

Júri Virtual I2TV: Uma Aplicação para TV Digital Interativa baseada em JavaTV e HyperProp

Valdecir Becker¹, Rafael Vargas¹, Günter Herweg Filho¹, Carlos Montez²

¹NURCAD – Núcleo de Redes de Alta Velocidade e Computação de Alto Desempenho

²LCMI – Depto de Automação e Sistemas

Univ. Fed. de Santa Catarina – Caixa Postal 476 - 88040-900 - Florianópolis - SC – Brasil
valdecir@nurcad.ufsc.br, (rsvargas, gunter)@inf.ufsc.br, montez@das.ufsc.br

Abstract

No Brasil atualmente existe uma demanda de pesquisas na área de conteúdo (aplicações) e de middlewares para TV Digital Interativa. O trabalho que envolve este artigo se situa neste contexto através do estudo de aplicações interativas e de serviços oferecidos por esses tipos de middlewares. Com esse propósito, dois protótipos de aplicação de Júri Virtual foram implementados, usando as API's de JavaTV e HyperProp. Este artigo apresenta as experiências relacionadas com essas implementações.

1. Introdução

No Brasil, desde 2002 as discussões sobre o sistema de TV digital a ser adotado mudaram de rumo. Até então restrito às tecnologias envolvidas e aos três padrões internacionais já estabelecidos, o debate foi redirecionado para assuntos como o alcance social da nova tecnologia, migração do padrão analógico para o digital considerando as especificidades regionais e nacionais e, pela primeira vez, considerou-se seriamente a possibilidade de desenvolver, se não um sistema de TV digital novo, pelo menos algumas partes importantes.

Resumidamente, pode-se classificar os componentes de um sistema de TV digital em cinco conjuntos de padrões: modulação, codificação, transporte, *middleware* e aplicação. Destes, a codificação e o transporte não devem ser muito diferentes dos padrões amplamente adotados no exterior, baseados nas especificações MPEG, da ISO. A modulação, o *middleware* e as aplicações podem ser mais facilmente adaptados às necessidades e especificidades do país [BECKER 2004].

Duas grandes oportunidades de pesquisa estão nas aplicações interativas (conteúdo para TV Interativa) e nos padrões de *middleware*, que vão dar o suporte para essas aplicações e para o acesso à web.

Na parte de conteúdo, atualmente, vem sendo dada atenção especial à produção e veiculação de programação televisiva. Apesar do Brasil ser hoje um dos maiores consumidores de televisão do mundo, isso não se reflete na produção de conteúdo televisivo. A produção nacional de conteúdo se resume a programas de estúdio, com poucas exceções. Além disso, a produção independente é quase nula. O brasileiro assiste mais à televisão do que quase todos os outros povos, mas se reconhece muito pouco nela [HOINEFF 2003]. Esse reconhecimento é fundamental para o sucesso da TV interativa, pois o sucesso dessa nova mídia dependerá da aceitação da programação e dos serviços interativos oferecidos [BECKER 2003].

Hoje há estima-se que existam mais de 65 milhões de aparelhos receptores do sinal televisivo, ultrapassando o patamar de 90% dos domicílios. Na maioria dos casos, a televisão é a única fonte de informação e de cidadania do brasileiro. Isso demanda estudos detalhados sobre que programas e que serviços devem ser oferecidos à população durante a transição para um modelo digital.

Como a idéia central do governo para o SBTVD é a inclusão digital [DOU 2003], o *middleware* deve fornecer esse serviço, respeitando as diferenças e objetivos de cada telespectador, inclusive os diferentes canais de interação que estiverem disponíveis. Como a penetração da telefonia no país ainda é pequena, outras alternativas devem ser usadas como canal de interação.

Além disso, o próprio *set top box* não deve ser homogêneo, respeitando e se adaptando aos diferentes ambientes e objetivos do telespectador. Dessa forma, torna-se possível ao mercado o oferecimento de

equipamentos com várias opções de *hardware* e *software*. A aquisição de *set top boxes* deve ser semelhante à compra de computadores pessoais, onde o usuário escolhe a capacidade da máquina segundo a necessidade de trabalho que ela vai realizar, ou segundo os recursos financeiros disponíveis.

Dentro desse contexto, tornou-se pertinente o estudo dos componentes de um sistema de TV digital. No projeto I2TV [I2TV 2004], desenvolvido por um consórcio de cinco universidades (UFSC, USP, PUC-Rio, UFPB e UFRN) e as respectivas TVs educativas de cada Estado, foram estudados vários componentes, desde o *hardware* adequado, até conteúdos específicos para essa nova mídia [BECKER 2003].

