A Context-Aware Framework for Media Recommendation on Smartphones

Abayomi M. Otebolaku and Maria T. Andrade

Abstract. The incredible appeals of smartphones and the unprecedented progress in the development of mobile and wireless networks in recent years have enabled ubiquitous availability of myriad media contents. Consequently, it has become problematic for mobile users to find relevant media items. However, context awareness has been proposed as a means to help mobile users find relevant media items anywhere and at any time. The contribution of this paper is the presentation of a context-aware media recommendation framework for smart devices (CAMR). CAMR supports the integration of context sensing, recognition, and inference, using classification algorithms, an ontology-based context model and user preferences to provide contextually relevant media items to smart device users. This paper describes CAMR and its components, and demonstrates its use to develop a context-aware mobile movie recommendation on Android smart devices. Experimental evaluations of the framework, via an experimental context-aware mobile recommendation application, confirm that the framework is effective, and that its power consumption is within acceptable range.

1 Introduction

The advancements in mobile computing technologies, wireless and mobile networks, and the proliferation of mobile devices such as smartphones have brought remarkable changes in the way we access online media items. Using smartphones via the Internet to access media content has become easier and ubiquitous. Moreover, with mobile phones now equipped with sophisticated video cameras and multimedia functionalities, user generated media content has become pervasive. Because of this development, much more media content is published every moment online. Consequently, mobile users can obtain myriad choices of

---

Abayomi M. Otebolaku · Maria T. Andrade
Telecommunications and Multimedia Unit, INESC TEC, Portugal
e-mail: abayomi.otebolaku@inescporto.pt

Maria T. Andrade
Faculty of Engineering, University of Porto, Portugal
e-mail: mandrade@fe.up.pt

L. De Strycker (ed.), *ECUMICT 2014*,
Lecture Notes in Electrical Engineering 302,
DOI: 10.1007/978-3-319-05440-7_8,

media items to consume. Nevertheless, this development comes with negative effects, owing to the exponential explosion of online-based media contents. Because of the huge volume of online-based media content, mobile users now waste valuable time hoping to find relevant ones to consume. Often, because mobile user's preferences are subject to contextual situations, they end up with irrelevant media items, which do not match their preferences [2-5, 7-8, 10-11].

Therefore, to assist mobile users make informed choices about relevant content, many works have been developed [1], [15]. The highly successful traditional approaches, based on personalized recommendations assist users to find relevant items by using evaluations of the previously consumed contents given by the target user (content based) or the evaluations given by users who are similar to him (collaborative based). These traditional methods fall short of getting relevant media items to users because they assume that users always give an evaluation of content they consume, which in practice rarely happens. Additionally, they do not consider contextual information as an important factor that affects user's preferences [2, 18].

Recently, however, context information has become integral part of the recommendation process. For example, context information is used to generate music recommendations for mobile users in [11]. In [12], context information is explored to recommend interesting movies to mobile users. However, these systems use context explicitly to recommend media content to users, thereby requiring user's constant interventions. Users would like the system to implicitly suggest content that match their preferences, without their interventions.

To deliver this rich personalized media experience to mobile users, and to address the weaknesses of the traditional approaches, especially collaborative and content based approaches [21], it is important to understand the relationship between a user's preference, context information, and media services. This relationship influences the media content consumption choices of mobile users.

As illustrated in Figure 1, the contribution of this paper is the provision of a generic framework with features that can automatically suggest content to users, based on their dynamic contexts, inferred from their smartphone-embedded sensors. The main functionality of the framework is the capturing and the use of context information, especially the mobile user activities, which it relates to the user preferences to select suitable content among available online media, personalizing it for specific users and their contexts. Furthermore, we have implemented the framework's components as RESTful web services [22], making it accessible to any platform, and we have evaluated its functionality and feasibility.

In the next section, we analyze briefly other research efforts for media item recommendations, and then describe the problems that the framework addresses in section 3. Section 4 presents the framework's conceptual design, and section 5 describes its implementation and some evaluations. Section 6 discusses the evaluation results. Section 7 concludes the paper and gives our future direction.

2 Related Work

Most popular work in context aware recommendation such as Pessemier et. al. [3], Chen [5], and Yu et.al. [10], focus on explicitly on using context information to suggest media items. Cinemappy [12] uses the location and movie consumption history of mobile users to recommend relevant movies. Our work integrates a context model that accommodates broader contextual information than just location information. In addition to user location, our work integrates the user as well as user environment information. Whereas, [11] uses user activities such as walking, running, studying, etc. only to recommend music to mobile users, its weakness is that it considers only user activity and not location information or other context information that influence user preferences.

