

FlexTV — Uma Proposta de Arquitetura de Middleware para o Sistema Brasileiro de TV Digital

Luiz Eduardo Cunha Leite

Carlos Eduardo Coelho Freire Batista

Guido Lemos de Souza Filho

Raoni Kulesza

Luiz Gustavo Pacola Alves

Graça Bressan

Rogério Ferreira Rodrigues

Luiz Fernando Gomes Soares

{leduardo,bidu,guido}@lavid.ufpb.br,

{raoni,luizgpa,gbressan}@larc.usp.br,

{rogerio,lfgs}@inf.puc-rio.br

Resumo

As diversas propostas de Sistema de Televisão Digital trazem consigo a especificação de middlewares que possibilitam a execução de programas de televisão interativos nos Receptores Digitais ou Terminais de Acesso, escondendo dos mesmos a complexidade dos mecanismos definidos pelos protocolos de comunicação, do sistema operacional e do hardware do equipamento. É o middleware que define a interface para tais programas e, por conseguinte, quais funcionalidades poderão ser oferecidas pelos aparelhos de televisão para seus usuários. Visando promover a inclusão social e digital da população brasileira, é extremamente importante que o middleware a ser adotado no Sistema Brasileiro de Televisão Digital permita o desenvolvimento de programas mais adequados à realidade desse público. Realidade esta diferente daquela encontrada em outros países que já definiram seus Sistemas de Televisão Digital. Não obstante, esse middleware também deve possuir alinhamento com padrões internacionais, de forma a possibilitar a exportação do conteúdo televisivo produzido no país, bem como a exibição de conteúdo produzido por outros países nos televisores do Brasil, promovendo, dessa forma, o desenvolvimento econômico e o intercâmbio cultural no país. Este trabalho apresenta a arquitetura do middleware FlexTV, que está sendo desenvolvido no contexto do projeto do Sistema Brasileiro de Televisão Digital. A arquitetura, que se baseia fortemente no uso de componentes de software, permite que o middleware adapte-se a diferentes tipos de aparelhos receptores e infra-estruturas de comunicação, através da adição ou remoção de componentes de software, de forma a torná-lo o mais adequado possível à realidade brasileira. A mesma característica também simplifica a tarefa de tornar o FlexTV compatível com os sistemas de televisão digital já em operação em outros países.

Palavras-chave: TV Digital Interativa, Middleware, SBTVD.

1. Introdução

A televisão, como os outros meios de comunicação de massa, segue a tendência mundial do movimento de digitalização através de um acelerado processo de substituição de suas plataformas analógicas por plataformas e tecnologias digitais interoperáveis.

A primeira onda de impacto, já sentida internamente por várias redes de TV brasileiras, é a necessária substituição dos equipamentos de captura, edição e transmissão interna de áudio e vídeo analógicos, por similares digitais, visando melhoria da imagem e som.

A segunda onda de impacto, a ser sentida pelo conjunto da sociedade, é a necessária adoção de um padrão uniforme de sistema para codificação, transmissão, modulação, difusão e recepção digital de programas de televisão. No Brasil, este impacto será maior nos sistemas de TV Digital Terrestre (DTT – *Digital Terrestrial Television*), comumente usados nos centros urbanos, onde o maior desafio é a escolha técnica-econômica-social-política do formato de modulação de sinais [37]. Está em discussão no país a definição do sistema a ser adotado para DTT. Tal discussão está sendo travada em torno do projeto do Sistema Brasileiro de Televisão Digital (SBTVD). Este projeto prevê o estudo dos sistemas já existentes – ATSC (americano), DVB (europeu) e ISDB (japonês) – e a proposta com base nas tecnologias já existentes, em extensões ou mesmo em tecnologias desenvolvidas no país, de um sistema que atenda aos requisitos socioeconômicos do Brasil. Os sistemas de televisão já definidos convergem com relação ao formato adotado para a codificação digital dos dados a serem transmitidos: o padrão ISO MPEG-2. Este padrão, que na realidade define não uma, mas um conjunto de normas, divide-se em MPEG Vídeo, MPEG Áudio e MPEG Sistema. O MPEG Áudio e o MPEG Vídeo especificam métodos para compressão e codificação de áudio e vídeo digital, respec-

tivamente, com diferentes níveis de qualidade. Para o caso do vídeo, estão, por exemplo: SDTV (*Standard Digital Television*) e HDTV (*High Definition Digital Television*). O MPEG Sistema define formatos e protocolos para montagem de pacotes para transmissão de fluxos elementares multiplexados. Os fluxos elementares transportam *streams* de vídeo, áudio ou dados.

A terceira onda de impacto da TV Digital (TVD), a ser sentida após a adoção do padrão de DTT, é a necessidade de desenvolvimento de novos modelos de negócio que: (i) estimulem a população a investir em equipamentos de TV de nova geração; e (ii) permitam às redes obter retorno sobre os investimentos efetuados. Segundo esta visão, os recursos aplicados pelas cadeias de TV e pela audiência televisiva devem ser encarados como investimentos que permitem a exploração de novos negócios. A maior expectativa de retorno reside na interação. O canal de dados do MPEG-2 torna possível a agregação de elementos de interação aos programas de televisão, tornando-os “interativos”. A possibilidade de interação obriga que o aparelho receptor de televisão tenha capacidade de processar o código que define os elementos de interação, o “código do programa de TV interativo” e, opcionalmente, enviar o resultado da interação através de um “canal de retorno” para a estação emissora do programa interativo. A estação, conseqüentemente, também precisa estar equipada com hardware e software adequados para dar suporte aos “programas interativos”. Espera-se também que a rede difusora de TV se associe às outras redes de transporte de informações, dando aos telespectadores oportunidade para influenciar diretamente nos programas assistidos. Em suma, quando a TV se tornar interativa (TV Digital Interativa - TVDI), espera-se que a mesma venha a associar imenso apelo e penetração com capacidade de interação instantânea com milhões de telespectadores e com uma vasta cadeia de produtores de conteúdo.

Portanto, é preciso entender a TVDI como um novo meio a ser explorado que, não apenas no Brasil, mas em todo o mundo, só obterá sucesso através do desenvolvimento de novas aplicações, reorganização das cadeias de produção televisiva, geração de negócios e de transformações sociais.

Todas as propostas de sistemas de Televisão Digital especificam *middlewares* sobre os quais as aplicações de TV Interativa podem ser executadas. *Middleware* é o neologismo criado para designar camadas de software que não constituem diretamente aplicações, mas que facilitam o uso de ambientes ricos em tecnologia da informação. A camada de *middleware* concentra serviços como identificação, autenticação, autorização, diretórios, certificados digitais e outras ferramentas para segurança [1].

No contexto de TV Digital, o *Middleware* vem a ser o software que controla suas principais facilidades (grade de programação, menus de opção), inclusive a possibilidade de execução de aplicações, dando suporte à interatividade [1]. O *middleware* é um elemento capaz de fornecer uma abstração do sistema para as aplicações e os usuários, escondendo toda a complexidade dos mecanismos definidos pelos hardware, software e interfaces de comunicação do aparelho receptor do sinal de televisão digital. Dessa forma, a padronização de uma camada de *middleware* permite a construção de aplicações independentes do hardware e do sistema operacional, executáveis em qualquer plataforma de qualquer fabricante.

Desenvolvimento baseado em (ou orientado a) componentes [3][4] é um paradigma de desenvolvimento de software onde o processo de produção de um sistema é feito através da construção de seus módulos (componentes) separadamente, sendo estes posteriormente interligados através de seus pontos de conexão. Na visão de Pereira [3], qualquer elemento de software pode ser considerado um componente, desde que possua uma interface bem definida. Pereira visualiza o desenvolvimento baseado em componentes como uma evolução direta do paradigma de orientação a objetos.

São várias as vantagens de se utilizar o paradigma de Desenvolvimento Orientado a Componentes. Por exemplo:

- **Possibilidade de reutilização de projetos e implementações** – dependendo da granularidade adotada, todo o projeto, ou parte dele, pode ser utilizado na produção de outros sistemas ou, até mesmo, de outros componentes;
- **Possibilidade de desenvolvimento em larga escala** – em função da possibilidade de reuso, uma grande quantidade de componentes pode ser desenvolvida a partir de outros;
- **Facilidade de testes** – as funcionalidades dos componentes podem ser testadas separadamente, possibilitando uma prematura detecção e correção de problemas;
- **Aumento de confiabilidade** – em virtude da facilitação dos testes, há a possibilidade de diminuição do tempo de detecção e correção de problemas;
- **Facilidade de atualização e crescimento funcional** – os componentes podem ter sua funcionalidade interna incrementada, sem acarretar danos aos usuários, apenas mantendo-se as antigas interfaces e incluindo outras novas;
- **Possibilidade de troca de componentes** - trabalhando-se em um ambiente autoconfigurável, caso componentes apresentem defeito, estes podem ser trocados por outros em perfeito funcionamento.

