Context-Aware User Profiling and Multimedia Content Classification for Smart Devices

Abayomi Moradeyo Otebolaku
Telecommunications and Multimedia Unit, INESC TEC
Faculty of Engineering, University of Porto, Portugal
abayomi.otebolaku@inescporto.pt

Maria Teresa Andrade
Telecommunications and Multimedia Unit, INESC TEC
Faculty of Engineering, University of Porto
Porto, Portugal
mandrade@fe.up.pt

***Abstract*— Current solutions for delivering adapted multimedia content to mobile users take into account only a limited set of contextual information, and can be seen as passive solutions. We propose a solution that anticipates user's needs based on the contexts of use and preferences, to deliver media content to users in mobile environments. This article describes the profiling approach of the proposed solution, and a context-aware content-based recommendation for smart devices. Recommendations are issued based on user history, driven by real-time contextual conditions.**

Keywords: Context; recommendation; mobility; user profile; vector space model; adaptation.

I. INTRODUCTION

The proliferation of multimedia-enabled smart devices, the online availability of large volumes of content, and the advances in wireless technologies have driven the desire of mobile users to gain access to content anytime, and anywhere with the best possible quality. However, the existing work on anywhere, anytime multimedia access, otherwise known as the Universal Multimedia Access (UMA) [1], considers only the challenge of explicitly requesting content, taking into account a limited set of contextual constraints. They generally overlook the aspect of anticipating user's needs. However, in mobile environments, user's consumption preferences are influenced by factors such as the current time of the day, location of the user, day of the week, the user's activities, and the type of terminal, etc. [6]. All these factors, and others, can be seen as contextual information, characterizing the situation of users when consuming content. In this paper, we describe the work endorsed in that direction, seeking a solution that combines content based recommendation for smart devices, adapted to the context of use.

The contribution of this work relies not only on combining these two techniques, but also on incorporating the contextual dimension in both processes as shown in Fig.1. This is accomplished by: 1) the development of contextualized user profiles, which classify user preferences on the basis of their contextual situations; 2) the use of the Vector Space Model (VSM) [2] to learn the similarities between candidate content and user preferences in specific contexts of use; and 3) the use of a knowledge-based approach to process context information.

The rest of the paper is organized as follows. Section II presents the related work, whereas section III describes the proposed context-aware content-based recommendation approach. In section IV, usage scenario and the system's operation are presented. The system architecture and some preliminary evaluation results of the recommendation engine prototype is described in section V. Section VI discusses the results, concludes the article, and gives our future direction.

II. RELATED WORK

UMA has been investigated extensively in recent years, aiming to realize the consumption of any content anywhere at any time, by adapting the content to the user, device, and network contexts [1,[3-4]. Though UMA uses context awareness in the adaptation decision process, this knowledge has been applied to adapt only an explicitly requested content to meet usage constraints and user preferences. Contextual information, however, can be used to anticipate users' needs [5]. Only very recently, and with limited range, has contextual information been considered as an important factor in recommendation processes [5-6], [7-8]. To anticipate and to generate recommendations, three main traditional approaches are explored for content recommendations [5]: 1) based on opinions and preferences of other users, designated as Collaborative Filtering (CF); 2) based on the previous user's consumption history, and descriptions of available candidate content, referred to as Content Based Recommendation (CBF); and 3) a combination of CF and CBF denoted as Hybrid Recommenders (HR) [5]. However, all these approaches depend on user ratings, which are not always available in practice.

Recently, contextual information has been incorporated in these approaches [5-6], [7] to address the problems of the traditional approaches stated above. In [8], Adomavicius et.al. present a multidimensional recommendation model, integrating context as one of the model's dimensions. A. Chen [9] presents a context-aware collaborative filtering approach, where recommendations are generated for users in a specific context, based on what other users with similar profiles have consumed in such contexts. In [10], A. Costa et.al. explore the synergy between recommender systems and context-aware computing, aiming at making service offers more efficient, personalized, and proactive. Z. Yu et.al. present in [11] an interesting work based on a hybrid recommendation approach, using context awareness to recommend and to adapt media for smart phones.

