Semantic Integration - Uma Extensão do Núcleo do *Middleware* Ginga

Cecília F. Silva[1], Priscilla Vieira[1], Natasha C. Q. Lino[1]

[1]Centro de Informática - Universidade Federal da Paraíba (UFPB)

João Pessoa, PB, Brasil

cecilia@lavid.ufpb.br, priscilla.vieira@lavid.ufpb.br, natasha@ci.ufpb.br

Abstract. *In this paper, we describe a Ginga middleware extension, developed in the context of the Knowledge TV (KTV) platform, a new layer to provide semantic services to the convergent TV environment (iDTV and Web), which aim is to share and reuse data, resources and service applications. We describe the Semantic Integration, an API of the Knowledge TV platform, that does the communication between the KTV server and the Ginga middleware. This is responsible for collecting metadata of TV schedule and send via return channel to the KTV server.*

Resumo. *O presente artigo descreve uma extensão do middleware Ginga, desenvolvida no contexto da plataforma Knowledge TV (KTV), que propõe uma camada semântica ao ambiente da TV Digital. É descrito o Semantic Integration, API da plataforma Knowledge TV, integrada ao middleware, responsável pela comunicação entre o servidor KTV e o middleware Ginga. Esta é responsável pela coleta de metadados da programação e o envio via canal de retorno para o servidor KTV.*

1.Introdução

A TV Digital interativa (TVDi) [Montez *et.al*, 2004] surgiu em meados da década de 90 nos Estados Unidos com o objetivo de substituir a TV analógica. A TVDi surgiu como uma nova tecnologia que propõe: (I) eliminação de ruídos, (II) distorções de imagens e (III) fantasmas. Entretanto, em pesquisas recentes, a potencialidade de interatividade é uma das características sendo mais investigadas para elevar o estado da arte dessa tecnologia emergente.

Tendo em vista esse cenário de TVDi, surgiu a proposta do projeto *Knowledge TV* (KTV) [Lino *et.al*, 2011] que, de forma geral, é uma camada que permite integrar serviços semânticos ao *middleware* brasileiro de TV Digital, o Ginga [Souza Filho *et.al*, 2007], permitindo interações com o usuário por meio de serviços e agentes inteligentes. O trabalho apresentado neste artigo está inserido no contexto do projeto KTV, e visa a criação de uma API (*Application Programming Interface*), a *Semantic Integration*, com o objetivo de fazer uma extensão do *middleware* Ginga, permitindo uma comunicação entre o Ginga e a plataforma KTV. Esta comunicação se dá por meio da captura das informações básicas da programação disponíveis no fluxo de programação que se encontram no *middleware*; e envio dos dados e metadados coletados a um servidor do KTV. Esta API foi validada em cenários de serviços de recomendação de conteúdo e consulta semântica do KTV.

3. Knowledge TV

O projeto KTV tem como objetivo principal adicionar serviços semânticos a TV Digital Interativa (TVDi), fazendo com que o *middleware* de TV compartilhe e reutilize dados, recursos e serviços de suas aplicações, potencializando a interatividade nas aplicações.

Conceitualmente, o KTV foi projetado com uma arquitetura cliente-servidor. Nesta arquitetura existe um módulo cliente (situado no Set-Top-Box e inserido no núcleo do *middleware*), e o módulo servidor, onde ocorre o processamento sofisticado de informações para dar suporte aos serviços oferecidos pela camada semântica provida pelo KTV.Esta arquitetura cliente-servidor está representada na Figura 1.

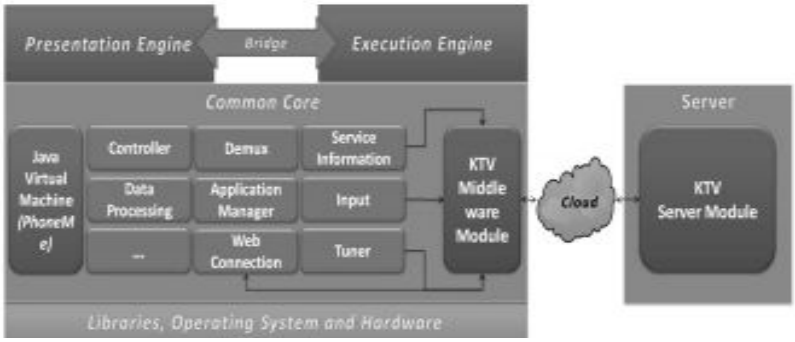

Figura 1. Arquitetura Conceitual do KTV

O módulo servidor é responsável pelos serviços avançados providos como a Consulta e a Recomendação Semântica de conteúdo por meio dos dados enviados pelo módulo cliente [Vieira and Lino, 2012]. O módulo cliente, por sua vez, está inserido na arquitetura do Set-Top-Box como uma camada integrada ao *middleware*.

