

Suporte ao Desenvolvimento de Sistemas de Recomendação em Ambientes de Televisão Digital Interativa

Paulo Muniz de Ávila¹, Sergio Donizetti Zorzo¹

¹ Universidade Federal de São Carlos – UFSCar, São Carlos – SP, Brasil
Lince – Laboratório para Inovação em Computação e Engenharia, UFSCar, São Carlos, Brazil
{paulo_avila, zorzo}@dc.ufscar.br

Abstract. With the dissemination of the technologies associated to the interactive multimedia, new applications and forms of interaction is appearing. One of the most promising technologies of interactive video – the Interactive Digital Television – has as one of its objectives to provide new interaction ways to the viewer. In this context, this work presents an architecture to support the development of recommendation systems in interactive digital television environments, compatible with the reference implementation of middleware Ginga. The implementation of the architecture demanded the inclusion of new features to the middleware Ginga none exists in the implementation of reference. Finally, we are reported the results obtained from the experiments with the system of recommendation implemented.

Keywords: Personalization, Multimedia, Profiling, Recommendation System, Digital TV, Middleware Ginga

1 Introdução

Uma das mais promissoras tecnologias de vídeo interativo - a televisão digital interativa - Tem como objetivo fornecer novas formas de interação ao telespectador. Uma das principais vantagens do advento da televisão digital interativa, especialmente no contexto brasileiro, é a possibilidade do aumento da interação entre o telespectador e os terminais de acesso e o aumento da capacidade de transmissão de novos canais com a mesma largura de banda. Ao mesmo tempo em que novos canais surgem em virtude do aumento da capacidade de transmissão, torna-se necessário a criação de meios que possibilitem aos telespectadores navegar entre estes canais.

O Guia de Programação Eletrônico (EPG) auxilia os telespectadores nesta tarefa. Porém, à medida que novos canais são disponibilizados é inevitável que ocorra sobrecarga de informação tornando os sistemas de EPG inadequados. Desta forma, os sistemas de EPG tradicionais tornam-se pouco atraentes porque os telespectadores necessitam de muito tempo para navegar nas dezenas de páginas disponibilizadas, a procura de seu programa predileto. É nesta situação que surge a necessidade dos sistemas de recomendação personalizados.

Diferentemente das funções do EPG que possibilitam navegação básica, um sistema de TV personalizado pode criar um perfil para cada telespectador, e recomendar programas que melhor enquadre neste perfil, evitando que o telespectador tenha que navegar por várias páginas do EPG a procura de seu programa predileto. Para que seja possível usufruir dos benefícios oferecidos pelo sistema digital (novos canais, aplicações interativas), os telespectadores que possuem seus equipamentos de TV analógicos necessitam adquirir um novo equipamento denominado set-top box (STB). STB é um dispositivo que trabalha conectado a TV e converte o sinal digital recebido dos provedores de serviços para áudio/vídeo que a TV analógica é capaz de apresentar. O STB para tirar vantagem dos recursos ofertados pela televisão digital necessita de um *middleware*, que é uma camada de software que faz a ligação entre o *hardware* e as aplicações interativas. Ginga [1][2] é o *middleware* do sistema brasileiro de TV digital (SBTVD), que permite através de seus componentes Ginga-NCL [1] e Ginga-J [2] executar aplicações declarativas e procedurais. Ginga-NCL executa aplicações declarativas escritas em *Nested Context Language* (NCL), enquanto Ginga-J é capaz de executar aplicações procedurais baseadas em Java, conhecidas como Xlets [3].

O objetivo desse trabalho é propor uma arquitetura para suporte ao desenvolvimento de sistemas de recomendação, através da inclusão de um novo módulo no *middleware* Ginga-NCL, intitulado *Recommender*. O objetivo principal do módulo *Recommender* é gerenciar o monitoramento do comportamento do usuário, persistir essas informações, realizar mineração de dados e persistir os resultados obtidos pela mineração, tornando-os disponíveis aos sistemas de recomendação que utilizem os serviços do módulo *Recommender*. Dessa forma, o módulo *Recommender* provê um conjunto de serviços, através de APIs (*Application Program Interface*) com propósitos bem definidos.