Though the work presented in this article is similar to the above-referred solutions, however, rather than relying on user ratings, which are not always available in practice, it relies on a contextual user profile model that tracks in real-time the

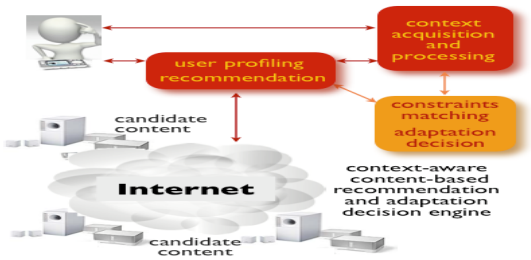

Fig. 1 - Functional view of the proposed system

user's contextual preferences [12]. It also relies on a context pre-filtering paradigm rather than on the context-modeling paradigm [8]. The context modeling requires a true integration of context processing in the recommendation process. The context pre-filtering paradigm promotes the use of contextual data with flexibility, providing the recommendation system with the freedom to use whatever, even none, contextual data. This approach does not require a user rating, and this can help to address the new user problem of CBF.

Finally, this work proposes an original and quite flexible user profile structure, introducing the concept of contextualized user profiles, which incorporates the context dimension whilst enabling easy update and usage.

III. I. PROPOSED CONTEXT-AWARE CONTENT BASED RECOMMENDATION

A. *System Overview*

The proposed solution leverages three important components as depicted in Fig.1. The first component is the context recognition process model, which runs on the user's phone, monitoring, learning, and predicting the user's contexts. It gathers events from the user's mobile device built-in sensors, preprocessing them, and inferring user's high-level context as described in [12]. The high-level context is then instantiated into a context ontology to realize a much more meaningful high-level contextual information as described in [13]. The recognized context is sent to the second component called contextual user profiling module, which consists of the preferences of the users and a process that relates them to the user's present and past contexts. It also learns the user's content consumption preferences and updates the profile accordingly. It stores the user's consumption history and preferences in a contextual profile repository. The third component, the recommendation service consists of a context-aware content-based algorithm and a media content profiling module. The former uses contextual information in a content based filtering process, using the latter to extract media metadata, which is used in the vector space model (VSM) to learn the contextual similarity between a target user and candidate media items.

B. *Contextualized User Profile*

A user profile summarizes the preferences of a user, normally based on the history of the user's actions [15]. In this work, it summarizes user's consumptions into a limited set of categories. Categories are characterized by genres, which in turn are characterized by a number of properties. Several genres may be associated with one category. Several properties may be associated with one genre. Additionally, it incorporates the contextual dimension, associating one or more inferred context with each *category-genre-property* concept. Accordingly, it can generate a contextualized user profile for each inferred context, thus obtaining the preferences of the user in each specific context. Examples of such categories, genres, properties, and contexts are listed in table 1. Though limited, they provide a representative sample of the type of content consumed by mobile users as well as the possible situations, in which they consume those content.

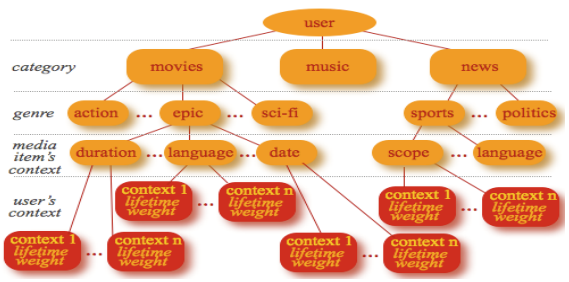

Fig. 2 - General tree structure for the user profile

TABLE 1 EXAMPLE BASIC INFORMATION IN USER PROFILE

Level	***Node***	***Possible values***
Category	Category	Movies(others are news and music)
Genre	Genre	action, animation, comedy, drama, documentary, epic, horror, politics, sci-fi, sports, thriller, other
Property/ media item's context	Language	English, French, German, Italian Portuguese, Spanish, other
	Country	Canada, France, Germany, Italy, Portugal, Brazil, Spain, UK, USA, other
	Date	old (>5 years), recent (<5 years, >1 years), present (<1 year)
	Duration	short (<30 mn), medium (>30 mn, <75 mn), long (>75 mn)
user's context	Context	@homeWeekMorningSitting,@homeWeekAfternoonLaying, @homeWeekEveningStanding,@homeWeekendMorningSitting,@homeWeekendAfternoonSitting, @homeWeekendEveningWalking,@officeMorningSitting, @officeAfternoonSitting,inTransitMorningStanding, inTransitAfternoonSitting, inTransitEveningStanding, walkingMorning, walkingAfternoon, walkingEvening, others