Este trabalho concentra-se no módulo cliente da plataforma KTV, com a função principal de recuperar informações da programação e do usuário do fluxo de programação disponível no Servidor de Informações (SI) e no *Tuner*, componentes do núcleo do Ginga.

Com o objetivo de executar as funcionalidades do módulo cliente, criou-se uma API titulada *Semantic Integration* especificada na linguagem de programação C++, uma vez que também é a linguagem de especificação do *middleware* Ginga. A API é a principal contribuição deste trabalho para o projeto *Knowledge TV* e será abordada detalhadamente na próxima sessão.

4. *Semantic Integration*

A API *Semantic Integration* proposta neste trabalho é responsável pelo módulo Cliente do KTV e é uma extensão do *middleware* Ginga. Esta API é formada por um conjunto de agentes que obtém no SI e no Tuner – componentes do *middleware* - informações necessárias (incluindo metadados) para os serviços de Consulta Semântica e Recomendação [Vieira and Lino, 2012] do KTV e enviam ao servidor de conhecimento do KTV. Dar-se o nome de API a um conjunto de métodos que juntos, executam funções de um software. Esses métodos em geral ficam abstraídos do usuário final que terá conhecimento apenas da utilização do serviço, e não da implementação em si. No caso desta API em específico, as seguintes funções são executadas:

- Descobrir, por meio do Tuner, quando o usuário sintoniza em um determinado canal e por quanto tempo permanece no mesmo;
- Obtenção de metadados da programação assistida por meio do SI;
- Gerar documentos XML (*eXtensible Markup Language)* com os metadados obtidos;
- Enviar os documentos XML para o servidor;
- Implementação e validação da API de acordo com a arquitetura do Ginga.

Uma vez enviado os documentos XML para o servidor, os metadados são traduzidos para o formato de ontologia e incluídos na *Core Ontology* [Araújo, 2011] - ontologia núcleo do projeto KTV para enfim poderem ser utilizados para os serviços semânticos do KTV.

A arquitetura do *Semantic Integration* é dividida em cinco agentes básicos: *Monitor* e *Provedor*, *ISemantic Agent*, *MMContent* e *ContentDescription,* onde os dois primeiros agentes comunicam-se diretamente com o Ginga através dos componentes *Tuner* e *SI* e o *ISemantic Agent* recebe os metadados capturados por eles. Os agentes *MMContent* e *Content Description* estão incluídos no agente *Provider Agent*. A arquitetura do *Semantic Integration* é descrita na Figura 2 e cada um desses agentes serão discutidos detalhadamente em seções individuais.

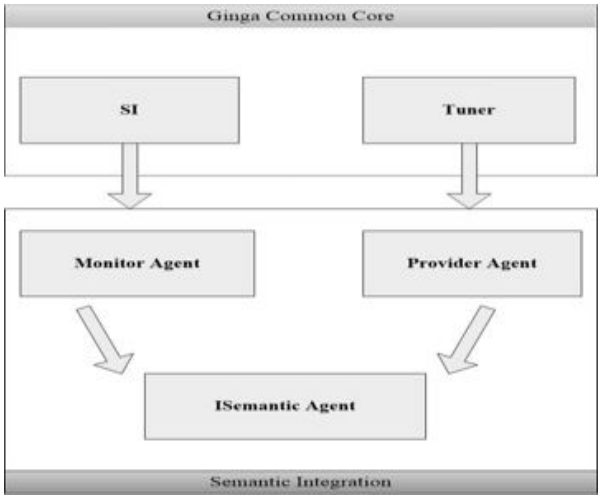

Figura 2. Arquitetura Semantic Integration

4.1 *Monitor Agent*

O Agente Monitor, desenvolvido neste trabalho, tem como função principal recuperar as informações básicas da programação assistida pelo usuário, – nome do programa, nome do canal, classificação e gênero - o intervalo de tempo que o programa foi assistido e o ID do usuário, localizado no *Set-Top-Box*. A recuperação das informações básicas da programação é obtida através do Servidor de Informações (SI) em conjunto com o *Tuner*, componentes do *Ginga Common Core*. O *Tuner* também é responsável por descobrir quando o usuário assistiu o programa e por quanto tempo permaneceu nele, além do ID do usuário. Os metadados obrigatórios, assim como o ID do usuário e as informações de tempo de duração em um determinado canal são capturados e enviados periodicamente ao servidor para tratamento semântico das mesmas.