Este trabalho está organizado da seguinte forma: a seção 2 descreve sobre os provedores de serviços e metadados, a seção 3 apresenta os trabalhos correlatos; a seção 4 discute a implementação do módulo *Recommender*, a seção 5 detalha as experiências e resultados obtidos e, por fim, a seção 6 apresenta a conclusão final desse trabalho.

2 Provedores de Serviços & Metadados

Nesta seção apresentamos conceitos importantes referentes à transmissão do sinal digital e os metadados providos através das tabelas SI (*Service Information*).

No sistema de Televisão Digital Interativa (TVDI) é previsto, além da transmissão de áudio e vídeo, o envio de dados ao telespectador. Os provedores de serviços podem enviar por *broadcast* aplicativos escritos em Java, conhecidos como Xlets ou aplicações NCL, ambos definidos no padrão do SBTVD. Além das aplicações, os provedores de serviços enviam tabelas que transportam informações ao STB.

Os sistemas abertos de televisão digital adotam o padrão MPEG-2 System – *Transport Stream* [9] para a multiplexação dos fluxos elementares. Para compreendermos o que são fluxos elementares, precisamos entender como é feita a construção do sinal digital. Inicialmente, o áudio capturado pelo microfone e o vídeo capturado pela câmera são enviados separadamente ao codificador de áudio e ao

codificador de vídeo; o fluxo de bits gerado pelos codificadores, separadamente, são denominados fluxos elementares. Uma vez multiplexados em um único fluxo de bits, os fluxos elementares passam a ser denominados fluxo de transporte. Podem ser multiplexados dois tipos de estruturas de dados em um fluxo de transporte: os fluxos elementares empacotados PES (*Packetized Elementary Stream*) e as seções. As seções são estruturas definidas para o transporte de tabelas que são conhecidas como SI (*System Information*) nos padrões Europeu [10], Japonês [11] e Brasileiro [12], e PSIP (*Program and System Information Protocol*) no padrão Americano [13].

Dentre as várias tabelas que compõem a SI, as mais importantes para este trabalho são: a EIT (*Event Information Table*) e a SDT (*Service Description Table*). A tabela EIT é usada para apresentar informações específicas de programas, tais como: nome do programa, hora de início, duração, etc. Ela é composta por campos de identificação e por descritores. Os campos de identificação são necessários para que o STB seja capaz de identificar dentro do fluxo de transporte a tabela EIT e, desta forma, reconstruir as informações que ela transporta, obtendo uma cópia idêntica a que foi originalmente transmitida pelo provedor de serviço. Os descritores da tabela EIT são estruturas que transportam informações adicionais e são encapsulados dentro da própria tabela, de tal forma que, um determinado serviço (canais no sistema analógico) possa ter associado a ele diversos descritores. Os descritores definem, por exemplo, a classificação etária do programa, o gênero e subgênero do programa, a sinopse do programa.

A tabela SDT tem como principal objetivo transportar informações sobre os provedores de serviços. De maneira análoga a EIT, a tabela SDT permite o uso de vários descritores.

3 Trabalhos Correlatos

Existem na literatura inúmeros sistemas de recomendação para Televisão Digital (TVD) construídos para oferecer um serviço diferenciado de personalização e auxiliar os telespectadores a lidar com a abundância de programas de TV. A seguir, serão apresentados sistemas que possuem alguma intersecção com o trabalho apresentado neste artigo.

O sistema AIMED proposto por [4], apresenta um mecanismo de recomendação que considera algumas características do telespectador como atividades, interesses, humores, histórico do uso da TV e informações demográficas. Esses dados são inseridos em um modelo de rede neural que infere as preferências de programas dos telespectadores. Diferentemente do trabalho proposto neste artigo, que utiliza a coleta implícita de dados, no sistema AIMED, a coleta de dados e a configuração do sistema, são feitas através de questionários. Essa abordagem é questionável, principalmente quando se considera as limitações impostas para entradas de dados em um sistema de TVDI.