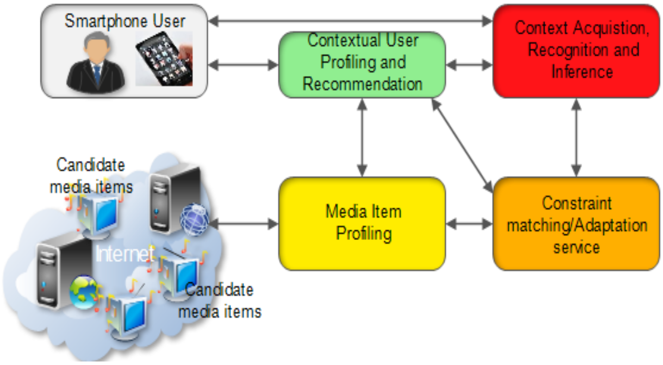

Fig. 1 Proposed framework's functional view

Our approach is different because the framework proposes a hybrid and generic system that provides both explicit and implicit media recommendations to mobile users. The context recognition process is exposed as a web service to be consumed by any application built on top of the framework. The contexts can be stored as historical data, which can be related to the present user contexts and the user preferences. Similarly, the framework exposes its recommendation processes, and contextual user profiles as web services, providing easy access from any mobile software and hardware platforms. It also includes an optional adaptation service that can tailor the presentations of the recommended contents to the device and network constraints.

3 Problem Definition and System Requirements

Enabling proactive contextual suggestion and delivery of relevant rich media items that satisfy the mobile user's preferences, in heterogeneous environments requires careful design considerations, because providing media items to users in such an environment is a challenging problem. This requires addressing several key issues, which are dynamic context and activity recognition, representation, contextual user profiling, preference management, an adaptation of the media item presentation to device constraints and network prevailing conditions, etc. Besides, developing a framework that can incorporate these aspects in a unifying framework has never been trivial. In this section, we examine the difficulties that characterize the design of the proposed framework.

3.1 Context Recognition and Representation

An item is relevant if it belongs to the same context as the user's active interaction [14]. Therefore, wrong context will lead to wrong recommendation. To explore contextual information for personalized content delivery, it is important to identify accurately, the user's contextual situations, which require context sensing capabilities. However, sensed data are generally vague and imprecise, because they contain noise, and in raw form, they do not make sense [18]. Additionally, several sensory data need to fuse to recognize more meaningful high-level contexts. These conditions necessitate a context recognition model, utilizing machine-learning approach that takes the low-level sensory signals as inputs and produces as output, accurate and meaningful high-level context information. Furthermore, recognized high-level context information must relate to other high-level context information to infer more semantically meaningful contextual situation of the user. The dynamic realization of this requirement remains a challenge in the overall process. Section 4.1 discusses how this issue is addressed by CAMR.

3.2 User and Media Content Profiling

A key requirement to offer truly contextual delivery of media content to mobile user is the user profile. The user profile should encode all desirable media content features, customized for the profile owner and in context. The user profile includes, among others, preferences of the user for 1) desired media presentation characteristics, (2) optional user identities such as names, gender and profession, 3) preferred location information, 4) usage history and, 5) other high-level context information such as user activities in relation to the user's preferences. Dynamic acquisition of user preference information, user consumption history, and their incorporation with user's dynamic context and activities to update the user's changing preferences and estimate their relevance and importance remains a challenge. Section 4.2 discusses the user profiling issue in CAMR.

3.3 Content Classification and Recommendation

The Web today contains a massive amount of digital contents that users explore using their mobile devices. The sheer size and the decentralization of these digital content make it difficult for mobile users to obtain those contents that suit their preferences and situations. Recommendation algorithms provide the advantage of guiding users in a personalized way to interesting content in a large space of possible options [1]. However, the recommended content must be provided according to the user's changing contexts, activities, and preferences. Therefore, content classification and recommendation need to be equipped with contextual information to assist mobile users to get rich media experience while on the move. Section 4.3 presents the classification and recommendation process in CAMR.

