Uma Arquitetura para Interoperablidade entre Aplicativos NCLua e Serviços da Internet

Ricardo Lucio Braga Reis
Laboratory of Advanced Web Systems – UFMA
Av. dos Portugueses, Campus do Bacanga
São Luís/MA – 65085-580 - Brasil
ricardo.l.b.reis@gmail.com

Mário Meireles Teixeira
Departamento de Informática – UFMA
Av. dos Portugueses, Campus do Bacanga
São Luís/MA – 65085-580 - Brasil
mario@deinf.ufma.br

ABSTRACT
The emergence of Digital TV brought in novel possibilities of interaction between TV viewers and TV programming. There is a clear tendency of convergence between TV and the web. One of the challenges in this scenario is to provide solutions to assist application development in this new media. This work defines and implements an open source architecture that not only streamlines the access of NCL applications to web content but also supports the so-called social networks, one of the strongest current trends.

Keywords
Interactive Digital TV, Applications, NCL, Ginga, Web, XMPP.

1. INTRODUÇÃO
Desde o início da TV, as emissoras tentam proporcionar interação com o telespectador através das mais diversas mídias. No início da TV analógica, essa interação acontecia através de cartas, forma mais tradicional de comunicação. A interatividade acompanhou o avanço tecnológico e logo passou para o telefone, seguida das mensagens SMS, e-mails e *chats*. No cenário atual, com o advento da TV Digital e a convergência entre tecnologias, tornou-se inevitável o surgimento de um elo entre web e TV.

Este artigo tem como objetivo especificar e implementar uma arquitetura para acesso e interação com serviços da internet a partir de mini-aplicativos (*widgets*) NCLua. Ao final, são apresentadas algumas aplicações desenvolvidas a fim de validar a arquitetura proposta.

O artigo está organizado como segue: a Seção 2 apresenta o modelo comumente utilizado no cenário de TV digital atual para integração com a web, além de definir a arquitetura aqui proposta. A Seção 3 discute alguns exemplos de aplicações desenvolvidas sobre essa arquitetura e, por fim, a Seção 4 discute as conclusões e trabalhos futuros.

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. VIII Workshop on Ongoing Undergraduate Research.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9650.
SBC - Brazilian Computer Society

2. ARQUITETURA DE INTEGRAÇÃO ENTRE APLICATIVOS DE TVDI E A WEB
Atualmente existe uma infinidade de conteúdos disponíveis na web, universalmente acessíveis a partir de *browsers*. Entretanto, considerando-se o cenário de aplicativos de TVDI, vê-se que este acesso é ainda restrito e geralmente tratado caso a caso, de maneira específica para cada tipo de aplicação.

Este trabalho, desenvolvido em um contexto de iniciação científica, tem como objetivo propor uma arquitetura genérica e escalável para acesso a conteúdo web a partir de aplicativos de TVDI, particularmente para o caso de *widgets* NCLua. Genérica a fim de permitir o acesso a qualquer conteúdo web disponível, de maneira transparente a partir da plataforma Ginga-NCL, isto é, sem que o desenvolvedor da aplicação NCLua tenha que produzir código específico para cuidar de detalhes de comunicação, dentre outros. E escalável, a fim de suportar múltiplos acessos e clientes concorrentes, sem sobrecarregar o ambiente Ginga-NCL do decodificador.

Deseja-se, em particular, permitir acessos do tipo *Publish/ Subscribe*, em que um cliente NCLua declara seu interesse em consumir informações fornecidas periodicamente por um dado site. Além disso, pretende-se também suportar aplicações de interação entre usuários, seja baseadas em mensagens instantâneas ou as novas redes sociais. Assim, será possível contemplar uma ampla gama das aplicações comuns da web a partir de *widgets* de TVDI.

2.1 Integração da TV Digital com Serviços da Internet
Atualmente a integração da TV Digital com serviços da internet é feita através de uma abordagem simples, que consiste em utilizar um *gateway* que faça a integração entre o aplicativo de TV Digital e o serviço desejado [5]. O *gateway* faz pleno uso da API (*Application Programming Interface*) fornecida pelo serviço da internet, a fim de gerenciar a troca de mensagens HTTP entre os ambientes de TV digital interativa e da web, funcionando como uma camada de tradução entre esses dois mundos. A Figura 1 mostra o caso de um aplicativo Ginga, transmitido por uma emissora de TV, conectando-se através do *gateway* integrador a uma rede social disponível na internet.

