Exploiting Contextual Information in Recommender Systems for Interactive Digital TV

Fábio Santos da Silva, Luiz Gustavo Pacola Alves, Graça Bressan
{fsilva, luizgpa, gbressan@larc.usp.br}
Polytechnic School of University of São Paulo (EP-USP)
AV. Prof. Luciano Gualberto, travessa 3, n158, sala C1-46, São Paulo,SP, Brazil

***Abstract*-The advent of Digital TV has provided the growth in the volume of TV programs offered by TV operators. 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. However, the context has rarely been explored during the recommendation process. Thus, this paper presents a software infrastructure in an Interactive Digital TV environment to support context-aware personalized recommendation of TV programs – entitled PersonalTVware. To demonstrate and validate the functionalities in a usage scenario was developed a context-aware recommender system as a case study which uses the PersonalTVware.**

I. Introduction

The process of digitalization of TV in several countries around the world has contributed to increase volume of TV programs, which leads to information overload problem [1]. Consequently, the difficulty in finding the content the user wishes in a transparent way among the available TV programs increased [15]. The traditional tool known as Electronic Program Guide (EPG), therefore, has not efficiently responded to the needs the user has for information. The EPG simply displays long lists of TV programs requiring the user to spend a great deal of time looking for information on his/her favorite TV programs. Thus, the recommender systems stand out as a possible solution. These systems filter relevant items according to user preferences or group of users who have similar profiles. However, most recommender systems for Interactive Digital TV have rarely considered the user's contextual information in carrying out the recommendation [3].

In many situations the user's interest may also depend on context. Thus, it becomes important to extend the traditional approaches by exploiting the context of the user, which may improve the quality of the recommendations [3]. Some questions related to context can be exploited such as, *who* is the user and *when* he attends a particular genre of TV programs? On Sunday morning or on Monday evening when does he come from office? *Where* and *how* the TV program will be seen? At home through a Digital TV receiver connected to full HD TV set or at school on portable TV? And *what* kind of TV program is considered relevant in that situation to user watching TV? Depending on his/her context, the user may have different preferences and needs.

Therefore, this paper presents the PersonalTVware, a software infrastructure to support the development of context-aware recommender systems for Interactive Digital TV. The PersonalTVware provides components that allow filtering TV programs, manage information regarding the context, user profile and TV programs, and the inference of contextual preferences. The remaining sections of this paper are structured as follows: section 2 discusses related works, the section 3 presents some fundamentals about context, section 4 presents the PersonalTVware infrastructure, section 5 illustrates its use through a case study and section 6 presents the conclusions.

II. Related Works

Recently in the scientific community it is possible to find some researches that address the use of context in the scenery of TV and recommender systems. Leite [5] proposes a service that can be used by content providers to evaluate how appropriate is a specific TV program, conforming to the context in which users will be insert. Alves, Silva and Bressan [12] propose a context-aware infrastructure with support for collaborative participation in an Interactive Digital TV environment. The difference of PersonalTVware is that considers the context to infer user preferences for genres and suggest TV programs of interest to the user.

Regarding recommender systems for digital TV, there are several systems. PTV [13] was one of the pioneering personalized recommendation projects for Digital TV, and it has became a reference landmark for many initiatives which investigate the Digital TV information overload problem. PTV is a system providing an Internet-based personalized TV listings service, differently of PersonalTVware designed for Interactive Digital TV environment. Zhang and Zheng [9] propose a recommender system for Digital TV based on fuzzy logic to infer preferences extracted from user's usage history. Blanco et al. [1] present an approach to personalized recommendation content that explores the concepts of the semantic Web to infer TV programs of interest to the user.

Zhiwen, Xingshe and Jinhua [10] propose an approach to recommend TV programs to multiple users through the union of their user profiles. In contrast with the PersonalTVware, the above-mentioned works fail to exploit the user's context during the recommendation process. Outside the field of TV Digital, Petry [7] presents a recommender system that considers the context, to recommend specialists. Differently of PersonalTVware, this system does not perform inferences of preferences from the user's context, and explores the context as additional information. Pessemier, Deryckere and Martens [6] present a proposal for a system that uses some contextual information (mood, location) obtained explicitly as extra information to filtering user-generated content on a social network site. No method of learning about the context was adopted, and context information over time was not considered as in PersonalTVware.

III. Context

According to Dey [2], context is any information that may be used to characterize the situation of an entity. An entity in turn, can be a person, place or object that is considered relevant in the interaction between a user and an application, including user and application themselves.

In the case of recommendation systems for Digital TV, it is also important to consider information about the context of the entity user [3]. Generally the contextual information can be identified from six basic contextual known as 5W + 1H [14]: *where* (location), *who* (identification), *what* (action or activity), *when* (time), *why* (motivation behind the action) and *how* (a way to identify how the elements of context are collected).