4 Context Aware Framework for Media Recommendation

In this section, we present CAMR as a conceptual framework for generating contextually relevant media content to mobile users. CAMR addresses the problems discussed in the last section. Figure 2 sketches the architecture of the framework, showing its major components. These components are described in this section.

4.1 Context Recognition

The ability of a system to identify contextual situations and respond to them is one of the most important functions of any context-aware system. However, context sensors emit data that are in low-level form, which are not suitable for mobile application. Context recognition is a process that collects raw data from sensors and transforms them to information that can be used by applications [18]. To provide dynamic contextual information about media items consumers, CAMR uses context recognition process to identify contextual information such as user activities, user location, and environment situation such as weather, illumination, and noise level from low-level data collected from smartphone-embedded sensors. To realize this, the context recognition service leverages four important processes. The next section presents these processes.

Sensor Data Collection and Preprocessing

CAMR gathers events from smartphone embedded sensors such as accelerometer, GPS, gyroscope, rotation vector, orientation, proximity, microphone and light sensors. It collects 128 samples of data from each axis of accelerometer, rotation vector, and orientation sensors in a continuous 3 seconds, with 64 samples from the previous 3 seconds overlapping the next 3 seconds. In the data preprocessing phase, we removed event outliers [9, 13]. This is done by removing samples from the beginning and the ending of each example to reduce the influence of noise in the data.

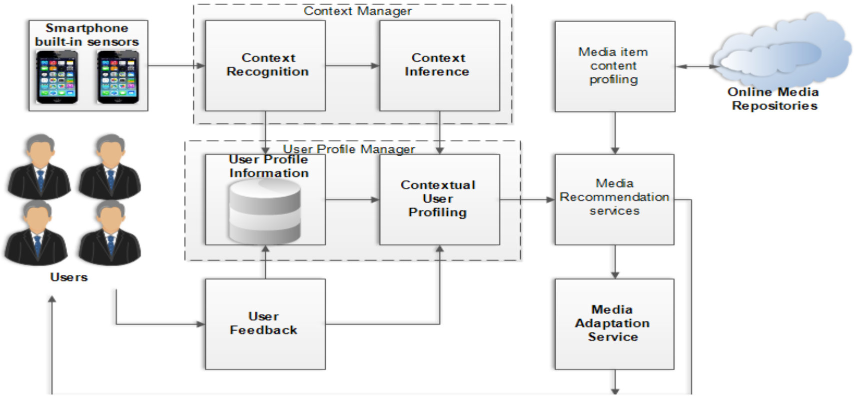

Fig. 2 CAMR architecture

Feature Extraction

User's dynamic contexts, such as activities, are performed in relatively lengthy period, in seconds or minutes, when compared with the sampling rate of the sensors. The sampling rates usually do not provide sufficient data to describe user activities. Therefore, activities are usually identified in the time window basis rather than sampling rate basis [13]. Comparing a series of time windows to identify activities is almost impracticable even if the signals being compared come from the same user performing the same activity context [18]. The feature extraction process addresses this problem by filtering relevant and obtaining quantitative data from each time window. Various approaches have been explored in literatures such as statistical properties and structural properties of the sensor signals. Structural features such as Fourier transform are quite complex and require more computational resources. This may not be ideal for resource starved devices. On the other hand, statistical features are simple and require less computational resources [9]. Thus, CAMR uses simple labeled statistical features [range, maximum, minimum, mean and standard deviation], which have been validated in our previous work to be very effective, to discriminate between time windows [9]. These features are extracted into feature vectors that are then used in the next process.

Context Classification

After extracting the time window features from the raw sensor signals, without deriving the context knowledge from them, the example features are meaningless. CAMR uses classification algorithms, in particular, Support Vector Machine, Neural Network (NN), Decision Trees, Nearest Neighbors (KNN), and BayesNet to derive high-level context from the statistical features. Details of the modeling and evaluations can be found in [9, 13, 18]. The present implementation integrates KNN in the recognition service to accurately obtain independent future activities and contexts of the mobile users. We implemented KNN based context recognition model because evaluations of various models in our previous report [9] confirmed its excellent performance in terms of accuracy and recognition time.