Na UFSC as pesquisas foram coordenadas pelo Núcleo de Redes de Alta Velocidade e Computação de Alto Desempenho (Nurcad), onde foram desenvolvidos estudos sobre conteúdo e *middleware*. Este artigo trata desses dois temas.

Em nossas pesquisas, foram estudados um *middleware* implementado a partir da implementação de referência do JavaTV, da Sun Microsystems [JavaTV 2004], e o formatador HyperProp, desenvolvido pela PUC-Rio [SOARES 2000], parceira no projeto I2TV. Em ambas as plataformas foi implementado um programa interativo, chamado de Júri Virtual I2TV.

A próxima seção conceitua *middleware* e apresenta os principais que atualmente estão no mercado. Nas seções três e quatro são descritos os *middlewares* baseado em JavaTV e o formatador HyperProp, respectivamente. A seção cinco descreve a aplicação implementada. A seção seis apresenta alguns trabalhos relacionados. Por último, a seção sete traz as conclusões sobre esse estudo e relaciona algumas sugestões de trabalhos futuros.

2. Middleware para TV digital interativa

Com o objetivo de lidar com um sistema complexo – como é o caso de um sistema de TV digital interativa – uma arquitetura de referência costuma ser representada. Essa arquitetura mostra os principais elementos desse sistema na forma de camadas de tecnologias (Figura 1).

Cada camada oferece serviços para a camada superior e usa os serviços oferecidos pela subjacente. Dessa forma, as aplicações que executam na TV digital interativa fazem uso dos serviços de uma camada de *middleware*, que intermedeia toda a comunicação entre a aplicação e o resto dos serviços oferecidos.

Exemplos de aplicações para TV digital são os Guias de Programação Eletrônica (EPG – *Electronic*

Program Guide), comércio eletrônico televisivo (T-Commerce), e acesso a Internet através da TV. Este artigo propõe uma aplicação de Júri Virtual, que será apresentada com mais detalhes na seção cinco.

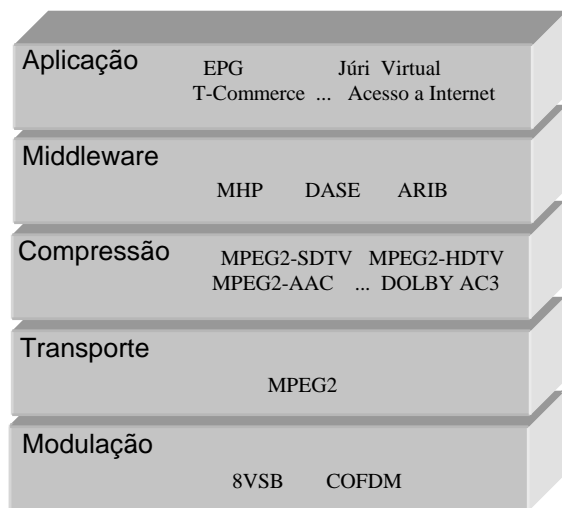


Figura 1. Camadas da TV Digital Interativa

A finalidade da camada de *middleware* – ou camada do meio – é oferecer um serviço padronizado para as aplicações (camada de cima), escondendo as peculiaridades e heterogeneidades das camadas inferiores (tecnologias de compressão, de transporte e de modulação). 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. Essa portabilidade é primordial em sistemas de TV digital, pois é muito difícil considerar como premissa que todos os receptores digitais sejam exatamente iguais.

Buscando evitar uma proliferação de padrões de *middleware*, os principais sistemas existentes de TV digital – norte-americano, japonês e europeu – adotam um padrão de *middleware* em seus receptores digitais. Esses padrões, DASE, ARIB e MHP, respectivamente, são apresentados sucintamente a seguir.

2.1 DASE

O DASE [DASE 2003] foi desenvolvido como um padrão norte-americano para a camada de *middleware* em *set top boxes* de TVs digitais. O DASE adota uma máquina virtual Java como mecanismo que facilita a execução de aplicações interativas baseadas em Java, mas também permite o uso de linguagens declarativas, usadas na *web*, como HTML e JavaScript.

2.2 ARIB

O middleware japonês é padronizado pela organização ARIB [ARIB 2001]. 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 (BML – *Broadcast Markup Language*) baseada em XML (*Extensible Markup Language*), e 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*), baseada no padrão europeu (MHP), e que permite a execução de aplicações interativas baseadas em Java.