Este trabalho apresenta uma proposta de *middleware* para o Sistema Brasileiro de Televisão Digital, denominado FlexTV, cuja arquitetura interna é baseada em componentes de software. Tal arquitetura possibilita que os componentes que constituem o FlexTV possam ser adicionados, removidos ou alterados, fomentando a escalabilidade funcional desse *middleware*. Ou seja, novas funcionalidades podem ser adicionadas ou removidas do FlexTV, de acordo

com a evolução dos Terminais de Acesso ou com o surgimento de novos requisitos por parte de seus usuários (telespectadores e emissoras).

Uma outra vantagem de adotar uma arquitetura baseada em componentes de software é que alguns dos componentes internos do FlexTV podem ser aderentes a padrões internacionais de *middleware* para Televisão Digital Interativa, enquanto outros podem ser inovadores e mais adequados à realidade brasileira. Os componentes do FlexTV que são aderentes a padrões internacionais possibilitam o desenvolvimento de aplicações que podem tanto funcionar no Sistema Brasileiro de Televisão Digital quanto nos sistemas de outros países; bem como a importação de aplicações desenvolvidas para outros sistemas de Televisão Digital Interativa. Já os componentes não aderentes a padrões internacionais possibilitam o desenvolvimento de aplicações que tirem proveito de características específicas do Sistema Brasileiro de Televisão Digital.

Para ilustrar os benefícios que a arquitetura baseada em componentes do FlexTV pode trazer, considere o caso de uma aplicação de comércio eletrônico sendo executada no terminal de acesso de um telespectador com deficiência visual. Nesse caso, os componentes do FlexTV responsáveis por fazer a exibição dos elementos gráficos das aplicações no televisor podem ser substituídos por outros componentes capazes de fazer a tradução de tais elementos em mensagens sonoras, possibilitando a interação com os usuários portadores de deficiência, sem que seja necessária qualquer modificação na aplicação original de comércio eletrônico que foi transmitida pela emissora de televisão.

Este trabalho está organizado da seguinte forma: seguindo a introdução, temos uma seção que visa embasar o leitor acerca da teoria fundamental para compreensão do texto (Seção 2 – Embasamento Teórico); na Seção 3 a especificação GEM (*Globally Executable MHP*) é apresentada; logo em seguida (Seção 4) uma análise das propostas de

middleware para TV Digital é exposta; na Seção 5 o *middleware* FlexTV é apresentado mais detalhadamente, e por fim (Seção 6) são descritas as conclusões do trabalho.

2. Embasamento Teórico

Um Sistema Básico de Televisão Digital (TVD) consiste de uma estação transmissora ou *head-end*, um meio físico sobre o qual o sinal de vídeo é transmitido, que pode ser o ar ou meios físicos guiados (cabo coaxial, fibra óptica etc.), e um Receptor Digital, incluindo um *Set-Top Box* ou Terminal de Acesso, responsável por receber o sinal transmitido, decodificá-lo e exibi-lo. Para garantir a compatibilidade entre esses elementos, é necessário que sejam estabelecidos padrões que normatizem todo o processo de captura, compressão, modulação e transmissão dos sinais de vídeo, além de todas as interfaces físicas entre os equipamentos envolvidos no processo. Para tanto, existem atualmente diversas propostas de Sistemas de Televisão Digital. Dentre elas podemos citar: o DVB [5] (proposta européia), o ATSC [6] (proposta americana) e o ISDB [7] (proposta japonesa).

Apesar de divergirem em vários aspectos, as principais propostas de padronização para TVD convergem em pelo menos um ponto: Todas seguem o modelo de referência União Internacional de Telecomunicações – ITU [22] para transmissão dos sinais de televisão digital. Este modelo divide as funcionalidades do sistema (de transmissão) em três camadas principais: (i) codificação do sinal-fonte; (ii) multiplexação; e (iii) modulação, adotando o padrão MPEG-2 como principal tecnologia nos processos realizados nas camadas (i) – em particular, a codificação do sinal-fonte de vídeo – e (ii), enquanto que nos processos da camada (iii) cada uma das propostas (DVB, ATSC e ISDB) adota uma solução diferente. As especificações MPEG-2 [23][24][25][26][27] basicamente descrevem regras sintáticas e semânticas que definem esquemas de

compressão, empacotamento e multiplexação de sinais individuais de áudio, vídeo e outros tipos dados digitais em um fluxo de transporte (TS – *transport stream*), permitindo transmitir em um único canal físico vários “fluxos elementares”, que em conjunto representam um ou mais serviços de TVD. Adicionalmente, para mapear e indexar o conteúdo do fluxo de transporte, cada padrão de TVD estende as tabelas PSI (*Program Specific Information*) do padrão MPEG-2 definindo um conjunto de estruturas que possuem dados descritivos (metadados) que carregam informações de serviços (SI – *Service Information*) específicas do domínio de TVD. Podemos citar como principais padrões de SI para TVD: DVB-SI [22], ATSC-PSIP [23], ISDB-SI [24], respectivamente, especificações do padrão europeu, americano e japonês.

O conceito de fluxos elementares multiplexados com informações de serviço num único fluxo de transporte permite que sejam codificados e transportados em paralelo um ou mais fluxos de vídeo, por exemplo, gerados por câmeras diferentes em uma partida de futebol; um ou mais fluxos de áudio, transportando, por exemplo, os seis canais de áudio utilizados na sonorização de *home theaters*; ou, generalizando, qualquer tipo de dado digital. A viabilidade de transportar e identificar qualquer tipo de dado em um fluxo elementar abriu novas possibilidades de expressão de informação nos conteúdos transmitidos na rede de televisão. Por exemplo, podem ser transmitidos textos com legendas; imagens com propaganda que podem ser exibidas em painéis virtuais; códigos definindo elementos de interação, que podem ser então associados aos programas de televisão tornando-os interativos; ou mesmo códigos de programação executáveis.

Uma vez entregue um fluxo de transporte, para que o mesmo possa ser demultiplexado, decodificado e exibido para o telespectador, se faz necessária a utilização de um Terminal de Acesso. Tal equipamento é uma combinação de elementos de arquitetura de um receptor de sinais digitais e um computador pes-

soal que essencialmente atua como um ponto central provedor de serviços para o usuário. A interação com o usuário pode ocorrer por dispositivos de entrada – tais como, controle remoto e teclado, e dispositivos de saída – tais como, aparelhos de TV analógico ou digital, monitores de computador e sistemas de som. Acessórios e outros equipamentos também podem ser conectados ao Terminal de Acesso. A presença desses aparelhos diretamente ligados à televisão na casa dos usuários e a possibilidade de receber qualquer dado digital, além do áudio e vídeo, motivou o desenvolvimento de aplicações (*software*), tais como: jogos, grades de programação eletrônica, etc. A essas aplicações, com as quais os telespectadores podem interagir através do aparelho de televisão, mas sem o envio de dados à estação transmissora ou *head-end*, damos o nome de Aplicações de Televisão Digital Interativa com Interação Local. Com o aprimoramento dos Terminais de Acesso, os mesmos começaram a ser dotados de interfaces que possibilitam o estabelecimento de canais de interação com os *head-ends*, por exemplo, modems telefônicos, interfaces de rede e etc. Os canais de interação possibilitam então que dados sejam enviados do usuário para o *head-end* e vice versa. O surgimento desses canais deu origem a um novo grupo de aplicações, as chamadas Aplicações de Televisão Digital Interativa com Interação Remota, ou simplesmente Aplicações de TV Digital Interativa – TVDI. São exemplos desse tipo de aplicações: aplicações de comércio eletrônico, acesso bancário, jogos multi-usuário etc.

Visando facilitar o desenvolvimento das aplicações, garantir a compatibilidade das mesmas com as diversas arquiteturas de Terminais de Acesso existentes e ocultar particularidades e diferenças do *hardware* e sistema operacional, as principais propostas de Sistema de TVD trazem, além da especificação das interfaces de meio físico e da codificação dos sinais de transmissão, propostas de *middlewares* para Televisão Digital com uma Interface para Programação de Aplicações

(*Application Programming Interface – API*) bem definida. Dentre as principais propostas de *middleware* para TVDI podem ser citados: o MHP [8] para o sistema DVB, o DASE [9] e o ACAP [10] para o sistema ATSC, o ARIB-STD 24 [7] para o sistema ISDB e o OCAP [11], definido pela CableLabs para o sistema de TV a cabo norte-americano. Todos esses *middlewares* oferecem mecanismos para carregar o código dos aplicativos recebidos a partir do *head-end*, e executá-los. Além disso, esses *middlewares* também oferecem serviços que permitem: (i) gerência de aplicações e recursos do *middleware*; (ii) comunicação entre aplicações e processos; (iii) processamento (decodificação e exibição) de mídias; (iv) sincronismo; (v) seleção e acesso e interpretação de fluxos elementares e informações de serviço; (vi) tratamento de eventos e elementos áudio-visuais de interface com o usuário; (vii) armazenamento, localização e recuperação de dados locais; e (viii) controle do ciclo de vida das aplicações. Outras funcionalidades podem ser oferecidas de acordo com o *middleware* e com os recursos presentes no Terminal de Acesso, tais como sistemas de acesso condicional, acesso a dados remotos, perfis de usuários, acesso ao fluxo de áudio e vídeo principal sendo exibido.

