

Chapter

1

Datacasting e Desenvolvimento de Serviços e Aplicações para TV Digital Interativa

Valdecir Becker, Carlos Piccioni, Carlos Montez, Günter H. Herweg Filho

Abstract

This short course introduces the interactive digital TV services and applications development. The main concepts around interactive digital TV, and the technology overview, are showed. It encloses since datacasting concepts until the applications development tools.

Resumo

Este minicurso introduz o tema desenvolvimento de serviços e de aplicações para TV digital interativa. Traz os principais conceitos envolvidos no tema e oferece uma visão geral sobre os componentes e fases envolvidos no processo. Abrange desde conceitos como datacasting até as ferramentas usadas no desenvolvimento dos aplicativos.

1.1. Apresentação

O presente minicurso enfoca um tema que usualmente não faz parte dos currículos de graduação: a TV digital interativa. O desenvolvimento de serviços e aplicações para TV digital interativa é um tema relativamente novo, com pouca literatura consolidada a respeito.

O principal objetivo é oferecer uma visão geral sobre os conceitos adjacentes ao desenvolvimento de aplicações para essa nova mídia, além de abordar as tecnologias envolvidas. Para tanto, enfatizaremos os aspectos sociais e técnicos relevantes tanto para entender a TV digital e a concepção dos serviços e das aplicações, assim como os aspectos estritamente técnicos de desenvolvimento, programação e publicação. São abordados temas usuais no desenvolvimento de temas para televisão, as ferramentas necessárias para desenvolver e transmitir as aplicações e os temas teóricos que interligam os dois aspectos anteriores. O enfoque do minicurso é técnico, explicando conceitos e descrevendo as tecnologias envolvidas, porém sem desconsiderar a evolução

e o alcance social da TV e os conceitos e recursos de interatividade, fundamentais para entender o que pode vir a ser TV interativa brasileira.

1.2. Televisão e internet

A televisão brasileira tem mais de meio século de vida. Quando, no final da década de 1940, Assis Chateaubriand iniciou as negociações com a americana RCA para importar a televisão para o país, pouca gente sabia o que era essa nova tecnologia. O próprio Chateaubriand inicialmente a considerava uma forma de ganhar dinheiro, mas no dia da inauguração saudou “o mais subversivo instrumento de comunicação deste século” (Loredo 2000). No entanto, “os anunciantes demoraram muito a acreditar no novo veículo (...) impedindo que Chateaubriand tivesse o esperado retorno do seu investimento” (Loredo 2000).

Loredo (2000) lembra que quando começaram as transmissões da televisão brasileira ninguém sabia o que era fazer televisão. “Tudo era uma interrogação. Estávamos todos ali aprendendo de improviso. A Tupi, tanto no Rio como em São Paulo, era a universidade da televisão brasileira, abrindo um caminho promissor para as futuras gerações”.

Apesar dessa incógnita do que era fazer televisão, esse novo meio de comunicação evoluiu e se tornou na principal fonte de informação da população brasileira. Pesquisas mostram que mesmo as classes sociais mais privilegiadas e com acesso a outros meios de comunicação, como os estudantes universitários, têm na TV a principal fonte de informação (Agência Estado 2005).

Esse papel começou a se delinear ainda na década de 1970, com o fortalecimento da Rede Globo e com a criação do Padrão Globo de Qualidade, que uniformizou as programações de todas as afiliadas da rede (Bial 2005). Além disso, a televisão tem ainda uma peculiaridade inexistente nas outras tecnologias, a exceção do rádio: ela conversa com as pessoas, incentivando-as a agirem (Crocomo 2004). Isso a torna uma ferramenta de coação e de incentivo a ação, permitindo que a sociedade reaja a um determinado impulso visto na TV. Essa característica é fundamental para estimular o processo de alfabetização digital através da TV digital, pretendido pelo governo no Sistema Brasileiro de TV Digital (SBTVD).

Machado (2003) explica que a televisão é “... um dispositivo audiovisual através do qual uma civilização pode exprimir a seus contemporâneos os seus próprios anseios e dúvidas, a suas crenças e descrenças, as suas inquietações, as suas descobertas e os vãos da imaginação. (...) a televisão é e será aquilo que nós fizemos dela”.

Dessa forma podemos concluir que a televisão é reflexo da sociedade e de seus anseios. Além disso, a televisão tem uma relação dialética com as pessoas. Na medida em que a sociedade se informa através dela, passa também a ser influenciada. Entender esse conceito é extremamente importante quando desenvolvemos programas para a televisão, e agora, com a interatividade, aplicativos.

Por outro lado, a internet, com pouco mais de 10 anos de vida no país, também já tem uma história de mudanças e adaptações, tanto a realidade brasileira como às novas possibilidades de mercado, advindas com a evolução da tecnologia digital. Os primeiros conteúdos que foram oferecidos na internet nada tinham de novo. Eram

simples adaptações e transcrições do que já existia em meio impresso. Com o passar do tempo, a abordagem e a arquitetura da informação foram se adaptando as características do novo meio, fazendo surgir produtos multimídia e hipermídia só possíveis em meios interativos.

É nesse contexto que surge no Brasil a televisão digital interativa, com todas as adversidades e incertezas de qualquer nova mídia. Segundo Becker e Montez (2005), a televisão interativa é uma nova mídia porque quebra os paradigmas da televisão analógica de unidirecionalidade da informação e da passividade do telespectador. Através dessa nova tecnologia, é possível interagir com a programação sem necessitar de uma terceira mídia, como o telefone ou a internet. Através do canal de interatividade, a televisão passa a ter uma forma de contato direto com os telespectadores, que deixam de simplesmente assistir a televisão para passar também a usá-la, tornando-se um usuário. Isso torna o telespectador ativo, usando a TV como uma ferramenta potencial resolvedora de problemas (BBC 2004)

A TV digital oferece opções muito superiores a TV analógica (com circulação unidirecional da informação, conhecida atualmente) (Srivastava 2002). A capacidade de transmissão de dados (*datacasting*), que possibilita a interatividade em diferentes níveis, traz potenciais benefícios e facilidades atualmente disponíveis apenas na internet (Whitaker 2001). Isso tem impactos diretos na circulação da informação, que pode-se tornar muito mais abrangente, atingindo setores da sociedade atualmente excluídos (Castles, 2003). Além disso, provoca conseqüências ainda não delineadas em inúmeros outros temas, como direitos autorais, comércio eletrônico e novos requisitos de segurança dos sistemas. Também pode se tornar uma ferramenta poderosa de educação a distancia, entretenimento, notícias, governo eletrônico e novos serviços (Piccioni et. al. 2003), não disponíveis atualmente ou ainda sequer imaginados (Becker e Moraes 2003). Tudo isso tende a permitir o uso da TV como ferramenta para a aquisição de informação, substituindo a forma passiva com que o telespectador recebia informações transmitidas unidirecionalmente (Becker e Montez 2004).

Porém, a simples junção da linguagem da internet com a televisão não representa nenhum atrativo especial para o telespectador, por que o comportamento do telespectador difere drasticamente do internauta (Nielsen 2000). Enquanto que o primeiro apenas recebe informações de forma passiva, sem interferir, o segundo determina os rumos da navegação. Assim, a internet caracteriza-se como um meio muito rico em informações baseado em um alto grau de iniciativa e participação: as pessoas criam sua própria experiência através de um fluxo de cliques que seguem os hipertextos.

Segundo Becker e Montez (2005), a interatividade, resultante, principalmente, da junção das linguagens da TV e da internet, pode ser dividida em sete níveis. No último nível, o usuário se confunde com o atual emissor de conteúdo. Essa possibilidade foi amplamente estudada por Crocomo (2004), que conclui ser plenamente possível para o telespectador produzir e enviar conteúdo televisivo para as emissoras. O grau dessa participação e a agilidade vão depender dos recursos tecnológicos envolvidos. Para Pagani (2003), essas novas possibilidades só são possíveis pela convergência de tecnologias, basicamente a TV, o computador pessoal e o telefone. Relacionando Pagani (2003) e Crocomo (2004), é possível perceber uma relação clara entre a evolução

tecnológica e as novas possibilidades de conteúdo televisivo e formatos de interatividade na TV.

1.3. Panorama mundial das aplicações interativas

A televisão interativa começou a se delinear no final da década de 1990. Desde então inúmeras emissoras têm iniciado a transmissão de aplicações e de serviços interativos, com foco em diversos tipos de interatividade. De uma maneira geral, a interatividade sem canal de retorno é a que tem feito mais sucesso. Por outro lado, aplicações de e-mail, banco e comércio eletrônicos ainda não deram o resultado esperado.

Bernardo (2002) divide os serviços oferecidos pela televisão interativa em 10 tipos: EPG, Comércio eletrônico, Banco eletrônico, E-mail, Internet, Portais de televisão interativa, Aplicações transversais aos canais, Programas interativos, Publicidade interativa. A seguir faremos uma análise de cada um desses serviços.

- 1- EPG (*Electronic Program Guide*) – uma relação com a programação dos diversos canais disponíveis, com datas e horários de início e término de cada programa. Muito comum na TV por assinatura, onde o operador da rede disponibiliza os dados sobre todas as emissoras disponíveis naquele pacote. Na TV aberta é mais comum cada emissora oferecer o seu EPG. Não necessita de canal de interatividade, uma vez que a atualização é feita por broadcast. É o maior sucesso da TV digital.
- 2- Comércio eletrônico – é a possibilidade de fazer compras usando a televisão. O serviço não difere muito do que é conhecido atualmente na internet; apenas oferece a comodidade da televisão com canal de interatividade para a realização das compras. Comumente está aliado à publicidade interativa.
- 3- Banco eletrônico – também muito semelhante aos serviços oferecidos na internet, o acesso ao banco através da TV foi uma grande aposta na Europa, porém até agora com poucos resultados. Como o serviço é similar à internet, a estrutura necessária por parte dos bancos também é similar, precisando apenas ser replicada para esse novo meio de comunicação.
- 4- E-mail – o acesso ao correio eletrônico também está no nível das apostas na maioria dos países que já adotaram a TV digital. Da mesma forma como o banco eletrônico, a estrutura de servidores e clientes é similar a internet, necessitando apenas uma adaptação dos softwares clientes.
- 5- Internet – o acesso à internet sem a necessidade de um PC é uma das principais apostas da televisão digital. O uso de um set top box, uma linha telefônica e a televisão como monitor há anos seduz os idealizadores da TV digital. Porém, até o momento não há registros de um sucesso absoluto dessa tecnologia, mesmo porque nos países pioneiros na implantação da TV digital a internet é uma constante na vida das pessoas. O Brasil aposta fortemente nessa possibilidade para levar a internet às pessoas que não tem condições de adquirir um computador.
- 6- Portais de televisão interativa – também conhecidos como *walled gardens*, são portais que disponibilizam uma relação dos conteúdos oferecidos por uma

determinada emissora. Normalmente representam um resumo dos serviços e aplicações que a emissora detentora do canal oferece.