Em [5], é proposto uma estratégia de recomendação de programa para múltiplos telespectadores baseada na junção do perfil do telespectador. O estudo analisou três estratégias para efetuar as recomendações de conteúdo e a análise proporcionou a escolha da estratégia baseada na junção do perfil. Os resultados comprovaram que a união do perfil dos telespectadores pode refletir de forma adequada nas preferências

da maioria dos membros de um grupo. A proposta apresentada neste artigo utiliza uma abordagem semelhante para um ambiente multiusuário. Entretanto, além da junção de perfil, consideramos o horário e dia da semana.

Em [6] é proposto um sistema de TV personalizado embarcado no STB compatível com o modelo *Multimedia Home Platform* (MHP) do padrão europeu de TVD. De acordo com os autores, o sistema foi implementado em uma solução comercial do *middleware* MHP e, para isto, implementou alterações e inclusões de novos módulos neste *middleware*. Neste sistema, para que seja possível oferecer recomendações duas importantes informações devem estar disponíveis: descrição dos programas e o comportamento de visualização do telespectador. Esse trabalho assemelha-se ao trabalho proposto nesse artigo. A coleta implícita de dados, aliada à inclusão de um novo módulo na arquitetura do *middleware*, são exemplos dessa semelhança. Entretanto, em [6] é proposto alterações no *middleware* e a vinculação dessas alterações a um único sistema de recomendação. O trabalho proposto nesse artigo permite que diferentes sistemas de recomendação utilizem os serviços providos pelo novo módulo acoplado ao *middleware*.

Em [7] é apresentado o sistema de recomendação AVATAR, compatível com o *middleware* Europeu MHP. Os autores propõem uma nova abordagem, onde o sistema de recomendação é distribuído por *broadcast* pelos provedores de serviços, como uma aplicação interativa. Essa abordagem, segundo os autores, permite aos usuários escolherem entre diferentes sistemas de recomendação existentes, o que não é possível quando se adquire um STB com um sistema de recomendação instalado em fábrica. O sistema AVATAR utiliza a abordagem de coleta implícita de perfil de usuário e propõem alterações no *middleware* MHP para inclusão do módulo de monitoramento. Como algoritmo de classificação é utilizado o *Naïve Bayes* [8] e uma das principais justificativas para sua adoção é o baixo consumo de recursos do STB. De maneira análoga ao sistema AVATAR, esse trabalho propõe que as aplicações interativas sejam distribuídas por *broadcast* pelos provedores de serviços.

4 Implementação

A arquitetura do *middleware* Ginga-NCL é composta por três camadas: 1) Aplicações residentes, responsável pela exibição (frequentemente chamada de camada de apresentação); 2) Ginga *Common Core*, conjunto de módulos responsável em oferecer vários serviços, inclusive demultiplexação e decodificação dos fluxos de transporte; 3) Camada pilha de protocolos, responsável pelo suporte de vários protocolos de comunicação, tais como: HTTP, TS, etc.

A arquitetura proposta neste artigo, para suporte ao desenvolvimento de sistemas de recomendação, foi implementada como um novo módulo denominado *Recommender* na camada Ginga *Common Core*. O módulo *Recommender* é composto por duas camadas bem delineadas denominadas: *Recommender Engine* e *Recommender API*.

A camada *Recommender Engine* é invisível aos sistemas de recomendação, e tem como principal objetivo realizar processamento de dados e comunicação com os demais módulos do *middleware* Ginga-NCL. É responsabilidade dessa camada coletar

e armazenar as informações de comportamento de visualização do telespectador em uma base de dados, realizar mineração de dados com essas informações e extrair padrões de comportamento.

A camada *Recommender API* é responsável em prover um conjunto de APIs que possibilitem aos sistemas ofertarem serviços de personalização. O *middleware* Ginga-NCL provê suporte para aplicações declarativas escritas em NCL. Entretanto, somente a utilização de NCL não é suficiente para prover recursos de personalização, é necessária uma linguagem procedural para permitir acesso aos serviços providos pelo módulo *Recommender*. Uma solução para esse impasse consiste em adicionar a linguagem declarativa algum suporte imperativo. Esse suporte foi provido pelo *middleware* Ginga-NCL com a adoção da linguagem Lua, através da criação de uma nova classe de objetivos de mídia os quais são chamados de *NCLua*. A Figura 1 ilustra a interconexão do módulo *Recommender* com o *middleware* Ginga-NCL.