Figura 1: Arquitetura para integração entre TVDI e serviços da internet

Tal solução, além de estar fortemente atrelada à API fornecida pelo serviço, o que compromete sua aplicabilidade de forma mais genérica, padece, ainda, de um grave defeito no que tange à sua escalabilidade. Note-se que ela traz consigo o problema do gargalo no *gateway* [5], dado que muitas solicitações simultâneas irão certamente sobrecarregá-lo, consequentemente aumentando o tempo de resposta percebido pelos usuários e podendo até mesmo inviabilizar o uso da aplicação em si .

No ambiente de televisão, não se fala numa escala de dezenas de milhares de usuários, mas sim de milhões ou dezenas de milhões, daí a preocupação que se deve ter desde o início com o desempenho da arquitetura concebida como um todo.

2.2 Arquitetura Proposta

Este trabalho propõe uma arquitetura genérica e escalável para permitir que usuários de TV Digital obtenham dados da internet, utilizando-se de seus serviços mais comuns, como sites, *blogs* e redes sociais. Como já foi comentado, o tradicional modelo requisição/resposta adotado pelo HTTP mostra-se ineficaz neste cenário, pois não é capaz de sustentar a alta audiência esperada dos programas televisivos.

Como alternativa para implementação da arquitetura, foi escolhido o protocolo XMPP [1][2], acrônimo de *eXtensible Messaging and Presence Protocol*, um padrão da IETF (*Internet Engineering Task Force*) para a comunicação em tempo real através de passagem de mensagens XML. Inicialmente chamado de *Jabber*, nome da comunidade que o desenvolveu, o XMPP surgiu para suprir a falta de um protocolo aberto de passagem de mensagens instantâneas. O protocolo XMPP utiliza uma arquitetura cliente-servidor descentralizada. O sistema consiste em uma rede de servidores que se comunicam entre si, permitindo assim a comunicação entre clientes conectados a servidores diferentes. A opção pelo XMPP, neste trabalho, deu-se pelo fato desse protocolo trabalhar com padrões abertos, o que garante a interoperabilidade da arquitetura proposta com os mais variados tipos de serviços e ambientes operacionais.

No cenário aqui proposto (Figura 2), interpõe-se um servidor, executando o protocolo XMPP, entre o *widget* NCL e o serviço da internet, servidor este que irá desempenhar com vantagens as funções do gateway integrador mencionado na Seção 2.2. A aplicação (*widget*) NCL executando no decodificador (*set-top box*) da residência do telespectador registra inicialmente seu interesse por uma informação específica, disponível no servidor XMPP, através do cliente XMPP Ginga. Este estabelece uma conexão com o servidor XMPP a fim de receber as mensagens (*stanzas*) com a informação solicitada. Do outro lado, o servidor XMPP é periodicamente atualizado com dados provenientes do cliente XMPP baseado na web, que os obtém de sites previamente registrados e os repassa ao servidor XMPP.

Note-se que o servidor XMPP representa um ponto de concentração de informações na arquitetura proposta, de um lado sendo alimentado por informações obtidas pelos clientes XMPP Web e, de outro, fornecendo-as aos consumidores (clientes XMPP Ginga) interessados. Os clientes Ginga devem se registrar no servidor XMPP em modo *push*, sinalizando seu interesse em receber informações periodicamente. O servidor XMPP alivia, assim, a carga nos servidores web originais, compilando e disponibilizando a informação em um formato mais conciso, estruturado (um documento XML) e de forma mais imediata do que se fosse obtido diretamente do servidor web original. A interação com os servidores web fica, portanto, a cargo dos clientes XMPP Web, que periodicamente consultam os servidores originais. Os *widgets* NCL, por sua vez, comunicam-se com o servidor XMPP por meio do cliente XMPP Ginga, liberando-se da tarefa (e da sobrecarga) de consultar diferentes sites da web.