- 7- Aplicações transversais aos canais – são serviços oferecidos pela TV digital mas que não estão atrelados a nenhum canal especificamente, como acontece com os portais. Normalmente são aplicações de utilidade pública ou de governo eletrônico, sem nenhuma vinculação com os conteúdos audiovisuais transmitidos pelos canais normais de televisão.
- 8- Programas interativos – são aplicativos diretamente relacionados ao conteúdo audiovisual, completando-o ou muitas vezes, alterando-o. São os aplicativos que mais aproximam a televisão do conceito ideal de televisão interativa, onde o usuário está englobado na emissão da programação. São inúmeros os exemplos de sucessos de programas interativos, principalmente em transmissões esportivas, programas infantis, educação, programas jornalísticos e de entretenimento. O mais comum nesse tipo de aplicativo é o oferecimento de informações adicionais à programação, que completam a mesma ou que permitem que a ela seja alterada na origem.
- 9- Publicidade interativa – uma extensão do comércio eletrônico na TV, permite que os produtos anunciados sejam comprados na mesma hora. Na Europa esse tipo de anúncio começa a ter somas significativas nos faturamentos das empresas de comunicação, com interesse principalmente por produtos esportivos e alimentícios.
- 10- Jogos e outras aplicações – apesar de pode ser incluído nos itens seis ou sete, os jogos representam um conjunto de aplicações de enorme sucesso em algumas plataformas de televisão. Especialmente desenhados para a resolução da TV, especialistas apontam uma possível convergência entre os game consoles e os *set top boxes* com canal de interatividade e acesso a internet. Na Inglaterra surgiu um novo mercado de empresas totalmente dedicadas ao desenvolvimento desse tipo de aplicação.

De uma maneira resumida podemos identificar uma série de serviços que ainda não deram a resposta comercial esperada pelos seus idealizadores. Fazendo uma relação entre eles, pode-se perceber uma característica comum: estão presentes na internet e foram simplesmente portados para a televisão. A pergunta que não feita na Europa, onde a internet é plenamente disseminada, é: se tem na internet, por que alguém vai querer usar a televisão para fazer a mesma coisa? Vale lembrar que a TV tem recursos muito mais limitados que qualquer computador pessoal, como resolução de monitor, processamento, memória e armazenamento. Por outro lado, tem uma penetração muito maior no Brasil do que a internet, o que a torna um meio atrativo para levar informações adicionais a pessoas atualmente fora do mundo digital.

1.4. Definição de serviços

Na difusão terrestre no Brasil, atualmente, em uma faixa de frequência VHF ou UHF de 6 MHz são transmitidas as informações necessárias para que um televisor consiga reproduzir o que atualmente é conhecido como canal. Dessa forma, o que ocorre quando o telespectador troca de canal em televisores analógicos, é a sintonia de uma dessas

faixas de frequência e exibição das informações audiovisuais transportadas, em PAL-M, no ecrã (*display*) da televisão.

A televisão digital por sua vez, parte do seguinte princípio: digitalização da informação audiovisual para sua difusão. Essa abordagem traz uma série de vantagens em relação à televisão analógica. A primeira é com relação à possibilidade de se aplicar diversas técnicas de compressão de vídeo e áudio nas informações a serem transmitidas, ou seja, mais informação significativa pode ser difundida em comparação com a tecnologia atual.

A segunda grande vantagem é que, como toda informação é digitalizada antes da transmissão, qualquer outro tipo de dado digital além do vídeo e áudio também pode ser difundido. Dessa forma, é possível a difusão de aplicativos a serem executados no receptor de TV digital. Assim, as portas para a interatividade na televisão digital são abertas.

Outras vantagens também podem ser citadas: mecanismos de recuperação de erros podem ser aplicados, de forma que pequenas alterações em alguns *bits* causados por interferências no sinal possam ser detectadas e revertidas. Outra característica da TV digital é a consistência da imagem reproduzida no televisor: na TV digital não existe chuvisco ou “fantasmas” como na televisão analógica: a qualidade da imagem exibida permanece constante até que a ação de interferências eletromagnéticas gere um número de erros dentro dos limites possíveis de correção. Tal interferência em um sinal de TV analógico causaria os problemas citados acima. Porém, caso a interferência sobre o sinal na TV digital ultrapasse esses limites aceitáveis, a informação digital a ser tratada pelos decodificadores passa a ser indecifrável, causando geralmente o congelamento da imagem e a interrupção do som.

As principais etapas da construção do sinal digital são ilustradas de forma simplificada na Figura 1.1. A primeira etapa é a codificação de vídeo, áudio e dados segundo padrões adotados por um sistema de TV digital. Todos os sistemas de TV digital abertos (DVB, ATSC e ISDB) adotam o padrão *MPEG-2 Video* (ISO/IEC 13818-2, 1995). Esse padrão de codificação e compressão prevê taxas em torno de 4 a 6 Mbits por segundo para um fluxo de vídeo em qualidade SD, ou *Standard Definition*, que é a qualidade similar a de um canal de televisão analógica.

O fluxo de dados de saída de cada codificador é denominado de fluxo elementar. Dessa forma são produzidos fluxos elementares de áudio, vídeo e dados, que são direcionados ao multiplexador, responsável por multiplexar todos os fluxos elementares em um único fluxo de bits, denominado fluxo de transporte. Esse mecanismo de multiplexação também segue um padrão bem definido, conhecido como *MPEG-2 Systems: Transport Stream* (ISO/IEC 13818-1, 2000), e adotado pelos três sistemas abertos de TV digital.

A Figura 1.1 ilustra apenas um codificador de vídeo, um de áudio e um de dados alimentando o multiplexador. Porém, a limitação de quantos fluxos elementares podem ser multiplexados em um mesmo fluxo de transporte é determinada pela taxa máxima de entrada (*bitrate*) do modulador. Apesar da informação ser codificada digitalmente, há a necessidade de uma etapa de modulação para a difusão terrestre, via satélite ou cabo. Na modulação terrestre a ser adotada no Brasil, por exemplo, o fluxo de transporte é

modulado na mesma faixa de frequência de 6 MHz utilizada por um canal na TV analógica.

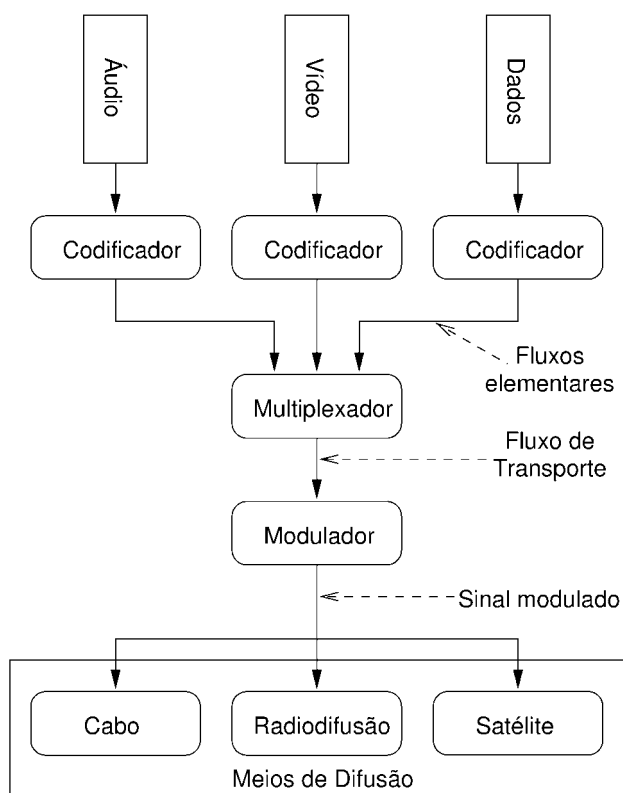


Figura 1.1: Etapas na construção do sinal digital.

Os parâmetros utilizados na modulação determinam a taxa máxima de transmissão, ou seja, a taxa de saída do multiplexador. Geralmente esses parâmetros buscam o melhor equilíbrio entre uma alta taxa de transmissão e uma significativa robustez no sinal. Em sistemas de difusão terrestre DVB, por exemplo, a taxa de transmissão geralmente utilizada é de 19.2 Mbits por segundo. Ou seja, como um fluxo elementar de vídeo MPEG-2 em qualidade SD pode ter uma taxa menor que $\frac{1}{4}$ dessa, e a taxa de um fluxo de áudio relacionado ao vídeo é relativamente menor, é possível a multiplexação de 4 fluxos de vídeo e 4 de áudio em um fluxo de transporte a ser modulado e difundido em uma mesma faixa de frequência que atualmente é ocupado por um único canal na televisão analógica.

Devido a essas características, a televisão digital traz novas nomenclaturas ao que atualmente é conhecido como canal na televisão analógica. Como, em uma mesma faixa de frequência de 6 MHz ocupado por um canal analógico no Brasil, é possível difundir um ou mais conjuntos de mídias audiovisuais, o conceito de canal por faixa de frequência torna-se obsoleto: é possível organizar logicamente fluxos de vídeo e áudio (e também dados) em um fluxo de transporte de forma que eles sejam reconhecidos como canais virtuais pelo *set top box*, nomenclatura definida pelo ATSC, ou como serviços, segundo o DVB.

Dessa forma, quando o receptor sintonizar e isolar determinado sinal em uma faixa de frequência de 6 MHz, ao invés de imediatamente exibir o que seria um canal na televisão analógica, ele deve seguir os passos inversos da Figura 1.1: demodular o sinal,

extraíndo dele o fluxo de transporte, e a partir dele determinar os serviços ou canais virtuais contidos no mesmo.

A seleção de qual serviço será exibido é realizada por parte do telespectador. Após determinado serviço ser selecionado, seus respectivos fluxos de vídeo e áudio (e em alguns casos dados) são direcionados aos respectivos decodificadores, sendo apresentados ao telespectador, que percebe o serviço de forma similar a um canal na televisão analógica.

1.5. Difusão de dados - *Datacasting*

Na seção anterior foram apresentadas algumas diferenças entre a televisão analógica e a digital. Uma delas é a possibilidade de transportar outros tipos de dados digitais no mesmo sinal utilizado para a transmissão de vídeo e áudio. A capacidade de difundir dados digitais que não sejam fluxos de áudio e vídeo nesse sinal é conhecida como *datacasting*, e é o pilar da interatividade na televisão digital. É o *datacasting* que permite às aplicações e demais dados correlacionados serem transmitidos.

1.5.1. MPEG-2 Systems: Transport Streams

Os sistemas abertos de televisão digital adotam o padrão *MPEG-2 Systems – Transport Stream* para a multiplexação dos fluxos elementares, como citado anteriormente. Para um melhor entendimento dos mecanismos de *datacasting* é necessário primeiro compreender como funciona este padrão de multiplexação.

Primeiramente, podem ser multiplexados dois tipos de estruturas de dados em um fluxo de transporte: os fluxos elementares empacotados (*PES - Packetized Elementary Streams*) e as seções. Os fluxos elementares empacotados nada mais são que os fluxos gerados pelos codificadores de vídeo e áudio (e em alguns casos, dados) divididos em pacotes. Cada pacote contém informações adicionais (*timestamps* de sincronismo, por exemplo) a respeito de uma unidade de informação que transporta, como por exemplo, um frame de vídeo ou determinado intervalo de áudio.