Dey [2] suggested that the dimensions *where*, *who*, *what* and *when* are, in practice, the most important. The information about the location is used to relate where the user is present, it could be a symbolic location (at home, work place, school, in a restaurant, among others).

The identification could be used to characterize the identity of the user, device or other entities. The information about the activities identifies the use of the device by the user, for example. And lastly, the time denotes the moment in which the user interacted with the system, being in agreement with the period of day (morning, noon, evening, midnight) or with a different granularity of time. The user activity is considered a complex (or high-level) context, so that must be the object of study for future wok.

According to Dey [2] a context-aware system uses context to provider relevant information and/or services to the user, where relevancy depends on the user's task. The PersonalTVware is a context-aware infrastructure since it uses similar context information of the other users, their corresponding profile information, to recommend, in a transparent way, other users in the choice of TV Programs.

IV. PersonalTVware Infrastructure

The PersonalTVware infrastructure is based on the hypothesis that the user may have different preferences as to content depending on his/her current and also past context. In the Digital TV scenario, to be aware of the contextual information of the user's interaction is relevant for the personalized selection and adaptation of contents to be presented to the user. The user's context can be managed in order to determine what genre of TV program is more adequate for recommendation [8].

A. Example Usage Scenario

In order to illustrate the applicability of the PersonalTVware, a use scenario will be initially presented in form of short stories.

"When arriving Monday at 07:30 P.M. in the gym, the student Fernanda begins a walk in the mat. She enjoys watching TV programs on sports in her portable device while she works out. Nevertheless, she does not wish to waste time looking for a TV program through various TV channels. Therefore, she accesses information about programming of Digital TV in a personal way by means of a recommendation system that presents a list of recommended TV programs in accordance with her profile and current context."

B. Context Identification

Through the usage scenario presented, it is possible to identify some context information implicitly present such as "Fernanda", "Monday at 07:30 P.M.", "gym", "treadmill work out", "sports", and "portable device". Such information has to do with contextual dimensions: *who* (identity), *when* (time), *where* (location), *what* (activity or genre of TV program) and *how* (a way to identify how the context elements are collected).

To efficiently exploit the user's context, a context model oriented the organization of the key context information abstracted from the usage scenario was specified. The model is used as a reference in the building of metadata structures in XML Schemes [15] used in the representation of context information. The Figure 1 presents a XML document describing the user's context. This XML document is based on the context model proposed by Goularte et al. [14].

```
<?xml version="1.0" encoding="UTF-8"?>
<ptw:ContextModel>
  <ptw:Context>
    <ptw:User>
      <ptw:Who xsi:type="ptw:UserIdentifyContextType">
        <ptw:UserID>1</ptw:UserID>
      </ptw:Who>
      <ptw:Where xsi:type="ptw:SpatialContextType">
<ptw:SystemLocation>189.99.139.117</ptw:SystemLocation>
<ptw:SymbolicLocation>home</ptw:SymbolicLocation>
<ptw:PhysicalLocation>BR-SP</ptw:PhysicalLocation>
      </ptw:Where>
      <ptw:When xsi:type="ptw:TemporalContextType">
        <ptw:SystemDateTime>2010-09-
05T18:15:47</ptw:SystemDateTime>
         < ptw:SymbolicTime>evening</ptw:SymbolicTime>
        <ptw:SymbolicDay>sunday</ptw:SymbolicDay>
      </ptw:When>
      <ptw:How xsi:type="ptw:InfrastructureContextType">
        <ptw:DeviceType>fixed</ptw:DeviceType>
      </ptw:How>
    </ptw:User>
    <ptw:TVProgram>
      <ptw:Who
xsi:type="ptw:TVProgramIdentifierContextType">
<ptw:TVProgramID>crid://bbc.com/lifev</ptw:TVProgramID>
      </ptw:Who>
    </ptw:TVProgram>
  </ptw:Context>
</ptw:ContextModel>
```

Fig.1.User's context.

