

UM FRAMEWORK PARA O MÓDULO DE SINTONIZAÇÃO DO MIDDLEWARE GINGA

Aluno: Henrique Siqueira Silva Junger
Orientador: Luiz Fernando Gomes Soares

Introdução

Nos últimos meses, o Brasil deu um importante passo na evolução da tecnologia para televisão, realizando suas primeiras transmissões digitais em TV aberta. Contudo, para que essa transmissão pudesse acontecer, uma padronização do sistema de TV digital teve de ser conduzida pelo governo brasileiro, suportado pelas pesquisas da academia e necessidades da indústria de eletroeletrônicos e de radiodifusão. Esse padrão, chamado de SBTVD (Sistema Brasileiro de Televisão Digital), foi definido combinando diversas características do sistema japonês com tecnologias mais recentes e inovações brasileiras, dentre as quais podemos destacar a camada de *middleware*, inteiramente desenvolvida no Brasil, num estudo conjunto entre as universidades PUC-Rio e UFPB. A este *middleware* deu-se o nome de Ginga.

O Ginga está dividido em três camadas: Ginga-NCL, Ginga-J e Ginga-CC. Esta última é responsável por oferecer serviços de infra-estrutura de TV Digital, abstraindo o Sistema Operacional e hardware para as duas outras camadas. Uma das responsabilidades do Ginga-CC (*Ginga Common-Core*) é sintonizar os diferentes tipos de canais, ou seja, ser capaz de receber um fluxo de áudio, vídeo e dados por diferentes interfaces de recepção. O módulo que realiza essa função é chamado de Sintonizador, e é o principal tema deste trabalho.

Objetivos

O objetivo do trabalho é desenvolver um *framework* para o módulo de sintonização, presente no Ginga-CC, que facilite possíveis especializações voltadas aos diversos tipos de interface de rede. Além disso, como prova de conceito, o framework será especializado para a recepção de fluxos de TV digital por duas interfaces de rede ainda parcialmente suportadas pela implementação de referência do Ginga-NCL: redes de difusão terrestre e redes pessoais Bluetooth. Em outras palavras, o Sintonizador da implementação de referência deve ser reorganizado para que seja genérico, facilitando futuras especializações. Essa característica é importantíssima para que o Sintonizador possa ser ampliado, oferecendo novos serviços e se adequando às futuras necessidades do mercado brasileiro.

Metodologia

Para alcançarmos os objetivos propostos, o trabalho foi dividido em duas partes. Na primeira, diversos estudos foram feitos para que houvesse uma melhor compreensão das tecnologias já existentes, de como o Ginga está estruturado e que alterações poderão ser feitas no módulo Sintonizador. Na segunda parte será realizada a implementação do trabalho, onde o *framework* será desenvolvido e as funcionalidades propostas implementadas.

Assim, foi feito um primeiro estudo sobre os principais sistemas de TV digital existentes, onde foi observado que DVB (sistema europeu), ATSC (americano) e ISDB (japonês) foram organizados em cinco camadas: modulação, transporte, codificação, *middleware* e aplicações. Na camada de Modulação, as técnicas COFDM, QPSK, QAM e 8-VSB são as utilizadas nos diferentes padrões e meios de transmissões. No Brasil, após vários testes, constatou-se que a modulação COFDM, utilizada pelo ISDB era a que melhor atendia os requisitos do mercado nacional, pois possui uma configuração mais inteligente dos segmentos, facilitando na transmissão para dispositivos móveis.

Na camada de Transporte, responsável pela multiplexação dos fluxos elementares (áudio, vídeo ou dados) em um único feixe, o padrão MPEG-2 é unanimidade entre todos os

sistemas estudados. Esses sistemas também dão preferência aos sistemas MPEG, na camada de Codificação de áudio e vídeo, o MPEG-2 Vídeo e MPEG-1 Áudio. Somente os americanos não utilizam apenas o sistema MPEG, preferindo o sistema Dolby AC-3 para codificação de áudio. No sistema brasileiro, o Fórum do SBTVD adotou as mais recentes tecnologias do MPEG, utilizando MPEG-4 para codificação de áudio e vídeo e MPEG2 para a camada de transporte.

Na camada de *Middleware*, principal foco do presente trabalho, foi possível observar as semelhanças e diferenças entre os diferentes sistemas, assim como os recentes esforços para que haja o máximo de compatibilidade entre eles, surgindo daí o padrão GEM (*Globally Executable MHP*). O GEM é uma tentativa de harmonizar os padrões de middlewares baseados em Java, a partir de APIs e funcionalidades em comum, extraídas do MHP (*Multimedia Home Platform*, o *middleware* europeu).

Fundamentado sobre todo esse conhecimento, foi feito um segundo estudo abordando as principais características do *middleware* Ginga, a inovação brasileira do SBTVD. Foi possível constatar como os *middlewares* já existentes influenciaram na sua arquitetura atual, tornando-o compatível com os padrões do GEM e, conseqüentemente, com os demais sistemas em conformidade com o GEM. Uma atenção especial foi dada à camada *Ginga Common-Core* (Ginga-CC), já que é onde se encontra o módulo de sintonização.

Por fim, foi estudado o Sintonizador atual da implementação de referência do Ginga-NCL, como ele está definido e implementado. Foi visto que o módulo só consegue tratar três tipos de interface para recepção de dados: *Unicast*, *Multicast* e arquivos locais. Isto é feito através de três classes (*Tuner*, *NetworkInterface* e *DataProvider*), que são responsáveis por receber o fluxo de TV digital e repassá-lo aos *listeners*, que são as entidades que irão utilizar esse dados recebidos.

O *Tuner* é responsável por gerenciar as *NetworkInterfaces* e os *listeners* repassando as mensagens e notificando os *listeners* sempre que necessário. O *DataProvider* é uma classe abstrata cujos filhos são todos os tipos de provedores. Esses filhos são os reais responsáveis por receber de diferentes tipos de provedores. Eles estão divididos em duas classes, sendo a primeira responsável por receber arquivos locais e a segunda uma classe abstrata que define a recepção de interfaces que trabalham com *sockets*. Essa classe gera mais dois filhos, que recebem os dados enviados via *multicast* ou *unicast*.

Conclusões

Após todos esses estudos, foi concluída a primeira parte do trabalho e adquiriu-se conhecimento necessário para se dar início à segunda parte. Para iniciar esta fase do projeto, o *framework* deverá ser estruturado, de uma forma que abstraia das classes *Tuner* e *NetworkInterface* todos os tipos de provedores.

Dessa forma, para adicionar novos provedores os programadores deverão apenas adicioná-los como um herdeiro do *DataProvider*, tornando o Sintonizador genérico e facilmente atualizável. Feito isso, restará adicionar os dois novos tipos de interfaces (ou provedores), na árvore de hereditariedade do *DataProvider*: um para transmissões via Broadcast e um para redes pessoais, cujo filho será o provedor para redes Bluetooth.

Referências

1 - Junger, Henrique Siqueira Silva. Um Framework para o Módulo de Sintonização do Middleware Ginga. Trabalho de Graduação. PUC-Rio. 2008;

2 - Moreno, Márcio F. – Um Middleware Declarativo para Sistemas de TV Digital Interativa. Dissertação de mestrado, Departamento de Informática, PUC-Rio. 2006.