A Web-like Personalized Advertising System for Interactive Digital TV

Marco Cristo, Ângelo Bitar, Bruno Ferreira Guimarães, Gabriel Penalber and Raiza Tamae Sarkis Hanada
FUCAPI - NEPCOMP
Av Danilo Areosa, 381, Distrito Industrial
Manaus, AM, Brazil
{marco.cristo, angelo.toy, bferreiraguimaraes, gabrielpenalber, raiza.hanada}@gmail.com

César Teixeira and Erick Melo
Federal University of São Carlos - LINCE
Rodovia Washington Luís, km 235, SP 310
São Carlos, SP, Brazil
{cesar, erick_melo}@dc.ufscar.br

ABSTRACT

As many other countries, Brazil is moving from an analogical to a digital TV system. Besides providing a better audio and video quality, an interactive digital TV (iDTV) system will make it possible to users interact with the TV in many ways. Such an interaction will contribute to the emergence of an array of new applications and possibilities, such as new approaches for interactive advertising. In particular, iDTV will provide an environment suitable for the placement of personalized advertisements which have the potential to be interesting for users and financing effective for advertisers and service providers. This kind of advertising approach has being successfully employed in another interactive environment, the Web, with large net gains for advertisers and content providers. Our aim in this work is to propose an architecture for a Web-like personalized advertising system (WPAS) and implement it for the Brazilian Digital Television System (ISDB-Tb), along with all the complementary mechanisms necessary in the ISDB-Tb. In particular, we present the design of the WPAS client module, describing the mechanisms necessary to the identification of the characteristics and interests of the users, the selection of the most appropriate ads, and their insertion into the content broadcasted by the service providers.

Categories and Subject Descriptors

J.m [Computer Applications]: Miscellaneous

General Terms

Algorithms, Design, Standardization.

Keywords

ISDB-Tb, Advertising, NCLua, Personalization, User signature, k-means

1. INTRODUCTION

TV advertising comprises a relationship among three main actors: (1) the TV user; (2) the service provider, that is, the entity which broadcasts the TV services/shows and the ads; and (3) the advertiser which provides the ads that will be broadcasted.

Interactive digital TV (iDTV) systems provide an environment suitable for the placement of personalized advertisements which have the potential to be interesting for users and financing effective for advertisers and service providers. Since many years, this kind of advertising approach has being successfully employed in the Web with large net gains for advertisers and content providers [1].

TV advertising systems are generally characterized by (a) a passive entertainment environment hard to be measured; (b) the use of interstitial ads, that is, ads which are placed in the interval of the shows; (c) the ads are targeted according to show time and content with less attention to personalization; (d) ad slot times are statically defined and exclusiveness is assured, which limits targetedness and personalization; (e) advertisers pay according to measured audience and not the investment return.

On the other hand, web advertising systems take great advantage of aspects such as targetedness, interactivity, and a collaborative environment. Such systems are characterized by (a) an active environment easy to be measured; (b) the ads are placed within the content; (c) ad targetedness is calculated according to user interests (high level of personalization), as well as a richer contextualization (spatial location, content, etc); (d) advertisers compete for ad slots dynamically without any guarantee of exclusiveness; (e) advertisers pay only when the user interacts with the ads.

As previously mentioned, the interactivity of digital TV systems makes it possible the use of advertising systems similar to those operated in the Web, with only a few adaptations. In these advertising systems, the most appropriate ad would be estimated almost at real time taking into consideration interests of users, service providers, and advertisers. To accomplish this, the information about users and ads could be transmitted using a return channel and synchronized in the set-top box[1] (STB) at show time, allowing maximum targetedness and personalization. As in the web, the ads might be obtained by means of an auction system.

Note that in such an advertising system, the service provider can be the company associated with one or more TV channels (for

[1] iDTV terminal.

instance, a network TV company), the distributor of multiple channels (for instance, a satellite or cable TV aggregator) or a information gatekeeper playing the role of an advertising agency (for instance, a web search engine company).