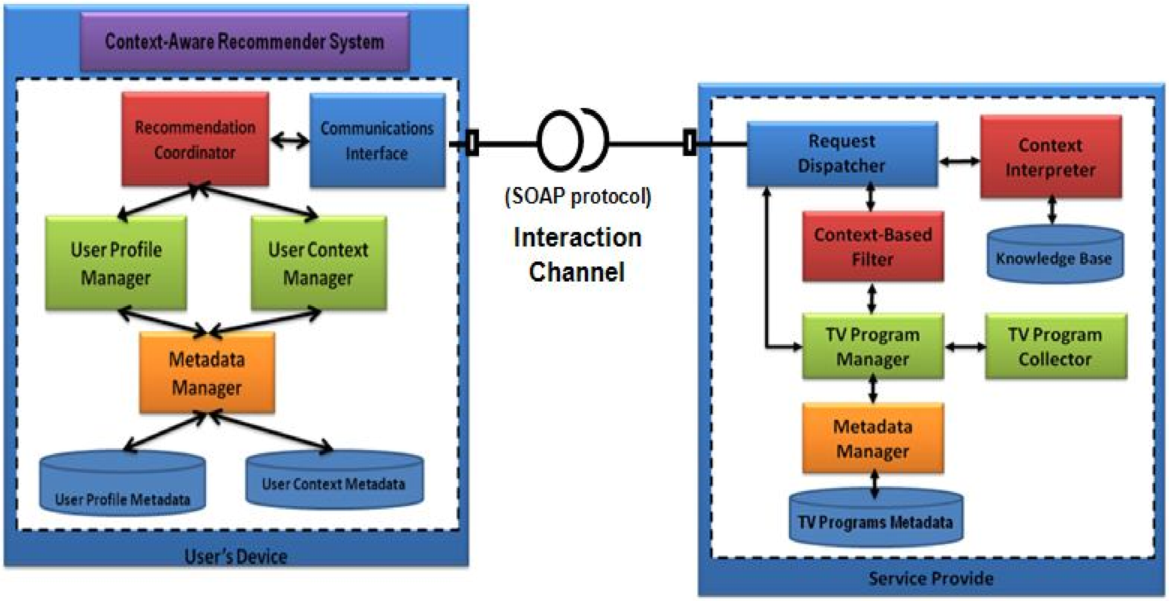

Fig. 2. PersonalTVware architecture.

C. *Solution Architecture*

The PersonalTVware architecture consists of two subsystems: user's device and service provider [8]. The Figure 2 outlines a detailed view of the overall architecture of PersonalTVware and their respective subsystems. In the subsystem user's device, the Recommendation Coordinator module connects a recommender system to PersonalTVware modules, and coordinates the recommendation process. Besides that, it is also responsible for the *Relevance Feedback* method [1] which computes interactively the *Contextual User Profiles* with instantiated genre information, and stores it in a knowledge base to be explored by inference methods.

A *Contextual User Profile* is the result of aggregation of contextual information from the user, personal data of the user profile and genre of TV program considered relevant in a certain context. At PersonalTVware, the *Relevance Feedback* method can be performed explicitly or implicitly. In the explicit way, users can select from the recommended TV programs those considered relevant to the context. The implicit way explores the ratio β between the time that the user watched a TV program (T_r) and its total duration (T_t), if the value obtained is greater than the threshold (50% in our experiments) the genre of the TV program will be considered relevant, and the *Contextual User Profile* is obtained.

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

The User Context Manager module manages the access, acquisition of automatic way and conversions of the user´s contextual information such as its identity, day of the week, period of the day, location and how to access recommendation through a fixed or portable device, and TV Program considered relevant. Such informations is obtained through calls to the operating system and represented by means of XML document (subsection B.). The User Profile Manager module is responsible for creating e updating the user profile through the explicit information provided by the user, for instance, the personal data (language, gender, age, occupation) and preferences (director, actor, author, keyword related to the subject of interest and a title of the TV program). The profile information will be described in accordance with the metadata of the TV-Anytime standard [9], making the representation structured and aligned to the Digital TV systems. Besides, due to privacy and security reasons, the user profile is just stored in the user's device.

The Context Interpreter module consists of components that enable the learning and inference of preferences by genres of TV programs. For that, upon request of the recommendation, PersonalTVware obtain the current *Contextual User Profile* without instantiated genre information being applied to the method used in the inference task. Hence, from the knowledge base the interpreter returns the genre(s). The PersonalTVware implements different methods of machine learning such as Case Based Reasoning (CBR), Bayesian Network, Decision Tree and Neural Networks [11].

The Context-Based Filter module is responsible for filtering of TV programs that are likely to be relevant to the user. The filtering process exploits the contextual information (date, time and location of origin of the TV program and user), explicit and contextual preferences (as the genre inferred by the Context Interpreter module), in addition to TV programs metadata. The filtering technique used is content-based filtering technique that involves comparing the user profile with the descriptions of items in order to obtain a customized list of items [3].