Observando um *middleware* como uma base de *software* que abstrai detalhes e permite às aplicações o acesso a recursos básicos do Terminal de Acesso, através de componentes, funções de alto nível, estruturas de dados e protocolos que representam um conjunto de APIs, temos a tecnologia Java¹ da Sun Microsystems como peça central desse conjunto de *software* (APIs) e que, atualmente, está presente em todos os padrões de *middleware* já citados. No contexto dos padrões DASE, ACAP, OCAP, ARIB STD 24 e MHP as principais APIs Java presentes são: (i) JavaTV [31], que viabiliza o desenvolvimento de aplicações interativas portáteis, que são independentes da tecnologia do hardware e da rede de difusão utilizados. Ela estende o pacote J2ME (Java 2 Platform, Micro Edition) e adiciona funcionalidades específi-

cas para o contexto de TVD; (ii) JMF [32] que define o comportamento e a interação de objetos de mídia contínua. A JMF é utilizada para capturar, processar e apresentar alguns tipos de mídias contínuas; (iii) DAVIC [34], que define requisitos de sistemas audiovisuais para prover interoperabilidade fim-a-fim; e (iv) HAVi [33], que estende o pacote gráfico padrão do Java (AWT) e adiciona funcionalidades para prover suporte a controle remoto, gráfico específico para TVD, entre outros.

As primeiras plataformas para serviços interativos de TV Digital foram implantadas no final da década de 90 de acordo com uma divisão em blocos macroeconômicos, políticos, geográficos. Conseqüentemente, esse cenário resultou em muitos padrões, tecnologias e equipamentos distintos e tornou-se o principal motivo da incompatibilidade, que não permite que uma aplicação, serviço ou conteúdo criado para uma plataforma específica seja portátil para outras plataformas do mesmo mercado, sem algum mecanismo de conversão técnica. Entretanto, nos últimos anos, vários esforços têm sido elaborados com o propósito de expandir o cenário de TVD, integrando-a assim aos cenários de convergência da tecnologia digital.

A convergência tecnológica permite a integração de serviços de propósitos complementares e sua disponibilidade para múltiplos terminais de diferentes características. Desta maneira, é possível integrar as redes de transmissão digital de telefonia celular, de dados (Internet) e de TVD e conseqüentemente, seus serviços e conteúdos. Para isso é necessário desenvolver, implementar e implantar uma série de itens tecnológicos, como: (i) *gateways* de conversão de: protocolos, modulação e formatos (resolução e qualidade das mídias); (ii) conteúdo com interatividade compatível para os diversos terminais; e (ii) e plataformas de execução (*middlewares*) compatíveis.

Como forma a prover uma padronização única para *middleware* de TV digital, está sendo desenvolvido

o padrão GEM (*Globally Executable MHP*), contendo um conjunto de funcionalidades básicas (por exemplo, a compatibilidade com a tecnologia Java e utilização de APIs comuns) a todos os outros padrões de *middleware*. Nota-se, portanto, que é desejável que qualquer *middleware* a ser desenvolvido para TV digital seja compatível com o modelo GEM, de forma a possibilitar a execução de um maior número de aplicações disponíveis e a serem desenvolvidas. Desta forma, na próxima seção detalharemos o GEM.

3. GEM (Globally Executable MHP)

Com a rápida evolução e popularização do *middleware* europeu MHP entre os diversos países que utilizam o padrão DVB, surgiram algumas iniciativas para sua implementação sobre outras plataformas internacionais. Certas entidades responsáveis pela padronização de sistemas de TV Digital fora da Europa manifestaram o interesse em ter parte do *middleware* MHP implementado em suas especificações, como forma de se aproveitar todo seu desenvolvimento tecnológico e manter uma compatibilidade que permitisse que as aplicações já existentes (e futuras) pudessem ser executadas ou apresentadas em terminais de acesso compatíveis com outros sistemas.

O padrão MHP é especificado com o objetivo principal de prover uma abstração do terminal de acesso e das redes de comunicação para as aplicações e serviços interativos, de forma que estes não se preocupem com os mecanismos e protocolos definidos no sistema. Por outro lado, por ter sido criado pelo grupo DVB, o MHP é uma implementação que se encontra atrelada aos protocolos e especificações próprias de terminais DVB. Alguns exemplos dessa dependência estão nos pacotes do MHP que fazem acesso às informações de serviços (DVB-SI) e naqueles que compõem o sistema de legendas (DVB-SUB).

Para que aplicações MHP pudessem ser utilizadas sobre outras plataformas, conforme requisitado por entidades tais como *CableLabs* (EUA) e ARIB (Japão), o grupo DVB tomou a decisão de propor uma especificação única chamada GEM (*Globally Executable MHP*) [12]. Além de capturar as interfaces e toda a semântica definidas pelo MHP (independentes da plataforma DVB), o GEM fornece também suporte às necessidades impostas por outros padrões internacionais. Ou seja, GEM é um acordo de harmonização. Por isso, a *CableLabs* participou da composição da primeira versão do GEM com o intuito de tornar o *middleware* OCAP (*Open Cable Application Platform*) [11] compatível com o GEM. Recentemente, o *middleware* japonês ARIB e o candidato a padrão americano ACAP também tornaram-se compatíveis com o GEM.

Formalmente, GEM não pode ser considerado uma especificação completa para terminais de acesso. O correto é dizer que GEM é um *framework* a partir do qual uma implementação de um terminal de acesso pode ser instanciada; ou ainda, que GEM é um padrão ao qual implementações existentes devem se adaptar para obter uma conformidade que garanta a execução global de aplicações. O padrão define, portanto, um conjunto de APIs, garantias de semântica e formatos de conteúdo com os quais as aplicações (agora globalmente interoperáveis) podem contar para a implementação de serviços interativos, executáveis em qualquer plataforma definida pelos padrões internacionais compatíveis. Para a criação de *middlewares* em conformidade, torna-se obrigatório referenciar por completo a especificação GEM, de forma a preencher todos os seus requisitos obrigatórios.

Os principais objetivos da especificação GEM são os seguintes:

- Maximizar a interoperabilidade entre especificações de terminais baseados no GEM e produzidos por diferentes organizações;

- Maximizar a presença de componentes MHP, promovendo economia de escala para toda a cadeia de difusão interativa, dado o parque de tecnologia MHP já instalado, o conhecimento já adquirido e o volume de conteúdo já produzido;
- Ser capaz de respeitar as restrições impostas pelas tecnologias de transporte e pelos modelos de negócio locais.

Por definir um *framework* baseado no padrão MHP, o documento de especificação do GEM é, na realidade, uma listagem de referências a outros documentos do consórcio DVB. Ele descreve as diversas partes do MHP que são independentes do padrão DVB e salienta aquelas que devem ser substituídas de acordo com a infra-estrutura de implementação das novas especificações compatíveis. Esses pontos de flexibilização do GEM devem ser, então, preenchidos por “equivalentes funcionais” – mecanismos que lançam mão de outras tecnologias (definidas pelas respectivas organizações de TV digital) para implementar uma funcionalidade análoga àquela proposta pelo DVB. Tais tecnologias passam a ser qualificadas como equivalentes funcionais somente após negociações entre o consórcio DVB e cada uma das organizações que requisitam a conformidade com o GEM.

A Figura 1 ilustra o cenário de convergência dos padrões de *middleware* de TV Digital em relação à especificação GEM. Os padrões OCAP, ACAP, ARIB e MHP definem cada qual seu conjunto de especificações para o gerenciamento de aplicações interativas e outros serviços importantes em TV Digital. Muitas dessas especificações se relacionam diretamente com os respectivos padrões de terminais de acesso, sendo óbvio, portanto, que boa parte delas esteja atrelada aos padrões internacionais de TV Digital. Porém, de uma forma geral, pode-se concluir que, em meio à heterogeneidade de definições, existe um conjunto de funcionalidades equivalentes, capturadas pelo GEM.