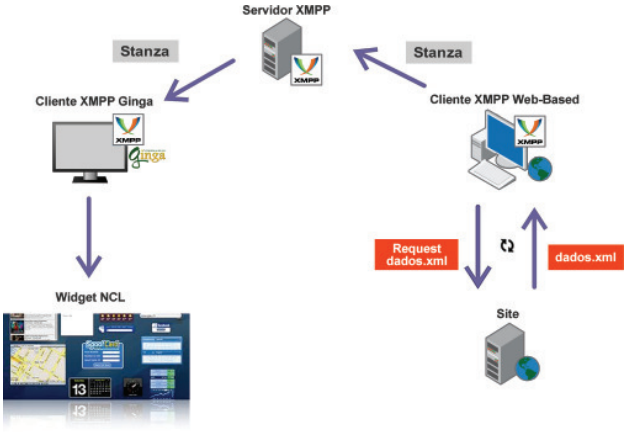

Figura 2: Arquitetura para integração entre *widgets* NCLua e redes sociais / web.

A característica genérica desta arquitetura advém do fato que qualquer site provedor de informações, que as disponibilize como *feeds*, pode utilizar a arquitetura aqui proposta para manter atualizado o seu *widget* NCL, no ambiente de TV Digital. E sua escalabilidade é garantida pela presença do servidor XMPP intermediário, que pode evoluir para um cluster, se assim for necessário.

2.2.1 Cliente XMPP Web

Nessa arquitetura, o cliente XMPP Web utiliza técnicas de AJAX (*Asynchronous Javascript and XML*) para fazer requisições automáticas, de tempos em tempos, visando obter o *feed* com dados atualizados do site provedor de informações.

O cliente XMPP Web utiliza um método de conexão que não requer a manutenção da conexão TCP aberta por muito tempo, embora o padrão para uma rede de passagem de mensagens seja manter a conexão TCP aberta enquanto o usuário estiver *on-line*. Esta otimização no tempo de uso das conexões é possível graças à opção feita pelo BOSH (*Bidirectional-streams Over Synchronous HTTP*) [6][4][3], um protocolo de extensão do XMPP.

Sistemas baseados em BOSH utilizam um gerenciador de conexões que atua como um tipo de *proxy* entre o cliente e o servidor XMPP [4]. Assim, a cada resposta obtida do site, o cliente XMPP Web encapsula o *feed* em uma *stanza* XMPP e a envia, encapsulada em um pacote HTTP, para o gerenciador de conexões, que extrai essa *stanza* e a repassa ao servidor. O servidor se encarrega de enviar esta *stanza* a todos os usuários interessados conectados a ele.

2.2.2 Cliente XMPP Ginga

No contexto da TV Digital, ao se ligar a televisão, o aplicativo abre um *socket* de conexão com o servidor XMPP e todo um processo de negociação e autenticação é executado para o cliente adquirir o direito de receber as *stanzas* de mensagens lá armazenadas.

Após a negociação e autenticação, o aplicativo fica à espera das mensagens enviadas pelo cliente XMPP Web, que alimenta o servidor XMPP com informações da web. Ao receber as *stanzas* de mensagem, o cliente Ginga extrai os dados da mensagem, trata-os e por fim os exibe, quando solicitado, através do *widget* NCLua no aparelho de TV.

3. EXEMPLOS DE APLICAÇÕES

3.1 Clima Tempo

Uma aplicação que acessa o site Clima Tempo[1] foi desenvolvida utilizando a arquitetura proposta, como forma de validá-la com um exemplo realista. O site do Clima Tempo disponibiliza um *feed* em formato XML que foi utilizado para alimentar o *widget* com informações sobre a previsão da tempo em todas as capitais do país.

A aplicação do clima tempo se encaixa perfeitamente na arquitetura, pois o *feed* do site é atualizado uma vez por dia, com isso foi possível desenvolver o cliente XMPP Web para enviar os dados ao *widget* na mesma frequência do site. Entretanto, esta abordagem obriga que o cliente XMPP Web tome a iniciativa de consultar o site e, caso isso tenha que ser feito várias vezes, pode acabar gerando uma sobrecarga desnecessária pelas muitas mensagens de *polling* e resposta. Um modo de diminuir essa sobrecarga por parte do cliente XMPP Web é utilizar um protocolo de extensão do XMPP do tipo *Publish/Subscribe* [7][4][3]. Ele consiste em um cliente que atua como um canal de notificação em tempo real, ou seja, é um nó que publica informações, onde qualquer pessoa pode assinar esse canal para receber as informações desejadas.