Thus, through this list, user can select the TV program considered relevant (*Relevance Feedback*). This action will allow the system to retain new *Contextual User profiles* with instantiated genre information. For reasons of limited resources of computing devices, both modules Interpreter Context and Context-Based Filtering are located in the

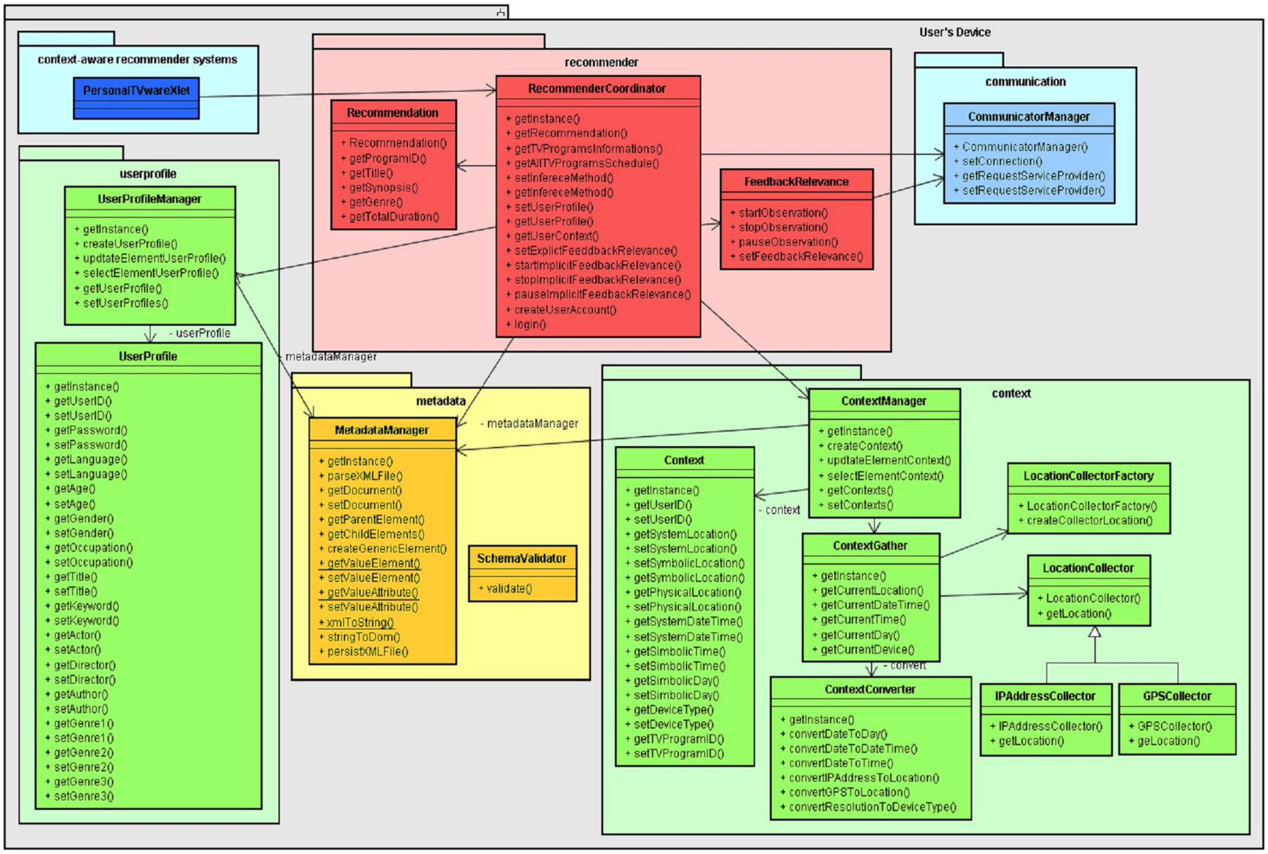

Fig. 3. Classes Diagram – User's Device

subsystem service provider. The <u>TV Programs Manager</u> module manages the handling of information regarding TV programs. Such information is also described according to the metadata of TV-Anytime standard [9].

Finally, the <u>Communication Interface</u> module enables communication between the user's device subsystems and service provider through a Web Service client that transmits requests via SOAP[1] (*Simple Object Access Protocol*) over the interaction channel. In the service provider, the <u>Request Dispatcher</u> module is a Web Services-based interface that receives requests coming from the user´s device subsystem, and transmits them to appropriate modules according to the type of the received request.

D. *Design and Implementation*

The main classes that model the PersonalTVware are presented in the UML diagrams of Figures 3 and 4. The <u>User Profile Manager</u> module is represented by `UserProfile` package and contains two classes: `UserProfileManager` and `UserProfile`. These classes are responsible for handling and delivery of user profiles explicit information. The `UserProfile` class represents a user profile that assists the class `UserProfileManager` responsible for creating, selecting and updating a user profile. The `Context` package refers to the <u>User Context Manager</u> module and implements the necessary classes for manipulating the contextual information. There are seven classes in the package, namely: `ContextManager`, `Context`, `ContextGather`, `ContextConverter`, `LocationCollector`, `LocationCollectorFactory` and `IPAddressCollector`. The `Context` class generalizes the representation of contextual information obtained implicitly through the logic sensors of class `ContextGather`. The class `ContextConverter`, functionally is an information converter, which performs the conversion of information from sources of context in readable data for the platform. The class `LocationSensorFactory` is a location collector factory represented by the class `LocationSensor`.