The user profile is represented as a four-level tree, as illustrated in Fig. 2. The root of the tree is user u_i with demographic information. The first level of nodes corresponds to the category; the second level represents the genre; the third level contains the properties of a given category-genre. This level provides the media item's context, characterizing at a finer level of detail the consumed content, and thus the user preferences. The leaf nodes provide information about the contexts in which the user's preferences occur. The leaf nodes have three fields - *type, weigh,t and lifetime* – whereas all other nodes have only the type field. In the leaves, it represents the type of context. The weight provides information on the number of times the user has consumed items of that category-genre-property in that specific context. The lifetime parameter provides an indication of the time elapsed since the last consumption. Its value is set to 1 when the user consumes a content of that category-genre-property, and periodically decrements it if the user does not consume such content. Smaller values indicate that the user has lost interest in that type of content, regardless of the value of its weight. In practice, it allows giving more importance to items consumed more recently. Above the leaves, nodes that have multiple children inherit the lifetime parameter of its child with the smallest value, whereas, the weight of property nodes is the sum of weights of the user's context nodes. The weight of a genre is the largest weight of its properties. The weight of a category is obtained by summing up the weights of all its direct children. These calculations are done using contextual information to filter out only the values assigned under that specific context. Alternatively, it is possible to neglect the contextual information and perform these calculations using all data available in the profile, providing flexibility to provide recommendations even when contextual information cannot be obtained. If the system is unable to acquire contextual information, it registers the values of the weight and the lifetime parameter under a generic context (with value "other").

This structure, thus provides sufficient flexibility to enable the system to provide recommendations at different levels, (items of a certain category-genre-property up to items in a certain category only) either with or without the contextual information. The process for calculating the weight and lifetime for any node when creating the contextualized user profile vector is exemplified in the next section, using a hypothetical user profile.

C. Contextualized User Vector

To classify candidate content, and to obtain a list of recommended items, a vector is created from the user profile. A global user vector has as many elements as the number of different category-genre-properties that appear in the complete user profile tree. A contextualized user vector typically has a much smaller number of elements, corresponding to different category-genre-properties associated with specific contexts under consideration. The contextualized user vector Vc is thus built using contextual data to filter the profile. The value of the current context of usage is compared with the leaves of the profile tree (context nodes) to identify the upper nodes that provide values for the elements of Vc. Only nodes whose leaves match the current context are selected and retrieved. Each element in the vector is a pair keyword-intensity. Keyword is the textual value of nodes (the type field); intensity is obtained by multiplying the values of the fields' weight and lifetime of the respective nodes. For instance, consider the hypothetical user profile of Mr. X represented in Fig. 3. Assuming that the system has inferred that Mr. X is in context C_1, the elements that will be included in the contextualized user profile vector are those that have leaves with context value C_1. The intensity of those elements in the media's context is calculated by summing up the products [*weight* x *lifetime*] of all their occurrences (e.g., the node with value "English" occurs three times for context C_1; therefore the intensity of "English" is the sum of the corresponding three intensities: 0.1 x 0.5 + 0.2 x 0.7 + 0.4 x 0.8 = 0.51). The intensity value of the retained elements at this level is obtained by visiting their child nodes, using a breadth-first traversal. The same process applies to the retained elements of the category level. The intensity of the elements that belong to the genre level is the largest value of their children. Accordingly, these values are obtained, using a depth-first traversal. Fig. 3 represents the resulting contextualized user vector.

$$V_C = [(\text{medium},0.08),\ (\text{english},0.51),(\text{old},0.06),(\text{recent},0.2),\ (\text{action},0.08),\ (\text{epic},0.14),(\text{sci-fi},0.32),(\text{movies},0.54)]$$

Fig. 3 –Contextualized User Profile Vector

D. Updating the User Profile

The user profile can be updated either explicitly or implicitly [16]. The former grants users the ability to modify the values assigned to their preferences by the system. The implicit approach involves preference learning through implicit feedback. The profile is implicitly updated whenever users consume any kind of media item. A context agent of the smartphone platform collects relevant contextual data from th user's mobile device embedded sensors (accelerometer, rotation and orientation sensors) as well as from the network connectivity. The system analyses these values to infer that the user is in context C_i. Once the user context is inferred, and the consumed content is characterized, the user profile update process is performed.

$$Sim\ (v_c, v_m) = \frac{V_c \cdot V_m}{V_c \times V_m} = \frac{\sum_{i=1}^{n} c_i \times \sum_{i=1}^{n} m_i}{\sqrt{\sum_{i=1}^{n} c_i^2} \times \sqrt{\sum_{i=1}^{n} m_i^2}} \qquad (1)$$

$$V_c = [c_1, c_2, \ldots, c_n]\ ;\ V_m = [m_1, m_2, \ldots, m_n]$$

V_C and V_M are the contextualized user profile and media item vectors respectively.