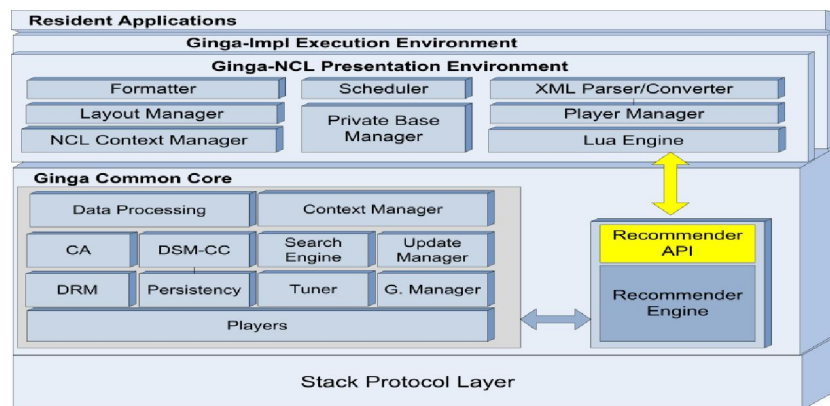


Fig 1. Middleware Ginga-NCL evidenciando a inclusão do módulo Recommender

Assim, o autor das aplicações interativas usa a forma declarativa (NCL) para desenvolver a interface gráfica do sistema de recomendação e a forma imperativa (LUA) para acessar as APIs e solicitar serviços providos pelo módulo *Recommender*.

4.1 Camada Recommender API

A camada *Recommender API* é composta por um conjunto de APIs que provêm serviços aos sistemas de recomendação executando na camada de apresentação do *middleware* Ginga-NCL. Basicamente, os sistemas de recomendação solicitam os serviços fazendo chamadas de funções escritas em linguagem C/C++. Para chamar as funções escritas em C/C++ é necessário o módulo *Lua Engine* que permite a comunicação entre a linguagem Lua e funções escritas em C/C++. A Tabela 1 apresenta algumas das funções disponibilizadas as aplicações interativas.

Tabela 1. Exemplos da API disponibilizada as aplicações interativas.

Métodos públicos disponíveis	Função do método
public string[] getUserDescription()	Retorna string com todo o histórico armazenado na base de dados
public string[] getResource(String resource)	Retornar string com o valor do <i>resource</i> .

4.2 Camada Recommender Engine

A camada *Recommender Engine* é composta por cinco módulos, sendo eles: *Scheduler Agent*, *Mining Agent*, *Local Agent*, *Filter Agent* e *Data Agent*. A figura 2 apresenta a arquitetura da camada *Recommender Engine*.

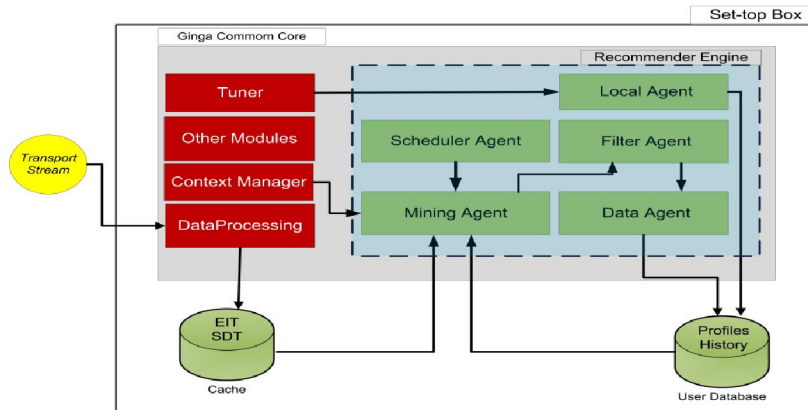


Fig. 2. Arquitetura da camada Recommender Engine

O módulo *Local Agent* é o responsável por monitorar constantemente os dispositivos de interação, por exemplo, o controle remoto e detectar qualquer interação do telespectador com o STB, persistindo em uma base de dados toda a interação. A Tabela 2 exibe uma amostra do histórico de comportamento do usuário.

