JavaTV*, além de uma implementação de referência para a mesma. O conceito de Xlets introduzido por ela tornou-se padrão de fato nessa área, além de vários outros recursos presentes na especificação.

Este artigo apresentou um estudo sobre plataformas de emulação para Xlets. Essas plataformas foram instaladas e testadas, e uma comparação qualitativa foi efetuada, considerando pontos como flexibilidade, portabilidade e facilidade de instalação.

Bibliografia

- [1] Digital Television. Consultado na Internet em 3 de novembro de 2003. <http://www.fcc.gov>
- [2] MHP Interactive. Consultado na Internet em 5 de dezembro de 2003. <http://www.mhp-interactive.org>
- [3] Televisão Interactiva. Consultado na Internet em 23 de janeiro de 2004. <http://pwp.netcabo.pt/>
- [4] JavaTV Technology. Consultado na Internet em 17 de novembro de 2003. <http://java.sun.com/products/javatv>

- [5] Centro de Pesquisa e Desenvolvimento em Telecomunicações. Consultado na Internet em 7 de novembro de 2003. <http://www.cpqd.com.br>
- [6] PersonalJava Technology. Consultado na Internet em 17 de novembro de 2003. <http://java.sun.com/products/personaljava>
- [7] Java 2 Micro Edition. Consultado na Internet em 20 de fevereiro de 2004. <http://java.sun.com/j2me>
- [8] SUN MICROSYSTEMS. JavaTV API Reference Implementation Porting Guide. 15 de novembro de 2000.
- [9] XleTVView. Consultado na Internet em 15 de dezembro de 2003. <http://sourceforge.net/projects/xletview/>
- [10] Home Áudio-Video Council. Consultado na Internet em 15 de janeiro de 2004. <http://www.havi.com>
- [11] Digital Áudio-Video Council. Consultado na Internet em 15 de janeiro de 2004. <http://www.davic.org>
- [12] Digital Vídeo Broadcasting. Consultado na Internet em 15 de janeiro de 2004. <http://www.dvb.org>
- [13] DeviceTop. Consultado na Internet em 17 de janeiro de 2004. <http://www.devicetop.com>
- [14] Espial Suíte. Consultado na Internet em 20 de fevereiro de 2004. <http://www.espial.com>
- [15] mhp4free. Consultado na Internet em 20 de fevereiro de 2004. <http://www.mhp4free.de>
- [16] Java 2 Micro Edition Community Forums. Consultado na Internet em 25 de fevereiro de 2004. <http://java.sun.com/j2me/community/forums/index.html>
- [17] CHAIN, et. al. @-gov.br: A Próxima Revolução Brasileira. Prentice Hall, São Paulo, SP, Brasil, 2004.