As seções, por sua vez, são estruturas que foram primeiramente utilizadas pelo *MPEG-2 Systems* para o transporte de tabelas contendo informações necessárias para a identificação correta de quais serviços estão presentes em um fluxo de transporte, assim como seus fluxos elementares. Tais tabelas são conhecidas como *PSI (Program Specific Information)*, e foram estendidas pelos sistemas de TV digital abertos de forma a incorporar mais informações a respeito dos serviços de um determinado fluxo de transporte. As seções possuem uma sintaxe mais simples em relação as *PES*, e não transportam informações temporais ou de sincronismo.

Tanto as seções como as *PES* são fragmentadas em pacotes de transporte no processo de multiplexação: esses pacotes de transporte são as unidades fundamentais de um fluxo de transporte. Cada pacote possui um tamanho fixo de 188 bytes. Os primeiros 4 bytes de cada pacote são destinados a um cabeçalho e os 184 bytes restantes para a carga do pacote. Em algumas situações, o cabeçalho de um pacote de transporte pode ser estendido de forma ocupar um número variável de bytes, podendo inclusive preencher os 188 bytes do pacote.

A informação mais importante do cabeçalho de um pacote de transporte é o PID (*Packet Identification*). Pacotes transportando determinado fluxo de vídeo, por exemplo, contêm o mesmo PID, assim como pacotes carregando seções de uma tabela PSI possuem em comum outro valor de PID. Dessa forma, o decodificador pode “filtrar” os pacotes de determinado PID de forma a reconstruir determinado fluxo elementar ou tabela.

A Figura 1.2 ilustra um fluxo de transporte contendo seções e PES multiplexadas. Os pacotes com PID igual a zero contêm seções que transportam a tabela conhecida como PAT, ou *Program Association Table*, uma das PSIs definidas pelo *MPEG-2 Systems*. A PAT é geralmente a primeira tabela alvo dos receptores: ela que contém as informações básicas sobre os serviços do fluxo de transporte.

Os pacotes que transportam a PAT sempre possuem o PID igual a zero. Isso se deve pelo seguinte motivo: quando o receptor começar a tratar um fluxo de transporte, a princípio não vai ter informações suficientes sobre quais e quantos serviços estão presentes no fluxo de transporte, assim como em que pacotes estão sendo transportados seus fluxos elementares. Dessa forma, o ponto de partida do *set top box* é filtrar os pacotes de PID igual a 0 de forma a encontrar a PAT e assim localizar as informações a respeito dos serviços.

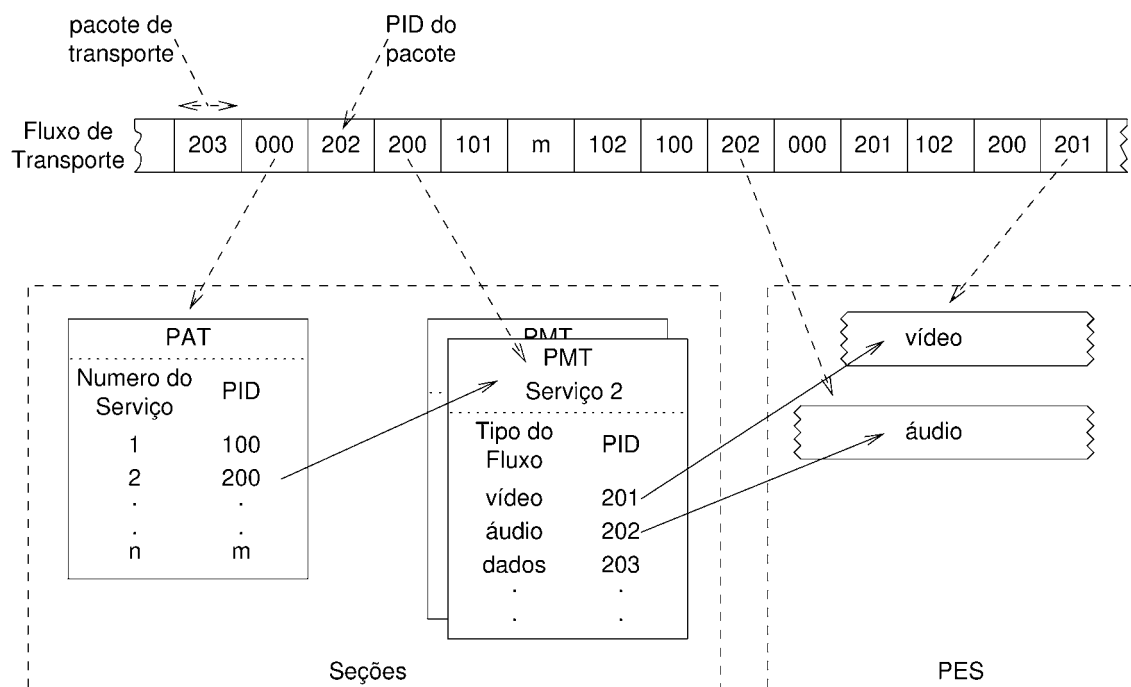


Figura 1.2: Exemplo de um fluxo de transporte contendo as tabelas PSI básicas, assim como fluxos elementares de vídeo e áudio.

Caso o serviço desejado seja o Serviço 2, como ilustra a Figura 1.2, o receptor passa a procurar por outra tabela, nos pacotes de PID 200, indicado pela PAT: a PMT, ou *Program Map Table*. Todo serviço possui uma PMT relacionada, responsável por indicar quais fluxos elementares fazem parte do serviço. Por exemplo, na Figura 1.2, fazem parte do Serviço 2 um fluxo de vídeo transportado nos pacotes de PID 201, um de áudio nos de 202, e por fim dados nos pacotes de PID 203. A partir desse ponto, o receptor passa a filtrar os pacotes com esses identificadores, reconstruindo os fluxos

elementares a partir da carga dos pacotes, e os direciona aos respectivos decodificadores.

1.5.2. Mecanismos de *Datacasting*

Como já apresentado, o *MPEG-2 Systems* é o padrão de fato de multiplexação dos sistemas de TV digital abertos. Dessa forma, os mecanismos de *datacasting* na TV digital fazem uso das facilidades e estruturas desse padrão: existem basicamente quatro mecanismos, ou áreas de aplicação (Tektronix, 2002): *Data Piping*, *Data Streaming*, *MPE* ou *Multiprotocol Encapsulation* e os Carrosséis (ETSI TR 101 202, 2003) (ATSC A/91, 2001), sendo esses últimos divididos em Carrosséis de Dados e Carrosséis de Objetos.

1.5.3. Data Piping

O *data piping* é o mecanismo mais simples de *datacasting*. Baseia-se na inserção de dados brutos diretamente na carga dos pacotes de transporte. A tarefa de um encapsulador de dados nessa abordagem é gerar pacotes de transporte, com um valor do PID conhecido pela aplicação destinatária, inserindo os dados nos 184 bytes da carga de cada pacote antes de enviá-los ao multiplexador.

O receptor, por sua vez, precisa apenas filtrar os pacotes com o PID determinado e extrair de sua carga os dados transmitidos. A vantagem dessa abordagem simplista do *data piping* reside no baixo custo de *hardware* e *software* necessários para a implementação, e é recomendado para soluções de *datacasting* proprietárias. Já a desvantagem do *data piping* é a falta de estruturas sintáticas mais poderosas para o encapsulamento de dados, problema resolvido nos outros mecanismos de *datacasting*.

1.5.4. Data Streaming

O *data streaming* por sua vez pode ser definido como uma área de *datacasting* onde dados são continuamente empacotados em PES antes de serem difundidos. Ao contrário do *data piping*, o *data streaming* permite *datacasting* síncrono devido ao uso das estampilhas de tempo no cabeçalho das PES. Essa é a sua grande vantagem em relação as outras formas de difusão de dados. O *datacasting* via PES, devido a essa capacidade de sincronização, é geralmente empregado em aplicações que requisitam forte acoplamento dos dados com outros fluxos elementares.

1.5.5. Multiprotocol Encapsulation

O *Multiprotocol Encapsulation*, MPE, é empregado no transporte de datagramas de protocolos de rede da terceira camada do modelo ISO/OSI (Tektronix, 2002). A estrutura utilizada para esse transporte é denominada de seção de datagrama, derivada das seções privadas, que por sua vez são seções de propósito geral do *MPEG-2 Systems* (de forma que outros tipos de dados além de tabelas possam ser difundidos por essas estruturas).

Sistemas abertos de TV digital otimizaram a sintaxe das seções de datagrama de forma a facilitar o transporte de datagramas IP. Dessa forma, é facilitado o uso do canal de difusão para oferecer serviços de internet (*downstream*).

Staffans (2004) aponta o *IP datacasting* através do MPE como solução para diversas aplicações utilizando o canal de difusão da televisão digital, principalmente no que se refere à recepção móvel.

A grande diferença com relação aos dois meios de *datacasting* apontados até então, o *data piping* e o *data streaming* reside no fato de que no MPE do tipo *IP datacasting* os dados podem ser facilmente endereçados a um *set top box (unicast)* ou a grupos de terminais de acesso (*multicast*). Não que isso não seja possível com os dois tipos anteriores, mas o endereçamento, nesse último caso, é uma característica intrínseca ao protocolo IP (e conseqüentemente, ao *IP datacasting*).

Outra diferença com relação ao *data streaming* através das PES é que as seções de datagrama não contêm informações temporais. Dessa forma, qualquer tentativa de sincronização deve ser de responsabilidade da implementação.

1.5.6. Carrosséis

Os carrosséis, por fim, são protocolos de difusão de dados definidos pelo padrão DSM-CC (*Digital Storage Media, Command and Control*) (ISO/IEC 13818-6, 1996), a sexta parte do conjunto de especificações MPEG-2. O DSM-CC foi originalmente desenvolvido com o objetivo de fornecer funções semelhantes às presentes em aparelhos de vídeo cassete para o controle de fluxos de áudio e vídeo de um fluxo de transporte (Fairhurst, 2005).

Posteriormente o DSM-CC foi estendido e dividido em várias partes, com o intuito de fornecer, entre outras, funções como seleção, acesso e controle de fontes distribuídas de vídeo e suporte para a transmissão de dados através de fluxos de transporte (Schwalb, 2003) (Morris, 2005).

Dividido em várias partes, o DSM-CC pode ser visto como um conjunto de ferramentas, apresentando várias configurações e funcionalidades. Dessa forma, vários sistemas, como os de televisão digital, adotam apenas algumas de suas partes. Os mecanismos de carrossel de dados e carrossel de objetos do DSM-CC são dois dos mecanismos mais utilizados para a difusão de dados nos sistemas DVB e ATSC. Dessa forma, serão apresentados em mais detalhes nas próximas seções.

O protocolo denominado carrossel de dados foi desenvolvido com o intuito de possibilitar a difusão de dados, de forma periódica, para *set top boxes*. A idéia básica desse protocolo é de módulos de dados difundidos ciclicamente, de modo que, quando o receptor necessitar determinado módulo, deve apenas aguardar o instante de sua próxima repetição no fluxo de dados.