In this work, we propose the architecture of a web-like personalized advertising system (WPAS) for a digital TV environment. In particular, we present the design of the WPAS client module, describing the mechanisms necessary to the identification of the characteristics and interests of the users, the reception of the most appropriate ads, and their insertion into the content broadcasted by the service providers. Note that in this work, we are not concerned with the strategy used by the WPAS server to select the most appropriate ads.

This paper is organized as follows. In Section 2, we present some concepts necessary to understand this work. In Section 3, we present the proposed architecture for the Advertising Local Module. In Section 4, we present the related work we found in literature. Finally, in Section 4, we present our conclusions and proposals for future work.

2. BACKGROUND

In this section, we present some background information necessary to understand this work. In particular, we present the ISDB-Tb, the Ginga-NCL, and a particular Ginga-NCL extension, the Recommender Module.

2.1 ISDB-Tb

ISDB-Tb or SBTVD (short for Sistema Brasileiro de TV Digital) is a technical standard for digital television broadcast used in Brazil and based on the Japanese ISDB-T standard [9]. It differs from the ISDB-T in aspects as video compression, presentation rate, and the middleware software layer used for interaction. In particular, ISTB-Tb adopts the middleware Ginga, composed by Ginga-NCL and Ginga-J modules.

2.2 Ginga-NCL

Ginga-NCL [11] is a subsystem of the Ginga middleware which provides a multimedia presentation environment for declarative applications written in the NCL (Nested Context Language) language and its scripting language, Lua. Ginga adopts the reference TV middleware architecture ITU-T J.200 [6]. The Ginga-NCL module, in particular, is a software component organized into a three-layer architecture. The first layer is the Presentation Layer responsible by running interactive NCL/Lua applications. The second layer is the Common Core, a set of software components which provide basic TVD resources such as video and audio reception and processing, and context management. The third layer is the Operating System Layer which provides management services for basic resource such as processor, memory and network. Ginga-NCL has a current reference implementation (GRI), released under the GPL license.

2.3 Recommender Module

Several extensions have been proposed to Ginga-NCL to provide additional services. For instance, the necessity of sophisticated TV data analysis has stimulated the development of data mining and recommendation modules, such as the Recommender Module (RM) [2]. RM provides a set of services for process scheduling, data persistence, access to contextual information and data mining algorithms. **Figure 1** presents a general description of the RM architecture and its interactions with other Ginga components.

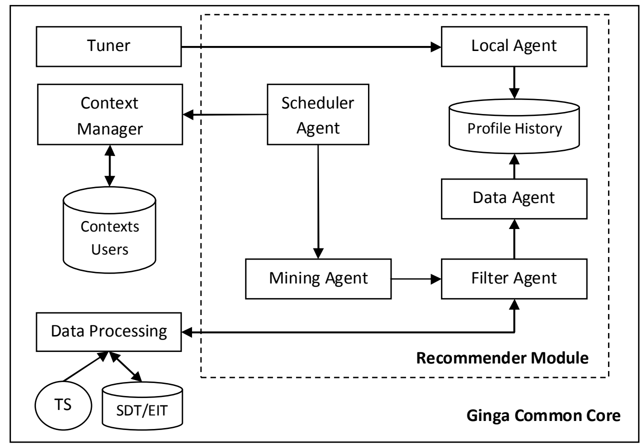

Figure 1: Recommender Module Architecture.

As we can see in **Figure 1**, the Recommender Module comprises five components: Scheduler Agent (SA), Mining Agent (MA), Local Agent (LA), Filter Agent (FA) and Data Agent (DA).

SA is responsible for dispatching mining processes. It communicates with Context Manager to determine if enough resources are available for the applications to be dispatched. MA provides mining algorithms for the most common mining tasks, that is, classification, regression, association, and clustering. Information about the interactions among the user and the STB and the time she/he spends watching a service is captured by the LA. The FA component filters out the results of MA, according to relevance criteria. It also provides an access interface to user interaction history and service metadata tables. The DA component is responsible for recommendation storing in a SQL database.