Assim, GEM se concentra na parte estritamente comum dos *middlewares*, especificando, por exemplo, um conjunto de protocolos de rede para canais de retorno e de difusão, formatos de conteúdo para exibição e o uso da máquina virtual Java e das APIs de extensão para TV. Para a parte funcionalmente comum, mas específica de plataforma, cabe aos padrões internacionais submeter seus equivalentes. Por exemplo, o padrão ACAP acumula funções do DASE e do OCAP para preencher as lacunas do *framework*. O *middleware* MHP, por ser a base do *framework*, funciona como alicerce para definição de equivalentes funcionais, que são sempre referenciados na especificação GEM. Os principais pontos de flexibilização redefinidos pelo ARIB, por sua vez, especificam um conjunto diferente de protocolos de rede e de informações sobre serviços. Vale ressaltar que, apesar de não ilustrado na Figura 1, cada *middleware* define seu próprio modelo de aplicações declarativas, porém esse item é considerado como “não-relevante” para a especificação GEM. Outro ponto digno de nota é a ausência do *middleware* DASE no cenário de convergência, que se deve ao fato de o ATSC ter partido para uma harmonização com o OCAP surgindo, então, o ACAP, como caminho para a definição de um *middleware* compatível com o GEM.

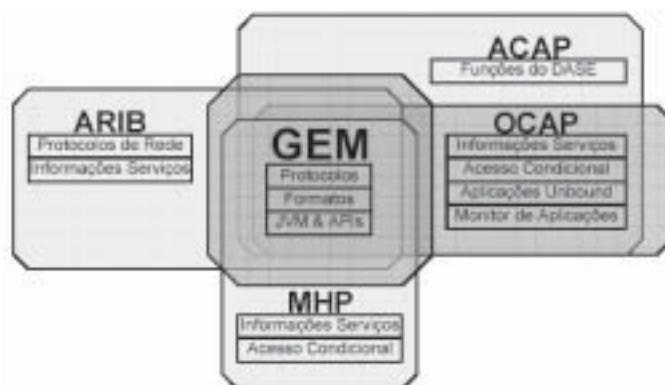


Figura 1. Relação entre a especificação GEM e outros padrões.

3.1. Definição de perfis GEM e requisitos de compatibilidade

Terminais GEM podem ser classificados pelos perfis herdados do padrão MHP. Os requisitos de compatibilidade são especificados apenas para os perfis *Enhanced Broadcast* e *Interactive Broadcast*.

Itens Obrigatórios em ambos os perfis

- Formatos estáticos: PNG, MPEG I-Frames, JPG, MPEG Audio Layers 1 e 2, MPEG-2 Video Drips e texto UTF-8
- Formatos de *streaming*: Pelo menos um formato de áudio e um formato de vídeo, a serem preenchidos por equivalentes funcionais.
- Fontes: *Teresias* e suporte a fontes carregáveis.
- Protocolos de difusão: Seções MPEG-2, Carrossel de Objetos (ou equivalente)
- DVB-J: APIs fundamentais, APIs de apresentação, APIs de acesso a dados encapsulados, APIs de acesso a informações sobre serviços, APIs de infra-estrutura e de segurança.

Itens obrigatórios para o perfil *Interactive Broadcast*

- Formatos estáticos: JPG sem restrições
- Protocolos de difusão: IP Multicast sobre difusão (recomendável)
- Protocolos para o canal de retorno: TCP/IP, UDP/IP, DNS
- DVB-J: APIs de suporte a IP e de segurança sobre o canal de retorno.

3.2. Equivalentes Funcionais

Uma extensa tabela de equivalentes funcionais é especificada em [12]. Ela define os pontos de flexibilização deixados em aberto pelo *framework* GEM e qual seu correspondente na arquitetura MHP. A Tabela 1 traz os equivalentes funcionais relevantes para este trabalho e indica a obrigatoriedade de preenchimento pelos padrões candidatos à conformidade com GEM.

Tabela 1. Equivalentes Funcionais GEM.

Nome	Definição no GEM	Definição no MHP	Preenchimento
Arquitetura	Alguma Arquitetura Básica	Arquitetura Básica MHP	Opcional
Carrossel	Algum sistema de carrossel para objetos	DSM-CC User-to-user Object Carousel	Obrigatório
SI	Algum sistema de informações sobre serviços	DVB Service Information	Obrigatório
	Alguma API para informações sobre serviços	DVB Service Information API	
	Integração com a API JavaTV-SI	Integração com a API JavaTV-SI	
Áudio	Algum padrão de áudio para <i>streaming</i>	MPEG-2 Audio	Obrigatório
Video	Algum padrão de vídeo para <i>streaming</i>	MPEG-2 Video SD 25Hz	Obrigatório
Legendas	Algum tipo de legenda	DVB Subtitles e Teletext	Opcional
Sinalização de Aplicações	Suporte à descrição de aplicações e capacidade de localização do código e dados	Tabela AIT e descritores específicos para aplicações DVB-J, protocolos de transporte, etc.	Obrigatório
Acesso Condicional	Algum subsistema de acesso condicional	MHP CA-System	Opcional
Resolução Gráfica	Uma resolução gráfica para SD	720x576 SD (4:3, 14:9, e 16:9)	Obrigatório

O resultado dos esforços do GEM originou a primeira versão da especificação técnica no início de 2003 (1.0.0). A versão atual (1.2.1) data de maio de 2004.

Há vertentes que garantem que o GEM atingiu uma maturidade no que diz respeito à sua especificação, e que em versões futuras as alterações serão relativamente simples, sem muitos elementos opcionais adicionais. No entanto, algumas mudanças podem surgir em relação a dois aspectos: funcionalidades como aplicações *Web*, pois devem ser estabelecidos fundamentos comuns entre OCAP, ACAP e MHP; e uso de APIs de *smart cards*.

4. Avaliação das propostas de Middleware para TV Digital

De uma forma geral, pode-se dizer que todos os padrões de *middleware* definem dois tipos de aplicações a serem executadas em terminais de acesso de sistemas de TV Digital: aplicações procedurais e aplicações declarativas.

As aplicações procedurais correspondem a programas de computador formados por comandos ou instruções a serem interpretados ou executados no Terminal de Acesso, para realizarem algum tipo de processamento de dados e interface com o usuário. Os padrões de *middleware* convergem para o uso da máquina virtual Java e de algumas APIs de extensão como modelo de execução de aplicações procedurais. Isso favoreceu o estabelecimento do *framework* GEM (*Globally Executable MHP*), conforme explicado na Seção 3, e sua adoção por vários padrões.

As aplicações declarativas são, na realidade, documentos hipermídia a serem apresentados no Terminal de Acesso, com intuito principalmente informativo, e que possibilitam o encadeamento de diversos documentos. Nesse ponto, no entanto, cada padrão de *middleware* estabeleceu uma linguagem diferente para a descrição de documentos. Cabe destacar que o *framework* GEM não se

preocupou com a definição do modelo de apresentação de documentos declarativos.

Independente do tipo de aplicação, cabe ao *middleware* fornecer mecanismos de abstração dos recursos e dos serviços de comunicação do terminal de acesso. Dessa forma, as aplicações ficam independentes da plataforma de hardware do terminal e, com a harmonização rumo ao padrão GEM, tendem a ficar independentes dos esquemas de transporte dos dados e metadados. Nesse contexto, torna-se clara a importância do uso de linguagens de programação procedurais e declarativas que facilitam tal interoperabilidade. Por essa razão, todos os sistemas de TV digital adotaram *middlewares* baseados na máquina virtual Java e em linguagens declarativas cujos interpretadores também são facilmente portáteis.

Podem-se levantar aspectos pontuais nos quais existem divergências entre os padrões de *middleware*. Por exemplo, para se adicionar novas funcionalidades ao *middleware* DASE é necessário o acréscimo de novos interpretadores para diferentes aplicações. Dependendo do tipo de aplicação (declarativa ou procedural) para o qual o interpretador foi projetado, o mesmo será alocado no respectivo ambiente de execução. Já no MHP, os interpretadores são chamados de *plug-ins* e têm funcionalidade análoga à dos interpretadores do DASE, podendo ser alocados em diferentes ambientes de execução. No ARIB, por sua vez, apenas *plug-ins* relacionados a aplicações procedurais podem ser adicionados ao *middleware*, pois o mesmo trabalha com a forma declarativa exclusivamente através da linguagem BML (*Broadband Markup Language*) [7].

A definição de perfis de plataforma também apresenta diferenças. O perfil *Enhanced Broadcast* do MHP corresponde ao DASE nível 1, ambos sem canal de retorno e destinados apenas à difusão de TV digital. O DASE nível 2 inclui suporte para critérios de segurança e pequenas condições de

interatividade, enquanto o perfil *Intercative Broadcast* do MHP possui boa interatividade, disponibilizando o suporte ao protocolo IP sobre o canal de retorno. O DASE nível 3 e o perfil *Internet Access* do MHP têm como foco a interatividade e a integração com a Internet, necessitando, como consequência, de um maior poder de processamento (processadores e memória) nos terminais de acesso.

O *middleware* ARIB não entra em detalhes sobre perfis em suas especificações disponíveis, mas sabe-se que existem perfis que definem como se dá apresentação de aplicações nos terminais de acesso [7]. Perfis de apresentação especificam a forma de exibição de aplicações para os usuários, considerando restrições como mobilidade e tipo de *display*. Para a adaptação correta de perfis, o ARIB define padrões de conformidade que se prendem aos tipos de funcionalidades que cada classe de hardware pode ter.