This design pattern enables that different location logical sensors are implemented in accordance with the hardware and software platform of the access device. In this project IP (Internet Protocol) was used through class `IPAddressSensor`. The `ContextManager` class, with the help of `Context` class, is responsible for supporting the creation, selection and updating of contextual information for a particular user. The `metadata` package (referring to the <u>Metadata Manager</u> module) contains two main classes, namely: `MetadataManager` and `SchemaValidator`. Any classes that need to conduct operations over XMLs files must use the class `MetadataManager`.

The class `SchemaValidator` performs validation of XML documents. The class `CommunicatorManager` of

[1] SOAP: http://www.w3.org/TR/soap12-part0/

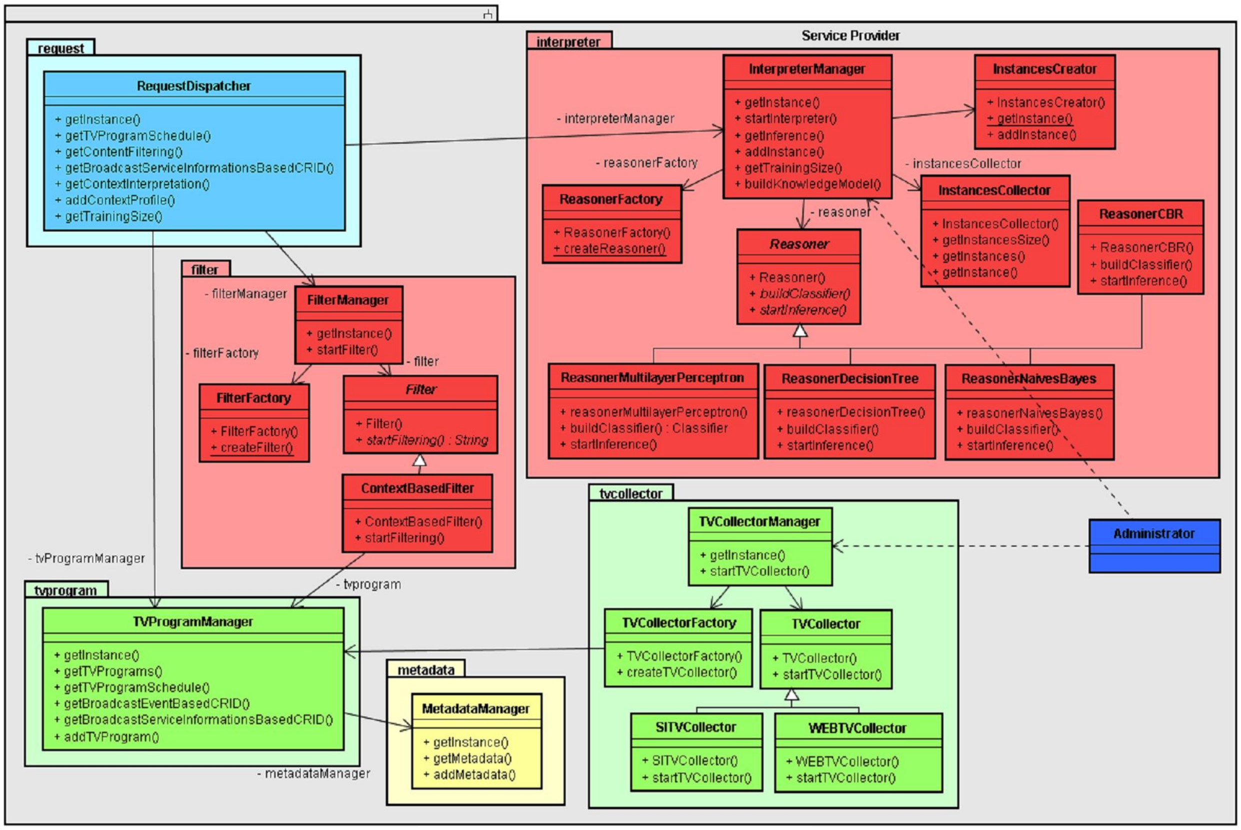

Fig. 4. Classes Diagram – Service Provider

the `Communication` package is a Web Service Client responsible for communication with the service provider subsystem via the interaction channel. Through the method (`setRequetsServiceProvider()`) of this class it is possible to send *Contextual User Profile* with instantiated genre information present in the device as well as receiving information (list of filtered TV programs, inferred genre, among others) originated from the services provider.