3. WPAS for iTVD

A complete a web-like personalized advertising system (WPAS) for a iTVD environment should support an auction system, take advantage of collaborative information derived from user and advertiser communities, and minimize transmission costs due to the probably low availability of a broadband return channel in the Brazilian TVD environment. To meet these requirements, the following key characteristics are proposed:

- User identification and part of the ad match processing takes place locally in the STB. This minimizes transmission costs and remote centralized processing;
- The ad recommendation is calculated in a centralized way, making it possible the use of collaborative recommendation approaches;
- The infra-structure for ad inclusion is web-based to make it easy the implementation of an auction system;

The general architecture of WPAS is described in **Figure 2**. As we can see in **Figure 2**, WPAS is composed by three main components: Advertiser Module, Service Provider Module, and User Module. Through Advertiser Module, interests and ads of advertisers are gathered and transmitted. User Module identifies users, receives and places ads. Finally, Service Provider Module aggregates information from advertisers, users, and show contents to find the most appropriate ads. From these modules, User Module is the only one to reside in the STB.

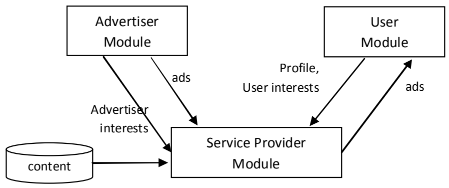

Figure 2: WPAS General Architecture.

In this paper, we present the detailed architecture of the WPAS User Module for the Brazilian Digital Television System (ISDB-Tb) using Ginga-NCL.

To work, the WPAS User Module has to identify the current TV user. Given the nature of a TVD environment, we do not expect that users identify themselves spontaneously. Thus, the STB would be able to estimate the current user based on his/her behavior characteristics. Since the current Ginga-NCL specification does not describe such an implicit identification service, we also propose an extension to GRI which supports user implicit identification.

In next subsections, we describe an extended version of the original Ginga Context Manager which supports user implicit identification and for which we refer to as Implicit Context Manager. We also describe the set of NCL/Lua application which will run at the top of Ginga-NCL responsible to receive and transmit user information and ads, as well as to place ads into the broadcasted content. We refer to this set of applications as Advertising Local Module.

3.1 Implicit Context Manager

Context management is one of the services provided by the Ginga Common Core layer through a component called Context Manager (CM). Ginga CM records information about user accounts and preferences. This data is recorded for the current authenticated user. If no user is authenticated, a generic user (*default*) is taken as the current user and data is collected for him. Since this is the most common situation (default is the current user), the contextual information gathered is more about the general TV usage pattern than about individual user behaviors. Thus, the availability of a service for individual user identification should minimize that problem, allowing for a higher level of service personalization.

In this section, we describe a new component of Ginga-NCL, the Implicit Context Manager (ICM). ICM extends the original Ginga CM to provide a service for guessing the current user based on its behavior characteristics.

Given the necessity of assessing the user identification from a large amount of user behavior data, the ICM uses several services provided by the Recommender module.

ICM is periodically launched by SA. Since user changes are more common during the transition of a show to another[2], the ICM continuously verifies such a transition event. Always that this event is observed, ICM starts the implicit user identification process. To accomplish this, from MA, ICM uses clustering algorithms. It obtains information about service schedules and the user history log from FA component. Finally, by using the services originally provided by Context Manager, the ICM gets information about authenticated users as well as saves information about implicit users, including the profiles which characterize them.

Note that all the information in ICM is available as context data in the Ginga presentation layer and, by extent, to any running NCL and Lua applications[3]. Further, the CM notification mechanism can be used to notify applications about implicit user changes.

