

PersonalTVware: Uma Proposta de Infraestrutura de Suporte a Sistemas de Recomendação Sensíveis ao Contexto para TV Digital Interativa

Fabio Santos da Silva
Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto,
travessa 3, n.158, sala C1-46
São Paulo, SP, +55 3091 5261
fsilva@larc.usp.br

Luiz Gustavo Pacola Alves
Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto,
travessa 3, n.158, sala C1-46
São Paulo, SP, +55 3091 5261
luizgpa@larc.usp.br

Graça Bressan
Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto,
travessa 3, n.158, sala C1-46
São Paulo, SP, +55 3091 5261
gbressan@larc.usp.br

RESUMO

O advento da TV Digital tem contribuído para o crescimento do volume de programas de TV. Conseqüentemente, aumentou a dificuldade do usuário para selecionar conteúdo relevante dentre as várias opções disponíveis. Diante deste cenário, os sistemas de recomendação destacam-se como uma possível solução. Assim, este artigo apresenta uma infraestrutura de software no cenário da TV Digital Interativa – intitulada PersonalTVware – de suporte a sistemas de recomendação sensíveis ao contexto. Com intuito de demonstrar e validar as funcionalidades em um cenário de uso foi desenvolvido um sistema de recomendação sensível ao contexto como estudo de caso que utiliza o PersonalTVware.

ABSTRACT

The advent of Digital TV has ensued the growth in the volume of TV programs. Consequently, the difficulty in finding the content the TV viewer wishes in a transparent way among the available TV programs increased. Within this scenario, the recommender systems stand out as a possible solution for this problem. Thus, this paper presents a software infrastructure in an Interactive Digital TV environment to support context-aware recommender system for Digital TV – entitled PersonalTVware. To demonstrate and validate the functionalities in a use scenario was developed a context-aware recommender system as a case study which uses the PersonalTVware.

Categories and Subject Descriptors

H.3.4 [Information Storage e Retrieval]: System and Software – Distributed systems.

General Terms

Algorithms, Design, Experimentation.

Keywords

Interactive Digital TV, Recommender systems, Learning Machine, Context-Awareness.

1. INTRODUÇÃO

O processo de digitalização da TV em diversos países do mundo tem contribuído para o aumento do volume de programas de TV, o que gera uma sobrecarga de informação [8], dificultando, assim, a seleção de conteúdo de maior relevância dentre as várias opções disponíveis. Desta forma, os sistemas de recomendação destacam-se como uma possível solução. Tais sistemas filtram itens relevantes de acordo com as preferências do usuário ou grupo de usuários que possuem perfis similares. Entretanto, em diversas situações o interesse do usuário pode depender também do seu contexto. Assim, torna-se importante estender as abordagens tradicionais de recomendação por meio da exploração do contexto do usuário, o que poderá melhorar a qualidade das recomendações [3].

De acordo com [2] “Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade por sua vez, pode ser uma pessoa, lugar ou objeto que é considerado relevante para interação entre um usuário e uma aplicação, incluindo o usuário e aplicação em si”. No caso dos sistemas de recomendação para TV Digital, torna-se importante também considerar informações sobre o contexto da entidade usuário. Por exemplo, **quem** é o usuário e **quando** ele assiste um determinado gênero de programas de TV? No domingo pela manhã? Ou na segunda-feira ao anoitecer quando chega do trabalho? **Onde** e **como** o programa de TV será visto? Na sua casa em um receptor de TV Digital fixo conectado em sua TV *FULL HD* ou na escola em sua TV portátil? E **qual** o gênero de programa de TV é considerado relevante naquela situação em que o usuário assiste TV.

Assim, este artigo apresenta o PersonalTVware, uma infraestrutura de *software* de suporte ao desenvolvimento e execução de sistemas de recomendação sensíveis ao contexto para TV Digital Interativa. O PersonalTVware fornece componentes que permitem a filtragem de programas de TV, o gerenciamento de informações sobre o contexto, perfil do usuário e programas de TV, e a inferência de preferências contextuais. As demais seções deste artigo estão estruturadas da seguinte forma. A seção 2 discute alguns trabalhos relacionados. A seção 3 apresenta a infraestrutura PersonalTVware. A seção 4

ilustra o seu uso através de um estudo de caso e a seção 5 traz as conclusões.