Context Inference

The contextual information obtained from the preceding process is usually semantically not adequate for recommendation process. For example, it is important to know what a user is doing, when, and where he is doing it. Nevertheless, without relating this information to provide situational context, this information will convey no useful sense. This is one of the weaknesses of the existing work. CAMR uses knowledge-based model on top of the classification process to relate different atomic context information to obtain contextual information at a higher semantic level. For example, having known that a user sitting at home is located in the living room, if we know that the TV is switched on (can be obtained from IR Blaster), and then one can relate the two information and conclude that the user is watching TV. The user is watching TV, obtained from sitting, home, living room, TV, etc. is inferred using the ontology based knowledge inference process of CAMR. The details of this knowledge based process have been presented in [6], and will not be discussed further in this article. CAMR can also determine such complex context, such as a user is "*jogging in the sport arena*". This situational information is crucial to offer a rich media experience to mobile users.

4.2 Contextual User Profiling Service

A user profile describes his preferences, normally based on the history of the user's actions [1]. CAMR's contextual user profile service (CUPS) summarizes the user's content consumptions into a limited set of categories. Categories are characterized by one or more genre, and a number of properties characterize the genre. Several genres can be associated to one category. Several properties can be associated with one genre. Additionally, it incorporates the contextual dimension, associating one or more inferred context to each *category-genre-property* concept. It presents each user profile as a four level tree, as shown in Figure 3 (a), with the root of the tree representing the user's optional 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 detail, the consumed content and thus the preferences of the user. A limited set of properties is used for each genre to obtain a good compromise between sufficient degree of characterization of content (hence, sufficient ability to make distinctions) and reasonable dimensions of the user profile. The leaf nodes provide information about the contexts where the user preferences have been observed. Leaf nodes have three fields – *type*, *intensity,* and *lifetime* – whereas all other nodes have only the type field. In the leaves, the types represent the type of context. The newly introduced concepts in the user profile, the intensity and lifetime track user's contextual consumption history.

Using these weighted parameters, the system is able to determine at runtime, the media contents that are important to the user, based on his contextual preferences. The *intensity* provides information on the number of times the user has consumed items of that *category-genre-property* in that specific context. The intensity (the dynamic preference of the user) of those elements belonging to the media's term is obtained by summing up the products [weight x lifetime] of all their occurrences. The intensity value of the retained elements at that level is obtained by visiting their child nodes in a breadth-first traversal. The same applies to the retained elements of the category level. The intensity of the elements belonging to the genre level is the largest value of their children. Accordingly, these values are obtained by performing a depth-first traversal. This way, the user profile can handle any category of media items such as movie, news, music, etc.

To classify the candidate media items, and to obtain a list of recommended items, a vector is created from the user profile. A global version of the user vector contains as many elements as the number of different *category-genre-properties* that appear in the complete user profile tree.

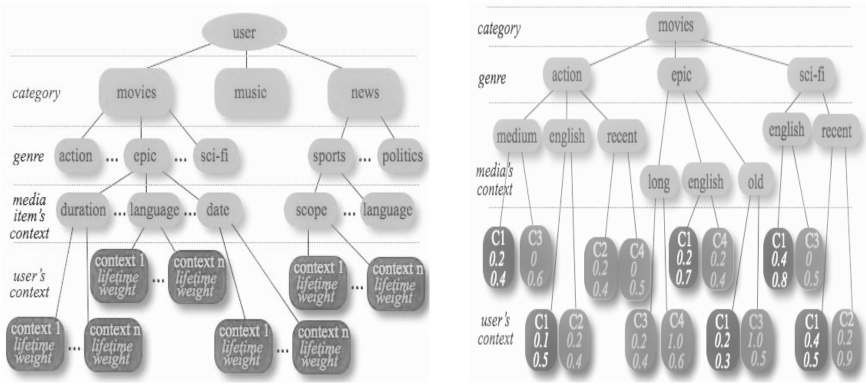

Fig. 3(a&b) Profile Structure for a hypothetical user

A *contextualized user vector* typically has a much smaller number of elements, corresponding to the different category-genre-properties associated with the specific context under consideration. The *contextualized user vector,* V_c is, thus, built using context data to filter the profile. The value of the current context of usage is compared to the leaves of the profile tree (context nodes) to identify the upper nodes that provide values for the elements of V_c. Only nodes whose leaves match the current context are used. 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 node. As an example, consider the hypothetical user profile of Mr. X represented in Figure 3 (b). 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 the ones that have leaves with context value C_1. The intensity of those elements belonging to 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 applies for the retained elements of the category level. The intensity of the elements belonging to the genre level is the largest value of their children. Accordingly, these values are obtained, using a depth-first traversal.