O sistema ACAP, por sua vez, define os perfis ACAP-X *only* e ACAP-X & ACAP-J, que, em conformidade com o GEM, são equivalentes funcionais aos perfis do MHP.

Apesar da grande quantidade de serviços e funcionalidades oferecidas pelos *middlewares* propostos pelos principais Sistemas de Televisão Digital existentes, inclusive o próprio GEM, a grande maioria deles não oferece os serviços inerentes aos *middlewares* de nova geração, tais como sensibilidade ao contexto, reconfigurabilidade, adaptabilidade, reflexibilidade, ubiquidade, mobilidade etc., ou mesmo alguns serviços comumente encontrados nos *middlewares* de propósito geral, como provisão de QoS, tolerância a falhas etc. Por este motivo, novas propostas de *middleware* para TVDI têm sido apresentadas na literatura contemplando esses serviços. No caso do FlexTV, tais serviços podem ser implementados através de componentes de software desenvolvidos e incorporados ao *middleware* com estas finalidades.

5. Middleware FlexTV

Da mesma forma que as várias propostas de *middleware* citadas na Seção 4, o *Middleware* de referência FlexTV, elaborado no contexto do Sistema Brasileiro de Televisão Digital, foi concebido com o objetivo de possibilitar a execução de aplicações computacionais nos Terminais de Acesso, isolando das mesmas as características específicas do hardware e do sistema operacional dos referidos Terminais.

O FlexTV possibilitará a execução de aplicações, sejam elas desenvolvidas em linguagem procedural ou em linguagem declarativa. No entanto, tais linguagens de desenvolvimento devem ser independentes de plataforma. O FlexTV incluirá implementações de uma máquina virtual e de um interpretador que possibilitem respectivamente a execução de aplicações procedurais e de aplicações declarativas.

Para funcionarem sobre o FlexTV, as aplicações procedurais devem ser desenvolvidas na linguagem Java [21]. Essa escolha é justificada principalmente pelos benefícios advindos do alinhamento com outros padrões internacionais, tendo em vista que essa é a linguagem de desenvolvimento suportada pelos principais *middlewares* para Televisão Digital existentes no mundo [8][9][10][6][11][12].

Já no que concerne ao desenvolvimento das aplicações declarativas, será utilizada a linguagem NCL [35]. Essa escolha é pautada pelo fato da mesma ser uma linguagem extremamente flexível, possibilitando o desenvolvimento de aplicações bastante complexas, mas ao mesmo tempo oferecendo um alto nível de abstração para os autores. Além disso, essa linguagem permite tratar as especificações declarativas constantes nas principais propostas de *middlewares* existentes no mundo como objetos (fragmentos) de um documento NCL que pode possuir relacionamentos de sincronização com outros objetos do documento. Assim sendo, aplicações declarativas que forem desenvolvidas para tais

middlewares poderão funcionar no *middleware* brasileiro. Aplicações que forem desenvolvidas para o FlexTV, mas também com vistas no mercado internacional, poderão funcionar em *middlewares* de outros países, com algumas restrições, através de conversões automáticas. Finalmente, as aplicações que forem desenvolvidas exclusivamente para o mercado nacional poderão se beneficiar de todas as facilidades apresentadas pela linguagem NCL.

As aplicações para televisão digital podem ser transmitidas através dos canais de televisão para os Terminais de Acesso, juntamente com os fluxos de áudio e vídeo do canal. Cabe ao *middleware* perceber a chegada de tais aplicações e apresentá-las aos usuários no momento adequado, de acordo com marcações transmitidas.

Visando proporcionar um bom desempenho, algumas aplicações mais usuais devem permanecer residentes no próprio Terminal de Acesso, mesmo quando ele for desligado. Tais aplicações devem estar disponíveis para o usuário no momento em que ele ligar o Terminal de Acesso e devem ser descorrelacionadas de qualquer canal de transmissão. Outras aplicações podem ser instaladas no Terminal de Acesso via canal de *broadcast* (quando forem disponibilizadas pela transmissora de televisão) ou via canal de interação. É responsabilidade do *middleware* notificar o usuário sobre a disponibilidade de novas aplicações e, quando autorizado pelo mesmo, tornar tais aplicações residentes. O *middleware* também deverá possibilitar que o usuário remova as aplicações pelas quais não tenha mais interesse; com exceção das aplicações pré-instaladas, que só devem poder ser alteradas mediante uma atualização do código do Terminal de Acesso.

As aplicações residentes podem ser desenvolvidas na linguagem Java, se beneficiando de todas as funcionalidades oferecidas pelas APIs do FlexTV, mas também podem ser desenvolvidas na linguagem nativa do Terminal de Acesso. Nesse caso, tais aplicações são denominadas aplicações residentes nativas. Para facilitar o desenvolvimento dessas

aplicações, o *middleware* deverá disponibilizar uma API reduzida em linguagem nativa.

Visando promover a inclusão social e digital da população brasileira, é extremamente importante que o *middleware* a ser adotado no Sistema Brasileiro de Televisão Digital permita o desenvolvimento das aplicações mais adequadas à realidade desse público, realidade esta que é diferente daquela encontrada em outros países que já possuem sistema de televisão digital definido. Não obstante, esse *middleware* também deve possuir alinhamento com outros padrões internacionais, de forma a possibilitar a exportação do conteúdo televisivo produzido no país, bem como a exibição de conteúdo produzido no exterior.

5.1. Arquitetura do FlexTV

Visando atingir os objetivos expostos no parágrafo anterior, uma arquitetura de referência para o *middleware* do Sistema Brasileiro de Televisão Digital é proposta neste trabalho. Conforme mencionado anteriormente, a principal característica dessa arquitetura é a utilização de uma abordagem baseada em componentes de software que permite que funcionalidades do *middleware* sejam acrescentadas, removidas ou modificadas, através da alteração de seus componentes internos. Essa abordagem permite que seja traçada uma estratégia de evolução para as funcionalidades a serem oferecidas pelo *middleware*. A estratégia adotada parte de uma configuração inicial para o *middleware* compatível com aquela especificada pelo *framework GEM - Globally Executable Multimedia Home Platform*; e prevê que tal configuração seja paulatinamente ajustada, através da substituição de componentes, de forma a tornar o *middleware* cada vez mais adequado às necessidades brasileiras.

A Figura 2 apresenta os principais elementos arquiteturais que constituem o FlexTV. Esses elementos por sua vez estão divididos de acordo com suas funcionalidades em sete grupos principais:

Componentes de Acesso a Fluxos de Baixo Nível

– Contêm os elementos responsáveis por acessar os fluxos de transporte, processá-los e demultiplexá-los nos vários fluxos elementares que os constituem.

Componentes de Processamento de Fluxos Elementares

– Contêm os elementos responsáveis por processar os fluxos elementares, decodificando-os e disponibilizando-os para as camadas superiores. Por exemplo, decodificadores de áudio, vídeo etc.

Componentes de Interface com Usuários

– Reúnem os elementos responsáveis por proporcionar a interação com os usuários através da apresentação de elementos audiovisuais, bem como através da coleta de eventos gerados pelos usuários.

Componentes de Comunicação – São responsáveis por promover a comunicação entre as aplicações que estão sendo executadas sobre o *middleware*, bem como das mesmas com outras aplicações (externas).

Componentes de Gerenciamento – Responsáveis por promover o gerenciamento do *middleware* (monitoramento de contexto, atualização de código etc.) e das aplicações (controle de ciclo de vida, controle de falhas etc.)

Componentes de Persistência – Responsáveis por armazenar dados de forma persistente.

Componentes de Acesso Condicional - Responsáveis pela segurança dos conteúdos de acesso restritos veiculados pelos canais de TV.

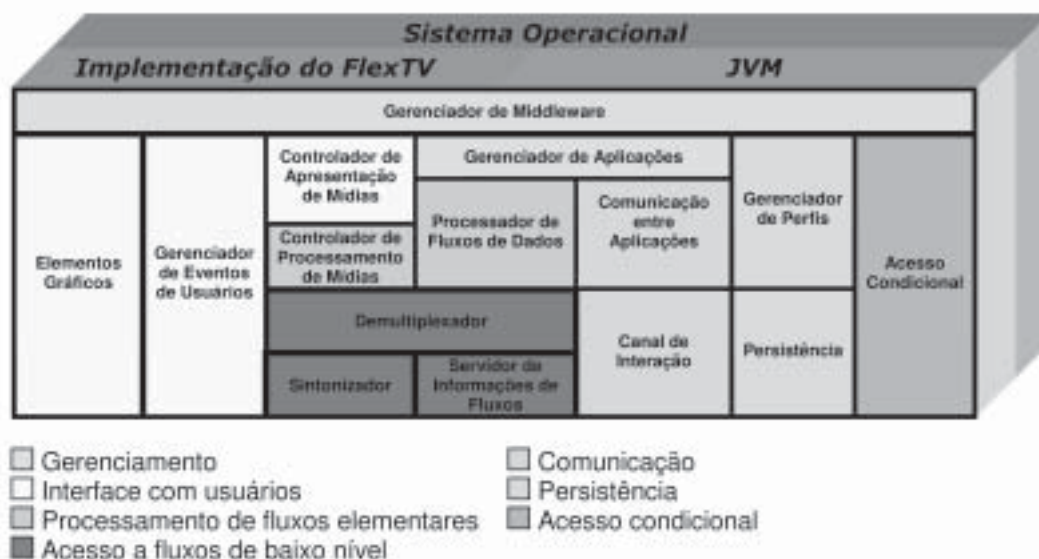


Figura 2. Arquitetura de Alto Nível proposta para o FlexTV.