2. TRABALHOS RELACIONADOS

Na comunidade científica é possível encontrar algumas pesquisas que abordam o uso de contexto no cenário da TV e em sistemas de recomendação. Em [5] é apresentado um serviço que pode ser usado por provedores de conteúdo para avaliar o quão apropriado é um programa de TV conforme o contexto no qual os usuários estão inseridos. Diferentemente do PersonalTVware que considera o contexto para inferir preferências por gêneros e sugerir programas de TV para os usuários. No que se refere aos sistemas de recomendação para TV Digital há diversos sistemas. Em [8] é descrito um sistema para recomendação de programas de TV baseados em lógica *fuzzy* para inferir preferências do usuário extraídas do histórico de uso. Em [1] é apresentada uma proposta de abordagem de recomendação personalizada de conteúdo que explora os conceitos da Web Semântica para inferir programas de TV de interesse do usuário.

Em [9] é proposto uma abordagem de recomendação de programas de TV para múltiplos usuários por meio da junção de seus perfis de usuários. Em contraste ao PersonalTVware, os trabalhos supracitados não exploram o contexto do usuário durante o processo de recomendação. Fora do domínio da TV Digital em [7] é apresentado um sistema de recomendação de especialistas sensível ao contexto para apoio à colaboração informal. De forma diferente ao PersonalTVware, o sistema não realiza inferências de preferências a partir do contexto do usuário, utilizando o contexto como informação adicional sobre a situação do usuário. Em [6] é apresentada uma proposta de um sistema que usa algumas informações contextuais (humor, localização) obtidas explicitamente como informação extra para filtragem de conteúdo gerado por usuários de sites de redes sociais. Nenhum método de aprendizagem sobre o contexto foi empregado, e não foram consideradas informações de contexto sobre o tempo como ocorre no PersonalTVware.

3. Infraestrutura PERSONALTVWARE

A arquitetura é composta por dois subsistemas: dispositivo do usuário e provedor de serviços. A Figura 1 apresenta uma visão detalhada da arquitetura geral do PersonalTVware e os seus respectivos subsistemas. No subsistema dispositivo do usuário, o módulo Coordenador de Recomendação faz a ligação entre um sistema de recomendação e os demais módulos do PersonalTVware, e coordena processo de recomendação. Além disso, também é responsável pelo método *Feedback Relevance* [1], que permite obter de forma interativa os *perfis contextuais* dos usuários com informação de gênero instanciada, sendo armazenados em uma base de conhecimento posteriormente explorada pelos métodos de inferência. Um *perfil contextual* é o resultado da agregação das informações contextuais do usuário, as informações do perfil do usuário e o gênero do programa de TV considerado relevante em certo contexto. No PersonalTVware o método *Feedback Relevance* pode ser realizado de forma explícita ou implícita. Na forma explícita o usuário pode selecionar dentre os programas de TV que foram recomendados aqueles considerados relevantes em certo

contexto. Enquanto que a forma implícita explora a β razão entre o tempo que o usuário assistiu a um programa de TV (T_r) e a sua duração total (T_t), se o valor obtido for maior que um limiar (50% em nossos experimentos) o gênero do programa de TV será considerado relevante, e o *perfil contextual* será obtido.

$$\beta = T_r / T_t \in [0,1] \quad (1)$$

O módulo Gerenciador de Contexto do Usuário gerencia o acesso, aquisição de forma automática e conversões das informações contextuais do usuário. São obtidas implicitamente informações tais como identificação do usuário, dia da semana, período do dia, localização, tipo de dispositivo de acesso, e o programa de TV considerado relevante. Tais informações são obtidas por meio de chamadas ao sistema operacional e representadas por meio de documento XML. Já o módulo Gerenciador de Perfil do Usuário gerencia o acesso e aquisição de forma explícita das informações que constituem o perfil do usuário. Por meio de seus componentes o usuário poderá especificar informações tais como dados pessoais, título de um programa e um termo relacionado ao assunto de interesse. As informações do perfil serão descritas de acordo com os metadados do padrão TV-Anytime [8], tornando a representação estruturada e alinhada aos sistemas de TV Digital. Além disso, por razões de privacidade e segurança, o perfil do usuário será armazenado no dispositivo do usuário.