Um módulo, em um carrossel de dados, é definido como um item simples e completo de dados. Por exemplo, um único arquivo de texto ou vários outros arquivos de texto podem ser definidos como um módulo. Cada módulo é dividido em um ou mais blocos, e cada bloco é inserido na carga de uma estrutura denominada *Download Data Block*, ou DDB, que é encapsulada em seções privadas.

Os carrosséis de objetos baseiam-se nas definições dos carrosséis de dados, porém passam a tratar a informação na forma de objetos. Para a difusão de dados, dois objetos são de maior importância em um carrossel deste tipo: objetos do tipo arquivo e do tipo diretório. Com esses dois tipos de objetos é possível formar um sistema de

arquivo simples. Dessa forma, o *set top box* pode acessar arquivos de um sistema de arquivos que está sendo difundido em um carrossel de objetos como se os mesmos estivessem disponíveis localmente. Devido a essa característica, o carrossel de objetos pode ser definido como um sistema de arquivos de difusão.

Os carrosséis de objetos são a escolha natural para a difusão de dados delimitados armazenados na forma de arquivos, como ocorre na maioria dos sistemas de arquivos atuais. Dessa forma, também é a solução mais empregada na difusão de aplicações assim como seus recursos.

1.6. Concepção de aplicações

Como já foi visto, no momento em que a TV interativa quebra os paradigmas da unidirecionalidade da TV e da inércia do telespectador, ela não representa uma simples evolução tecnológica, mas uma nova mídia, com problemas e especificidades diferentes das da TV analógica (Becker e Montez 2005). Uma das principais questões referentes a esse tema é o conteúdo desse meio, que deve passar por um processo de adaptação para conviver com os novos serviços oferecidos graças a inúmeras revoluções e convergências tecnológicas, além da digitalização da transmissão dos sinais audiovisuais.

Mesmo após quase oito anos da implantação dos primeiros canais digitais de radiodifusão na Inglaterra e parte da Europa, poucas emissoras têm programas interativos na grade de programação. Programas estes ainda com uma interatividade restrita apenas a reação a algum estímulo prévio, com pouca iniciativa por parte do telespectador (Gawliniski 2004). Esse é um dos principais motivos pela pouca aceitação inicial da TV digital nos países em que ela foi implantada, levando com que as pessoas não façam distinção entre as novas potencialidades da TV digital interativa e não vejam motivos para adquirir um *set top box* ou uma TV digital (Maclin 2001).

Como a programação interativa ainda é uma incógnita, com apostas que vão desde a crença de que a TV interativa não será mais do que um consumo interativo, até utopias que prevêem a total democratização ou desregulamentação do setor de radiodifusão (Becker et. al. 2004) um estudo mais aprofundado sobre a participação do telespectador na programação da televisão brasileira pode contribuir para clarear alguns pontos da programação dessa nova mídia. Além disso, torna-se pertinente analisar as iniciativas das emissoras no que se refere a participação direta e indireta do telespectador na programação (Becker e Montez 2004). Isso pode ajudar a delinear que programas a população gostaria de ver na televisão. A análise sobre a aceitação por parte do público pode ser feita relacionando o tempo em que os programas ficaram no ar e se tiveram uma segunda edição ou segunda temporada. Afinal, para um programa ficar no ar na televisão brasileira, exceção feita à TV pública, pouco representativa do ponto de vista da audiência, é preciso uma boa receita publicitária. E os anunciantes só apóiam os programas que têm aceitação popular.

Apesar dessas indefinições, é possível se espelhar no que deu certo nos países pioneiros na implantação da TV digital e na história da TV brasileira para definir o escopo e o alcance das primeiras aplicações a serem desenvolvidas no país. Tomando como base que na Europa os principais sucessos estão nas aplicações especialmente desenvolvidas para televisão, e não simplesmente copiadas da internet, e que no Brasil

quase 90% da população não tem acesso à internet, as aplicações e serviços da TV digital devem estar voltados para públicos alvo específicos.

Apesar de Crocomo (2004) dizer que há um preceito jornalístico segundo o qual a televisão é para todos, a maioria dos programas tem público alvo bem definido, que pode variar entre faixa etária, poder aquisitivo e formação intelectual e cultural. Uma das poucas exceções a essa segmentação da programação está nos programas jornalísticos, principalmente os telejornais, que tem uma abrangência maior.

Esse dado precisa ser levado em consideração quando pensarmos em aplicações e serviços para a TV digital brasileira. Pelos estudos em andamento no contexto do SBTVD, o país deverá ter uma série de modelos de *set top boxes*, variando inclusive os recursos disponíveis. Para a alfabetização digital o governo exige que tenham no mínimo interatividade local, o que quer dizer que pode ter *set top boxes* no mercado sem canal de interatividade. Isso tem impacto direto no desenvolvedor de aplicações. A mesma aplicação deve ser atrativa tanto para quem tem canal de interatividade de banda larga, linha discada ou mesmo para quem sequer tem canal de interatividade.

Portanto, as aplicações devem ser desenvolvidas com público alvo específico, que varia principalmente segundo o poder aquisitivo de comprar um *set top box* mais potente e de manter um canal de interatividade. Isso falando em aplicações. Se estendermos o raciocínio para o desenvolvimento de serviços, precisamos englobar ainda as pessoas sem *set top box*, ou seja, que ainda não tem TV digital, que assistem a televisão pelo sinal analógico.

Tendo visto isso, cabe ainda ressaltar a diferença entre TV interativa e internet na TV. Como vimos, em todos os momentos em que se fala em TV, fala-se em áudio e vídeo. E até o momento não há nada que indique que na TV interativa vá ser diferente. Dessa forma, não existe televisão sem vídeo. A TV interativa agrega inicialmente (não podemos nos esquecer da evolução das linguagens e da própria tecnologia) alguns recursos interativos ao vídeo, que passam a interagir com ele. São as aplicações correlacionadas com o vídeo, que não tem razão de existir senão atreladas a um determinado programa televisivo. Numa evolução desse tipo de aplicação, espera-se que estas possam se tornar cada vez mais relevantes, com a interatividade e a autonomia sobre o vídeo aumentando consideravelmente.

Por outro lado, as aplicações aqui chamadas de internet na TV são aquelas que atualmente existem da rede mundial de computadores e são simplesmente adaptadas à televisão. São desconectadas do vídeo, existindo, portanto, independente dos programas audiovisuais em transmissão. Como vimos na seção 1.3, esse tipo de aplicação não obteve grande sucesso na Europa.

Identificados o contexto da concepção dos serviços e aplicações interativos, apresentamos agora algumas características consideradas essenciais em cada serviço e aplicação para TV digital interativa:

1. Realidade nacional – o desenvolvedor de aplicações deve compreender que existem *set top boxes* com diferentes especificidades e pessoas que vêem televisão pelo sinal analógico.
2. Usabilidade avançada – a sociedade brasileira é pouco alfabetizada, com pouquíssimo acesso a computadores pessoais e a internet. Isso requer uma

interface simples e intuitiva, usando a linguagem atual da televisão, conhecida e compreendida por todos.

3. Compreensão imediata – a aplicação deve ser imediatamente compreendida como tal pelo usuário, que não a deve confundir com GC ou qualquer outro recurso audiovisual. Quando a mesma for sinalizada ou se aparecer na tela automaticamente, o usuário deve ter ciência de que se trata de uma aplicação interativa e que ele tem a liberdade de interagir ou não.
4. Público alvo específico – assistir televisão é uma experiência coletiva; interagir com a televisão é uma experiência individual. Apenas uma pessoa tem o controle remoto na mão. Além disso, nem todo mundo que assiste a televisão naquele momento está disposto a interagir.
5. Compreensão do público alvo – além de identificar o público alvo do serviço ou da aplicação, é extremamente importante entender o que esse público espera de uma aplicação interativa, e caso não espere nada ou desconheça essa nova funcionalidade, o que o atraia à interatividade.
6. Agregação de valor à informação – a aplicação interativa deve trazer uma experiência a mais para o usuário, não possível com o simples uso do áudio e do vídeo.
7. Relevância e pertinência – o conteúdo da aplicação deve estar condizente com o serviço todo, acrescentando informações a este, sem destoar do tema central apresentado pelo audiovisual.
8. Curiosidade do usuário – o usuário da televisão digital deve se sentir incentivado a interagir, tendo certeza de que será uma experiência agradável sob todos os aspectos. Para isso, um dos recursos muito usados é o despertar da curiosidade no telespectador, agora usuário, que dessa forma se sente incentivado a participar.
9. Foco no mercado – a televisão analógica brasileira é custeada pela publicidade. Nada indica que a digital seja diferente. Portanto, a aplicação deve ser interessante o suficiente para atrair anúncios para ela própria ou para o serviço no qual está embutida.
10. Ineditismo – a aplicação ou o serviço não precisam totalmente inéditos, uma vez que atingir esse grau de qualidade em todas as aplicações é extremamente difícil. Em todo caso, deve tender para tal, pelo menos em cada situação em que a aplicação ou o serviço forem transmitidos, até atingir uma receita de sucesso e manter o programa no ar por mais tempo.

1.7. Arquitetura de *set top boxes* e *middlewares*

As seções 1.4 e 1.5 deram os subsídios básicos para o entendimento das tecnologias de base para a difusão de dados na TV digital. O *datacasting* serve como base para interatividade no momento em que dados difundidos possam ser usados por aplicações residentes no *set top box*, ou possam ser as próprias aplicações difundidas. Nesta seção serão apresentados os conceitos básicos necessários para o início do desenvolvimento de aplicações e de como estas se inserem na arquitetura dos sistemas de TV digital.

Um *set top box* possui arquitetura semelhante à de um microcomputador. A princípio, possui uma menor capacidade de processamento e armazenamento, porém conta com componentes de *hardware* específicos para um ambiente de televisão digital, como sintonizador, demodulador, demultiplexador, decodificadores de mídia, entre outros. Alguns inclusive podem ser implementados por software. A Figura 1.3 ilustra a arquitetura básica de um *set top box* sem canal de retorno (ETSI ES 201 812, 2003).

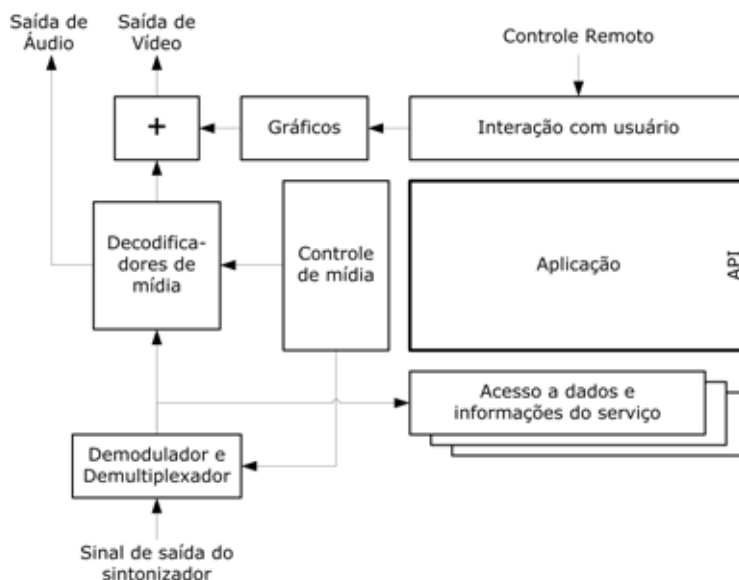


Figura 1.3: Arquitetura básica de um *set top box*.