Esses grupos, por sua vez, são constituídos pelos seguintes elementos:

Acesso a fluxos de baixo nível

Sintonizador – Elemento responsável por selecionar um canal físico bem como um dos fluxos de transporte que estão sendo transmitidos sobre o canal selecionado.

Servidor de Informações de Fluxos – Responsável por identificar e informar aos elementos das camadas superiores quais os fluxos elementares que estão presentes no fluxo de transporte selecionado pelo componente Sintonizador. Esse componente também deverá ser capaz de prover para as camadas superiores a meta-informação acerca dos fluxos elementares que é transmitida pelo *head-end*. Por exemplo, informação sobre agendamento de programas, sobre as opções de legenda disponíveis para um dado programa etc. O projeto interno desse componente deve possibilitar a fácil inclusão de sub-componentes capazes de interpretar os *bitstreams* de informação de diversos padrões de TV Digital, tornando o *middleware* mais independente dos possíveis padrões de transmissão a serem adotados.

Demultiplexador – Responsável por disponibilizar, para as camadas superiores, de acordo com suas necessidades, os fluxos elementares presentes no fluxo de transporte. Esses fluxos elementares podem ser disponibilizados para algum componente de software do sistema (por exemplo, alguma aplicação ou componente de processamento de fluxo elementar) ou podem ser diretamente entregues para algum componente de hardware (por exemplo, um decodificador de vídeo ou áudio).

Processamento de fluxos elementares

Controlador de Processamento de Mídia – Responsável por controlar o processamento dos elementos multimídia e disponibilizá-los para as camadas superiores. É através da API desse componente que se pode, por exemplo, selecionar

quais fluxos de áudio, vídeo e legenda devem ser decodificados, iniciar e parar a decodificação de mídias, acelerar e desacelerar o processamento das mídias etc. Também é de responsabilidade desse componente o processamento de mídias como: HTML, XML e NCL. Na realidade, nesse caso, o componente possui sub-componentes que atuam como formatadores ou processadores específicos para cada um desses tipos de mídia. Operações de escrita direta nos decodificadores por hardware devem ser providas, objetivando a exibição de mídias cujas fontes de dados são fornecidas pelas aplicações. Métodos de pré-carregamento de mídia devem ser desenvolvidos com o intuito de manipular arquivos oriundos do DSM-CC [27] e/ou de alguma outra forma de transporte que utilize canal de retorno. A API para controle de processamento de fluxos multimídia contínuos deve ser baseada no JMF 1.0, uma vez que este é um ponto de convergência da maioria dos padrões de TV Digital, o que representa um meio para manter a compatibilidade e uma forma de integração com estes padrões.

Processador de Fluxos de Dados – Responsável por acessar, processar e disponibilizar para as camadas superiores os fluxos elementares de dados tais como carrossel de dados, fluxos IP transmitidos via canal de broadcast etc. Esse componente deverá prover APIs através das quais as camadas superiores poderão acessar os fluxos de dados (carrossel de dados, fluxos IP etc.) transmitidos em fluxos elementares do fluxo de transporte; bem como poderão ser notificadas da ocorrência de eventos como: chegada de aplicações, eventos síncronos e assíncronos DSM-CC etc.

Interface com usuários

Controlador de Apresentação de Mídia – Elemento responsável pela apresentação dos fluxos multimídia (áudio, vídeo, imagens, texto, documentos HTML, NCL etc.) para o usuário. A exibição das mídias visuais deverá ocorrer em planos de apre-

sentação organizados hierarquicamente: Plano de Fundo, Plano de Vídeo e Plano Gráfico. É de responsabilidade desse componente prover e controlar o acesso a tais planos.

Gerenciador de Eventos de Usuários – Responsável por capturar os eventos gerados pelos usuários, por exemplo, acionamento de teclas de controle remoto, teclado, mouse etc., e repassá-los para as aplicações interessadas, bem como para o gerenciador de aplicações.

Elementos Gráficos – Implementam os principais elementos gráficos que constituem uma aplicação, tais como: botões, caixas de texto etc. Tais componentes deverão poder ser organizados em cenas. Níveis de transparência podem ser associados aos elementos, de forma a possibilitar a exibição dos elementos visuais que estão dispostos hierarquicamente sob o elemento gráfico em questão.

Comunicação

Canal de Interação – Componente responsável por prover interfaces que poderão ser utilizadas pelas camadas superiores para acessar o canal de interação (canal de dados bidirecional que pode ser utilizado pelas aplicações locais para se comunicarem com aplicações externas). Esse componente deverá considerar a utilização de canais de interação intermitentes. Esse componente deverá também prover APIs que permitam: que dados sejam transmitidos assim que o canal de comunicação esteja disponível; que dados sejam transmitidos em horários pré-estabelecidos pelo usuário; ou mesmo que os dados sejam transmitidos em horários informados pelas aplicações. Nos dois últimos casos, o componente deverá configurar e colocar em funcionamento o canal de interação caso o mesmo não esteja disponível.

Comunicação entre Aplicações – As interfaces desse componente podem ser utilizadas pelas aplicações que estão sendo executadas sobre o *middleware* para se comunicarem entre si.

Gerenciamento

Gerenciador de Aplicações – Responsável por: carregar, configurar, instanciar e executar as aplicações que irão funcionar sobre o FlexTV, sejam elas procedurais ou declarativas; controlar o ciclo de vida dessas aplicações, gerenciando seus estados; atribuir níveis de prioridade às mesmas; e identificar e tratar eventuais falhas nas aplicações;

Gerenciador de Middleware – Responsável por possibilitar a atualização de código em tempo de execução do *middleware*, permitindo que componentes sejam substituídos para correção de eventuais erros de implementação ou mesmo alteração de funcionalidades. Esse componente também é responsável por fornecer às aplicações informações acerca do contexto atual do Terminal de Acesso. Contexto esse que é definido em termos de: capacidade, utilização e disponibilidade de recursos como CPU, memória, dispositivo de persistência, rede etc.

Persistência

Gerenciador de Perfis – O objetivo básico desse componente é fornecer para os demais componentes internos do *middleware*, ou mesmo para as aplicações, informações acerca das preferências dos usuários que estão correntemente assistindo a TV. Essas informações podem ser relativas às preferências de horário para uso do canal de interação, bloqueio e desbloqueio de canais, autorização para uso de aplicações tarifadas ou de canais de TV pagos etc.

Persistência – Serviços destinados a permitir que um objeto possa ser acessado após processo que o criou ter sua execução encerrada.

Acesso condicional

Acesso Condicional - O Acesso Condicional é um módulo do *middleware* responsável pela segurança dos conteúdos de acesso restritos veiculados pelos canais de programação.

5.2. Classificação das APIs do FlexTV

A abordagem baseada em componentes para o desenvolvimento da arquitetura do FlexTV permite que cada um de seus componentes seja composto por uma série de sub-componentes. Considere por exemplo o caso do componente Apresentador de Mídias. Esse componente que, como já foi dito anteriormente, é responsável por realizar a apresentação dos diversos tipos de mídias no Terminal de Acesso, é composto na realidade por uma série de sub-componentes, onde cada sub-componente pode ser utilizado para apresentar um determinado tipo de mídia. Por exemplo, um sub-componente para apresentar vídeo, outro para áudio, outro para legendas, e assim por diante. Cada sub-componente apresenta uma API apropriada para a fun-

ção que desempenha. O conjunto das APIs implementadas por todos os sub-componentes constituem a API do seu componente. O conjunto das APIs de todos os componentes, por sua vez, constituem a API do *middleware*.

A abordagem baseada em componentes possibilita que alguns sub-componentes do FlexTV apresentem APIs que são especificadas por outros padrões de *middleware* para TV digital. De maneira análoga, outros sub-componentes podem possuir as APIs que são específicas para o contexto brasileiro e, portanto, não são contempladas por outros padrões. Dessa forma, a API do FlexTV pode ser categorizada em três grupos, representadas pelas cores verde, amarela e vermelha, conforme mostrado na Figura 3.