The key element of ICM is the algorithm used to estimate the current implicit user, depicted in **Figure 3**. Every time ICM is called, independently of observing service transitions, it collects statistics about the content watched by the user. In our prototype, in particular, each user is represented by the time he/she spends watching each one of the many available service subgenres (news, sports, comedy, drama, etc)[4]. We refer to this user representation as an implicit user profile. Implicit user profiles are recorded by ICM.

The statistic collected in the last service transition is accumulated into the implicit profile of each implicit user candidate recorded by the algorithm. For each new subgenre transition, a new candidate is created. If a new user *u* is authenticated or if the degree of certainty on the implicit profile of *u* is greater than a threshold, all the previously candidates created during the active section are discarded[5]. At the same time, that implicit profile is associated with the user account of *u*. Note that each user with an account (that is, all the users,

---

[2] This information was obtained from a TV audience log, provided by a company that collects TV ratings in Brazil.

[3] Context variables *default.curiprof* and *default.iprof*, accessible to any NCL/Lua application, indicate the current implicit user id and the corresponding implicit profile, respectively. Variable *default.curiprof* has a format *userid*:*profileid*, where *userid* corresponds to the current user id and *profileid* corresponds to the current profile id of this user. Variable *default.iprof* presents a series of numbers separated by commas which correspond to the values of the attributes used to represent the implicit user profile.

[4] Note that the profile might be composed by any available evidence other than service subgenres, such as channels, demographic data (age, gender, zip code, etc) or a combination of them. Also, no change is necessary in the algorithms if other set of evidence is used to represent the user profiles. As previously mentioned, variable *default.iprof* stores the list of attribute values used to represent the user profile, separated by commas.

[5] If the degree of certainty calculated by ICM for a user *u* is greater or equal to 0.95, the ICM internally authenticates *u*.

except the *default*) can have only one implicit profile in the system.

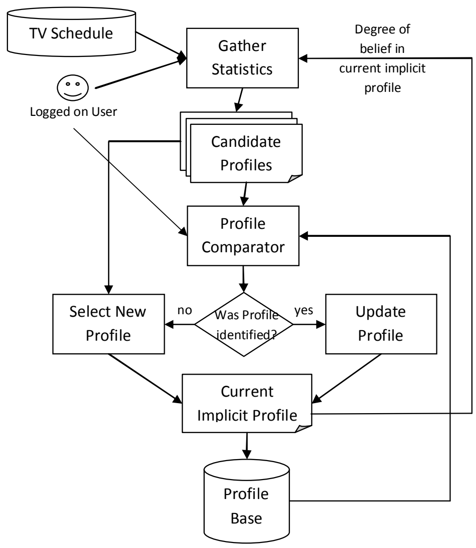

Figure 3: Estimating the current implicit user.

ICM, then, compares each one of the candidate implicit profiles with all the profiles in the profile database (the profiles are saved as context variables in the STB)[6]. In the case in which no database profile is identified[7], one of the candidates is selected as the current user profile and inserted into the profile database[8]. In the case in which a database profile is identified, its statistics are updated. If the identified profile *p* differs from the current implicit profile, *p* becomes the current one. NCL/Lua applications can note such profile change by monitoring variable *default.curiprof*.

As the time goes on, the profile base converges to a set of profiles corresponding to the users of that TV. Note that certain profiles may characterize user groups and not only individuals.

The profile base is started with generic profiles. These profiles correspond to typical user behavior classes (children, teens, adults, and seniors; men and women; A, B, and C social classes).

To avoid a large number of profiles in this base and, thus, to keep it with a reasonable size, compatible with a typical family, ICM periodically identifies very similar profiles and groups them[9]. Further, profiles not used for a long time are removed from the base.

3.2 Local Advertising Module

As seen in the previous section, to determine the local context of each user, it is necessary to process a large amount of information on user behavior and service content. In a completely centralized architecture, this would imply both on larger transmission costs and server processing time. To avoid this, our web personalized advertising system (PWAS) is based on a client-server architecture, where the client resides in the STB, it is able to receive ads and insert them into the broadcasted content, as well as to send to the server the contextual information necessary to the selection of the most appropriate ads. This client, the Local Advertising Module (LAM), is described in this section.