O surgimento desses receptores de TV Digital com capacidade de processamento fez surgir no mercado uma nova atividade: desenvolvimento (ou programação) de aplicações interativas. Os desenvolvedores desse novo tipo de aplicação, a princípio, teriam de resolver um sério problema. Como não é razoável supor-se que todos os receptores digitais ou *set top boxes* são exatamente iguais, haveria necessidade de implementar diversas versões de aplicações interativas. Além disso, essas aplicações, de alguma forma, precisariam ser difundidas simultaneamente e cada *set top box* selecionaria a aplicação adequada para ser executada nele. Essa abordagem atrapalharia o desenvolvimento de novos tipos de *set top boxes*, e seria de difícil implementação.

Felizmente, não há necessidade de empregar essa abordagem ineficiente. Sistemas de TV digital interativa adotam uma camada de *middleware* que esconde as peculiaridades e complexidades não apenas do *hardware*, mas também do sistema operacional, *device drivers*, *softwares* e *hardwares* responsáveis pela decodificação do sinal (Figura 1.4).

Um *middleware* intermedeia toda a comunicação entre a aplicação e o resto dos serviços oferecidos pelas camadas inferiores. O uso do *middleware* facilita a portabilidade das aplicações, permitindo que sejam transportadas para qualquer receptor digital (ou *set top box*) que suporte o *middleware* adotado.

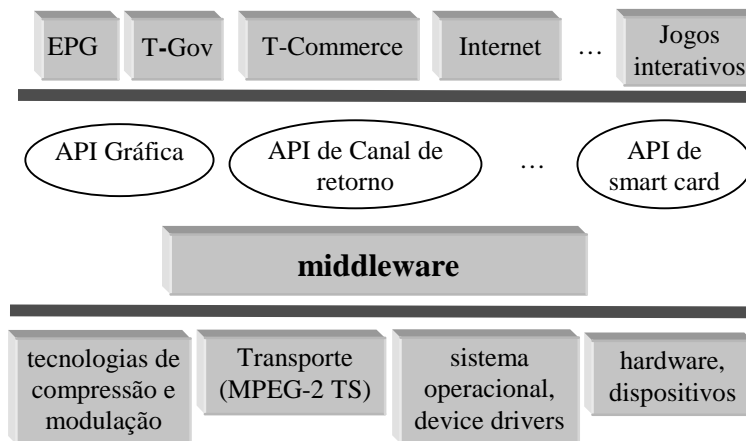


Figura 1.4: Middleware como uma plataforma de programação de aplicações.

Apesar de ser representado na Figura 1.4 como apenas um bloco, um *middleware* costuma ser formado por uma pilha de *softwares*. Esses *softwares* oferecem APIs (*Application Program Interface* – Interface de Programação de Aplicações, representado na Figura 1.3) padronizadas e, juntos, se constituem em uma plataforma de programação para aplicações interativas.

No final dos anos 1990, quando os principais *middlewares* começaram a ser definidos, várias bibliotecas e APIs para aplicações multimídia e para dispositivos embutidos (*embedded*) já existiam. Esses padrões já existentes foram prontamente adotados pelos *middlewares* recém-criados. Dentre estes, destacam-se HAVi e DAVIC. Além disso, simultaneamente houve uma iniciativa da SUN em propor uma biblioteca Java que facilitasse a adoção desse padrão em ambientes de TV digital. O JavaTV, resultado desse esforço, também foi adotado pelos principais *middlewares* de TV digital. Esses padrões – JavaTV, HAVi e DAVIC – que ajudam a compor as APIs dos *middlewares* de TV digital, são descritos a seguir de forma sucinta.

1.7.1. JavaTV

Um programa Java que pode ser difundido, recebido, executado remotamente em *set top box*, em conformidade com a biblioteca JavaTV, recebe o nome de *Xlet* (Figura 1.5).

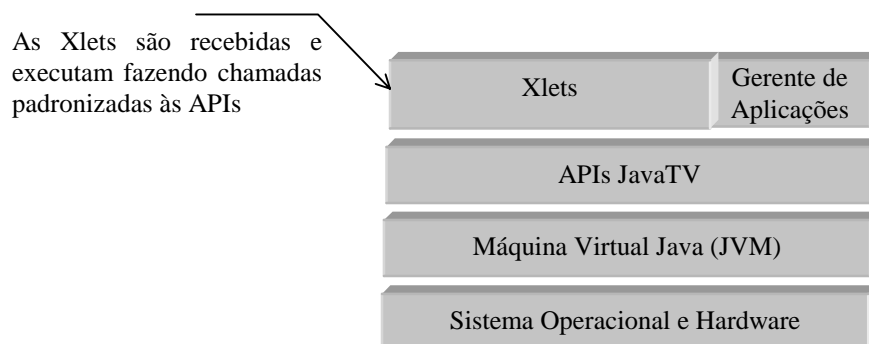


Figura 1.5: Xlets, Gerente de Aplicações e máquina virtual Java.

Para manipular esse tipo de aplicação, o JavaTV provê a API *javax.tv.xlet*. Outras APIs importantes do JavaTV são: *javax.tv.service.guide* – com facilidades para suportar EPGs; *javax.tv.service.selection* – para selecionar serviços ou programas de televisão; e *javax.tv.graphics* – para lidar com gráficos.

Buscando controlar as *Xlets*, cada *set top box* possui um Gerente de Aplicações (*Application Manager*) instalado. Um gerente de aplicações lida com os estados de cada *Xlet*, permitindo iniciar sua execução, destruir, pausar e retomar sua execução. Esses estados são necessários, pois uma aplicação pode ser pausada momentaneamente se esta for ocultada por um vídeo de TV; ou ainda pode ser destruída caso o usuário troque de canal. A *Xlet* precisa ser notificada quando seu estado muda (por exemplo, quando é pausada) e, assim, pode lidar com seus recursos (ex. liberar memória), se desejar. Mais sobre *Xlets* será visto na seção 1.8.1.

1.7.2. HAVi

O HAVi (*Home Audio Video Interoperability*) é um padrão especificado por uma organização homônima, formada por grandes companhias de produtos de consumo audiovisuais. Foi originalmente criado para prover um padrão para interoperabilidade entre dispositivos e equipamentos digitais audiovisuais. O objetivo principal era o de facilitar o oferecimento mútuo de serviços entre esses dispositivos, tais como TVs, DVD *players* e DV *camcorders*.

Esse padrão foi adotado pelos sistemas de TV digital, principalmente por prover um conjunto APIs com suportes específicos para televisão, tais como funções para lidar com ecrãs e gráficos de TV. Esses componentes são resumidamente botões, campos de textos, campos de entrada de textos, dentre outros, que facilitam o desenvolvimento das interfaces interativas das aplicações.

1.7.3. DAVIC

DAVIC (*Digital Audio Visual Council*), por sua vez, foi uma associação criada em 1994, que teve uma duração de apenas cinco anos, mas que conseguiu agregar 222 companhias espalhadas em 25 países.

A API DAVIC fornece acesso a funcionalidades específicas de um ambiente de TV digital (como filtro de seções, por exemplo), além do gerenciamento de recursos escassos nesse ambiente (como componentes de *hardware* específicos). As extensões DVB foram criadas de forma a suprir as necessidades não cobertas por essas outras APIs e detectadas como necessárias no MHP (*Multimedia Home Platform*). Algumas especificações do DVB foram fortemente influenciadas pelo DAVIC, tais como o DVB-C (padrão para transmissão de TV a cabo) e DVB-RCC (canal de retorno por cabo)

1.7.4. MHP, OCAP, DASE e ARIB

No final dos anos 1990, o grupo DVB começou a especificar seu padrão de *middleware*, que, em 2000, deu origem à plataforma MHP 1.0 e, abril de 2001, MHP 1.1. O MHP busca oferecer um ambiente de TV interativa aberto e interoperável, para receptores e *set top boxes* de TV digital. Seu ambiente de execução é baseado no uso de uma máquina virtual Java e APIs JavaTV. Essas APIs possibilitam aos programas escritos em Java o acesso a recursos e facilidades do receptor digital de forma padronizada. Uma

aplicação DVB usando API Java é denominada aplicação DVB-J. A especificação MHP 1.1 introduziu a possibilidade de usar uma linguagem de programação semelhante ao HTML denominada DVB-HTML.

Aplicações DVB-J e DVB-HTML possuem a capacidade de fazer *download* de aplicações interativas, através de um canal de interatividade; armazenar aplicações em memória persistente (ex. disco rígido); acessar leitores de *smart cards*; e controlar aplicações de internet, tais como navegador *web*.

O OCAP (*OpenCable Applications Platform*) é um *middleware* voltado para sistemas de TV a cabo norte-americano. Sua primeira versão foi liberada em 2001, e seguida pela versão OCAP 2.0, em abril de 2002. Nessa última versão houve uma preocupação em definir APIs mais próximas possível das especificações MHP.

O ATSC, sistema norte-americano, desenvolveu o DASE como sua camada de *middleware*. De forma similar ao MHP, o DASE adota uma máquina virtual Java como mecanismo que facilita a execução de aplicações interativas. Também de forma similar ao MHP, o DASE permite o uso de linguagens declarativas, usadas na *web*, como HTML e JavaScript

O *middleware* do sistema japonês ISDB é padronizado pela organização ARIB. Esse *middleware* é formado por alguns padrões, como o ARIB STD-B24 (*Data Coding and Transmission Specification for Digital Broadcasting*) que define uma linguagem declarativa denominada BML (*Broadcast Markup Language*). Essa linguagem, baseada na linguagem padrão de serviços web XML (*Extensible Markup Language*), é usada para especificação de serviços multimídia para TV digital.

Outra especificação do *middleware* é o ARIB-STD B23 (*Application Execution Engine Platform for Digital Broadcasting*), que é baseada na especificação MHP, através da adoção do padrão GEM, que é mostrado na seqüência.

1.7.5. GEM

A necessidade de padronização é uma questão recorrente em sistemas computacionais; e em sistemas de TV Digital não é diferente. Padronização facilita a portabilidade e interoperabilidade entre componentes do sistema, reduzindo os custos de sua implantação no mercado. Como exemplo, a adoção padronizada de MPEG-2 TS na camada de transporte, em sistemas de TV digital, não só possibilitou o emprego de componentes de *hardware* e *software* já existentes, como também facilitou o desenvolvimento, testes e depuração de novos componentes. Desenvolvedores de sistemas que utilizam tais componentes também acabam se beneficiando com a redução de seus riscos, já que muitos destes componentes foram usados e testados ao longo de anos em aplicações multimídia.