The system compares the media metadata with the list of properties it uses. Every match triggers a search into the current user profile. If a node with the same value as the current context is found, the corresponding *weight* and *lifetime* parameters are updated. Otherwise, a new node is created with an associated leaf capturing the contextual information. It is possible that a matching node already exists but with leaves indicating different user contexts. In this case, only an additional leaf, representing the identified context, is created with the new *weight* and *lifetime* parameters. The assignment and update of the values of the *weight* w and *lifetime* γ parameters are performed using equations 2 and 3 presented.

$$w = (1-\alpha)\cdot w + \alpha \cdot \beta \qquad (2) \qquad \gamma = 1 - \left(\frac{t}{45}\right)^5 \qquad (3)$$

The factor α has a value in the interval [0;1], assigning less or more importance to new data being introduced in the node. For newly created nodes, α assumes the value 1; otherwise its value, determined based on experiments, is 0.5. The factor β is the score given by the system and can assume the values 1 or -1. The value 1 indicates that the user has consumed a new item having the characteristics as described by that node (i.e., indicates a preference of the user). The value -1 is used to indicate that the user has rejected an item with those properties.

The *lifetime* parameter is always set to 1 for matching nodes, by assigning the value 0 to the factor t. The factor t represents the number of days elapsed since the last time the user has consumed an item with the characteristics described by the node. With equation (2), the relative importance of the node remains above 0.9 during the first 30 days after it has been visited, rapidly decreasing to zero after that period (non-negative values are automatically converted to zero). For all other nodes, the update of the *lifetime* parameter is performed daily by linearly increasing the value of t. This way, nodes that represent items that have not been seen for a long period, will have a smaller, possibly none, impact on the user preferences evaluation.

E. Media Item Profile Model and content classification

The system classifies candidate content as being relevant to the user by measuring its similarity to the contextualized user profile vector V_C. It is therefore necessary to create a media vector, V_M, for each candidate media item. To describe media items, the proposed system relies on the availability of semantic metadata using the MPEG-7 MDS semantic tools. Given that in practice most of the media resources available online lack this kind of metadata, the proposed system incorporates alternative methods for obtaining semantic descriptions of the candidate content. One of such alternatives is to query the Internet using existing open source API such as the Internet Movie Data Base (IMDB). For each media item, a vector V_M is initially created as an exact replica of V_C. Then, for each element of V_M, the system inspects the MPEG-7 metadata for a match. If it finds a match, it retains the intensity value of the matching element in V_M. Otherwise, it allocates the value 0 (zero). Fig. 5 shows an example MPEG-7 metadata of a candidate media item. Fig.6 illustrates the corresponding media vector, built using the contextualized user vector presented in Fig.4. This idea was adapted from [12] but implemented in a simpler way, without defining an additional vector of attributes and weights. Instead, the intensity values in V_C are directly used. Additionally, it assigns varying levels of importance to the elements, according to the contribution they provide in the

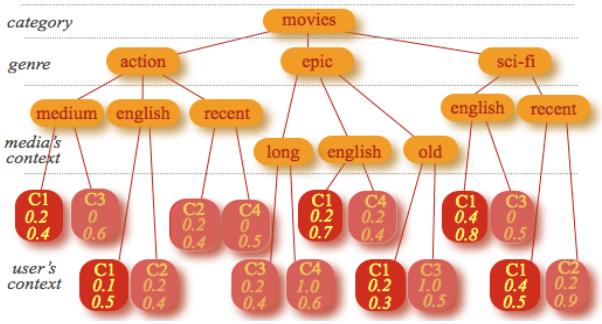

Fig. 4 – Hypothetical user profile

characterization of the content or the user's preferences. For example, for the category "movies", the property "genre" is the one considered more relevant that say, "keywords" or "duration".

```
<MPEG7>
...
<DescriptionMetadata>
<Genre>action</Genre>
<Language>English</Language>
<MediaDuration>PT0h55M30S</MediaDuration>
<CreationTime>2005-10-10T19:45:00+09:00</CreationTime>
<Textannotation>
...
</MPEG7>
```

Fig. 5 - Excerpt of hypothetical MPEG-7 metadata

$$V_M = [(\text{medium},0.08),\ (\text{english},0.51),\ (\text{old},0),\ (\text{recent},0.2),\ (\text{action},0.08),\ (\text{epic},0),\ (\text{sci-fi},0),\ (\text{movies},0.54)]$$