O módulo Interpretador de Contexto é composto por componentes que permitem o aprendizado e inferência de preferências por gêneros de programas de TV. Para isso, no momento da solicitação da recomendação, o PersonalTVware obtém o *perfil contextual* corrente sem informação de gênero instanciada sendo aplicado ao método utilizado na tarefa de inferência. Daí então, a partir da base de conhecimento o interpretador retorna o(s) gênero(s) previsto(s). O PersonalTVware implementa diferentes métodos de aprendizagem de máquina como o Raciocínio Baseado em Casos (RBC), Rede Bayesiana, Árvore de Decisão e Redes Neurais [10]. Já o módulo Filtro Baseado em Contexto é responsável pela filtragem dos programas de TV que provavelmente serão relevantes para o usuário. O processo de filtragem explora as informações de contexto (dia, horário e localização de origem do programa de TV), o perfil do usuário, e as preferências contextuais, como os gêneros inferidos pelo módulo Interpretador de Contexto, além dos metadados dos programas de TV.

A técnica de filtragem empregada é a técnica baseada em conteúdo que consiste em comparar o perfil do usuário com as descrições dos itens, visando obter uma lista personalizada de itens [8]. Assim, por meio desta lista o usuário pode avaliar dentre os programas de TV filtrados aqueles considerados relevantes. Esta ação permitirá ao sistema reter novos *perfis contextuais* com informação de gênero instanciada. Por questões de limitação de recursos computacionais dos dispositivos de acesso, ambos os módulos Interpretador de Contexto e Filtro Baseado em Contexto estão localizados no subsistema provedor de serviços. Já o módulo Gerenciador de Programas de TV gerencia a manipulação das informações referentes aos programas de TV. Tais informações também são descritas de

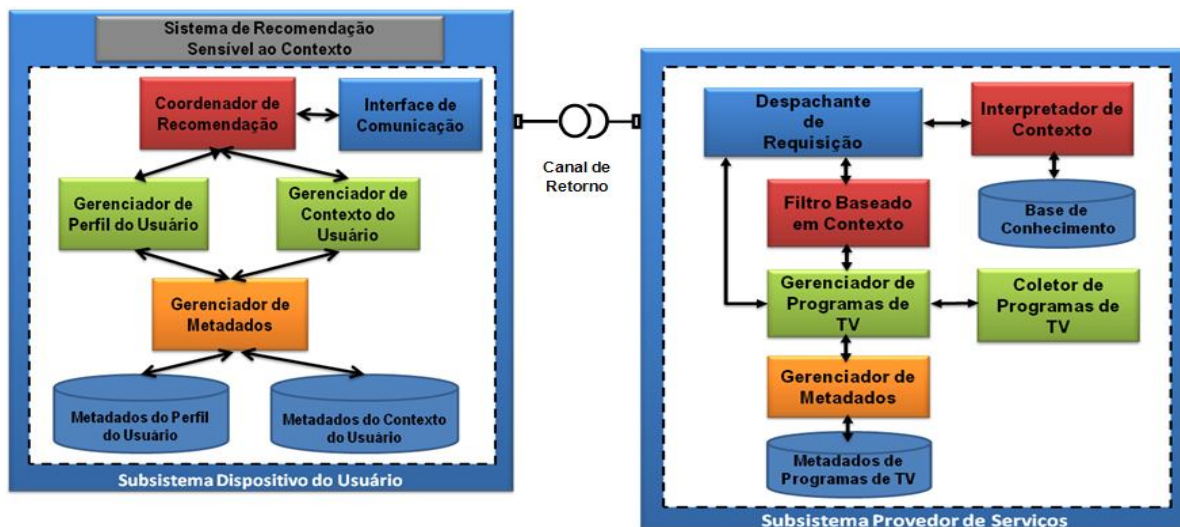


Figura 1. Arquitetura do PersonalTVware

acordo com os metadados do padrão TV-Anytime. Em ambos os subsistemas, o módulo Gerenciador de Metadados fornece suporte aos demais módulos para o gerenciamento do acesso, inserção, e validação dos metadados representados em documentos XML. O módulo Coletor de programas de TV é composto por componentes que oferecem suporte para coleta de informações sobre os programas de TV.