Além disso, caso essa padronização fosse estendida a outros sistemas de TV digital existentes, em um âmbito mundial, esses benefícios tenderiam a se ampliar devido à redução de custos, causada pela maior escala de produção e do aumento da concorrência. Com a camada de *middleware* não é diferente. A indústria de TV é geralmente uma indústria globalizada, e a possibilidade de desenvolver aplicações interativas e conteúdo que possam ser portados facilmente em sistemas de diferentes países é um forte atrativo.

A partir da década de 1990, com o surgimento dos primeiros sistemas de TV Digital, em um primeiro momento houve um movimento no sentido de cada organização desenvolver o seu próprio padrão de *middleware* (MHP, DASE, ARIB, OCAP, etc.). Contudo, a identificação de várias características em comum, tal como a adoção de APIs de JavaTV, HAVi, etc., e o reconhecimento das vantagens de uma padronização entre os *middlewares* levou a diversos esforços no sentido de harmonizar esses padrões.

Por ser o padrão de *middleware* mais maduro, o MHP foi escolhido como referência para essa tentativa de estabelecimento de conformidade. Contudo, o MHP possui diversas especificidades do DVB e, por conseguinte, sua utilização pura e simples não é adequada em sistemas de outros países que adotam, por exemplo, ATSC ou ARIB. Por conseguinte, GEM (*Globally Executable MHP*) (ETSI TS 102 819 – 2005) foi criado para facilitar a criação de especificações de *middlewares* baseados em MHP.

A primeira versão da especificação do GEM, foi publicada em fevereiro de 2003 (GEM 1.0) e atualmente existe uma versão “rascunho” que foi publicada em maio de 2005 (TS 102 819 V1.2.1, 2005).

As especificações GEM listam partes das especificações MHP que podem ser encontradas em um mercado específico. Contudo, ela pode ser substituída onde necessário, desde que sua funcionalidade seja equivalente à original (equivalentes funcionais). Isso é necessário devido às características específicas de cada sistema. Por exemplo, as tabelas de sinalização do ATSC são incompatíveis com as do DVB. Outro exemplo citado de equivalente funcional é o fato do ARIB B.23 usar carrossel de dados do DSM-CC em vez de carrossel de objetos como é feito no MHP.

É importante observar que GEM não é simplesmente um subconjunto do MHP. Cada padrão de *middleware* que usa GEM, o faz de forma personalizada, adicionando extensões e, também, colocando restrições na especificação básica do GEM (Figura 1.6).

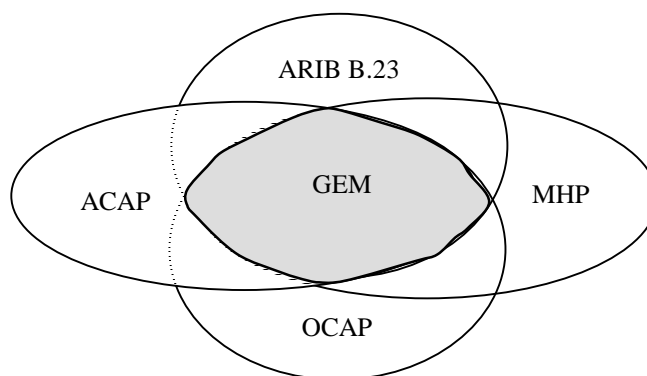


Figura 1.6: Relacionamento do GEM com os outros *middlewares* (ETSI TS 102 819 2005).

A Figura 1.6 também mostra o ACAP – um padrão criado com objetivo de tentar harmonizar os sistemas norte-americanos de transmissão terrestre (DASE) e por cabo (OCAP). Essa busca de harmonização culminou na adoção do GEM como base de implementação desses *middlewares*.

1.8. Desenvolvimento de aplicações para TV digital

Como vimos nos capítulos anteriores, existem várias arquiteturas de *middleware* predominantes no mercado mundial, e cada uma delas com suas próprias peculiaridades. Infelizmente não existe um padrão mundial no que diz respeito ao desenvolvimento de aplicações para televisão digital. Para cada arquitetura uma nova forma de estruturação e, às vezes, um novo padrão de desenvolvimento.

No entanto, as arquiteturas abordadas neste capítulo têm muito em comum, pois todas têm a mesma base tecnológica: Java e MPEG, e é nelas que focaremos a análise e explicação do desenvolvimento dos aplicativos.

As aplicações para televisão digital podem ser desenvolvidas usando linguagem Java ou HTML, porém a segunda é opcional e não suportado por alguns *set top boxes*. Por este motivo os exemplos deste capítulo serão exclusivamente aplicativos escritos em Java. Todavia, o fato da aplicação ser escrita nesta linguagem, não implica que ela seja considerada como uma aplicação Java normal. Aplicações para TV digital usam um conjunto de APIs diferente, por vezes mais enxutas e com modificações, e por outras com APIs exclusivas para desenvolvimento para a televisão digital.

No contexto da arquitetura de software do ambiente em que a aplicação executa, podemos destacar três elementos que participam ativamente do processo de execução de uma aplicação. Todo sistema que é capaz de executar aplicações para TV digital, deve prover um gerenciador de aplicações, que é responsável por detectar e instanciar uma nova aplicação emitida pelo operador, e disponibilizar para aplicação um *contexto*, que é outro elemento da arquitetura e que será detalhado mais adiante. O terceiro elemento da arquitetura é a própria aplicação, o foco deste capítulo.

1.8.1. Introdução a *Xlets*

Uma aplicação para televisão digital é chamada de *Xlet*. Ela forma o conceito básico de aplicação em todos os sistemas baseados no JavaTV, como os abordados por este capítulo. A idéia básica de uma *Xlet* é funcionar como uma máquina de estados em cima da máquina virtual que a está executando. Desta maneira duas *Xlets* podem coexistir na mesma máquina virtual sem que uma interfira na outra, bastando para isso um gerenciamento dos estados de cada uma, e é neste ponto que o *contexto* se torna importante neste ciclo.

As *Xlets* implementam a interface *Xlet* que provê um ciclo de vida para a aplicação através da implementação de seus métodos. Cada método da interface provê uma transição a um estado da aplicação. Os métodos da interface a serem implementados, *initXlet*, *startXlet*, *pauseXlet* e *destroyXlet*, são detalhados mais adiante. Existem quatro estados principais em que uma *Xlet* pode estar: *carregada*, *parada*, *em*

execução e destruída. Os estados *não carregada* e *inválido* também são considerados em alguns contextos.

A figura 1.7 mostra a máquina de estados de uma *Xlet*, com seus possíveis estados e os métodos invocados para que ocorra a transição entre eles.

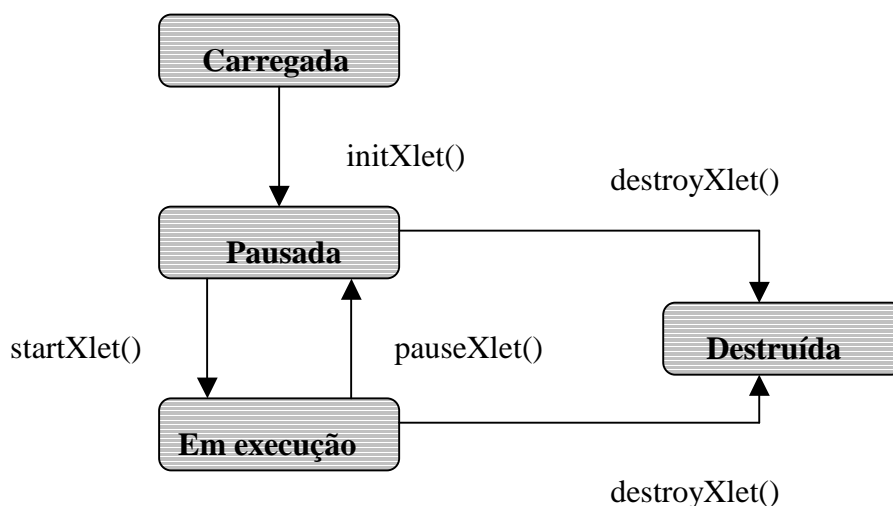


Figura 1.7: Ciclo de vida da *Xlet*.

Inicialmente o receptor recebe a sinalização de uma nova aplicação, o ciclo de vida da *Xlet* então começa e é caracterizado como *não carregada*. A partir deste ponto, o gerenciador da aplicação, que está no receptor, deve detectar a classe principal e criar uma instância dela. Feito isso, o estado passa a ser caracterizado como *carregada*. Uma aplicação que já foi carregada está apta a ser inicializada pelo método *initXlet*, que executa suas tarefas e deixa a aplicação no estado *pausada*. Neste ponto, a *Xlet* já está pronta para ser executada, pois já carregou e inicializou todos os elementos que eram necessários para começar a executar. O método *startXlet* é chamado e a aplicação de fato começa a funcionar. O método *destroyXlet* é chamado toda vez que a aplicação termina suas tarefas.

Veamos a interface *Xlet*:

```
Public interface Xlet{
    public void initXlet( XletContext ctx );
        throws XletstateChangeException;
    public void startXlet();
        throws XletstateChangeException;
    public void pauseXlet();
    public void destroyXlet( Boolean incondicional);
        throws XletstateChangeException;
}
```

startXlet: Método chamado para fazer a aplicação executar sua função principal, este método deve implementar todos procedimentos necessários para que a aplicação entre no estado *em execução*. É durante este estado que a *Xlet* gerencia seus componentes internos, manipula os objetos previamente criados e está apta a escutar todos os eventos internos e externos a aplicação.

destroyXlet: Neste método devem ser implementados todos os procedimentos necessários para que a *Xlet* termine sua execução de forma correta. Geralmente é usado para remover componentes gráficos da tela, liberar recursos com E/S de dados ou salvar dados pertinentes. Sua condição somente é satisfeita se o parâmetro condicional for verdadeiro.

pauseXlet: Sinaliza a aplicação para esta parar de prover seu serviço e entrar para o estado de *pausa*.

initXlet: Método com função parecida a de um construtor de classe, toda instância de objetos ou variáveis podem ser inicializados dentro do escopo deste método, um container pode ser criado para exibir interfaces gráficas ou conexões de rede necessárias podem ser abertas, em outras palavras, prepara a aplicação para ser executada. *InitXlet* é chamado somente uma vez no ciclo de vida da *Xlet*. É neste método que a aplicação recebe como parâmetro o *contexto*.

1.8.2. O XletContext

O *contexto* implementa a interface *XletContext* que faz parte do pacote *Xlet*, e de uma maneira geral serve para isolar a aplicação do resto da máquina virtual.

Entre outras, o contexto permite a *Xlet* descobrir informações a respeito do ambiente de execução através do método *getXletProperty()*. É possível para o *contexto* acessar propriedades do sistema em que a aplicação está executando. Grande parte dessas propriedades podem ser usadas por aplicações desenvolvidas em JavaTV, MHP e OCAP.