4.3 Contextual User Profile Information Store

The contextual user profile information store (CUPIS) is the persistence store where user profile data such as lifetime and intensity of each user preference, contexts, etc. are kept. Consequently, interactions of the user with the systems, including his feedback are likewise stored for subsequent recommendation processes. User feedback manager (UFM) and CUPS rely on CUPIS for persistence.

4.4 User Feedback Manager

It is important to track user's consumption behavior to improve the system's classification accuracy. The user profile can be updated using two approaches, explicit and implicit methods [1]. The former grants the users the ability to modify the values assigned to their preferences by the system. The implicit approach involves preference learning without direct user intervention, such as updating the profile when the user has spent a certain amount of time on a given item. The CAMR's user feedback manager (UFM) updates the user profile whenever the user interacts with the system; it tracks both consumption and non-consumption of content by the user to learn the contextual preferences of the user for any kind of media item. To implicitly learn the user preferences, UFM runs an intermittent background service monitoring the interactions of the users. Additionally, it explicitly updates the profile whenever a user consumes any content item by obtaining a feedback from such user. Equations *(1&2)* are used for the implicit user profile update model, allowing associating importance to the finer details of content and corresponding contexts of consumption. The model associates weight $w_i m_i \in [0,1]$ that relates the relevance of a consumed media content to the context in which it is consumed, learning the user preference for the content he consumes or those he does not consume. A weight w_{ij} represents the relevance of content m_j belonging to the media content category k_i that a user u_i consumes in context c_i. Whenever the recognition model detects that the user is in a context c_i, for a continuous period of time [0, T] and the user consumes one or more of content m_1, $m_2, \ldots m_n$ recommended by the recommendation service, then weight $w_i m_i \in [0,1]$ is associated with this content, which at time T is updated as follows:

$$w_i m_i = w_i m_i + \gamma(\alpha - w_i m_i)\ i = 1,2,3,\ldots,n. \quad (1)$$

Then for those content b_1, b_2, $b_3,\ldots,b_n$ with associated weights w_1b_1,w_2b_2, $w_3b_3,\ldots,w_nb_n$ not consumed by the user, these weights are updated as follows:

$$w_i m_i = w_i m_i + \gamma(\alpha - w_i m_i)\ i = 1,2,3,\ldots,n\ . \quad (2)$$

γ is a learning parameter whose value is obtained by: $\alpha \in [0,1]$ its value is set to 1 in (4) and 0 in (5)

Factor t in equation (3) represents the number of days elapsed since the last time the user has consumed an item with the characteristics described by his profile nodes. With equation (3), the parameter γ for each 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, its value is linearly increased daily. This way, nodes that represent items that have not been seen for a long period, will eventually have no impact on the user preferences evaluation.

$$\gamma = 1 - \left(\frac{t}{45}\right)^5 \quad (3)$$

4.5 Media Item Content Profile Service

The media item content profile service (MICPS) is responsible for crawling the Internet for candidate media items. It retrieves and processes the media item metadata into a form that can be processed by the recommendation service. Usually, it filters the metadata and scores the terms in the metadata to generate a media item profile vector, which is fed into the recommendation services. It is thus necessary to create a media vector, V_M, for each media item. To describe the media items, MICPS relies on the availability of semantic metadata using the MPEG-7 MDS semantic tools [20]. Given that in practice most of the media resources available online lack this metadata, our system incorporates alternative methods to obtain the semantic descriptions of the candidate content. One of such alternatives is the Internet Movie Data Base (IMDB) service API, Last.fm API etc. For each media item, a vector V_M is initially created as an exact replica of V_C. Then, for every 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 a V_M. Otherwise, it assigns zero to the element.