2.3 MHP

O MHP [MHP 2003] – padrão europeu de *middleware* para TV digital – busca oferecer um ambiente de TV interativa, independente de *hardware* e *software* específicos, 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 em um conjunto APIs que 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. Além do uso da API Java, o MHP 1.1 introduziu a possibilidade de usar uma linguagem de programação semelhante ao HTML, denominada DVB-HTML.

2.4 Outros middlewares

Existem muitos esforços no sentido de definir um *middleware* interoperável entre os sistemas japonês, norte-americano e europeu. Como exemplo, o GEM (*Globally Executable MHP*) [GEM 2002] é uma tentativa de definir um padrão baseado no MHP, que permita a interoperabilidade entre os *middlewares* DASE, ARIB e MHP.

Apesar desses esforços de padronização poderem levantar dúvidas no sentido de parecer um contrasenso qualquer tentativa de estudar e/ou definir um *middleware* que seja diferente desses padronizados, as tecnologias relacionadas à TV digital ainda não estão totalmente estabelecidas, encontrando-se em fase de estudos no mundo inteiro.

Nas próximas seções apresentamos os nossos estudos sobre a adoção de JavaTV [JavaTV 2004] e sua API e do formatador HyperProp [MUCHALUAT-

SAADE 2003] para o desenvolvimento de uma aplicação de Júri Virtual.

3. O JavaTV

O JavaTV [JavaTV 2004] é uma plataforma para o desenvolvimento e distribuição de serviços para televisão digital interativa, composto por um conjunto de APIs. Por se tratar de uma API Java, possui um alto nível de controle e flexibilidade sobre o aspecto e comportamento das aplicações, tornando possível o desenvolvimento de aplicações interativas.

Desenvolvido pela Sun Microsystems, em parceria com empresas do ramo da televisão digital, é uma extensão da plataforma Java. Visa atender às funcionalidades dos receptores (*set top boxes*) de televisão digital interativa, como:

- *streaming* de áudio/vídeo;
- acesso mediante identificação (CA - *Conditional Access*);
- acesso às Informações de Serviço (SI - *Service Information*);
- controle e troca do canal do receptor;
- controle dos gráficos na tela.

Uma vez que se situa entre a camada do sistema operacional e as aplicações interativas, JavaTV pode ser visto como um *middleware* de TV digital. Os desenvolvedores de serviços interativos podem criar o conteúdo escrevendo uma única vez o código, que será compatível com todos os aparelhos que usam o JVM. Isso evita preocupações com o processador ou sistema operacional de cada receptor. Esta, por sinal, é a principal característica da linguagem de programação Java: sua portabilidade e compatibilidade. O *middleware* desenvolvido a partir dela pode rodar em qualquer aparelho cujo sistema operacional suporte a linguagem. Além disso, as aplicações desenvolvidas serão compatíveis com qualquer receptor que esteja rodando esse *middleware*.

O JavaTV provê em sua API recursos de uso geral, sempre em um alto nível de abstração, tornando possível o desenvolvimento de aplicações sem conhecimento específico das camadas de baixo nível, como o protocolo do SI (*Service Information*), os protocolos de transmissão ou mesmo a rede transmissora. Provê também meios de seleção de serviços, acesso a um banco de dados com as informações do SI, controle do *media player* interno do receptor, e acesso a dados transmitidos no sinal de televisão.

Alguns itens não especificados podem ser encontradas em outras APIs Java, como o suporte a um

canal de retorno ou a reprodução de conteúdo multimídia. O primeiro pode ser encontrado no pacote *java.net* e o segundo, nas APIs do Java Media Framework. Dos recursos não disponibilizados pelo JavaTV e que não possuem nenhum equivalente na linguagem Java, os mais notáveis são a falta de um sistema de gerenciamento de recursos, assim como o fato de não existir suporte a nenhum *broadcast filesystem*, que seria o sistema de arquivos utilizado nas transmissões digitais.

As aplicações desenvolvidas em Java para o *middleware* JavaTV são chamadas de Xlets. Os Xlets são especificados para prover um modo de controle rígido do seu ciclo de vida. Ou seja, possuem meios para que o *middleware* possa saber e controlar se determinado aplicativo estará rodando ou não em determinado ponto da programação, bem como determinar o tipo do aplicativo. Existem quatro tipos básicos de aplicativos: os EPGs, ou *Electronic Program Guides*, que são os guias da programação da rede/canal; as aplicações dependentes de programa, como uma enquete em um programa de auditório; as aplicações independentes, como um aplicativo que mostre cotações da bolsa de valores, vindas de uma rede secundária e que pode manter-se ativa mesmo ao trocar de canal; e as propagandas, que poderiam dar acesso a outros produtos da mesma marca, bem como à possibilidade de efetuar a compra on-line.