As informações devem ser obtidas de fontes externas como a Web ou tabelas SI (*Service Information*) [8]. Finalmente, o módulo Interface de Comunicação viabiliza a comunicação entre os subsistemas dispositivo do usuário e provedor de serviços por meio de um cliente de serviços Web que envia requisições via o canal de retorno. No provedor de serviços, o módulo Despachante de Requisição é um serviço Web que recebe requisições oriundas do subsistema dispositivo do usuário, e as encaminha aos módulos adequados de acordo com o tipo de requisição recebida.

3.1 Projeto e Implementação

As principais classes que modelam o PersonalTVware são apresentadas no diagramas UML da Figuras 2 e 3. O módulo Gerenciador de Perfil do Usuário é representado pelo pacote `userprofile` e contém duas classes: `UserProfileManager` e `UserProfile`. Estas classes são responsáveis pela manipulação e entrega das informações explícitas de perfis de usuários. A classe `UserProfile` representa um perfil de usuário que é auxiliar a classe `UserProfileManager` responsável pela criação, seleção e atualização de um perfil de usuário. Já o pacote `context` se refere ao módulo Gerenciador de Contexto do Usuário e implementa as classes necessárias para manipulação das informações contextuais. Existem sete classes no pacote, a saber: `ContextManager`, `Context`, `ContextGather`, `ContextConverter`, `LocationCollectorFactory`, `LocationCollector` e `IPAddressCollector`. A classe `Context` generaliza a representação das informações contextuais obtidas de forma implícita por meio dos sensores lógicos da classe `ContextGather`.

A classe `ContextConverter`, funcionalmente é um conversor de informações, que realiza a conversão de informações de fontes de contexto em dados legíveis para a plataforma. A classe `LocationSensorFactory` é uma fábrica de coletores de localização representados pela classe `LocationSensor`. Este padrão de projeto viabiliza que sejam implementados diferentes sensores lógicos de localização de acordo com a plataforma de *hardware* e *software* do dispositivo de acesso. Para este projeto foi utilizado IP (*Internet Protocol*) por meio classe `IPAddressSensor`. Já a classe `ContextManager` com o auxílio da classe `Context` é responsável pelo suporte a criação, seleção, atualização de informações contextuais de um determinado usuário.

O pacote `metadata` (referente ao módulo Gerenciador de Metadados) contém duas classes principais, sendo elas: `MetadataManager` e `SchemaValidator`. Quaisquer classes que necessitem realizar operações comuns em arquivos XMLs devem utilizar a classe `MetadataManager`. A classe `SchemaValidator` realiza a validação de documentos XMLs. A classe `CommunicatorManager` do pacote `communication` é um cliente de serviços Web sendo responsável pela comunicação com o subsistema provedor de serviços via canal de retorno. Por meio do método (`setRequetsServiceProvider()`) desta classe é possível o envio de *perfis contextuais* com informação de gênero instanciada presentes no dispositivo, bem como o recebimento de informações (lista de programas de TV filtrados, gêneros inferidos, dentre outros) oriundas do provedor de serviços.

Finalizando o diagrama referente ao subsistema dispositivo do usuário, há o pacote `recommender` que corresponde ao módulo Coordenador de Recomendação. Existem três classes no pacote `recommender`, a saber: `RecommenderCoordinator`, `Recommendation` e `FeedbackRelevance`. A classe `Recommendation` é auxiliar a classe `RecommenderCoordinator`, pois representa as informações básicas sobre os programa de TV obtidos dos metadados oriundos da lista de programas de TV recomendados. A classe