Fig. 6 - Hypothetical candidate media vector

Table 2 -Usage Environment Context Information

Terminal Context	Maximum Frame Rate	25fps
	Battery Time Left	2 minutes
	Display Size	WVGA(480 x 800)
Network Context	Available Bandwidth	142 kbps
Environment Context	Noise Level	85.5 dB
User Preference	Frame Rate	30fps
	Frame Size	Xhdpi(720 x 1280)

Table 3 -Usage Environment Context Information User

Media Visual Coding	**Frame Size**	**Xhdpi(720 x 1280)**
	Frame Rate	30fps
	Bit Rate	1024 kbps

Table 4 - Adaptation Decision Parameters

	Input Values	**Output Values**
Frame Rate	30fps	20
Bit Rate	1024 kbps	142kbps
Spatial Resolution	720 x 1280	480 x 800
Volume Level	0.9	N/A

As such, the system uses pre-defined relative importance values to adjust the relevance of the computed elements in the media vector. These pre-defined multiplicative values can be adjusted based on experiments. Based on the experiments performed, the values [3, 3, 2, 2, 1, 1] were assigned to the elements [category, genre, language, duration, country, date]. Applying these multiplicative factors to the media vector in Fig.6 results in the final vector represented in Fig.7.

Identification of content to recommend to the user under a specific context is then performed once V_c and V_M have been constructed.

$$V_M = [(\text{medium},0.16),\ (\text{english},1.02),\ (\text{old},0),\ (\text{recent},0.2),\ (\text{action},0.24),\ (\text{epic},0),\ (\text{sci-fi},0),\ (\text{movies},1.62)]$$

Fig. 7 - Final hypothetical candidate media vector

The traditional cosine distance correlation, providing a measure of similarity between the vectors, obtained as expressed in the equation (1), is used in the classification process. A recommendation list is built by ordering the candidate media items in descending order of magnitude of the computed similarity values.

IV. USAGE SCENARIO AND MODE OF OPERATION

A example scenario has been developed to illustrate the type of contextual information used by the system, as well as the sequence of operations performed to arrive at a list of recommendations.

Waiting at the metro station on her way home from the University on Friday at 8:30 PM, Susan enjoys watching video clips of recently released movies on her PDA. She relies on the system to provide her with some recommendations. While accessing one of the recommended video clips, the system realizes that her device battery will soon run out. The system then chooses to deliver, not the full video clips, but only a summary with key frames and synopsis.

This scenario allows identifying the contextual information necessary to provide the system with. Specifically: user identification (who: "userID#"); basic timing data (when: "Friday at 8:30PM"), allowing to derive more information related with time (evening, weekday, weekend); location information (where: Latitude: 41°10'40.30"N, Longitude: 8°35'54.29"W), which contributes to infer more information related with location (metro station, outside/inside, public space); type of terminal device (terminal: "mobile device"). The system incorporates knowledge-based mechanisms (ontologies, inference rules and reasoning tools) to establish semantic relationships between these low-level context data to derive additional knowledge, namely a high-level description of the context of use, such as the ones listed in table 1. This process has been addressed in [13]. In the current paper, the high-level context is inferred using our existing context recognition framework [12].

The high-level description of the context of use is labeled by the system as *context i*. To generate recommendations for Susan, the system performs the set of operations described as follows.

A. Context Capturing phase

When Susan starts to consume content, or when her context changes, the system acquires high-level context data from Susan's mobile phone, in the form of MPEG-21 UED files. Context values extracted from these UEDs are instantiated into a context ontology model; high-level context is inferred, as described in [13], and sent to the recommendation engine. An example of such inference is to infer from Susan's WI-FI, the knowledge that she is at home; or deducing that it is morning of a weekday from low-level temporal information.

B. Profile Update

Whenever Susan consumes a given content, that content is identified and described with a given category and features and is associated with the inferred context. The process of updating the data in the user profile is conducted as described in section III.D of this article.

C. Filtering/Recommendation

High-level context information from A) is used to find a match between currently inferred high-level context and contexts that have already been inserted into the user profile.

1) A match is found

If the system finds a match (i.e. If Susan had already consumed something in that context), it generates the corresponding contextualized user profile vector and retrieves Susan's preferences in that context. The system then initiates a search on the Web using the triplet "*category-genre-property*" with the largest weight as keywords to find candidate content; upon receiving the results, it builds the media profile vector for each result, adopting the process as described in section III.E, computing its similarity with the contextualized user vector.