4. O Formatador HyperProp

O formatador HyperProp [MUCHALUAT-SAADE, 2003] foi desenvolvido pelo laboratório Telemídia da PUC-Rio para controlar a apresentação de documentos multimídia/hipermídia especificados na linguagem NCL. No atual estágio, o formatador é totalmente desenvolvido em Java e utiliza o JMF, API Java para tratamento de documentos de mídia.

A NCL (*Nested Context Language*) [SOARES 2000] é uma linguagem declarativa para desenvolvimento de documentos hipermídia com sincronização temporal e espacial entre seus documentos e que segue o modelo conceitual NCM (*Nested Context Model*).

A linguagem é baseada em DTM's (*Document Type Definition*) XML, o que caracteriza a simplicidade e, principalmente, a sua facilidade de uso, pois torna a linguagem mais legível, já que incorpora, assim, toda a política de estruturação da meta-linguagem XML.

A NCL está especificada de forma modular. Está dividido em onze áreas funcionais, particionadas em módulos, o que permite sua combinação com outras

linguagens modulares, como o SMIL ou o XHTML. Algumas funcionalidades do NCL são idênticas às do SMIL, o que permite que alguns módulos deste sejam usados por NCL. Como exemplo podemos citar o módulo *BasicMedia*, que define os tipos básicos de objetos de mídia que um documento NCL pode conter. Esse módulo é idêntico ao módulo *BasicMedia* do SMIL, que define os elementos *animation*, *audio*, *img*, *text*, *textstream*, *video* e *ref*.

A linguagem NCL está no momento em sua segunda versão. Foi desenvolvida pela equipe do laboratório Telemídia, da PUC-Rio [TELEMIDIA 2004], com o objetivo de criar uma linguagem genérica, sem limitações, que suprisse melhor e com mais eficiência a especificação de documentos hipermídia com sincronização espacial e temporal.

A principal característica da linguagem reside no fato dela tratar de relações hipermídia como entidades de primeira classe, através da definição de conectores hipermídia. A versão 2.0 também traz algumas características não encontradas em outra linguagem hipermídia. A área funcional *Connectors* permite a definição de conectores hipermídia, através do elemento *xconnector*, e de bases de conectores, através do elemento *connectorBase*. Um conector hipermídia representa uma relação que será usada para criar elos entre nós de um documento e é definido por um conjunto de pontos de interface, chamados *roles*, que especificam a função de um participante da interação, e outro elemento filho chamado *glue*, que descreve como os participantes interagem, completando a definição da semântica da relação. O *XConnector* permite a definição de conectores representando relações de sincronização e de referência.

Se considerarmos como parâmetro padrões já consagrados para autoria de documentos hipermídia, como o SMIL, a NCL traz diversas novas facilidades. É possível, por exemplo, combinar os módulos de NCL entre si, o que permite a criação de linguagens mais próximas das necessidades dos usuários. A linguagem também permite a definição de elos que representam o relacionamento complexo entre dois ou mais componentes do documento.

Apesar da linguagem fornecer um excelente suporte ao ambiente hipermídia, ela possui algumas limitações. Baseado em experiências práticas, como o programa interativo Júri Virtual, descrito a seguir, pode se notar certa limitação em relação a suas funcionalidades quando se tenta aprofundar uma aplicação, pois a NCL não oferece suporte a operações aritméticas, nem estruturas de controle.

5. A aplicação: Júri Virtual

Para testar com uma aplicação prática os *softwares* em desenvolvimento, criamos o programa interativo Júri Virtual I2TV. Pensamos num programa flexível e adaptável aos diferentes requisitos de software e que pudesse ter uma boa aceitação na sociedade como programa televisivo.

Atualmente já há várias iniciativas na televisão em que o telespectador julga alguma atração, votando pelo telefone ou e-mail. Nesses programas, bastante populares sob o ponto de vista da audiência a interação é indireta, precisando de um terceiro meio de comunicação para que a resposta do telespectador chegue a emissora. Como exemplo, pode-se citar os *reality shows* e o programa Você Decide, da Rede Globo, que ficou no ar por quatro anos [BECKER 2003b].