Tabela 2. Amostra histórico de comportamento do usuário

User	Cód. Ginga	Programa	Gênero	Sub	Data	Hora	Min
1	KEY_POWER	EU A PATROA E AS CRIANCAS	0x5	0x00	19/03/09	14:05	11
1	KEY_UP	CINEMA EM CASA	0xC	0x00	19/03/09	14:16	15

O *Local Agent* comunica constantemente com o módulo tuner implementando no *middleware* Ginga-NCL. O módulo *tuner* é responsável em efetuar trocas de canais. Desta maneira, qualquer ação do telespectador é detectada pelo *Local Agent* e persistida na base de dados de comportamento do telespectador.

Algumas das dificuldades encontradas para as aplicações que executam no STB são a escassez de recursos computacionais, sendo que uma das atividades que mais

demandam recursos é a tarefa de mineração de dados, tornando inviável a sua execução a todo o momento.

O módulo *Scheduler Agent* é o responsável em disparar o processo de mineração, sempre que os recursos necessários (cpu e memória) estiverem disponíveis .

O módulo *Mining Agent* é o responsável por processar a mineração de dados. O módulo *Mining Agent* encapsula o algoritmo de classificação necessário para produzir recomendação de programas televisivos. O módulo *Mining Agent* tem acesso ao histórico de comportamento do usuário e também acesso às informações das tabelas SI, incluindo a EIT e a SDT que juntas transportam informações sobre a programação atual e futura. Com essas informações, o algoritmo de classificação tem dados suficientes para produzir recomendação de programas televisivos.

O módulo *Filter Agent* recebe uma mensagem assim que o módulo *Mining Agent* executa seu algoritmo. Os dados resultantes do processo de mineração são consequentemente processados através de um *parser* implementado no *Filter Agent*, somente as informações relevantes como: provedor de serviço, horário de início do programa e programas recomendados são considerados. O módulo *Data Agent* gerencia a inserção das recomendações (nome do provedor, horário de início, duração, programa) na base de dados e, posteriormente, a exclusão dessas, assim que se tornam obsoletas.

Para persistir as informações obtidas, foi adotado o banco de dados relacional *Sqlite* escrito em linguagem C. A escolha do banco de dados *Sqlite* foi motivada por três facilitadores: ter sido escrito em linguagem C; ter sido projetado para trabalhar em dispositivos embarcados; e permitir acesso às informações armazenadas através de consultas SQL.

4.3 Algoritmos de Mineração

Para determinar qual algoritmo de mineração implementar no módulo *Mining Agent* foram realizados testes com os seguintes algoritmos: C.45, *Naive Bayes* e *Apriori* [8]. Os testes dos algoritmos foram realizados com uso de conjuntos de dados fornecidos pelo IBOPE¹ Mídia. O IBOPE Mídia é a empresa do Grupo IBOPE responsável pelas pesquisas de comunicação, mídia, consumo e audiência. O IBOPE Mídia já é conhecido por suas pesquisas na área de audiência, mas atua também na área de investimento publicitário e em pesquisas quantitativas em todos os tipos de veículos, seja Televisão, Rádio, mídia impressa e mídia alternativa.

Os dados são referentes a 8 famílias com perfis socioeconômicos variados. O comportamento de visualização foi coletado durante 4 semanas, minuto a minuto, em cada domicílio. Para a escolha do algoritmo duas características dos STBs foram consideradas: quantidade de memória e capacidade de processamento.