4.6 Media Item Recommendation Service

The recommendation service explores three recommendation algorithms for context-aware media recommendations. The content base (CBF), the collaborative based (CF) and a hybrid based recommendation algorithms. In this article, we will only elaborate on the hybrid approach, which is based on context-aware content-based collaborative process. The traditional collaborative recommendation generates predictions for the target user based on the item previously rated or viewed by other users, and the content-based approach, which uses the consumption history of the user. These two approaches suffer from the so-called overspecialization and new user/item problem respectively, which excludes casual users or those whom the system does not have enough information to generate recommendations [1,16], or always suggesting the items similar to those consumed in the past by the user. Hybrid recommenders combine one or more of the conventional recommendation processes to overcome their individual weaknesses to gain better performance [1]. To take advantage of the hybrid technique, we built context-aware content based collaborative recommendation (CACBR), a hybrid recommendation that combines the context-aware CF and CBF. Basically, it uses the contexts in which other users have consumed the content previously to find users that are similar to the target user by comparing the active user context history and the target user's present context. This way, CACBR can address the new user/item problem of collaborative process and the overspecialization problem of the content based process. This is achieved in three phases. In the first phase, it identifies every user (neighbor) that is similar to the target user by searching through each user's profile tree, looking for context that matches the target user's recognized context. For every user profile with a match, the intensity value p_{ni} of the *category-genre-property* nodes in the user profiles is retrieved into a vector. The vector is then used to calculate the similarities, using equation (5), between all users. After this calculation, it then selects the *top n* most similar users, called neighbors or friends of the target user who have consumed content in the same or similar contexts to the target user's current context. In the second phase, it ranks the candidate content for each neighbor by obtaining vectors V_c and V_m corresponding to contextual user profile of every neighbor and candidate media profile vectors respectively.

$$Sim\,(v_c, v_m) = \frac{V_c . V_m}{V_c \times V_m} = \frac{\sum_{i=1}^{n} V_{c_i} \times V_{m_i}}{\sqrt{\sum_{i=1}^{n} (V_{c_i})^2} \times \sqrt{\sum_{i=1}^{n} (V_{m_i})^2}} \tag{4}$$

By applying the cosine formula (4), it calculates the distance between the contextual user profile vector (CUPV) V_c and the media item profile vector (MIFV) V_m.

$$Sim(P_u, P_n) = \frac{\sum_{i \subset CP_{u,n}} (p_{u_i} - \overline{p}_u)(P_{ni} - \overline{p}_n)}{\sqrt{\sum_{i \subset CP_{u,n}} (p_{u_i} - \overline{p}_u)^2} \times \sqrt{\sum_{i \subset CP_{u,n}} (p_{n_i} - \overline{p}_u)^2}} \tag{5}$$

In the third phase, it generates the preference prediction value for each of the content from (5) for the target user, using Resnick [17] prediction formula (6). In this formula, C_i is the intensity to be predicted for each candidate content *i* generated from formula (5) for the target user and p_{ni} is the intensity by the

1) *Sense and classify target user present context*
2) *Search target user's profile to find if current context matches any context in his profile*
   a. *If it matches:*
      i. *Generate his contextual profile vector*
      ii. *Generate contextual profile vector for every other user in the profile repository*
      iii. *Generate all users similar to the target user(using the above contextual profile vector and Pearson correlation) based on equation(5)*
      iv. *Retrieve media metadata from the Internet*
      v. *Generate media content profile vectors*
      vi. *For every similar user in(iii),generate its similarity value with every media content obtained using (4)*
      vii. *Rank the candidate contents for each similar user*
      viii. *Generate a preference prediction for target user for every media item using modified Resnick formula(6)*
      ix. *Rank the TopN candidate content from (vii) for the target User*
   b. *If no match is found:*
      i. *Generate non-contextual user profile vector*
      ii. *Generate non-contextual user profile vector for every user in the profile repository*
      iii. *Generate all friends of the active user(using the non-contextual user profile vectors)*
      iv *Repeat a(iv-x) above*
   c. *Present suggested items to target user*
   d. *record feedback(implicit/explicit)*
   e. *Log the present context and update his user profile*

Fig. 4 Context aware content based collaborative process

$$C_i = \overline{C} + \frac{\sum_{i \subset neighbors(u)} (p_{n_i} - \overline{p}_n) Sim(c, p)}{\sum_{i \subset neighbors(u)} |sim(c, p)|} \tag{6}$$

$\overline{C}$ is the average intensity of all terms in target user's profile, whereas $\overline{p}_n$ is the average intensity of all terms in neighbor *i* profile. The *sim (c, p)* is the similarity measure between profiles of the target user *c* and neighbor *p*, calculated using formula (4). The modified Resnick prediction discounts the contribution of every neighbor's intensity according to its degree of similarity with the target user so that more neighbors have a large impact on the final intensity predictions [17].

Finally, recommendation list is built by ordering the candidate media items in descending order of magnitude of the computed prediction values. Figure 4 summarizes the entire hybrid content classification (CACBR) process.