No júri desenvolvido pelo projeto I2TV, o juiz, o réu, os advogados, tanto de defesa como a promotoria, e as testemunhas estão em um fórum realizando o julgamento de um suposto crime. Todo o julgamento decorre normalmente, com uma diferença: os jurados estão em suas próprias casas, assistindo tudo pela televisão. No protótipo implementado, eles podem se comunicar por *chat* e acessar as provas do crime e o inquérito policial através da TV. Ao final do julgamento, quando o juiz concita os jurados a decidirem pela culpa ou inocência do réu, há um espaço de cinco minutos para o debate e ao término desse tempo, todos devem votar inocente ou culpado. O próprio *software*, seja JavaTV ou formatador HyperProp, faz a soma dos “votos” e os remete ao juiz, que declara o réu inocente, caso a maioria dos jurados tenha assim considerado, ou culpado. Em caso de empate dos votos, o réu é considerado inocente.

Como o objetivo foi testar o *middleware* em desenvolvimento, previmos no roteiro das filmagens duas possibilidades, ambas gravadas: o réu poderia ser considerado inocente ou culpado pelos jurados. Na prática o roteiro seria diferente, pois a aplicação seria ao vivo, e a parte final, culpado ou inocente, só seria rodada após o parecer dos jurados.

A princípio o protótipo foi pensado como uma ferramenta de entretenimento, propiciando lazer aos telespectadores. Mas analisando o resultado final, chegamos a conclusão de que também pode ser usado como uma ferramenta de governo eletrônico, ou t-gov (governo televisivo), como preferem chamar alguns [BECKER 2004]. Nesse caso, não só os jurados poderiam estar em diferentes localidades, mas também os advogados, as testemunhas e inclusive, o réu e o juiz.

Em ambas as implementações, descritas a seguir, foram usados PCs como *set top boxes*, e a própria Internet como meio de difusão e canal de interação.

5.1. O Júri implementado em JavaTV

De posse da implementação de referência do Java TV, esta teve que ser modificada pra se adequar às particularidades do “receptor”, neste caso, um PC comum rodando Linux. Como o objetivo do trabalho era desenvolver e testar o *middleware*, num primeiro momento não haveria a transmissão propriamente dita, mas o SI apontaria para um arquivo armazenado em um servidor HTTP. Uma conexão à internet funcionaria como canal de retorno. Algumas outras dificuldades tiveram de ser contornadas, como a falta de um codec no JMF para MPEG2, o que fez com que fosse necessário utilizar outro codec de vídeo compatível com o JMF. No caso a opção foi pelo Cinepak.

A aplicação baseia-se em um servidor *broadcast multithread* para o *chat*, que recebe as conexões e envia os dados (do *chat*) para todos os clientes conectados. É o cliente do *chat* que se conecta ao servidor e conseqüentemente comunica-se com todos os outros clientes. O cliente do *chat*, necessita de dois parâmetros para efetuar a conexão: o nome do jurado e o local onde ele está. Possuindo esses parâmetros, se conecta (por TCP/IP) ao servidor e recebe a lista dos outros clientes já conectados, enquanto estes outros recebem a informação de que mais um se conectou. Todas as informações (falas) enviadas ao servidor são difundidas entre os clientes, de modo que todos possam ler o que cada um escreveu.

Em determinada altura do programa, quando é necessário que os jurados votem a inocência ou a culpa do réu, o servidor envia o comando, e o cliente, recebendo este comando, pergunta ao usuário, que retorna a resposta para servidor. Este faz a contagem e envia o veredicto para o juiz e para os clientes. O cliente, de posse do veredicto, troca o serviço corrente, passando a apontar para o vídeo do final do julgamento. Nesse caso, há dois serviços, um apontando para o vídeo referente ao veredicto culpado e outro ao veredicto inocente. Como é sabido de antemão quais são os serviços necessários, o programa requer ao SI um apontador para o vídeo, e em seguida requer ao JMF um *player* para o mesmo. Finalmente coloca o *player* na tela principal da aplicação e inicia o vídeo.

Esses dois serviços foram implementados porque o programa está pré-gravado. Num programa ao vivo seriam desnecessários.

5.2. O Júri implementado em NCL

Durante todo o processo de criação do júri virtual implementado em NCL, desde seu estudo até a implementação, pudemos perceber o grande potencial da NCL em seu domínio. Foi possível organizar de forma muito clara e simples a aplicação dentro da estrutura da linguagem e de forma relativamente rápida conseguimos uma primeira versão do júri. Porém, como a idéia original do júri foge um pouco do escopo atual da finalidade da NCL e de seu formatador, algumas dificuldades logo apareceram.