O primeiro algoritmo testado foi o C.45. Trata-se de um algoritmo de classificação, baseado em árvores de decisão, utilizando o conceito de dividir e conquistar. O segundo algoritmo foi o *Naive Bayes* que é um classificador baseado em estatística, rápido e eficiente quando aplicado a um conjunto grande de dados. O último algoritmo comparado é o *Apriori*. Trata-se de um algoritmo de associação aplicado

¹ <http://www.ibope.com.br>

para descobrir padrões escondidos em conjunto de dados. O algoritmo procura afinidade entre itens e a expressa na forma de regras, por exemplo, “70% do tempo de visualização às segundas-feiras no período entre 19:00 hs. e 20:00 hs. é noticiário”. Outro eficiente algoritmo encontrado na literatura é o SVM (*Support vector machines*) [8]. Para a proposta desse artigo, o SVM não foi utilizado, devido a limitações impostas pelo *hardware* do STB. A seguir, apresentamos os resultados da comparação dos algoritmos considerando velocidade de processamento e a precisão das recomendações. A precisão é calculada usando a seguinte fórmula:

$$= - 100 \quad (1)$$

Onde δ corresponde à eficiência do sistema e varia de 0 a 100, β é o número de recomendações visualizadas pelos telespectadores e α é o número de recomendações efetuadas. A tabela 3 apresenta os resultados obtidos após analisar os históricos das 8 famílias durante 4 semanas. Os algoritmos executaram em um STB equipado com uma CPU de 233 Mhz e uma capacidade de memória de 256 MB.

Tabela 3. Comparação entre Algoritmos executando em um STB de 200 Mhz

Algoritmo	Tempo Médio (s)	Precisão (Maior Valor Obtido entre os 8 domicílios)
C.45	65	71.22
<i>Naive Bayes</i>	54	72.10
<i>Apriori</i>	62	70.45

A conclusão é que os três algoritmos têm desempenho próximo. Entretanto, com o grande volume de dados analisados, em torno de 43 mil tuplas, o algoritmo *Naive Bayes* teve um desempenho superior em tempo de processamento e precisão. Assim sendo, optamos por sua inclusão como algoritmo de classificação no módulo *Mining Agent*.

5 Resultados

Para validação do módulo *Recommender* foram utilizados os dados fornecidos pelo IBOPE. A avaliação adotou a fórmula de precisão apresentada em (1).

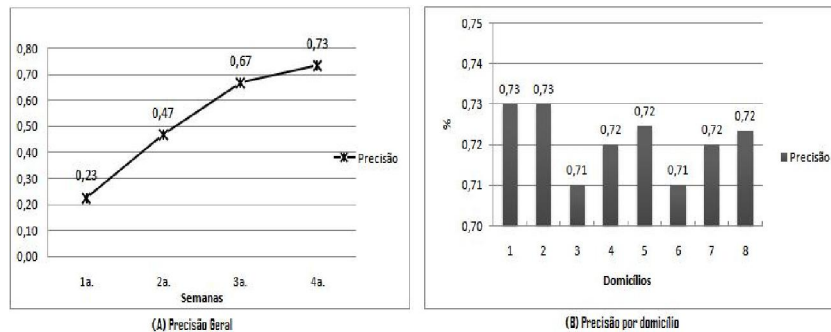


Fig 3. Precisão do algoritmo naïve bayes

A Figura 3a apresenta os resultados obtidos após 4 semanas de monitoramento, considerando o melhor valor obtido entre os oito domicílios investigado. Fica evidente que nas primeiras semanas, como são poucos os dados coletados, o algoritmo *Naïve Bayes* não extraiu informações relevantes das preferências do grupo. Com o aumento dos dados no histórico de visualização na terceira e quarta semana, o algoritmo obteve melhores resultados e o índice de aceitação das recomendações aumentou. A figura 3b apresenta a precisão por domicílio. A principal característica dos domicílios é a diferença socioeconômica existente entre eles. A conclusão é que o algoritmo *Naïve Bayes* obteve um bom desempenho, independente do perfil socioeconômico dos usuários.

Para utilizar os serviços do módulo *Recommender* foi implementado uma aplicação NCL/LUA. A figura 4 apresenta a interface gráfica da aplicação em NCL recomendando 4 programas aos usuários.