LAM comprises three components: Ad Schedule Receiver (ADSR), Ad Synchronizer (ADSY) and Ad Placer (ADPL). ADSR is the component which receives ad schedules and the ads. ADSY synchronizes each ad according to the received ad schedule. Further, it monitors user changes and notify them to the advertising server. Finally, ADPL places the ads within the content being broadcasted. All these components were implemented using Lua, which makes it possible the server modifies its local advertising strategy by sending new versions of these applications.

Note that in the descriptions in this section, we assume that:

- The local advertising module is restarted when the channel is changed. Thus, none of the described components needs to be notified about channel changes. This does not imply that the received schedules are discarded. They can be useful if the user comes back to a channel for which a valid schedule is still available;
- The location of the advertising server is sent along with the local advertising components by the service provider which also knows the channel which the TV is tuned in. So, we can say that the advertising server is dedicated to that channel. Note that this does not prevent the creation of large advertiser/user communities and facilitates the adoption of several client schedule approaches. It also facilitates the maintenance of local caches of ads for popular channels;

Figure 4 shows the architecture for the proposed local advertising module and its interactions with other Ginga components and the advertising server. As we can see in **Figure 4**, the ADSR component listens to the advertising server waiting for new ad schedules and ads. After receiving new ads, ADSR stores them locally and notifies ADSY about them. Note that ADSR is responsible for implementing politics to save space and

[6] Comparisons are carried out using a simple vector comparison method, based on the cosine similarity metric [3]. In the future, we intend to use a more sophisticated probabilistic method which takes into account aspects as subgenre popularity and user assiduity.

[7] More precisely, if no candidate profile matches a database profile with a certainty degree of similarity greater than a minimum threshold, we consider the observed behavior is new.

[8] The selection criterion adopted in our prototype is the choice of the oldest candidate profile in the current watching section. As alternative, we could choose the most distinct candidate or a combination of candidates.

[9] In particular, we used an iterative clustering algorithm able to forecast the number of clusters. The basic clustering method we used is *k-means* [12]. The forecasting of the number of clusters is performed by finding the *k* value for which the Bayesian inference coefficient (BIC) is not able to present significant gains anymore [12]. This *k* value can be obtained by applying the L method [10].

network band. To accomplish this, ADSR stores ads until to occupy a maximum limit, when it starts to remove the least frequently used ads. It also asks to the server to send only ads not present in the STB. The information in the schedule corresponds to the ad insertion time (date and time), the exhibition time, the ad close button coordinates, the ad identification, and the channel for which the ad is valid (useful for cache strategies). The ads are transmitted as unique compressed files containing the corresponding Lua script and media files.