Figura 3. Classificação das APIs do FlexTV

A API verde está alinhada com outros sistemas internacionais e visa garantir a compatibilidade do FlexTV com tais padrões. Dessa forma, as aplicações que forem desenvolvidas para o SBTVD e que façam uso apenas da API verde poderão também ser executadas nos *middlewares* que disponibilizem APIs compatíveis com a API verde do FlexTV. No sentido inverso, a API verde possibilita também que

aplicações desenvolvidas em outros países, para outros *middlewares* de TV Digital, possam ser executadas no FlexTV, desde que a API do *middleware* para o qual a aplicação tenha sido desenvolvida seja compatível com a API verde do FlexTV. Em suma, é a API verde do FlexTV que possibilita o intercâmbio de aplicações entre os países que adotam algum sistema de televisão digital. A API verde

é fortemente baseada na API do padrão GEM, tendo em vista que vários outros Sistemas de Televisão Digital em outros países têm procurado adotar o GEM como *middleware* de referência (Seção 3).

Os telespectadores brasileiros possuem características sócio-culturais e econômicas extremamente heterogêneas. Algumas características divergem bastante daquelas apresentadas por outros países que já adotaram algum sistema de televisão digital. Por esse motivo, as aplicações de televisão digital desenvolvidas para o público brasileiro podem apresentar recursos extremamente peculiares e demandarem do terminal de acesso e do *middleware* requisitos bastante específicos. No caso do *middleware* FlexTV, a API verde pode não ser suficiente para o desenvolvimento de aplicações que melhor atendam às demandas do público brasileiro. Assim sendo, se faz necessário que o *middleware* FlexTV disponibilize APIs de desenvolvimento que, mesmo não sendo totalmente compatíveis com aquelas constantes em outros sistemas de TV digital, são mais adequadas à realidade brasileira. No FlexTV, essas APIs são classificadas pelas cores amarelo e vermelho, conforme ilustra a Figura 3. É importante destacar que a presença dessas APIs no FlexTV não inviabiliza o intercâmbio de aplicações com outros países que já adotaram algum sistema de TV digital. Desenvolvedores de aplicações para TV digital devem fazer uso apenas da API verde quando houver a intenção de que essas aplicações sejam exportadas para outros países. Num outro cenário, em que as aplicações são desenvolvidas unicamente para o Brasil, os desenvolvedores podem se beneficiar das funcionalidades disponibilizadas pelas APIs amarela e vermelha.

As APIs amarela e vermelha diferem basicamente no que diz respeito ao grau de compatibilidade com a API verde, ou seja, com as APIs apresentadas por outros Sistemas de Televisão Digital. No caso da API amarela, adaptadores de software podem ser desenvolvidos para possibilitar que aplicações

desenvolvidas para tal API sejam executadas em outros sistemas de televisão digital. Um exemplo de funcionalidade que poderia ser disponibilizada pela API amarela seria a apresentação de elementos gráficos mais complexos que os previstos no GEM. Aplicações desenvolvidas no Brasil e que utilizem tal funcionalidade podem ser exportadas para outros países mediante o acoplamento de um adaptador de software que possibilite que os elementos gráficos apresentados pela API amarela sejam construídos a partir dos elementos gráficos apresentados pela API verde, ou seja, pela API do país destino da exportação.

A API vermelha do FlexTV não possui nenhum grau de compatibilidade com APIs disponibilizadas por outros sistemas de televisão digital. Dessa forma, as aplicações desenvolvidas utilizando tal API devem funcionar apenas no Brasil, inviabilizando a exportação das mesmas. Normalmente, a API vermelha está associada às características específicas do Terminal de Acesso. Por exemplo, essa API poderá estar associada a funcionalidades avançadas como reconfiguração dinâmica do *middleware*.

5.3. Aplicações residentes

Independente dos sistemas e padrões para televisão digital, é prática comum a implementação de aplicações residentes no terminal de acesso. Um exemplo deste tipo de aplicação encontrado na quase totalidade dos terminais de acesso é o Guia Eletrônico de Programação (EPG, do inglês *Electronic Program Guide*). Visando atender os requisitos definidos pelo governo brasileiro nos editais que definiram o Projeto do Sistema Brasileiro de Televisão Digital, o consórcio formado para propor o *middleware* FlexTV projetou e está implementando as seguintes aplicações residentes para o terminal de acesso de referência:

Navegador de Aplicações – Aplicação responsável por carregar as outras aplicações residentes, armazenar (instalar) e excluir (desinstalar) aplica-

ções com a anuência dos usuários. Também é função do navegador apresentar meta-informações referente às aplicações residentes.

Navegador de Áudio e Vídeo – Guia de programação Eletrônico que permite o acesso à grade de programação dos canais disponíveis.

Gerenciador de Perfis de Usuários – Aplicação que permite que os usuários do Terminal de Acesso configurem suas preferências no que se refere à utilização deste equipamento. Essa aplicação deve funcionar como interface entre o componente *Gerenciador de Perfis*, anteriormente mencionado, e os usuários do sistema.

Correio Eletrônico – Aplicação de Correio Eletrônico (*E-Mail*) adaptada para televisão. O recebimento de mensagens via canal de difusão (*broadcast*) não deve ter o compromisso de confirmação de entrega (eventualmente enviada para um grupo de usuários). Essa aplicação deve prever a possibilidade de criação de pastas de mensagens personalizadas pelos usuários. Como a conexão pode ser intermitente, deve ser possível realizar uma discagem agendada para enviar e receber as mensagens.

Navegador Web (HTML 1.1) – Navegador Web adaptado para Televisão.

Navegador NCL – Aplicação responsável por receber descrições de documentos multimídia/hipermídia segundo o modelo conceitual hipermídia NCM (*Nested Context Model*) [36] e controlar suas apresentações.

Os navegadores de áudio e vídeo e de aplicações, funcionando em conjunto com o gerenciador de perfis de usuários, fornecem para o usuário de televisão um guia de programação sofisticado. O navegador Web foi introduzido com o intuito de permitir o folheio de páginas HTML simples, o que vem ao encontro de um dos objetivos do Projeto do Sistema Brasileiro de Televisão Digital: a promoção da inclusão digital através da possibilidade de usuários navegarem em documentos HTML simples trans-

mitidos via broadcast. A aplicação de correio eletrônico vai além no processo de inclusão digital, uma vez que permite a atribuição de endereços eletrônicos aos milhões de brasileiros que hoje não tem acesso à Internet. Esses brasileiros passarão a “fazer parte” do “mundo digital”. Obviamente as aplicações de folheio de documentos HTML e correio eletrônico não terão todas as funcionalidades das aplicações semelhantes executadas em computadores conectados à Internet, principalmente no caso dos usuários que não dispõem de recursos para custear o canal de interação. Todavia, mesmo com funcionalidade limitada, essas aplicações poderão representar um importante papel como primeiro contato a estas tecnologias para os que hoje estão excluídos do mundo digital.

O navegador NCL visa fornecer aos projetistas de apresentações multimídia interativas a possibilidade de projetar e executar suas aplicações em aparelhos de televisão, lançando mão de recursos sofisticados de sincronismo e controle da apresentação de mídias que não estão disponíveis em navegadores HTML.

6. Conclusões e Perspectivas Futuras

Este trabalho apresentou o FlexTV, uma proposta de *middleware* para o Sistema Brasileiro de Televisão Digital, cuja arquitetura interna é baseada em componentes de software. A abordagem baseada em componentes permite que as funcionalidades disponibilizadas pelo FlexTV evoluam ao longo do tempo, de acordo com o surgimento de novas tecnologias ou de novos requisitos por parte dos telespectadores ou mesmo dos provedores de conteúdo televisivo.

Visando promover a compatibilidade do FlexTV com os *middlewares* de outros Sistemas de Televisão Digital e, dessa forma, possibilitar o intercâmbio de conteúdo televisivo entre os países que adotam tais

sistemas, o FlexTV segue o *framework* GEM – Globally Executable MHP. O *framework* GEM é adotado pelos *middlewares* das principais propostas de sistemas de televisão digital atuais, como é o caso do ACAP (americano), ARIB (japonês) e MHP (europeu).

As funcionalidades dos componentes do FlexTV são disponibilizadas para as aplicações através de suas interfaces de programação (*Application Programming Interface* – API). As APIs do FlexTV são classificadas de acordo com a sua aderência ou não a padrões internacionais. Assim sendo, as APIs verdes do FlexTV são completamente aderentes ao *framework* GEM e as aplicações que utilizarem apenas essas APIs poderão ser executadas em outros *middlewares* que sejam compatíveis com o GEM. Já as aplicações que utilizem as APIs amarelas do FlexTV poderão ser exportadas desde que sejam desenvolvidos adaptadores de software que permitam a sua execução em outros *middlewares*. As APIs vermelhas do FlexTV, por sua vez, disponibilizam as funcionalidades que servirão para atender as demandas específicas do Brasil. Assim sendo, aplicações que façam uso das APIs vermelhas apenas poderão ser executadas no FlexTV. Cabe aos desenvolvedores decidir quais APIs as suas aplicações devem utilizar, de forma a possibilitar ou não que as mesmas sejam executadas em outros Sistemas de Televisão Digital.