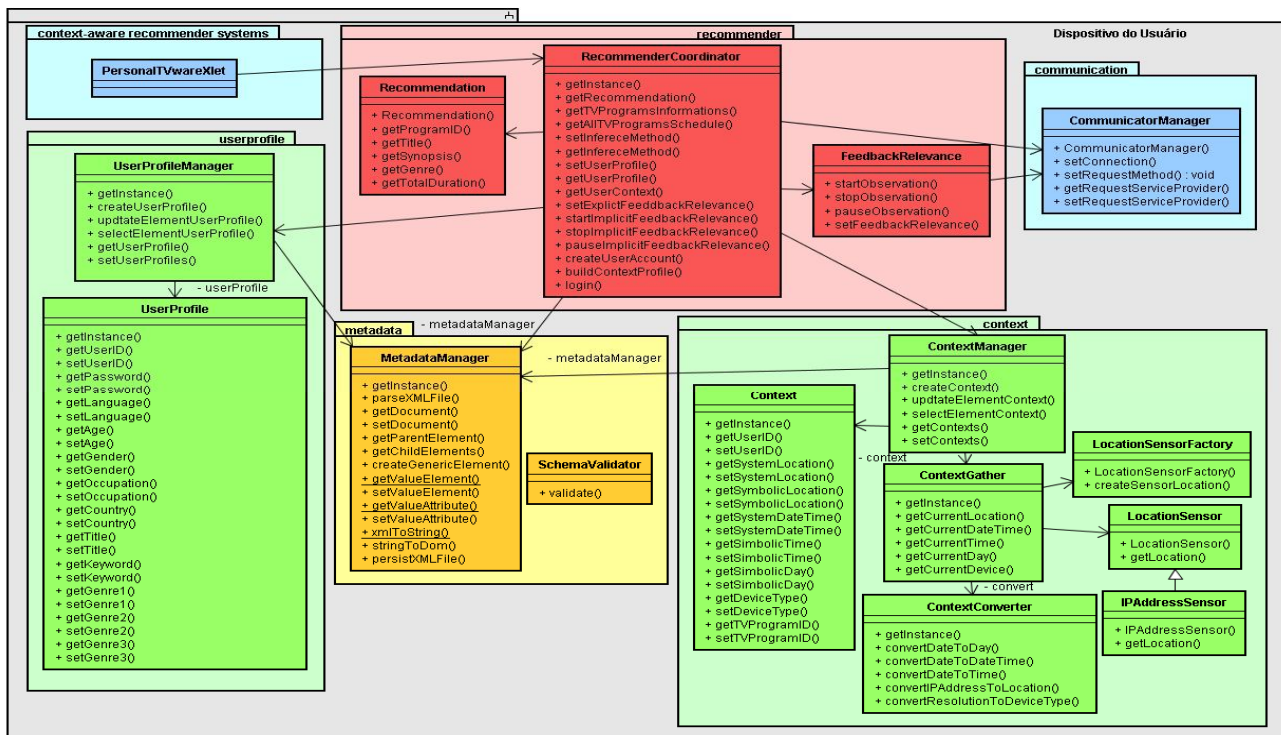


Figura 2. Digrama de Classes – Dispositivo do Usuário

FeedbackRelevance é responsável por monitorar de forma implícita ou explícita a interação do usuário. Quando um usuário interage de forma explícita por meio da avaliação de um programa de TV recomendado pelo sistema, é criado o *perfil contextual* através do método (`setFeedbackRelevance()`). Já na forma implícita o controle da totalização do tempo de observação de um programa de TV é realizado intrinsecamente pelos métodos (`startObservation()`), (`stopObservation()`) e (`pauseObservation()`). Vale dizer que a identificação do programa corrente dá-se através do seu CRID (*Content Reference Identifier*) que, segundo o TV-Anytime, significa um identificador que referencia um conteúdo. Por fim, a classe `RecommenderCoordinator` é incumbida de coordenar e controlar as operações do PersonalTVware de acordo com o que for requisitado pelo sistema de recomendação. Por exemplo, a lista de recomendação de programas de TV pode ser obtida por meio do método (`getRecommendation()`) que executa o processo de recomendação baseando-se no *perfil contextual* sem informação de gênero instanciada, que será obtida pelo método de inferência corrente.

No diagrama referente ao subsistema provedor de serviço há o pacote interpretar que é composto por um conjunto de classes que juntas formam o serviço de interpretação de contexto (inferência de gêneros a partir do *perfil contextual*). Pelo diagrama da Figura 3 é possível observar que a infraestrutura suporta a utilização de quatro métodos de aprendizagem: algoritmo J48 (árvore de decisão) – `ReasonerDecisionTree`; *perceptron* multicamadas com algoritmo *back-propagation* (redes neurais) – `ReasonerMultilayerPerceptron`; classificador bayesiano ingênuo (aprendizagem bayesiana) – `ReasonerNaiveBayes`; ou raciocínio baseado em casos – `ReasonerCBR`. O pacote `request`, refere-se ao módulo

Despachante de Requisição. A classe `RequestDispatcher` implementa o serviço Web para obtenção das requisições submetidas pela classe `RecommenderCoordinator`, e a chamada do método correspondente a operação solicitada. O pacote `filter` é formado por um conjunto de classes que viabilizam o serviço de filtragem de programas de TV. A classe `FilterFactory` é uma fábrica de filtros representados pela classe `Filter`. Este padrão de projeto permite a implementação de filtros baseados em abordagens distintas. Para este projeto foi implementado a filtragem baseada em contexto por meio da classe `ContextBasedFilter`.