Como a aplicação exige interação do usuário (para realizar a votação), tornou-se necessário um meio para que o jurado informasse seu voto, ou seja, precisava-se de uma interface. A NCL não implementa interfaces gráficas. A solução foi usar um applet Java para realizar essa tarefa.

Durante todo processo de implementação do júri contamos com uma importante colaboração da equipe da equipe do laboratório Telemidia da PUC-Rio, que supria, na medida do possível, as nossas necessidades com o formatador. Após relatar os nossos objetivos com a aplicação, o formatador HyperProp foi adaptado para suportar o sincronismo a Applets Java. Isso tornou possível a implementação do sistema de voto feito em Java. Foi implementado, também em Java, um *chat* para comunicação entre os jurados

A aplicação, em sua versão final, usa além do NCL e Java, um banco de dados MySQL, que guarda as informações sobre um julgamento, e os votos para posterior contagem. Todo o sincronismo entre a apresentação das provas, a liberação/bloqueio do sistema de voto, a gerência dos vídeos, ficou por conta do NCL.

6. Trabalhos relacionados

Inúmeras aplicações têm sido desenvolvidas para TV digital. Porém a idéia da interatividade televisiva não é nova. O primeiro programa interativo testado na televisão data da década de 1950 [GAWLINSKI 2003]. Era chamado de “Winky Dink and you”. Consistia num desenho animado em que o personagem (Winky Dink) enfrentava algum problema de locomoção. A interatividade residia numa prancheta adaptada à televisão, em que a criança desenhava as soluções para resolver os problemas do personagem.

As aplicações interativas evoluíram. Na Europa os aplicativos de TV interativa de maior sucesso são a avaliação da atração televisiva por intermédio de enquetes e o pedido de pizzas pela TV [GAWLINSKI 2003]. Em ambos os casos a linha telefônica serve de

canal de retorno. Para o Brasil, [WAISMAN 2002] acredita que os principais estudos sobre TV digital interativa devam se centrar nas aplicações socialmente mais relevantes, como: tele-educação, telemedicina, inclusão digital e governo eletrônico.

Para desenvolver essas aplicações, é fundamental conhecer o *middleware* que dará suporte as aplicações em desenvolvimento, pois não há uniformidade numa mídia ainda em desenvolvimento. O passo seguinte é ter claro o objetivo do programa e a que público se dirige. Baseado nessas variáveis, tanto o formato como as ferramentas a serem usadas podem mudar. [HARTMAN 2002]. Segundo [MAGUIRE 2000] a acessibilidade também é um fator muito importante durante o desenvolvimento de qualquer aplicação interativa, pois se for mal planejada, pode impedir que certas pessoas tenham acesso ao serviço.

No Brasil alguns grupos investigam questões ligadas ao desenvolvimento de *middlewares* para tempo real. Ao longo deste artigo foi citado o HyperProp, um trabalho desenvolvido em [TELEMIDIA 2004]. Além desse grupo, outros trabalhos de pesquisa também vêm investigando questões na área. Em [RIBEIRO 2003], por exemplo, são relatadas experiências da análise da expressividade da API JavaTV em aplicações de TV digital e interativa. O uso de componentes para desenvolver *middlewares* adaptativos é abordado em [BORELLI 2003].

7. Conclusões e trabalhos futuros

Baseado nas experiências dos estudos realizados com o Júri Virtual I2TV e considerando os objetivos do SBTVD, ponderamos que o *middleware* ideal deva ser portátil entre as diferentes plataformas de *hardware* e *software*, para não limitar a indústria e o mercado a um único modelo de *set top box*. Além disso, deve ter canal de interação, que é pré-requisito para a inclusão digital proposta pelo governo. Nesse caso, o acesso à web também se torna necessário. Também nos parece certo que o *middleware* deva saber gerenciar os fluxos elementares, vindos do demultiplexador e do decodificador, dando os destinos adequados ao vídeo, ao áudio, e caso necessário, aos dados.

Como em todos os estudos trabalhamos com diferentes plataformas de *hardware* para *set top box* e diversos cenários mercadológicos e sociais, consideramos fundamental que o *middleware* se adapte a essas diferentes realidades.