Fig 4 Aplicação NCL/LUA

6 Considerações Finais

O domínio de TVD contribui para o surgimento de novas aplicações na área de personalização de conteúdo e, este trabalho apresentou uma dessas possibilidades. Foi desenvolvido um novo módulo para prover suporte ao desenvolvimento de sistemas de recomendação para ambientes televisivos. A inserção desse novo componente permite aos desenvolvedores de aplicações interativas investirem em *designer* e navegabilidade do sistema, deixando questões de baixo nível como: gerenciamento de histórico de comportamento, persistência de dados, serviços de classificação, entre outros, como responsabilidade do módulo *Recommender*.

Os resultados obtidos demonstraram a melhor adequação do algoritmo de classificação *Naïve Bayes*, considerando as limitações impostas pelo *hardware* do STB. Como prova de conceito, implementou-se um sistema de recomendação

desenvolvido em NCL/LUA fazendo uso dos serviços providos pelo módulo *Recommender*.

Nossa expectativa para trabalhos futuros é estender as funcionalidades já implementadas, permitindo a interoperabilidade com outros dispositivos através do protocolo UPnPTM/DLNA num ambiente de redes doméstica.

7 Agradecimentos

Nós gostaríamos de agradecer a FAPESP pelo suporte no projeto TIDIA-AE (processo 05/60653-1) e FINEP pelo suporte no projeto “Avaliação do Middleware Ginga” (01.07.0110.00).

Referências

1. Soares, L., Rodrigues, R., Moreno, M.: “Ginga-NCL: The declarative Environment of the Brazilian Digital TV System”. In: Journal of the Brazilian Computer Society. V.12, n.4, p.37-46 (2007)
2. Souza, G., Leite, L., Batista, C.: “Ginga-J: The Procedural Middleware for the Brazilian Digital TV System”. In Journal of the Brazilian Computer Society, v. 12, n. 4, p. 47-56 (2007)
3. Sun Microsystems.: Java TV API 1.1 (JSR-927). Available: <http://java.sun.com/javame/reference/apis/jsr927> (2009)
4. Hsu, S. H., Wen, M. H., Lin, H. C., Lee, C. C. and Lee, C. H.: AIMED – A personalized TV Recommendation System. In Proceedings of the Interactive TV: A Shared Experience, pages 166-174, Vol 4471, Springer Berlin / Heidelberg (2007)
5. Zhiwen, Y., Xingshe, Z., Yanbin, H. and Jianhua, G.: TV program recommendation for multiple viewers based on user profile merging. In Proceedings of the User Modeling and User-Adapted Interaction, pages 63-82. Publishing Springer Netherlands (2006)
6. Zhang, H.; Zheng, S. Yuan J.: A personalized TV guide system compliant with MHP. In: Consumer Electronics, IEEE Transactions on , vol.51, no.2, pp. 731-737 (2005)
7. Blanco-Fernandez, Y., Pazos-Arias, J., Gil-Solla, A., Ramos-Cabrer, M.,Lopes-Nores, M., Barragans-Martinez, B.: AVATAR: a Multi-agent TV Recommender System Using MHP Applications. In: IEEE International Conference on E-Technology, E-Commerce and E-Service (EEE '05), pp. 660-665. (2005)
8. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z., Steinbach, M., Hand, D. J., and Steinberg, D.: Top 10 algorithms in data mining. Knowl. Inf. Syst. 14, 1 (Dec. 2007), 1-37. DOI=<http://dx.doi.org/10.1007/s10115-007-0114-2> (2007)
9. ISO/IEC 13818-1. “Information technology — Generic coding of moving pictures and associated audio information: Systems”, ISO/IEC (2000)
10. DVB Document A038 Rev. 3. “Specification for Service Information (SI) in DVB systems”, DVB (2007)
11. ARIB STD-B10. “Service Information for Digital Broadcasting System”, ARIB (2005)
12. ABNT NBR 15603-1:2007. “Televisão digital terrestre - Multiplexação e serviços de informação (SI) - Parte 1: Serviços de informação do sistema de radiodifusão”. (2007)
13. ATSC A/65b. “Program and System Information Protocol”, ATSC. (2003)