Para realizar as funções propostas pelo agente, foi construída a classe *UserMonitorAgent*, responsável por monitorar as mudanças de canal a partir da

integração do mesmo com o *Tuner* e buscar no SI as informações mínimas sobre o programa. Este agente torna-se um dos mais importantes da API uma vez que é o único que tem uma comunicação direta com o *Tuner,* podendo o servidor, através das informações de tempo capturadas neste agente, descobrir se o usuário de fato gostou ou não do programa.

4.2 Provider Agent

O Agente Provedor tem como função principal recuperar as informações da programação transmitida pela televisão. Isso pode ser feito de duas formas: i) capturando as informações disponíveis no fluxo que são processadas no SI, de forma semelhante ao que é feito no componente *Monitor* e; ii) obtendo as informações a partir da Web, que podem ser adquiridas através de busca em repositórios de dados. Para este trabalho, será abordado apenas a captura das informações a partir do fluxo de programação processado no SI.

Os metadados recuperados são utilizados pelos agentes *MMContent* e *ContentDescriptio*. Sendo assim, a classe *ProviderAgent*, uma superclasse, e os métodos da mesma são os mesmos das classes-filhas *ContentDescription* e *MMContent*. Alguns destes metadados serão abordados nas próximas subssessões. A maior contribuição deste agente está no envio de informações mais detalhadas da programação contidas apenas no SI.

4.3 *Content Description*

O agente *Content Description* tem como função principal recuperar a descrição completa do conteúdo transmitido. Tal descrição, assim como ocorre na classe *MMContent*, não é recuperada diretamente pelo SI (componente do *Ginga Common Core*), e sim pelo *Provider Agent*.

É importante ressaltar que a criação de uma classe apenas para obter toda a descrição de conteúdo dar-se com o objetivo de abstrair outros metadados ou funções disponíveis no *Provider* (como as que são usadas para a classe *MMContent*) e enviar para o servidor apenas a descrição em um arquivo *XML*, o que facilita as atividades no módulo servidor. Além disso, o uso do *Provider* para capturar metadados no lugar de duas classes distintas (*Content Description* e *MMContent*) no SI evita a sobrecarga do mesmo.

Alguns dos principais metadados recuperados por esta classe a serem implementados além dos metadados obrigatórios são: gênero, subgênero, palavras-chave, classificação etária e idioma.

4.4 MMContent

O componente *MMContent* representa uma instância de um programa multimídia na TV, ou um evento existente em um serviço. As informações capturadas nesta classe não são obtidas diretamente pelo Ginga e sim pelo *ProviderAgent*. Cada objeto desta classe gera uma instância que representa o conteúdo capturado pelo provedor de informações (*Provider*). Além disso, o componente deve conter também um método responsável pela coleta de todos os metadados contidos na classe *ContentDescription*, que podem ser transmitidos de forma opcional, diferentemente dos outros métodos que estão vinculados ao agente *Provider*. Esta classe recuperará metadados que contém a sinopse da mídia e o nome.

4.5 ISemantic Agent

Por fim, o *ISemantic Agent* é o agente principal da API *Semantic Integration* e é responsável por integrar e validar os componentes do KTV na arquitetura do *middleware*. Sua função principal é realizar o tratamento dos dados, extraindo os metadados das classes *UserMonitorAgent*, *MMContent* e *ContentDescription* e armazená-los em arquivo texto *XML* e enviá-los ao servidor de serviços semânticos do KTV. A Figura 3 representa as funções do agente.

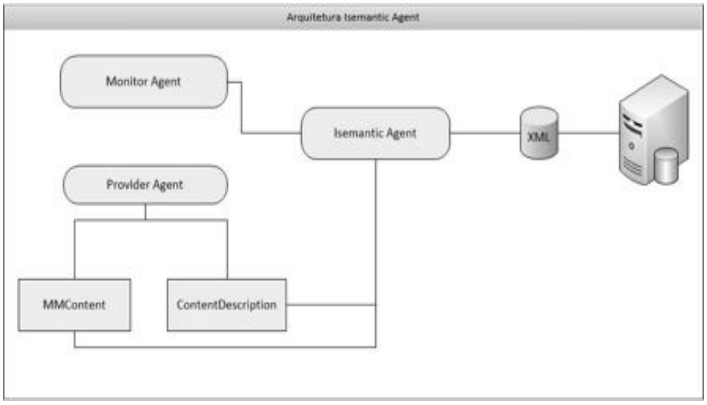

Figura 3. Arquitetura ISemantic Agent

Sendo assim, a classe *ISemanticAgent* criada é responsável por obter e encapsular os metadados das classes *UserMonitorAgent, Content Description e MMContent*, armazená-los em arquivos XML e enviar estes arquivos XML para o servidor do KTV, além de validar e integrar a API ao Ginga seguindo a arquitetura interna adotada pelo *middleware*.