In the diagram referring to the user's device subsystem (Figure 2), there is the `recommender` package that corresponds to the Coordinator Recommendation module. There are three classes in the `recommender` package, namely `RecommenderCoordinator`, `Recommendation` and `FeedbackRelevance`. The `Recommendation` class assists the `RecommenderCoordinator` class to access the basic information about the TV program metadata obtained from the list of recommended TV programs. The class `FeedbackRelevance` is responsible for monitoring the implicit or explicit user interaction. When a user interacts through the explicit evaluation of a TV program recommended by the system, the *Contextual User Profile* is created through the method (`setFeedbackRelevance()`). In the implicit way, the computation of the aggregated time of watching a TV program is carried out by methods (`startObservantion)`), (`stopObservation()`) and (`pauseObservation()`).

It is worth saying that the current TV program identification takes place through its CRID (*Content Reference Identifier*) which, according to TV-Anytime is an identifier used to identify content. Finally, the class `RecommenderCoordinator` is responsible for coordinating and controlling the operations of PersonalTVware according to what is required by the recommendation system. For example, the recommendation list of TV programs can be obtained through the method (`getRecommendation()`) that performs the recommendation process based on *Contextual User Profile* without genre information instantiated, which will be obtained by the method of inference.

In the diagram of the service provider subsystem (Figure 4) there is the `interpreter` package consisting of a set of classes that together form the context interpretation service (genres inference from the *Contextual User Profile*). It can be observed that the infrastructure supports the use of four methods of learning: C4.5 algorithm (decision tree) - `ReasonerDecisionTree`; multilayer perceptron with back-propagation algorithm (neural networks) - `ReasonerMultilayerPerceptron`; naive Bayesian classifier (Bayesian learning) - `ReasonerNaiveBayes`, or case-based reasoning - `ReasonerCBR`. The package `request` refers to the Request Dispatcher module.

The class `RequestDispatcher` implements the Web Service to obtain the requests submitted by `RecommenderCoordinator` class, and method invocation corresponding to the requested operation. The `filter` package consists of a set of classes that enable the filtering service of TV programs. The class `FilterFactory` is a filter factory represented by the class `Filter`. This design pattern the implementation of filters based on different approaches. The context-based filtering was implemented via class `ContextBasedFilter`. The `tvprogram` package consists of the class `TVProgramManager`. With this class it is possible to perform basic operations like add or retrieve information about TV programs represented by the metadata of the standard TV-Anytime [9]. Operations under the metadata have been implemented by means of language XQuery[2]. The `tvcollector` package consists of a set of classes that form the service of collecting information about TV programs. It is possible to notice that classes `SITVCollector` and `WebTVCollector` were specified to support two types of collections.

The class `SITVCollector`, despite being represented in the proposed architecture was not implemented in this work because there is no platform for Digital TV available for data transmission via the distribution channel. Therefore, the collection of metadata of TV programs was performed over Internet. Finally, the `metadata` package `MetadataManager` has class that supports the other classes of the subsystem to perform common operations on XML documents. The implementation of PersonalTVware was done in Java. In terms of Web Service technology, it was adopted Apache Axis 1.4[3]. The interpretation and validation of XML documents are performed by a parser that uses the DOM interface (*Document Object Model*).

For implementation of the module responsible for the inference task it was used the Lucene[4] framework, and the APIs of Weka[5] tool. It contemplates series of machine learning methods implemented. The Weka uses a file in ARFF format (*Attribute-Relation File Format*) that contains a set of records to be used in the inference task (knowledge base with the *Contextual User profile*). For Metadata manipulation on the server side and TV program filtering, it was used the language XQuery by mean of XML Database Exist[6].

E. Recommendation Process

Assuming that the user-defined user profile, the process begins when the Coordinator Recommendation module receives a request through the method (`getRecommendation(userID)`) to retrieve a TV program listing. The User Context Manager module is activated to capture and represent the user's current contextual information such as user's identification, location, day and time of the interaction and type of access device, which should generate a history of the user's contexts. Later, the Coordinator Recommendation module, having the user's ID, checks its respective information from the current context and the personal data via the User Context Manager and the User profile Manager modules. Such information is aggregated to create the current *Contextual User profile* without instantiated genre information being applied to the method used in the inference task. Thus, the *Contextual User profile* forwarded by Coordinator Recommendation module to the Context Interpreter module that should infer contextual preferences by TV program genres by means of a machine learning method selected. Therefore, the user can receive personalized recommendations of TV programs which are adequate to his/her current context. The Coordinator Recommendation module receives the contextual preferences inferred from the Context Interpreter module and then joins the explicit preferences defined in the user profile and user´s contextual information to send them to the filtering module.

Finally, the Context-based Filter module carries out the filtering of the TV programs by comparing the contextual information (date, time and location of origin of the TV program and user), explicit and contextual preferences with the descriptions of the TV programs obtained through the TV Program Manager module. Therefore, upon receiving a request featuring the profile and inferred preferences from a specific user as input, the module executes the filtering of TV programs in order to obtain a list of TV programs in decreasing order based on the similarity value which will later be forwarded to the Coordinator Recommendation module.