O pacote `tvprogram` é formado pela classe `TVProgramManager`. Com esta classe é possível realizar operações básicas como adicionar ou obter informações sobre os programas de TV representados por meio de metadados do padrão TV-Anytime. As operações sob os metadados foram implementadas por meio da linguagem XQuery¹. Já o pacote `tvcollector` é composto por um conjunto de classes que formam o serviço de coleta de informações sobre os programas de TV. É possível notar que as classes `WEBTVCollector` e `SITVCollector` foram especificadas para suportar dois tipos de coletas. A classe `SITVCollector`, apesar de estar representado na arquitetura proposta, não foi implementada neste trabalho por não haver uma plataforma de TV Digital disponível para transmissão de dados via canal de difusão. Logo, a coleta dos metadados dos programas de TV foi por meio da Web. Por fim, o pacote `metadata` tem a classe `MetadataManager` que oferece suporte as demais classes do subsistema para realizar operações comuns em documentos XMLs.

¹ XQuery: <http://www.w3.org/standards/techs/xquery>

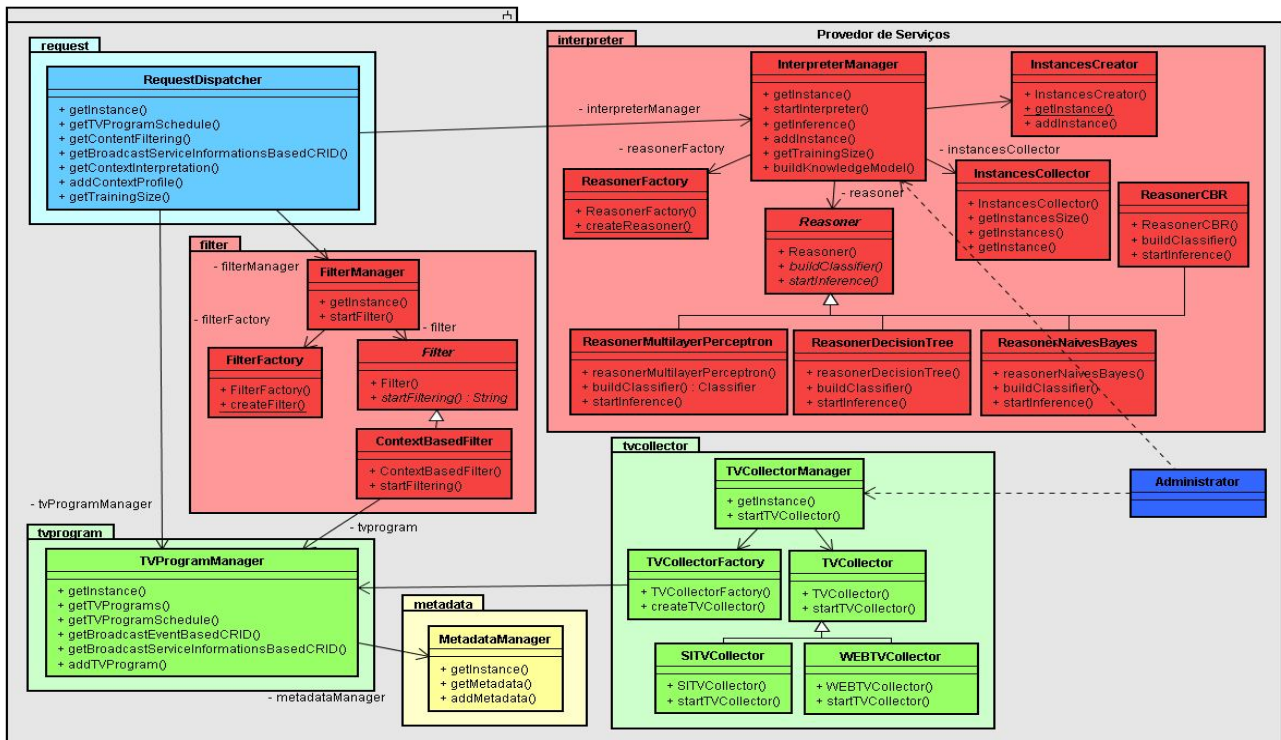


Figura 3. Digrama de Classes – Provedor de Serviços

A implementação do PersonalTVware foi feita em Java. Em termos de serviço Web adotou-se a tecnologia Apache Axis² v. 1.4. A interpretação e validação de documentos XMLs são realizadas por um *parser* que utiliza a interface DOM (*Document Object Model*). Para o módulo responsável pela tarefa de inferência foram utilizadas as APIs da ferramenta Weka³ que contempla uma série de métodos de aprendizagem de máquina implementados e o *framework* Lucene⁴. A Weka utiliza um arquivo no formato ARFF (*Attribute-Relation File Format*) que contém um conjunto de registros que servem para serem utilizadas na tarefa de inferência (base de conhecimento com os *perfis contextuais*). Para manipulação de metadados no lado servidor e filtragem de programas de TV foi empregada a linguagem XQuery por meio do XML Database Exist⁵.

4. ESTUDO DE CASO

O estudo de caso consistiu no desenvolvimento de um sistema de recomendação sensível ao contexto para TV Digital, que utiliza a API do PersonalTVware para prover recomendações de programas de TV. O sistema de recomendação foi implementado como um aplicativo Xlet e executado por meio de emulação de ambiente de TV Digital em computadores tais como *notebooks* e *desktops*, que simularam receptores (fixo e portátil).