2) A match is not found

If the system does not find a match (i.e., if Susan has never consumed anything in that particular context), it ignores the context information, and builds a complete user profile vector. It then searches the Web as in step C-1).

3) Recommendations obtained

Susan then receives a filtered and personalized list of recommended items having similarity values above a threshold or generating the top *n items* [17], from which she selects one or more specific item. The selection prompts the update process

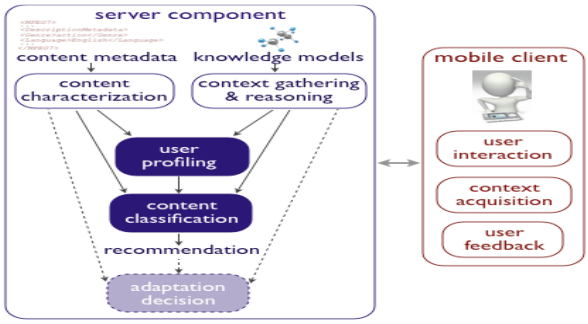

Fig. 8 – System high-level architecture

Fig.11 Recommendation Interface

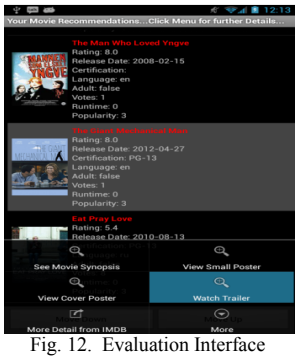

Fig. 12. Evaluation Interface

,by sending information about the selected item to the server.

D. Adaptation Decision

The indication of the selected item is optionally forwarded to the adaptation decision engine to check if its format satisfies the constraints imposed by the current usage context. If not, the item is subject to adaptation. In the scenario, the system discovers the low battery problem. One possible adaptation to perform is a summarization of the video clip. For example, the context information needed by the adaptation engine is given in table 2. The information related to the content's visual data is given in table 3. From these tables, it is evident that some form of adaptation is required to provide acceptable quality that satisfies those constraints in the table 4 for successful playing of the movie clip.

V. Initial Implementation and Experiments

A. Implementation

We have developed a client-server prototype as illustrated in Fig.8. The client side was developed for the Android platform running on Samsung Galaxy S I9000 smartphone. It takes advantage of a number of sensors made available by the Android platform, collecting contextual data whenever a user interacts with the application and passing these data to the server. The server registers user actions and performs all the necessary processing for user profiling, update, and recommendation generation. The recommendation engine and the user profile management have been implemented, utilizing multiple technologies such as Java EE (JPA, EJB, etc.), as well as RESTful Web Services, which are deployed on the Glassfish server [18]. A MySQL database acts as a persistent repository, hosting the user profiles. Figs. 9 and 10 present some screenshots of the mobile client, showing the user profile and the recognized user context. Fig.11 and Fig.12 shows the recommendation visualization of the context-aware mobile recommendation application.

B. The Experimental Data

To test the feasibility of our system, preliminary experiments have been carried out using two sets of data. First, the candidate dataset was obtained by crawling over 4500 movie metadata records from the Movie Database (themoviedb.org), further enhanced with additional metadata retrieved from the IMDB. This metadata set contains 23 separate movie genres, and each record contains on the average, 3 different genre labels. Genres are characterized by language, cast, country, duration, and release date. These terms, thus constitute the media item's context in our user profile model (as illustrated in Fig.2). Second, we solicited user data from an online survey we conducted to create over 135 different user profiles, each having 19 separate entries in the genre level (the entry in the category level was the same for all users – movies). High-level contexts, such as the ones presented in Table 1, were associated with these entries.

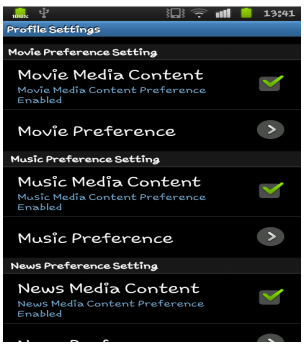

Fig. 9 - User profile settings

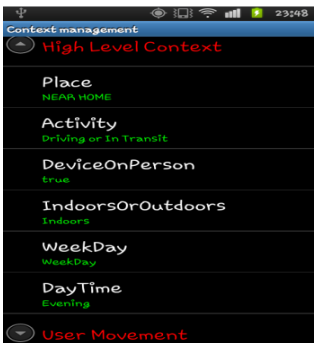

Fig. 10 - Context browser

C. Experimental Setup

To assess the quality of recommendations, given that many of the users were anonymous and thus were not available to provide on-device feedback in the initial experimentation, we devised a technique to allow marking of recommended items either as *relevant* or as *irrelevant.* This allows us to simulate the acceptance or rejection of those items recommended by the user as shown in Fig. 12. In this simulated approach, an item is marked by the system as relevant if 2/3 of the terms that appear in its metadata record (but not less than 2 terms), also appear in the user profile with a weight larger than the average weight of all terms in the user profile. Otherwise, it is marked as irrelevant. This approach was adopted because it was observed that the classification of candidate items was influenced by the number of terms contained in their metadata records.