V. Case Study

The case study consisted of the development of a context-aware recommender system for digital TV, which uses the API PersonalTVware to provide recommendations for TV programs. The recommender system has been implemented as an interactive application (Xlet) and run over a Digital TV emulation environment on computers such as notebooks and desktops, which simulated receivers (fixed and portable). For the subsystem service provider, it has been installed on a server. Figure 5 shows the recommendation screen with a personalized list of TV programs and information about the current context of the user.

Fig.5. Recommender System Screen.

---

[2] XQuery: http://www.w3.org/standards/techs/xquery

[3] Apache Axis: http://ws.apache.org/axis

[4] Lucene: http://lucene.apache.org/java/docs/index.html

[5] Weka: http://www.cs.waikato.ac.nz/ml/weka

[6] XML Database Exist: http://exist.sourceforge.net/

Through this screen the user can select TV programs rated as relevant and also detailed information (Figure 6) of each television program.

Fig.6. Detailed information about the TV Program Screen.

A. *Experiments and results summarized*

In order to test and evaluate the PersonalTVware, a number of experiments were conducted. Experiments were carried out with 10 users who used the system with various *Contextual User profiles* during the 1-month period corresponding to a total of 2197 *Contextual User profiles* recorded in the knowledge base. The experiments involved a collection of TV programs metadata describing titles, synopsis, genre, cast, among others. In total, this contained metadata for 2426 of 14 channels corresponding to one complete week's TV programming. After a period of system training in the lasted one week the knowledge base was generated.

The first experiment carried out by users, consisted in selecting TV programs considered relevant among those who were recommended by each method of machine learning supported by PersonalTVware. The objective was to evaluate which method provided the most appropriate recommendation to the *Contextual User profile*.

In the second experiment were not employed the methods of machine learning. Therefore, only the user profile defined explicitly was used. The objective was to evaluate the quality of the recommendations without the use of context.

According to [4] the efficiency of a recommendation system can be measured by the metrics: *precision*, *recall* and *F-measure*. The *precision* can be used to measure the ability of the system to present only relevant items. While *recall* may be used to measure the ability of the system to provide all the relevant items, the *precision* and *recall* metrics are conflicting by nature i.e., when the system has a high r*ecall* the *precision* is low and vice versa.

Thus, it can be used *F-measure* in a single formula that combines the *precision* and *recall* metrics. According to [4] the closer to one is to measure the value of *F-measure* the more efficient the system. Table 1 presents the averages of the metrics obtained from recommendations based on learning machine methods and that using only the user profile.

TABLE I
COMPARISON OF THE AVERAGE OF METRICS

Method	Precision	Recall	F-measure
C4.5 algorithm	0,861	0,620	0,680
Naive Bayesian classifier	0,797	0,593	0,636
Multilayer perceptron	0,774	0,582	0,606
Case-based reasoning	0,834	0,920	0,841
User profile	0,511	0,174	0, 247

The results indicate that the quality of the recommendations was better when the method employed CBR. According to Table 1, the average *F-measure* of the CBR method is closer to 1. Therefore, the system showed superior performance as compared to the other tested methods. The reason for this result was less discrepancy between the measures of *precision* and *recall* achieved by the system.

But the algorithm C4.5 showed a slightly better *precision* compared to other methods because returned fewer TV programs not relevant to the user's context, but its *recall* was less than the CBR method, resulting in the reduction of the value of its *F-measure*. This result was due to the fact that CBR method returns a greater number of TV programs relevant to the user's context. It can be observed that the quality of the recommendations was lower when only the user profile was employed without the use of context, which shows that the lack of exploration of the context directly impacts the quality of recommendations. It was also noted that there were wide variations among the other learning methods.

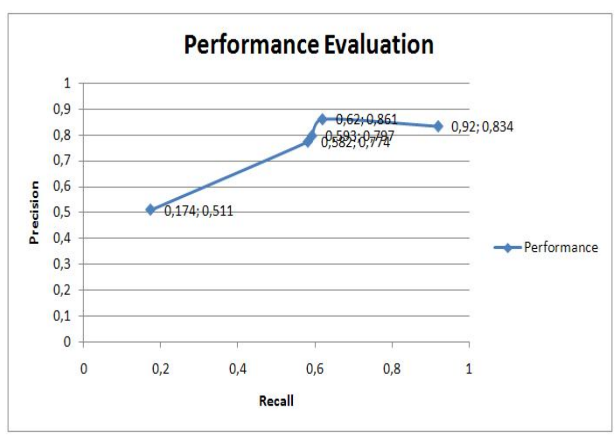

Fig.7. Recall-precision graph.