O Brasil tem inúmeros problemas que impedem o alcance global da tecnologia, devendo esta se adaptar às necessidades de cada região, classe econômica e

interesses pessoais dos usuários. Isso leva a conclusão de que o canal de interação não será uniforme e que o *middleware* mínimo pode não ter suporte a Java para baratear os custos dos equipamentos. Mas isso não impede que aplicações produzidas em Java e que necessitam da máquina virtual sejam exploradas. Vai depender do *middleware* tratar essas questões, como a de não exibir as aplicações que exijam canal de interação ou Java, caso o *set top box* não tenha esses recursos. É o que chamamos de adaptabilidade aos diferentes cenários. Isso implica em programas televisivos mais sofisticados, capazes de atrair os telespectadores, mesmo que não possuam canal de interação.

O *middleware* desenvolvido a partir do JavaTV tem suporte a todos os requisitos considerados ideais, se caracterizando como uma boa alternativa para o país. Tem código aberto, o que é outro diferencial importante para uma economia já desgastada, entre outras coisas, pela incessante aquisição de tecnologias, implicando num crescente pagamento de *royalties* [BECKER 2004].

Porém, sua implementação de referência é incompleta, o que num primeiro momento pode representar alguns obstáculos ao desenvolvimento. Além disso, necessita de um grande poder de processamento por executar código interpretado, o que representa um ponto negativo, pois eleva o preço dos *set top boxes*. Para um país que precisa urgentemente baratear a tecnologia e torná-la disponível, isso pode representar um ponto crítico.

Pelos testes feitos no Nurcad, o *middleware* desenvolvido a partir do JavaTV precisa no mínimo de um processador Pentium III, 500 MHz, com 64 MB de memória e 100 MB de espaço em disco, além de uma placa de vídeo com 32 MB de memória, ou uma placa decodificadora MPEG-2. Porém, o desempenho ideal apenas é alcançado com 128 MB de memória e 300 MB de espaço em disco.

O formatador HyperProp ainda não tem todos os recursos necessários para implementação do programa proposto. Também tem restrições aos recursos que consideramos ideais no *middleware*. É portátil, pois usa a JVM, e gerencia bem os fluxos elementares. Porém não tem adaptabilidade, nem canal de interação, o que não permite o acesso à web.

Ao contrário do JavaTV, o formatador HyperProp não precisa de muitos recursos computacionais. A versão testada, que usa o JMF para a exibição dos vídeos, necessita apenas de um processador 200 MHz e 64 MB de memória, e uma placa de vídeo com 32 MB ou decodificadora MPEG-2. 30 MB de espaço em disco são suficientes.

Apesar desse aspecto positivo que permitiria *set top boxes* com preços reduzidos, acreditamos que a maior qualidade do formatador HyperProp não está na tecnologia, mas na origem dela. Por ser totalmente nacional e ainda estar em desenvolvimento, estando, portanto, em constante evolução, a linguagem pode ser adaptada para fornecer os serviços considerados ideais num *middleware*.

A aplicação desenvolvida mostrou-se complexa no primeiro momento da implementação, pois tanto o JavaTV como o formatador HyperProp tiveram que sofrer algumas alterações para suprirem todas as demandas do programa. Além disso, mostrou-se adaptável a diferentes cenários, como ambientes sem canal de interação ou com canais de interação de alta velocidade. Em ambos os casos, o telespectador pode acompanhar a programação, atingindo o objetivo da aplicação, que é o entretenimento.

O próximo passo no estudo dos dois *middlewares* apresentados aqui é aumentar a interatividade do júri. No momento só é possível a comunicação dos jurados por *chat*. Num segundo momento essa comunicação deve ser por áudio, e num terceiro momento, por vídeo, o que caracterizaria a interatividade máxima possível nessa aplicação. Considerando, é claro, um cenário ideal, onde o canal de interação tem velocidade suficiente para o envio de vídeo, necessário nesse nível de interatividade. Além disso, os quatro níveis devem coexistir. Num cenário sem canal de retorno, o telespectador assiste ao julgamento, sem as ferramentas de interação. Caso o canal de interação seja de baixa velocidade, a interação fica restrita ao *chat*. Num cenário com canal de interação de média velocidade, a interação pode ser por *chat* e áudio, acrescentando o vídeo num canal de interação de alta velocidade.

Além disso, torna-se pertinente o estudo e a implementação desse mesmo protótipo no MHP, não estudado ainda por falta de uma implementação de referência aberta. Só após completados esses estudos será possível fornecer uma visão mais abrangente sobre qual *middleware* é melhor para o país dentro do contexto dos objetivos previstos pelo SBTVD.

Referências

[ARIB 2001] ARIB STANDARD STD-B24 Version 3.2 VOLUME 1 Version 3.2. Novembro, 2001.