Para uma prova de conceitos, os principais componentes do FlexTV foram implementados de forma a permitir a execução de algumas aplicações compatíveis com o GEM. Outros componentes estão sendo paulatinamente adicionados à implementação de acordo com novos requisitos colhidos a partir da submissão do sistema ao uso por parte de usuários selecionados. Esses requisitos servem também para o refinamento dos componentes já existentes.

Na prova de conceitos, algumas aplicações mais comumente utilizadas são implementadas na lin-

guagem de programação nativa do Terminal de Acesso e mantidas residentes nesse aparelho, enquanto outras são implementadas em Java e carregadas dinamicamente pelo *middleware*, após terem sido transmitidas pela emissora de televisão. Atualmente estão sendo realizados estudos para se determinar quais tipos de aplicações devem permanecer residentes e quais devem ser transmitidas para os Terminais de Acesso.

Como perspectiva futura deste trabalho prevê-se a adição de componentes que possibilitem que sejam incorporadas ao FlexTV características inerentes aos *middlewares* de nova geração, tais como: sensibilidade a contexto, adaptabilidade, tolerância a falhas, entre outras. Tais características podem permitir, por exemplo, que o Terminal de Acesso seja capaz de perceber quais telespectadores estão fazendo uso da televisão em um determinado momento e gerar um modelo de preferências desses usuários com base nos seus históricos de utilização. A partir desse modelo, o Terminal de Acesso seria capaz customizar as aplicações interativas para os seus usuários correntes. Esse é apenas um dos vários cenários possíveis com a evolução do FlexTV.

Referências Bibliográficas

- [1] RNP, GT Middleware. Disponível em <<http://www.rnp.br/pd/gts2004-2005/middleware.html>>. Acessado em Maio de 2005.
- [2] SET, Textos Importantes – SET. Disponível em <http://www.set.com.br/textos_tvdigital.htm>. Acessado em Maio de 2005.
- [3] Pereira, R., “Suporte ao desenvolvimento e uso de frameworks e componentes”. *Tese de Doutorado*, UFRGS. Brasil. Março, 2000.
- [4] Klabunde, C., “Um Estudo sobre Componentes de Software”. *Relatório Técnico*, UFRGS. Brasil. 1999.

- [5] Digital Vídeo Broadcasting. Disponível em <<http://www.dvb.org>>. Acessado em Maio de 2005.
- [6] Advanced Television System Committee. Disponível em <<http://www.atsc.org>>. Acesso em Maio de 2005.
- [7] Association of Radio Industries and Business. Disponível em <<http://www.arib.or.jp>>. Acessado em Maio de 2005.
- [8] Multimedia Home Platform. Disponível em <<http://www.mhp.org>>. Acessado em Maio de 2005.
- [9] DTV Application Software Environment. Disponível em <<http://www.dase.org>>. Acessado em Maio de 2005.
- [10] CS/101A: Advanced Common Application Platform (ACAP). ATSC Proposed Standard, fevereiro, 2004. Disponível em <http://www.atsc.org/standards/cs_documents/cs_101a.pdf>. Acesso em Outubro de 2004.
- [11] OpenCable – Specifications. Disponível em <<http://www.opencable.com/specifications>>. Acessado em Maio de 2005.
- [12] ETSI. TS 102 819: Globally Executable MHP (GEM). ETSI Standard, maio, 2004. Disponível em <http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=19737>. Acessado em Outubro de 2004.
- [13] Rodrigues R.F. “Formatação e Controle de Apresentações Hipermedia com Mecanismos de Adaptação Temporal”. *Tese de Doutorado, Departamento de Informática, PUC-Rio*, Março de 2003.
- [14] Soares L.F.G., Rodrigues R.F., Muchaluat-Saade D.C. “Modeling, Authoring and Formatting Hypermedia Documents in the HyperProp System”. *ACM Multimedia Systems Journal*, Springer-Verlag, 8(2), 2000, pp. 118-134.
- [15] Rodrigues R.F., Soares, L., “Provisão de QoS Inter e Intra Objetos de Mídia em Formadores Adaptativos”, *Simpósio Brasileiro de Multimídia e Web*, Salvador, Novembro de 2003.
- [16] Gomes A.T.A., Colcher S., Soares L.F.G. “Modeling QoS provision on Adaptable Communication Environments”. *IEEE International Conference on Communications*, Helsinque, Finlândia, junho de 2001.
- [17] Elias, G., Lopes, A., Borelli, F., Magalhães, M. F., “Exploring an Open, Distributed Multimedia Framework to Design and Develop an Adaptive Middleware for Interactive Digital Television Systems”. *9th ACM Symposium on Applied Computing (SAC)*, Nicósia, Chipre, 2004.
- [18] Lopes, A., Borelli, F., Elias, G., Magalhães, M., “A Component-Based Configuration Framework for Open, Distributed Multimedia Systems” *18th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Fukuoka, Japão, 2004
- [19] Nelson, M. N., Linton, M., Owicki, S., “A highly available scalable ITV system”, *ACM SIGOPS Operating Systems Review*, Proceedings of the fifteenth ACM symposium on Operating systems principles, December, 1995.
- [20] Omojokun, O., Isbell, C., “User modeling for personalized universal appliance interaction”, *ACM conference on Diversity in computing*, October, 2003.
- [21] Sun, Java Platform Specification. Disponível em <http://java.sun.com/j2ee/j2ee-1_4-fr-spec-license.html>. Acesso em Maio de 2005.
- [22] ITU-R: “A Guide to Digital Terrestrial Television Broadcasting in the VHF/UHF Bands” – Document 11-3/3-E, ITU, Janeiro de 1996.
- [23] ISO/IEC 13818-1. Information technology — Generic coding of moving pictures and associated audio information: Part 1: Systems, 2000.

[24] ISO/IEC 13818-2. Information technology — Generic coding of moving pictures and associated audio information: Part 2: Video, 2000.

[25] ISO/IEC 13818-3. Information technology — Generic coding of moving pictures and associated audio information: Part 3: Audio, 2000.

[26] ISO/IEC TR 13818-6. Information technology — Generic coding of moving pictures and associated audio information — Part 6: Extensions for DSM-CC, 1998.

[27] ISO/IEC 13818-10. Information technology — Generic coding of moving pictures and associated audio information — Part 10: Conformance extensions for Digital Storage Media Command and Control (DSM-CC), 1999.

[28] ETSI, ETSI EN 300 468: Specification for Service Information (SI) in DVB systems, novembro, 2000. Disponível em <http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=8191>. Acesso em Agosto de 2005.

[29] ATSC. A/65B: Program and System Information Protocol for Terrestrial Broadcast and Cable, 2003. Disponível em <http://www.atsc.org/standards/a_65b.pdf>. Acesso em Agosto de 2005.

[30] ARIB. ARIB STD-B10: Service Information for Digital Broadcasting System, ARIB Standard, vs. 3.8, junho, 2004. Disponível em <<http://www.dibeg.org/aribstd/STD-B10-v3.8e.pdf>>. Acesso em Outubro de 2004.

[31] JavaTV. Java Technology in Digital TV Disponível em: <http://java.sun.com/products/javatv/>. Acesso em 10 ago de 2005

[32] JMF. Java Media Framework (JMF). Disponível em: <<http://java.sun.com/products/jmf/>>. Acesso em Agosto de 2005

[33] HAVi *Level 2 Graphical User-Interface*. Specification of the Home Audio/Video Interoperability (HAVi) Architecture. HAVi, Inc. 2001. Disponível em: <<http://www.havi.org>>. Acesso em agosto de 2005.

[34] DAVIC 1.4 Part 2 – DAVIC Specification Reference Models and Scenarios, 1998. Disponível em: <<http://www.havi.org>>. Acesso em 10 ago de 2005.

[35] Soares, L.F.G.; Rodrigues, R.F. “Nested Context Model 3.0: Part 5 – NCL (Nested Context Language)”. Relatório Técnico de Pesquisa da série de Monografias do Departamento de Informática da PUC-Rio. 2005.

[36] Soares L.F.G., Rodrigues R.F. “Nested Context Model 3.0: Part 1 – NCM Core”. Relatório Técnico de Pesquisa da série de Monografias do Departamento de Informática da PUC-Rio. No. 12. abril de 2005.

[37] Broadcast Papers. “TV Transmission Papers”. Disponível em <<http://www.broadcastpapers.com/tvtran/tvtran.htm>>. Acesso em junho de 2004.

Nota

-
- ¹ A tecnologia Java inclui desde a linguagem de programação, um conjunto de classes organizadas em bibliotecas (API Java), um formato de arquivo para as classes (arquivo class) e uma máquina virtual, até os ambientes de execução e desenvolvimento, também chamada de plataforma Java
-