The Figure 7 illustrates the Recall-Precision Graph [15] [16] which integrates *precision* and *recall*, to evaluate system performance. In the graph, each dot is a pair of recall-precision value. The curve closest to the upper right-hand corner of graph (where *recall* and *precision* are maximized) indicates the best performance. Comparing the results obtained from recommendations based on learning machine methods and that using only the user profile, we can note that the CBR method was superior to the other methods. This can be seen from performance curve is closer to the upper right-hand corner of the graph. This can be explained by the facto the CBR method to present the greatest number of relevant TV programs among all relevant TV programs available to the user's context. We can note that without the notion of context, the user's profile can only provide general recommendations resulting in the reduction of system performance.

VI. Conclusion

It was presented a software infrastructure entitled PersonalTVware that supports the development of context-aware recommender systems for Interactive Digital TV. Based on these experimental results, we conclude that the exploitation of context can improve the performance of a recommender system, especially when methods of machine learning were employed. Regarding the task of favorite's genre inference from the context and user profile, it is noteworthy to note that CBR method provides recommendations of best quality. Users who participated in the experiments evaluated positively the results of the recommendations of TV programs presented. In future wok, we will investigate new recommendation techniques that take advantage of other categories of context (infrastructure, system, application, domain and environment) [17] to improve recommendation quality.

References

[1] Y. Blanco, J. Pazos, A. Gil, M. Ramos, M. Lopes, J. García, R. Redeondo, A. Vilas, B. Martínes and M. Nores, "AVATAR: Advanced Telematic Search of Audiovisual Contents by Semantic Reasoning". in *Proc. of the Personalization of Future TV Workshop*, 2004.

[2] A. K. Dey, "Understanding an Using Context" in *ACM Personal and Ubiq*. Comp. Journal, vol.5, n.1, pp.4-7, 2001.

[3] A. Gediminas and A. Tuzhilin, "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, pp. 149-156, 2005.

[4] Jiangshan, X. et al. The Development and Prospect of Personalized TV Program Recommendation Systems. In Proc. of the IEEE 4th Int. Symp. on Multimedia Software Engineering, 2002.

[5] L.Leite, "Uma Arquitetura de Serviço para Avaliação de Contextos em Redes de TV Digital" in SBRC R,2007.

[6] T. D. Pessemier, T. Deryckere and L. Martens,"Context-Aware Recommendations for User-generated Content on a Social Network Site" in *Proc. of the 7th European Interactive Television Conference*, 2009.

[7] H. Petry, "Um Sistema de Recomendação de Especialistas Sensível ao Contexto para Apoio à Colaboração Informal" in *Simpósio Brasileiro de Sistemas Colaborativos*, p.38-47, 2006.

[8] F. S. Silva, L. G. P. Alves and G. Bressan,"PersonalTVware: A Proposal of Architecture to Support the Context-aware Personalized Recommendation of TV Programs" in *Adjunct Proceedings of the European Interactive TV Conference (EuroITV 2009)*, Leuven, Belgium, pp.39-42, 2009.

[9] H. Zhang and S. Zheng, "Personalized Recommendation based on TV-Anytime Metadata" in *Proc. Of 9th Int. Symp. on Consumer Electronics*, ISCE, 2005.

[10] Y. Zhiwen, H. Xingshe and G.Jianhua, "TV Program recommendation for multiple viewers based on user profile merging" in *Proc. User Model User-Adap Inter*.p.63-82, 2006.

[11] I. H. Witten and E. Franck, "Data Mining: Practical Machine Learning Tools and Techniques", Elsevier, 2005.

[12] L. G. P Alves, F. S. Silva, and G. Bressan, "CollaboraTVware: A Context-Aware Infrastructure with Support for Collaborative Participation in an Interactive Digital TV Environment" in *Int. J. Advanced Media and Communication*, vol. 3, n4, 2009.

[13] P. Cotter and B. Smyth, "PTV: Intelligent Personalized TV Guides" in *Proceedings of the 12th Innovative Applications of Artificial Intelligence (IAAI-2000) Conference*. AAAI Press, 2000.

[14] R. Goularte, M. G. C. Pimentel, and E. S. Moreira, "Context-Aware Support in Structured Documents for Interactive-TV", in *Multimedia Systems*, 11(4):367-382, 2006.

[15] Z. Yu, X. Zhou, X., "TV3P: an adaptive assistant for personalized TV", in *IEEE Transactions on Consumer Electronics*, vol.650, Issue 1,pp. 393-399, 2004

[16] Z. Zaier, R. Godin, and L. Faucher, "Evaluating Recommender Systems", in *International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution*, 978-0-7695-3406-0/08, 2008 IEEE.

[17] R. Goularte, "Personalização e adaptação de conteúdo baseadas em contexto para TV Interativa", doctoral thesis. University of São Paulo, 2003