D. Experiments

We defined three scenarios to experiment to evaluate the quality of the recommendation system.

1) The first experiment involves generating user profiles with specific bias for certain content in the category-genre-property-context. In this case, users want to consume content at a specific contextual situation. In practice, this translates into using larger variance for generating values of the weights assigned to genres and properties. Some are assigned higher weights and others are assigned smaller weights or even zero, depending on the user's preferences for such genres in specific contextual situations. This scenario we call biased contextual user profiling.

2) The second experiment is similar to the first. However, unlike in the first experiment, the situation in this experiment is opposite. In other words, this is a situation where contexts do not play essential role in a user's choice of content. The system makes recommendation without considering contextual information.

3) In the third experiment, we generated the user profiles with no specific bias for any specific category-genre-property-

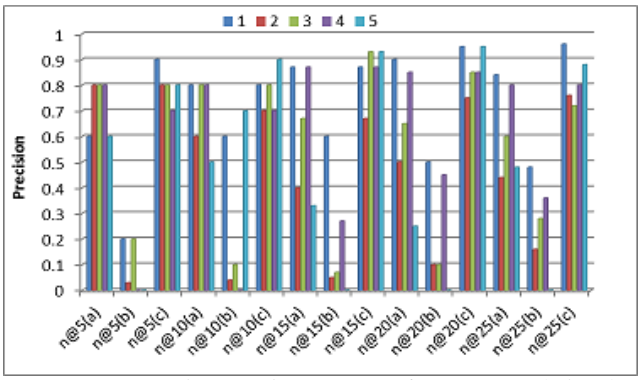

Fig.13. *Recommendation Quality Comparison for Experiments (a, b & c)*

contexts but with a user's preferences having a uniform weight for preferred genres in associated contexts.

The experiments above have been performed to classify the candidate content as *described in section III (E)*, using their corresponding metadata. The recommended items were obtained from an ordered list, using a combination of top *n* neighbors and threshold approaches [17]. The former selects *n* items with the highest similarity. The latter imposes a minimum value of similarity to those *n* items to remain included in the recommendation list. Precision, a traditional information retrieval evaluation metric was used to determine the performance of the system in each scenario.

The precision is calculated as the ratio of the number of recommended items marked as relevant to the total number of recommended items (*n*). Fig.13 shows the precision values obtained for 5 test user profiles (1,2,3,4,5, representing each user), for the three scenarios that were evaluated. The size of the recommendation list *n* is set at 5,10,15,20, and 25 (@5,@10,@15,@20,@25 respectively).

The results show that contextual information can improve recommendation quality. In scenarios a and c for example, where contexts were considered, there is relatively high precision compared with the scenario b. This result promises the importance and significance of contexts in media content delivery and consumption.

4) User satisfaction test: We have also conducted a preliminary user satisfaction evaluation of the proposed system. In the experiment, we asked 5 individuals whose profiles and context information have been used in the system to evaluate the system on the basis of how satisfied they are with the recommendations that the system generated for them. Using the online-based movie metadata that we have crawled from the Web, we generated recommendations for the users and then asked them to evaluate their degree of satisfaction of the recommendations. They were able to give satisfaction scores ranging between 1 and 5. Where 5 is the highest satisfaction and 1 being the lowest satisfaction. In the final analysis, those scores between 3 and 5 were considered satisfactory whereas, 1 and 2 were considered unsatisfactory.

VI. DISCUSSION

The conducted experimentations show promising results. Data from the user survey were used to generate user profiles, with some users having well-defined preferences for certain types of content (contextual preferences). However, as described in the last section, three different situations have been evaluated, imposing different contextual constraints on the system. The results we got in the experiment *(b)* show a clear example of where recommendations without contextualization would not give good quality. This happened because the system considered irrelevant user preferences that were not related to the user's current situation. Experiments *a* and *c* show how this type of situation can be avoided, significantly enhancing recommendations, eliminating preferences not relevant in the current contextual situations. It consistently provides better performances as depicted in Fig. 13.