Vejamos a interface *XletContext*:

```
Public interface XletContext{
    public static final String ARGS = "javax.tv.xlet.args";
    public void notifyDestroyed();
    public void notifyPaused();
    public void resumeRequest();
    public Object getXletProperty( String key );
}
```

Também é possível ao contexto notificar ao *middleware* sobre alterações no estado da *Xlet*, os métodos do contexto que tratam desta questão são importantes para o caso de haver mais de uma *Xlet* executando ao mesmo tempo na máquina virtual do sistema, *notifyDestroyed()* e *notifyPaused()* informam ao receptor que controla a aplicação, que ela deseja terminar e parar a execução, respectivamente, assim, o controlador pode saber exatamente em que estado cada aplicação está e evita que ocorra alguma mudança de estado não permitida. O método *resumeRequest()* é usado para informar ao controlador que a aplicação saiu do estado *pausada* para entrar em novamente no estado *em execução*.

1.8.3. Desenvolvimento de aplicativos

Quando se desenvolve aplicações para televisão digital interativa, sempre se deve ter em mente que *Xlets* são aplicativos feitos para executar em um ambiente televisivo, onde recursos de imagem, como vídeos e interfaces gráficas, reinam e devem ser um dos requisitos melhor aproveitados em uma aplicação. Devido a este fato, daremos ênfase na

explicação de recursos e aplicativos que exploram as APIs gráficas disponíveis para o desenvolvimento.

As APIs disponíveis para o desenvolvimento de *Xlets*, como comentado anteriormente, são diferentes das APIs do Java tradicional. No MHP é usada uma versão reduzida da API AWT do Java SE, como definida em Java 1.1.8, e juntamente com a API HAVi UI formam o conjunto de recursos gráficos de uma *Xlet*. Para exibição de vídeos a API JMF(*Java Media Framework*) dá suporte ao padrão.

O início da implementação deve estar baseado na classe que implementa a interface *Xlet*, pois é ela a classe inicial da aplicação. Tal classe provê um construtor público e sem argumentos, que geralmente não é declarado no corpo na classe, pois o gerenciamento feito pelo *middleware* é mais bem aproveitado se as funções do construtor ficarem a cargo dos métodos *initXlet()* e *startXlet()*.

O primeiro método a ser chamado depois que a aplicação foi carregada e instanciada é o *initXlet()* , que, como já mencionado, também provê o contexto para a *Xlet*, o *XletContext*, que é passado como parâmetro na chamada do método.

O método *startXlet()* pode ser considerado como o par do método *pauseXlet()*, já que tudo o que um inicia, o outro deve parar e vice versa. Geralmente torna os componentes visuais visíveis ao usuário e é invocado se deseja que a aplicação comece a executar suas funções. Todos os componentes visuais implementados pelas APIs gráficas são apresentados por padrão na camada gráfica de representação.

No MHP existem três camadas que representam dispositivos de hardware, camada de background, de vídeo e a camada gráfica.

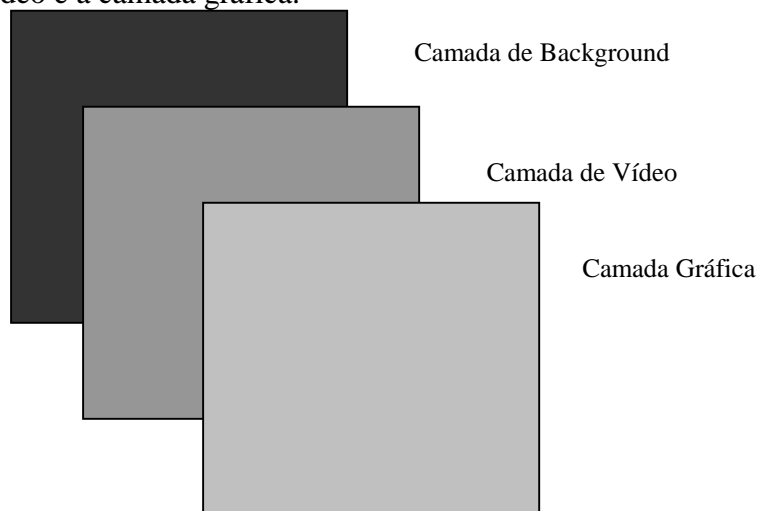


Figura 1.8: Camadas de apresentação no MHP.

A camada de *background* é usada quando se deseja uma imagem estática que sirva de fundo para a aplicação. A camada de vídeo reproduz o fluxo de vídeo vindo do operador e a camada gráfica, que se sobrepõe a todas as outras, mostra os componentes gráficos que caracterizam uma aplicação.

Quando se trabalha em aplicações desenvolvidas em Java da forma tradicional, geralmente é criado um *container* principal como o “*java.awt.Frame*” para podermos inserir componentes gráficos e exibí-los na tela. Para *Xlets*, um componente equivalente

ao *Frame*, é a *HScene*, que serve como um *container* principal disposto em uma determinada área da tela, para que componentes gráficos sejam inseridos e exibidos. Em *Xlets* é permitido o uso dos tradicionais componentes AWT, como “java.awt.Container” ou “java.awt.Componet”, porém, HAVi disponibiliza uma versão mais adaptada deste componentes para serem usados em ambientes de *Xlets*, que são, “org.havi.ui.HContainer” e “org.havi.ui.HComponent”.

Um dos cuidados básicos ao se desenvolver interfaces gráficas para televisão é no tamanho da visualização da tela, uma vez que duas possíveis formas serão consideradas, tamanho de tela 4:3 ou 16:9 que corresponde à transmissão de alta definição (*HDTV – High definition television*). Muitas vezes a solução pode ser a de usar uma definição intermediária 14:9, que contemplaria uma razoável visualização em telas 4:3 e 16:9.

Análise do projeto

Em um ambiente de TV digital, onde os recursos de *hardware* disponíveis são mais escassos, deve se levar em conta aspectos como processamento, tamanho da aplicação, e ao fato da aplicação ser acessada por um sistema de carrossel onde a latência é gigantesca.

Escolher uma abordagem orientada a componentes tem algumas vantagens, pois componentes são implementados de forma a desempenhar um papel específico dentro de um sistema e, de certa forma, serem substituíveis facilmente. Um componente pode desempenhar vários serviços dentro do sistema, porém isto torna alto o nível de acoplamento, ou seja, alta dependência entre os componentes. Estes fatores podem comprometer o fator tempo de processamento, que geralmente é crítico, por outro lado, beneficia o fator tamanho e tempo de latência da aplicação.

Para se tomar tais decisões, é preciso conhecer muito bem a estratégia de gerenciamento das aplicações feita pelo *middleware* que pode, por vezes, beneficiar uma estratégia de implementação de baixa latência e por outras uma estratégia de otimização de processamento.

1.8.4. Um exemplo

Para um primeiro exemplo, é normal esperarmos alguma funcionalidade que seja visual para podermos ter certeza do funcionamento da *Xlet*, no exemplo a seguir faremos o uso de tais recursos, como também aplicar alguns dos conceitos vistos até agora e introduzir de forma prática outros recursos necessários para se desenvolver uma aplicação para televisão digital. Dividiremos o exemplo a seguir em seções e ao final de cada seção daremos uma breve explicação do código apresentado.

Devido ao fato de que o emulador de distribuição gratuita que usaremos para executar as *Xlets*, o *XletView*, juntamente com a principal e mais completa ferramenta de emulação de *Xlets*, *IRT*, estarem baseadas em *MHP*, os recursos utilizados serão preferencialmente otimizados para tal, ressaltando que a maior parte dos recursos que serão apresentados podem ser aplicados para qualquer plataforma apresentada neste capítulo.

Vejamos o exemplo:

```
public class Exemplo1 extends HContainer implements Xlet, KeyListener{
```

Declaração da classe que entende *HContainer* e implementa as interfaces *Xlet* e *KeyListener*, o que possibilita para a classe controlar eventos do controle remoto ou de teclado.

```
private HScene cena;  
private XletContext ctx;  
private HStaticText texto;  
private HDefaultTextLayoutManager manager;  
private HTextButton botao1, botao2;  
private Font fonte = new Font("Tiresias",Font.PLAIN,21);
```

Declaração de todas as variáveis da classe, dois botões, um campo de texto, a cena, o contexto e o formato da fonte utilizada na aplicação. É recomendado para televisão que a fonte utilizada seja a “Tiresias” e com um tamanho mínimo 21.

```
public void initXlet(XletContext context) throws XletStateChangeException {  
    ctx = context;  
    cena = HSceneFactory.getInstance().getDefaultHScene();  
    setBounds(cena.getBounds());  
    manager = new HDefaultTextLayoutManager();  
}
```

Este foi o corpo do método *initXlet()*. Notem que não foi declarado nenhum construtor para classe e que o trabalho que seria realizado por ele foi transferido para dentro do corpo do método *initXlet()*.

Uma parte importante neste trecho de código foi a da obtenção da cena. Normalmente a cena precisa ser devidamente configurada para se adaptar as peculiaridades da aplicação, porém usaremos para exemplo a cena *default*, obtida através da *HSceneFactory* que é a entidade responsável por devolver instâncias da cena.

```
public void startXlet() throws XletStateChangeException {  
    cena.setVisible(false);  
    texto = new HStaticText("", 10,150,700,200,fonte,  
    Color.black,new DVBColor(255, 255, 255, 200),manager );  
    botao1 = new HTextButton("OK", 200,400,150,50,fonte,  
    Color.white,Color.red,manager);  
    botao2 = new HTextButton("Cancela", 400,400,150,50,fonte,  
    Color.white, Color.blue ,manager);  
    cena.add(this);  
    cena.addBefore(botao1, this);  
    cena.addBefore(botao2, this);  
    cena.addBefore(texto, this);
```

Até este momento foram inicializados os componentes que fazem parte da cena, um campo de texto estático e dois botões. Note que tais componentes foram adicionados através do método *addBefore()*, este método é uma das diferenças entre *HContainer* e *Container*. Ele controla a ordem de visualização na cena, neste caso os componentes devem aparecer na frente do container, para que o fundo pintado por *paint()*, não se sobreponha aos componentes.

```
String aux = revelaConfiguracoes();  
texto.setTextContent(aux, HState.ALL_STATES);  
otao1.setVisible(true);  
botao2.setVisible(true);
```

```

        texto.setVisible(true);
        botao1.setFocusTraversal(null, null, botao2, botao2);
        botao2.setFocusTraversal(null, null, botao1, botao1);
        botao1.addKeyListener(this);
        botao2.addKeyListener(this);
        botao1.requestFocus();
        cena.setVisible(true);
    }

```

Aqui termina a implementação do método *startXlet()*, todos os componentes foram ditos visíveis através do método *setVisible()*, isto é importante para garantir a visualização deles na tela. Nas linhas seguintes foi definido a navegação entre os botões através de *setFocusTransversal()*, que informa a aplicação qual o componente que deve obter o foco no caso de eventos nos botões direcionais do controle remoto ou teclado. Este método foi derivado de *setMove()*, que especifica o foco para qualquer ação de controle remoto ou teclado. Também foi adicionado um *listener* a cada botão, para tratar dos eventos gerados por eles.

```

    public void pauseXlet() {
        System.out.println("Estado de Pausa");
        cena.setVisible(false);
    }