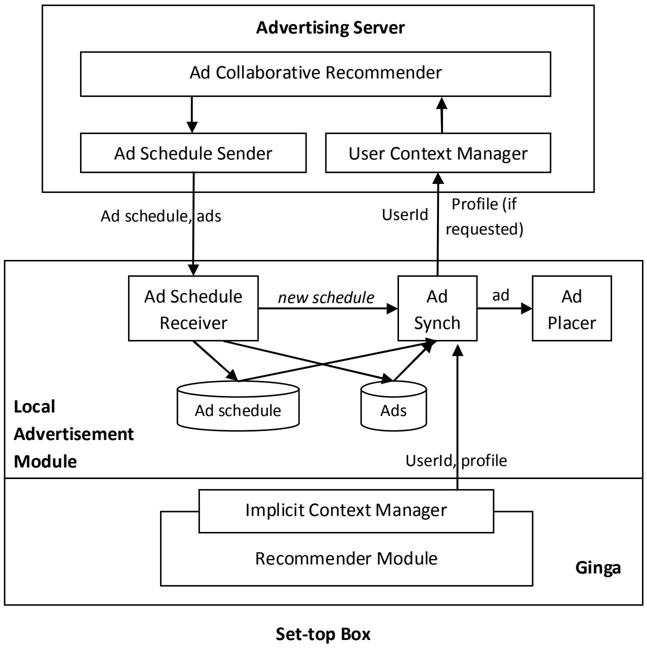

Figure 4: Local Advertising Module

ADSY wakes up every half second to verify if a new ad has to be placed into the content according to the current schedule. If a new ad is available, ADSY removes the corresponding schedule entry, retrieves the ad from the database, decompresses it and notifies ADPL to place it into the content. The ADSY notification to ADPL also includes the ad close button coordinates.

ADSY also monitors the ICM component, waiting for an implicit user change. When this event is observed, ADSY collects the new implicit user identification and its corresponding implicit profile. It then stops ad placements using the current schedule and notifies the advertising server, through the return channel, about the new current user. As an answer to this notification, the server can request the user profile, which is sent to it by ADSY. Once stopped, ADSY only restarts ad placements after receiving a notification from ADSR (about a new valid schedule).

Finally, ADPL is the component that places the ad into the content. It shows the ad close button in the requested position and runs the Lua script corresponding to the ad. If the user closes the ad, ADPL finishes the execution of the corresponding Lua script. **Figure 5** depicts an ad placed at the top of a video screen. In this figure, the close button is displayed as a small rectangle with a white down arrow. Note that this example ad includes an image and some text.

Figure 5: Ad placed at the top of the video area.

```
local img   = canvas:new('propmedia/Prop2.png')
local dx,dy = img:attrSize()
local fundo = {img=img,x=0,y=0,dx=dx,dy=dy}

local img1   = canvas:new('propmedia/danca.png')
local dx,dy = img:attrSize()
local foto = {img1=img1,x=10,y=0,dx=dx,dy=dy}

local menorsz=20

require 'desenha-ad'

frase1="lulu sabongi - a arte da danca do ventre"
frase2="site completo de pesquisa e desenvolvimento,
cultura arabe, videos didaticos, "
frase3="cds e milhares de dicas para a mulher. aulas e
workshops. inicio imediato."

calculasz(frase1,20)
calculasz(frase2,20)
calculasz(frase3,20)

l1=2
l3=l1+20+menorsz
while (l3 > 300)do
        menorsz = menorsz-1
        l3=l1+20+menorsz
end

desenhafundo()
escrevetexto(frase1,20,125,l1)
escrevetexto(frase2,menorsz,125,l2)
escrevetexto(frase3,menorsz,125,l3)
```

Figura 6: Lua script corresponding to the ad depicted in Figure 5.

Figure 6 presents the Lua application corresponding to the ad shown in **Figure 5**. By allowing the execution of any Lua script, the ADPL makes it possible the placement of many ad formats (including interactive ones) at different regions in the video.

4. RELATED WORK

Many works in literature have addressed the problem of recommending ads considering interests of users, advertisers, and service providers. The works most similar to ours are these focusing on video advertising in the web and advertising architectures for iDTV.

Regarding web video advertising, a very interesting study was proposed by the authors in [7]. They suggested an algorithm for determining the best ads and video insertion points based on video content. More specifically, they use evidence surrounding the video available in the site where the video is located. In fact, the use of this kind of evidence for ad selection is also explored in other works, such as [5]. In that work, the authors also consider user interests when estimating the best ads such that their approach can be considered a personalized one and, by extent, more similar to ours. However, unlikely our study, both works deeply rely on a Web 2.0 environment. In such environments, much collaborative textual evidence is available, explicit authentication is very common, and a typical user prefers to watch TV alone.