The preliminary user satisfaction test shows that test users expressed overall satisfaction of 65% for all the recommendations provided to them by the system.

In the future, further tests will be carried out, especially the usability test of the Android application because some of the users were not impressed the recommendations. In addition, further performance evaluation of the system still needs to be carried out to compare it with other traditional recommendation approaches such as collaborative recommendations.

Finally, integration of the adaptation and the recommendation engines to be performed.

ACKNOWLEDGMENT

The work presented in this paper was partly supported by: Portuguese Foundation for Science and Technology within project FCT/UTA-Est/MAI/0010/2009; the North Portugal Regional Operational Progra (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF)

REFERENCES

[1] A. Vetro,"MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access," IEEE Multimedia Vol.11, No1, pp. 84- 87, (2004).

[2] S. Wong and V. Raghavan, "Vector Space Model of Information retrieval – a reevaluation," In Proc. of SIGIR, pages 167–185,Cambridge, England, (1984).

[3] C. Timmerer," MPEG-21 Digital Item Adaptation," FDAM/2, Dynamic and Distributed Adaptation, 78th ISO/IEC JTC 1/SC 29/WG 11 (MPEG) Meeting, Hangzhou, China, (2006).

[4] D.Jannach, k. Leopold, C.Timmerer, H. Hellwagner,"A Knowledge-Based Framework for Multimedia Adaptation" Journal of Applied Intelligence, Vol.24 , No.2, pp.109-125, (2006).

[5] G. Adomavicius, A. Tuzhilin, "Towards the next Generation of Recommender Systems: A survey of the State-of-the-art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, Vol 17, No 6, pp 734-749, (2005).

[6] G.Adomavicius , A.Tuzhilin , "Context-aware recommender systems". In: Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, Berlin, pp 217-256(2011).

[7] K. Xu , M. Zhu , D. Zhang, T. Gu, "Context-Aware Content Filtering and Presentation for Pervasive and Mobile Information Systems," In Proceedings of the 1st international Conference on Ambient Media and Systems, pp.1-8, Quebec, Canada (2008).

[8] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems using a Multidimensional Approach," ACM Transaction in Information System., Vol. 23 No.1, pp.103-145, (2005).

[9] A. Chen, "Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment."Lecture Notes in Computer Science, Vol 3479, pp. 244-253, (2005).

[10] A. Costa, R.Guizzardi, J. Filho, "COReS: Context-aware, Ontology-based Recommender system for Service recommendation," In Proceedings of the 19th international conference on advanced information systems engineering (CAiSE'07), Trondheim, Norway, pp 11-15, (2007).

[11] Z. Yu, X. Zhou, D. Zhang,. C-Y Chin, X. Wang, J. Men, "Supporting Context-Aware Media Recommendations for Smart Phones," IEEE Pervasive Computing, Vol.5, No.3, pp.68-75, (2006).

[12] A.M Otebolaku and M.T.Andrade., Recognizing High-Level Contexts from Smartphone Built-in Sensors for Mobile Media Content Recommendation, in Proc of the 14th IEEE International Conference on Mobile Data Management, Milan Italy (2013).

[13] A. M. Otebolaku, and M. T. Andrade, "Context Representation for Context Aware Mobile Multimedia Recommendation," In Proceedings of the 15th IASTED International Conference on Internet and Multimedia Systems and Applications, Washington, USA, (2011).

[14] A. B. Benitez, D. Zhong, S.F. Chang, J.R. Smith, " MPEG-7 MDS Content Description Tools and Applications" W. Skarbek (Ed.): Computer Analysis of Images and Patterns, Lecture Notes in Computer Science 2124, Springer, (2001).

[15] J. Wang, Z. Li, J. Yao, Z. Sun, M. Li, W. Ma, " Adaptive User Profile Model and Collaborative Filtering for Personalized News," In Proceedings of APWeb, pp.474-485, (2006) .

[16] M. Pazzani, D. Billsus, "Content-based Recommendation Systems" In: The Adaptive Web, Vol.4321, pp.325-341, (2007)

[17] M. Deshpande, G. Karypis, "Item-based Top-N Recommendation Algorithms" ACM Transactions on Information Systems. Vol.22, Issue 1, pp. 143 - 177, (2004).

[18] http://www.oracle.com/technetwork/java/javaee/tech/index.html[last accessed on September 2013].