```

Neste caso, o método *pauseXlet* apenas torna a cena invisível, pois nenhum outro recurso precisa ser parado, como alguma *thread* ou entrada e saída de dados.

```

    public void destroyXlet(boolean arg0) throws XletStateChangeException {
        System.out.println("Estado de Pausa");
        ctx.notifyDestroyed();
    }
    public void paint( Graphics g ){
        g.setColor( new DVBColor( 0,0,0,100));
        g.fillRect(0,0,720,576);
        super.paint(g);
    }

```

O método *paint()* é muito conhecido em ambientes Java tradicionais. Ele foi mantido na API AWT para TV digital e pode ser muito útil para aproveitar todo potencial gráfico de uma aplicação. Neste caso ele pinta um fundo preto transparente que permite que a camada de vídeo seja visualizada no para o caso de haver um vídeo em execução. A transparência nas cores é um recurso muito importante no desenvolvimento das *Xlets* e pode ser conseguido com a classe *DVBColor* que possui um parâmetro para transparência, além dos parâmetros RGB normais.

```

    public String revelaConfiguracoes(){
        String conf = "";
        conf += "Dimensoes da cena: " + cena.getBounds();
        conf += "\n" + "Dimensoes do Container: " + getBounds();
        conf += "\n" + "Numero de componentes na Cena: " +
            cena.getComponentCount();
        return conf;
    }
    public void keyTyped(KeyEvent e) {}
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == HRcEvent.VK_ENTER ){
            if(botao1.isSelected()){

```

```

        System.out.println("Acao OK");
    }
    else{ System.out.println("Acao Cancela");}
}
}
public void keyReleased(KeyEvent e) {}
}

```

Os três últimos métodos se referem ao controle de eventos externos. Neste caso é necessário o controle sobre dois botões. Quando o usuário da aplicação apertar o botão do controle remoto que corresponde à tecla de confirmação, tecla *ok* por exemplo, o tratador de eventos deve ser capaz de verificar qual dos botões está com o foco e disparar a ação correspondente.



Figura 1.9: Saída do programa no XletView

1.8.5. Emuladores e ferramentas de desenvolvimento.

XletView

XletView é uma ferramenta de emulação em PC, baseada em Java, para Xlets MHP e está atualmente na versão 0.3.6 e sendo constantemente atualizada. É um projeto de software livre licenciado sob a "Gnu Public Licence". Está acessível para *download* a partir do site da Fource Forge: <http://xletview.sourceforge.net>.

Para executar o XletView em um computador é recomendável que você tenha instalado a Máquina Virtual Java 1.4 ou maior.

O XletView usa o JMF2.1.1 (*Java Media Framework*) para exibição de vídeos e aceita exclusivamente o formato de vídeo AVI (*Audio Video Interleaved*) compactado com compressão Cinepack.

IRT

Middleware baseado na implementação MHP. Por ser usado como *middleware* em set top boxes, é o ambiente de emulação para MHP mais completo do mercado. Foi desenvolvido na Alemanha em parceria com Suíça, e pode ser solicitado para *download* após o pagamento da licença. Pode ser instalado no PC sob Windows ou Linux para simulação de Xlets MHP.

Cardinal

Cardinal é uma ferramenta de autoria de *Xlets*. Provê uma maneira simplificada de criação de aplicações através da utilização de componentes e *frameworks*. É baseado em Java, porém o código gerado é fechado. Pode-se obter uma versão para teste no *site* da desenvolvedora.

1.9. Conclusão e continuação dos estudos

Um bom começo para quem deseja se aprimorar no desenvolvimento de aplicações para televisão digital é ler a bibliografia do capítulo, até este momento, um dos livros mais indicados para iniciantes no assunto é o livro de Steven Morris. Existem na internet boas fontes de leitura, do básico ao avançado. No endereço <http://interactivetvweb.org/> pode-se encontrar tutoriais completos sobre televisão digital, MHP, JavaTV e desenvolvimento de aplicações, incluindo exemplos de código que podem ser obtidos do próprio *site*.

Em (Becker e Montez 2004) pode-se obter uma visão sistêmica da televisão digital e a relação das tecnologias envolvidas com a sociedade brasileira. Além disso, recomendamos também o livro “O Guia Prático da Produção de Televisão Interactiva”, de Nuno Bernardo, que faz um apanhado geral dos conhecimentos necessários para identificar as ferramentas e os meios atrelados ao desenvolvimento de produtos para TV interativa. Para ver exemplos do que está sendo produzido pelo mundo afora, um bom *site* é Broadband Bananas (<http://www.broadbandbananas.com>), que tem uma série de exemplos de aplicativos produzidos pelas principais emissoras de televisão digitais.

Também é altamente recomendável uma leitura nas APIs disponíveis, como a HAVi e a DAVIC, para ter noção do que é possível ser feito em um ambiente de televisão. Você pode desenvolver seus aplicativos em alguma IDE Java como o Eclipse (www.eclipse.org), e importar as bibliotecas necessárias, neste caso, é importante verificar a API disponível, que é diferente da de Java tradicional, e simulá-las em um dos emuladores disponíveis, Xletview (<http://xletview.sourceforge.net/>) e OpenMHP (www.openmhp.org) são boas opções de código livre e podem ser facilmente achadas na internet e integradas ao Eclipse.

Mais informações sobre *datacasting* pode ser encontrada em trabalhos acadêmicos, como (Piccioni 2005) e (Staffans 2004). O primeiro detalha os mecanismos de carrosséis de dados e de objetos, assim como implementa um protótipo fim a fim de transmissão de dados. Já o segundo introduz o *IP Datacasting*, que está servindo como base para a difusão de dados em sistemas de TV digital móveis como o DVB-H.

Referências

- Agência Estado. Web é 2.^a fonte de informações para universitários. Disponível em <<http://www.estadao.com.br/educando/noticias/2005/ago/17/93.htm>>. Acesso em 4/10/05.
- ATSC A/91 - Advanced Television Systems Committee. ATSC Recommended Guidelines for the ATSC Data Broadcasting Standard, 2001.
- BBC, BRITISH Broadcast Corporation. “Progress towards achieving digital switchover: a BBC report to the Government”. BBC, Londres, 2004.
- BECKER, Valdecir e MORAES, Áureo. “Do analógico ao digital: uma proposta de comercial para TV digital interativa”. In: III Simpósio Catarinense de Processamento Digital de Imagens. 2003, Florianópolis, 2003.
- BECKER, Valdecir; VARGAS, Rafael; GUNTER FILHO,; MONTEZ, C. B. Júri Virtual I2TV: Uma Aplicação para TV Digital Interativa baseada em JavaTV e HyperProp. In: WEBMÍDIA 2004, Ribeirão Preto. 2004.
- BECKER, Valdecir; MONTEZ, Carlos. TV Digital Interativa: conceitos e tecnologias. In: Minicursos Webmidia 2004. Ribeirão Preto, 2004.
- BECKER, Valdecir e MONTEZ, Carlos. “TV digital interativa: conceitos, desafios e perspectivas para o Brasil”. Editora da UFSC, Florianópolis, 2005.
- BERNARDO, Nuno. “O guia prático da produção de televisão interactiva”. Centro Atlântico, Porto, 2002.
- BIAL, Pedro. “Roberto Marinho”. Jorge Zahar Editor, Rio de Janeiro, 2005.
- CASTELLS, Manuel. “A sociedade em rede: a era da informação: economia, sociedade e cultura”. Paz e Terra, São Paulo, 2003.
- CROCOMO, Fernando Antonio. “TV digital e produção interativa: a comunidade recebe e manda notícias”. Trabalho apresentado ao Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina para o exame de qualificação, requisito parcial para obtenção do título de doutor em Engenharia de Produção. Florianópolis, 2004.
- ETSI ES 201 812 V1.1.1 - European Telecommunications Standards Institute. Digital Video Broadcasting: Multimedia Home Platform Specification 1.0.3, 2003.
- ETSI TS 102 819 V1.2.1 - “GEM – Globally Executable MHP: A Guide to Platform Harmonisation”, DVB Project Office White Paper on iTV Platform Harmonisation, May 2005.
- ETSI TR 101 202 - European Telecommunications Standards Institute. Digital Video Broadcasting: Implementation guidelines for Data Broadcasting, 2003.
- FAIRHURST, G. Data transmission using MPEG-2 and DVB, 2005. URL <http://www.erg.abdn.ac.uk/research/future-net/digital-video/dsm-cc.html>. Último acesso em 3 de outubro.
- GAWLINSKI, Mark. “Interactive television production”. Oxford, Focal Press, 2003.

- ISO/IEC 13818-2 - International Organization for Standardization. Coding of Moving Pictures and Associated Audio – MPEG-2 Video, 1995.
- ISO/IEC 13818-1 - International Organization for Standardization. Coding of Moving Pictures and Associated Áudio - MPEG-2 Systems, 2000.
- ISO/IEC 13818-6 - Organization for Standardization. Coding of Moving Pictures and Associated Audio- Extension for Digital Storage Media Command and Controls, 1996.
- LORÊDO, João. “Era uma vez... a televisão”. Alegro, São Paulo, 2000.
- MACLIN, Bem. “What Every Marketer Needs to Know about iTV”. New York, eMarketer Analyst Brief, 2001.
- MACHADO, Arlindo. “A televisão levado a sério”. Senac São Paulo, São Paulo, 2003. 3ª edição.
- MORRIS, Steven and Smith-Chaigneau, Anthony “Interactive television standards”, Edited by Elsevier Inc., United States, 2005.
- MORRIS, S. MHP interactive, 2005. URL <http://www.mhp-interactive.org/tutorial/mhp/index.shtml>. Último acesso em 3 de outubro.
- NIELSEN, Jakob. “Projetando websites”. São Paulo, Campos, 2000.
- PAGANI, M., Multimedia and Interactive Digital TV: Managing the Opportunities Created by Digital Convergence. IRM Press, 2003.
- PICCIONI, C. A., Modelo e Implementação de um Serviço de Datacasting para Televisão Digital. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2005.
- PICCIONI, Carlos A; BECKER, Valdecir; MONTEZ, Carlos; VARGAS, Rafael. “Juri virtual: uma aplicação de governo eletrônico usando televisão digital interativa”. In: IV Simpósio Catarinense de Processamento Digital de Imagens. 2003. Florianópolis, 2003.
- SCHWALB, E.M. iTV Handbook: Technologies and Standards. Prentice Hall PTR, 2003.
- SRIVASTAVA, Hari Om. “Interactive TV technology and markets”. Boston: Artech House, 2002.
- STAFFANS, L. Internet protocol datacasting, a technology overview. Master’s thesis, Helsinki University of Technology, 2004.
- Sun Microsystems, “Programmers Guide”, Santa Clara, California, 2002.
- Tektronix. A Guide to MPEG Fundamentals and Protocol Analysis, 2002. URL http://www.tek.com/Masurement/App_Notes/25_11418/eng/25W_11418_4.pdf. Último acesso em 3 de outubro.
- WHITAKER, Jerry. “Interactive Television Demystified”. McGraw-Hill, Nova York, 2001.