² Apache Axis: <http://ws.apache.org/axis>

³ Weka: <http://www.cs.waikato.ac.nz/ml/weka>

⁴ Lucene: <http://lucene.apache.org/java/docs/index.html>

⁵ XML Database Exist: <http://exist.sourceforge.net/>

subsistema provedor de serviço, o mesmo foi instalado em um servidor com conexão permanente a Internet. Por meio da Figura 4 é possível observar a tela de recomendação de programas de TV que apresenta a lista personalizada de programas de TV e informações sobre o contexto corrente do usuário. Por meio desta tela o usuário pode selecionar os programas de TV avaliados como relevantes e também obter informações detalhadas de cada programa de TV.



Figura 4. Sistema de Recomendação – Tela de recomendação

4.1 Experimentos e Resultados Sumarizados

Foram realizados experimentos com 10 usuários que utilizaram o sistema com diversos *perfis contextuais* durante o período de 1 mês totalizando 2197 *perfis contextuais* registrados na base de conhecimento. Os experimentos envolveram uma coleção de 2426 programas de TV de 14 canais correspondente a uma semana de programação da TV. Após uma etapa de treinamento do sistema que durou 1 semana foi gerada a base de conhecimento. O primeiro experimento realizado pelos usuários consistiu em selecionar os programas de TV considerados relevantes dentre aqueles que foram recomendados por meio de cada método de aprendizagem de máquina suportado pelo PersonalTVware. O objetivo foi analisar qual método proporcionava a recomendação mais adequada ao *perfil contextual* do usuário. No segundo experimento não foram empregados os métodos de aprendizagem de máquina. Logo, somente o perfil do usuário definido de forma explícita foi utilizado. O objetivo foi verificar a qualidade das recomendações sem o uso do contexto.

Segundo [4] a eficiência de um sistema de recomendação pode ser mensurada por meio das métricas: precisão, cobertura e medida F. A precisão pode ser usada para mensurar a habilidade de o sistema apresentar somente itens relevantes. Enquanto, que a cobertura pode ser usada para mensurar a habilidade de o sistema apresentar todos os itens relevantes. A precisão e cobertura são métricas por natureza conflitantes, ou seja, quando o sistema apresenta uma cobertura alta a precisão é baixa e vice-versa. Assim, pode ser utilizada a medida F que combina em única fórmula as métricas precisão e cobertura. Segundo [4] quanto mais próximo de 1 for o valor da medida F mais eficiente será o sistema.