[BECKER 2003] BECKER, Valdecir. **Algumas indagações sobre o futuro**. Observatório da Imprensa, 2003. Disponível em <http://observatorio.ultimosegundo.ig.com.br/artigos/qtv300920031.htm>.

- [BECKER 2003b] BECKER, Valdecir e MOARES, Áureo. Do analógico ao digital: uma proposta de comercial para TV digital interativa. In: III Simpósio Catarinense de Processamento Digital de Imagens. 2003, Florianópolis. Florianópolis: Simpósio Catarinense de Processamento Digital de Imagens, 2003. p. 122-134.
- [BECKER 2004] BECKER, Valdecir e MONTEZ, Carlos. **TV digital interativa: conceitos, desafios e perspectivas para o Brasil.** Florianópolis, I2TV, 2004.
- [BORELLI 2003] BORELLI F., ELIAS, G. **AdapTV: Um Middleware Adaptativo para Sistemas de Televisão Digital Interativa.** (2003) In: Workshop de Teses e Dissertações do Webmidia2003 - Simpósio Brasileiro de Sistemas Multimídia e Web.
- [DASE 2003] **DTV Application Software Environment** - ATSC Standard A/100. Março 2003.
- [DOU 2003] Diário Oficial da União. Decreto-lei 4.901, de 26 de novembro de 2003. Institui o Sistema Brasileiro de Televisão Digital – SBTVD, e dá outras providências. **Diário Oficial da República Federativa do Brasil**, Brasília, 27 de nov. 2003. Seção 1, Pág. 7.
- [GAWLINSKI 2003] GAWLINSKI, Mark. **Interactive television production.** Oxford, Focal Press, 2003.
- [GEM 2002] ETSI – European Telecommunications Standards Institute. **Globally Executable MHP (GEM).** TS 102 819, Novembro 2002.
- [HARTMAN 2002] HARTMAN, Annesa. **Producing interactive television.** Massachusetts, Charles River Media, 2002.
- [HOINEFF 2004] HOINEFF, Nelson. **Produção de conteúdo, eis a questão.** Observatório da Imprensa, 2003. Disponível em <<http://observatorio.ultimosegundo.ig.com.br/artigos/qtv300920031.htm>>. Acesso 17/05/2004.
- [I2TV 2004] INFRA-ESTRUTURA Internet2 para Desenvolvimento e Teste de Programas e Ferramentas para TV Interativa. Florianópolis, I2TV, 2004. Disponível em <<http://www.i2tv.ufsc.br>>. Acesso em 17/05/2004.
- [JavaTV 2004] SUN Microsystems. **JavaTV(TM) API: Technical Overview.** [S.I sn], 2000. Disponível em <http://java.sun.com/products/javatv/jtv-1_0-spec_overview.pdf>. Acesso 11/05/2004.
- [MAGUIRE 2000] MAGUIRE, Martin. **Accessibility of Telecommunications Services.** HUSAT Research Institute – Loughborough University. Leicestershire, 2000.
- [MHP 2003] ETSI – European Telecommunications Standards Institute. **Multimedia Home Platform (MHP) Specification ver. 1.1.1.** TS 101 812, Abril 2003.
- [MUCHALUAT-SAADE, 2003] MUCHALUAT-SAADE D.C., SILVA H.V., SOARES L.F.G. Linguagem NCL versão 2.0 para Autoria Declarativa de Documentos Hiperídia. In: simpósio Brasileiro de Sistemas Multimídia e Web. 2003, Salvador. Salvador. Webmídia, 2003. P. 229-245.
- [RIBEIRO 2003] RIBEIRO, L. A., BECKER, Valdecir, MONTEZ, Carlos, SCHULTER, Alexandre, MELO, Edison, FROHLICH, Antônio A M. Infra-estrutura para recepção de TV Interativa baseada em Set Top box para o Projeto I2TV. In: IV WORKSHOP WRNP2, 2003, Natal. 2003.
- [SOARES 2000] 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**, 8(2):118-134, Março 2000.
- [TELEMIDIA 2004] LABORATÓRIO de Redes de Computadores e Sistemas Multimídia. Rio de Janeiro, Telemídia, 2004. Disponível em <<http://www.telemidia.puc-rio.br>>. Acesso em 17/05/2004.
- [WAISMAN 2002] WAISMAN, Thais. **TV digital interativa na educação: afinal, interatividade para quê?** São Paulo, Escola do Futuro da USP, 2002. Disponível em <http://www.futuro.usp.br/producao_cientifica/artigos/itv.pdf>. Acesso em 20/05/2004.