Regarding digital TV advertising, similarly to our work, the authors in [4] use behavior information to guess the user in front of the TV and recommend ads based on that evidence. Unlike that work, however, we also propose an architecture for the placement of interactive ads and implement such an architecture for the ISDB-Tb. Other work on ad recommendation that takes into consideration user personalization was proposed in [8]. In that one, a mobile device is used as additional source of user information and as a means for advertising. Note that our proposal is different from that one since we guess the user and do not rely on external devices to gather statistics or store profiles.

5. CONCLUSIONS

In this work, we proposed an architecture for a Web-like personalized advertising system (WPAS) and implemented it for the Brazilian Digital Television System (ISDB-Tb). In particular, we presented the design of the WPAS client module, describing the mechanisms necessary to the identification of the characteristics and interests of the users, the selection of the most appropriate ads, and their insertion into the content broadcasted by the service providers. As future work, we consider to improve several aspects of our architecture and algorithms, as well as, to test it on an actual TVD environment. Regarding our architecture, we intend to extend it to take advantage of implicit user information to recommend interstitial ads. Our current architecture is focused only on placing ads into the broadcasted content. Regarding our algorithms to guess the current user, we intend to study different information for representing the users and evaluate several strategies to minimize the number of profile candidates. We also intend to evaluate more sophisticated profile comparison methods. In particular, we will use a probabilistic method that take into consideration aspects as genre popularity and the total time the user spends watching TV.

6. REFERENCES

[1] Aris Anagnostopoulos, Andrei Z. Broder, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel . "Just-in-time contextual advertising". CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM. November 2007.

[2] Paulo Muniz de Ávila and Sérgio Donizetti Zorzo. "Recommender TV - A Personalized TV Guide System Compliant with Ginga". SIGMAP 2009: 149-156, 2009.

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. "Modern Information Retrieval". Addison Wesley, first edition, 544p.1999.

[4] K.-C. Chen and W.-G. Teng, "Adopting User Profile and Behavior Patterns in a Web-TV Recommendation System," Accepted by Proceedings of the 13th IEEE International Symposium on Consumer Electronics, Kyoto, Japan, May 25-28, 2009. (EI)

[5] Rebeca P. Díaz Redondo, Ana Fernández Vilas, José J. Pazos Arias, Alberto Gil Solla, Jorge García Duque. "Sponsored Advertising for IDTV: A Personalized and Content-Aware Approach" 27th International Conference on Consumer Electronics (ICCE) - jan 2009

[6] ITU. "ITU-T Recommendation J.200: Worldwide common core - Application environment for digital interactive television services", 2001.

[7] Tao Mei, Xian-Sheng Hua, Linjun Yang, Shipeng Li. "VideoSense - Towards Effective Online Video Advertising," ACM International Conference on Multimedia (ACM MM), pp. 1075-1084, Augsburg, Germany, Sept. 2007.

[8] Katrien De Moor, Istvan Ketyko, Wout Joseph, Tom Deryckere, Lieven De Marez, Luc Martens and Gino Verleye. "Proposed architecture and algorithm for personalized advertising on iDTV and mobile devices" IEEE Transactions on In Consumer Electronics, IEEE Transactions on, Vol. 54, No. 2. (2008), pp. 709-713.

[9] NHK. "Overview of ISDB-T specifications". Available at Internet in www.nhk.or.jp/strl/open99/de-2/shosai-e.html. 2010.

[10] Stan Salvador and Philip Chan. "Determining the number of clusters/segments in Hierarchical Clustering". Tools with Artificial Intelligence. ICTAI 2004. 16th IEEE International Conference on (17 November 2004), pp. 576-584, 2004

[11] Luiz Fernando Gomes Soares, Rogério Ferreira Rodrigues e Márcio Ferreira Moreno. 2007. Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. Journal of the Brazilian Computer Society. 2007, Vol. v12, pp. 37-46.

[12] Qinpei Zhao, Ville Hautamaki, and Pasi Franti. "Knee Point Detection in BIC for Detecting the Number of Clusters". Lecture Notes In Computer Science, vol. 5259. Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems. Juan-les-Pins, France, pp. 664 – 673, 2008.