Tabela 1. Comparação das médias das métricas

Método	Precisão	Cobertura	Medida F
Algoritmo J48	0,861	0,620	0,680
Classif. bayesiano ingênuo	0,797	0,593	0,636
Perceptron multicamadas	0,774	0,582	0,606
Raciocínio baseado em casos	0,834	0,920	0,841
Perfil do usuário	0,511	0,174	0,247

A Tabela 1 apresenta as médias das métricas obtidas das recomendações baseadas em métodos de aprendizagem máquina e somente em perfil do usuário. Os resultados indicam que a qualidade das recomendações foi melhor quando empregado o método RBC. Analisando a Tabela 1, nota-se que a média das medidas F do método RBC está mais próxima de 1. Conseqüentemente, o sistema apresentou um desempenho superior quando comparado aos demais métodos experimentados. A razão deste resultado foi a menor discrepância entre as medidas de precisão e cobertura obtidas pelo sistema. Já o algoritmo J48 apresentou uma precisão ligeiramente melhor em relação aos outros métodos, pois retornou menor número de programas de TV não relevantes ao contexto do usuário, contudo a sua cobertura foi inferior ao método RBC, o que implicou na

redução do valor da sua medida F. Tal resultado foi devido ao fato do método RBC retornar maior número de programas de TV relevantes ao contexto do usuário. É possível notar que a qualidade das recomendações foi inferior quando empregado somente o perfil do usuário sem o uso de contexto, o que demonstra que a ausência da exploração do contexto impacta diretamente na qualidade das recomendações. Também foi constatado que não houve grandes variações entre os outros métodos de aprendizagem.

5. CONCLUSÃO

Foi apresentada uma infraestrutura intitulada de PersonalTVware que oferece suporte ao desenvolvimento de sistemas de recomendação sensíveis ao contexto para TV Digital Interativa. Diante dos resultados obtidos, conclui-se que a exploração do contexto pode melhorar o desempenho de um sistema de recomendação, principalmente, quando empregados métodos de aprendizagem de máquina. Em relação à tarefa de inferência de gêneros preferidos a partir do contexto e perfil do usuário, vale destacar que o uso do método RBC proporcionou recomendações de melhor qualidade. Os usuários que participaram dos experimentos mostraram satisfeitos com recomendações de programas de TV apresentadas.

6. REFERÊNCIAS

- [1] Blanco Y. et al. 2004. AVATAR: Advanced Telematic Search of Audiovisual Contents by Semantic Reasoning. In Proc. of the Personalization of Future TV Workshop.
- [2] Dey, A. K. 2001. Understanding an Using Context. In ACM Personal and Ubiqu. Comp. Journal, v.5, n.1, p.4-7.
- [3] Gediminas, A. and Tuzhilin, A. 2005. Toward the Next Generation of Recommender System: A Survey of State-of-the-Art and Possible Extensions. In IEEE Trans. On Knowledge and Data Engineering, v. 17, n.6, p. 149-156.
- [4] Jiangshan, X. et al. 2002. The Development and Prospect of Personalized TV Program Recommendation Systems. In Proc. of the IEEE 4th Int. Symp. on Multimedia Software Engineering.
- [5] Leite, L. E. et al. 2007. Uma Arquitetura de Serviço para Avaliação de Contextos em Redes de TV Digital. In SBRC.
- [6] Pessemier, T. D., Deryckere, T. and Martens, L. 2009. Context-Aware Recommendations for User-generated Content on a Social Network Site In Proc. of the 7th European Interactive Television conference.
- [7] Petry, H. et al. 2006. Um Sistema de Recomendação de Especialistas Sensível ao Contexto para Apoio à Colaboração Informal. In: Simpósio Brasileiro de Sistemas Colaborativos, p.38-47.
- [8] Zhang, H. and Zheng, S. 2005. Personalized Recommendation based on TV-Anytime Metadata. In Proc. Of 9th Int. Symp. on Consumer Electronics, ISCE.
- [9] Zhiwen, Y., Xingshe, H and Jianhua G. 2006. TV Program recommendation for multiple viewers based on user profile merging In Proc. User Model User-Adap Inter.p.63-82.
- [10] Witten, I. H., Franck E. 2005. Data Mining: Practical Machine Learning Tools and Techniques, Elsevier.