No caso do cliente web convencional, a freqüência de requisições ao servidor pode se tornar alta, gerando uma diferença significativa de desempenho se muitos clientes estiverem ativos simultaneamente [4]. Já com o uso do modelo *Publish/Subscribe*, os clientes irão receber as informações (por meio de uma notificação) quando o site publicador as disponibilizar, economizando assim processamento e largura de banda.

3.2 Google Talk

A arquitetura descrita neste artigo não suporta somente aplicativos do tipo *Publish/Subscribe*, podendo ser utilizada também para aplicações interativas. Como estudo de caso, decidiu-se utilizar o Google Talk, serviço de mensagens instantâneas da Google, cujo código está disponibilizado para desenvolvedores[2].

Para tanto, algumas modificações se fazem necessárias. O servidor do Google Talk toma o lugar do site provedor de informações e o cliente XMPP Web continua sendo o mesmo, só que com uma funcionalidade diferente, pois passa a funcionar como um *gateway*, repassando mensagens entre o servidor XMPP e o serviço do Google Talk. O cliente XMPP Web, para atuar como *gateway*, conecta-se com dois servidores XMPP, o do GTalk e o nosso servidor XMPP intermediário, logo para uma mensagem chegar ao seu destino, esta precisa passar por pelo menos dois servidores.

Dessa forma, utilizando a arquitetura de integração, um usuário dos sistemas do Google poderá se conectar à rede Google Talk por meio de um *widget* da TV Digital interativa. Note que, para o *widget* NCL, não faz nenhuma diferença se as informações recebidas do servidor XMPP são mensagens instantâneas ou provenientes de um site como o Clima Tempo, o que demonstra o caráter genérico da arquitetura aqui detalhada.

4. CONCLUSÃO

Este trabalho discutiu e implementou uma arquitetura genérica para integração entre aplicativos de TV digital interativa e serviços comuns da internet, visando assim suprir uma demanda geral por acesso a conteúdo e aplicações da internet a partir da nova plataforma Ginga-NCL de TVDI.

A solução arquitetural apresentada utiliza-se de um servidor executando o protocolo XMPP, atendendo aos critérios de generalidade e escalabilidade colocados como premissa inicial deste trabalho. Um conjunto de aplicativos foi desenvolvido a fim de validar a funcionalidade da arquitetura, sendo aqui destacados dois deles, um segundo o modelo *publish/subscribe* e outro, de mensagens instantâneas. Ambos foram implementadas como *widgets* NCLua, com resultados bastante satisfatórios.

Como trabalhos futuros, pretende-se investigar algumas variações na arquitetura, com vistas a melhorar ainda mais seu desempenho e escalablidade. E ainda, deseja-se realizar sua integração com o Facebook e Twitter, duas das redes sociais mais populares no momento, ambas com APIs livremente disponibilizadas na web.

REFERÊNCIAS

[1] SAINT-ANDRE, Peter. “Extensible Messaging and Presence Protocol (XMPP): Core”. 10/2004. http://xmpp.org/rfcs/rfc3920.html.

[2] SAINT-ANDRE, Peter. “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence”. 10/2004. http://xmpp.org/rfcs/rfc3921.html.

[3] MOFFITT, Jack. Professional XMPP Programming with Javascript and JQuery. Wrox. 2010.

[4] SAINT-ANDRE, Peter; SMITH, Kevin; TRONÇON, Remko. XMPP: The Definitive Guide. O’Reilly. 2009.

[5] C. GHISI, Bruno; F. LOPES, Guilherme; SIQUEIRA, Frank. Integração de Aplicações para TV Digital Interativa com Redes Sociais. WEBMEDIA 2010. Belo Horizonte – MG. 2010.

[6] PATERSON, Ian; SMITH, Dave; SAINT-ANDRE, Peter; MOFFITT, Jack. [XEP-0124] Bidirectional-streams Over Synchronous HTTP (BOSH). http://xmpp.org/extensions/xep-0124.html.

[7] MILLARD, Peter; SAINT-ANDRE, Peter; MEIJER, Ralph. [XEP-0060] Publish-Subscribe. http://xmpp.org/extensions/xep-0060.html.

[1] http://www.climatempo.com.br/

[2] http://code.google.com/intl/pt-BR/apis/talk