5. Validação

Para a validação da API proposta neste trabalho foram utilizados estudos de caso, com dois serviços semânticos no contexto de ambientes de convergência digital (TV e Web): recomendação e consulta semântica.

No primeiro foi realizada recomendação de conteúdo no contexto da plataforma KTV, o componente *Semantic Intregration* foi primordial, pois se faz necessário os metadados do histórico do usuário, para posterior integração semântica com a CoreKTV [Vieira and Lino, 2012].

No segundo, antes de realizar o enriquecimento semântico para as consultas, se faz necessário a identificação do contexto no qual o usuário está ao solicitar o serviço. Neste contexto o *Semantic Integration* é responsável pela captura dos metadados do conteúdo no qual se tem interesse [Amaro *et.al*, 2012] . Desta forma, é possível identificar cenários em que o *Semantic Integration* possui função fundamental, abstraindo dificuldades do *middleware* para desenvolvedores.

6. Trabalhos Relacionados

O trabalho RecommenderTV [Ávila, 2010], apresenta algumas interseções com a abordagem apresentada neste trabalho, adicionando agentes ao núcleo comum do *middleware* Ginga, responsáveis pela execução das operações de coleta de informações do usuário e do conteúdo. No entanto as principais diferenças para o cenário levantado

neste trabalho reside na não utilização de padrões que promovem a interoperabilidade entre sistemas e o compartilhamento de informações, tal como a ontologia CoreKTV.

7. Conclusões

Este artigo apresentou uma extensão do *middleware* Ginga para a plataforma *Knowledge TV* (KTV). Este trabalho teve como contribuição principal prover à plataforma KTV a integração com o *middleware* Ginga, realizada por meio da captura dos metadados do fluxo de programação enviados pela emissora para o Ginga. A integração do *middleware* Ginga com o KTV tem-se sua importância uma vez que o Ginga é o *middleware* brasileiro de TVDi, sendo assim, a construção de uma API que integre o mesmo e o KTV insere o projeto em um contexto real e muito utilizado no Brasil. Esta extensão está sendo validada e utilizada através dos serviços avançados de Consulta Semântica e Recomendação sendo construídos na plataforma KTV.

8.Trabalhos Futuros

Como trabalhos futuros, planeja-se um maior refinamento dos agentes por meio da implementação de métodos que capturam metadados que até o momento não estão presentes, mas que proveriam mais informações sobre o que o usuário está assistindo e geraria mais dados para o servidor de conhecimento do KTV. Além disso, planeja-se também a exploração de um agente Provider que obtém informações a partir da Web, no lugar de um fluxo de informações disponíveis no middleware de TV Digital.

Referências

Amaro, M. A. L.; Lino, N. C. Q.; Siebra, C. A. and Guedes, A. L. V. (2012). The Knowledge as a Service Metaphor and its Use for Building Convergence Environments.*WebMedia 2012 - 18th Brazilian Symposium on Multimedia and the Web*, São Paulo, Brazil, 2012.

Araújo, J. P. C. (2011). CoreKTV - Uma infraestrutura baseada em conhecimento para TV Digital Interativa: um estudo de caso para o middleware Ginga. Master of Science Dissertation at Programa de Pós-Graduação em Informática (PPGI), Centro de Informática, Universidade Federal da Paraíba (UFPB), Setembro, 2011.

Ávila, P. M. RecommenderTV: Suporte ao Desenvolvimento de Aplicações de Recomendação para o Sistema Brasileiro de TV Digital. Dissertaçao de Mestrado. Universidade Federal de São Carlos, Brasil, 2010.

Lino, N. C. Q.; Araújo, J. P.; Anabuki, D.; Patrício Junior, J. C.; Batista, M.; Nóbrega, R.; Amaro, M.; Siebra, C. A. and Souza Filho, G. L. de. (2011). Knowledge TV. Proceedings of the 9th EuroITV Conference (EuroITV 2011), Lisbon, Portugal, 2011.

Montez, C; Becker, V. TV Digital Interativa: Conceitos e Tecnologias. In: WebMidia e LA-Web 2004 – Joint Conference. Ribeirão Preto, SP, Outubro de 2004.

Souza Filho, G. L. de; Leite, L. E. C.; Batista, C. E. C. F. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. Journal of the Brazilian Computer Society. 2007.

Vieira, P. K. M. and Lino, N. C. Q. (2012). Recomendação de Conteúdo em Ambientes de Convergência Digital Incorporada ao Middleware Ginga. *Workshop de Teses e Dissertações - WebMedia 2012 - 18th Brazilian Symposium on Multimedia and the Web*, São Paulo, Brazil, 2012.