5 Framework Implementation and Evaluation

This section presents the implementation of CAMR and its evaluation via a mobile movie recommendation application that was built on top of it. First, we present the framework integration, second its implementation and third, its evaluation.

5.1 Framework Component Integration

We describe, in this section, the integration of different aspects of CAFMR. Figure 5 shows the component model of CAMR. It provides a detailed version of the high-level architecture shown in Figure 2. In this figure, all the components

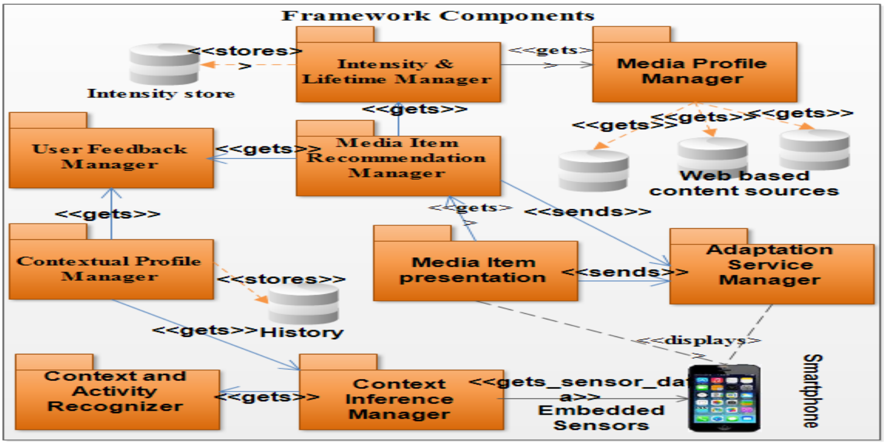

Fig. 5 CAMR Component Model

interact via interfaces. *Media Item Recommendation Manager* is the central component of the framework, establishing direct or indirect connections between the other components. It provides the service that executes the task that generates the recommendations. The *Media Profile Manager* connects the online-based databases such as YouTube, IMDB, etc. to extract media metadata descriptors. The *Context and Activity Recognizer* connects with various sensors on the smartphone to obtain raw sensor data, which it then processes to obtain high-level context information. The *Context Inference Manager* is responsible for relating two or more recognized context information to derive semantically expressive high-level context information. The *Contextual User Profile Manager* collaborates with *Context Inference Manager* and *Context and Activity Recognizer*, taking the context information, user preferences, and establishing a relation between the context information and user preferences.

The *User feedback manager* is responsible for learning the user's interaction with the framework; it collects either explicitly or implicitly, the user's contextual feedback in order to improve the future recommendations, which truly reflects the user's contextual preferences. *The intensity and lifetime manager* is responsible for scoring the contextual preferences of the user, and for tracking those media items that users have consumed before, which are no longer interesting to them. The *Media presentation manager* is responsible for determining the appropriate format of the recommended media items to display by the smartphone. The *adaptation manager* is responsible for determining if the presentation format of the recommended media items can be played by the smartphone. If it cannot be played, it then determines the appropriate adaptation mechanism to be executed to display the media item.

5.2 CAMR Implementation

To demonstrate the feasibility of CAMR, we have implemented its prototype consisting of eight services, representing the operations offered by its components. The context recognition component is implemented as a mobile client service that can be deployed on smartphones. Other components of Figure 5, including the media item recommendation and the contextual user profile managers have been implemented using Java technologies, integrating Java EE 7 (EJB, JPA) and RESTful Web Services and deployed on the Oracle Glassfish server [19]. MySQL 5.6 database [22] serves as the backend database, hosting user profiles and user context history. The contextual user profile information service registers user actions and performs all the necessary processing such as the user profiling, update, whereas the recommendation generation is performed by media recommendation service, at the same time working in tandem with the context service and recommendation client service on the Android device. The client-server implementation architecture is depicted in Figure 6. Figures 7&8 presents some screenshots of the mobile client. Figure 7 (a) shows the user profile management interface, where a user can optionally manage her profile. In the application, three categories of contents (movies, music, and news) are presently supported. Figure 7 (b) is the context browser where contextual information can be managed on the device by the users. It shows the high-level context indicating, for example, that the user is at (*Home)* location. The user's activity is inferred as *Sitting* while she is *indoors,* and time of the day inferred as *night.* This context

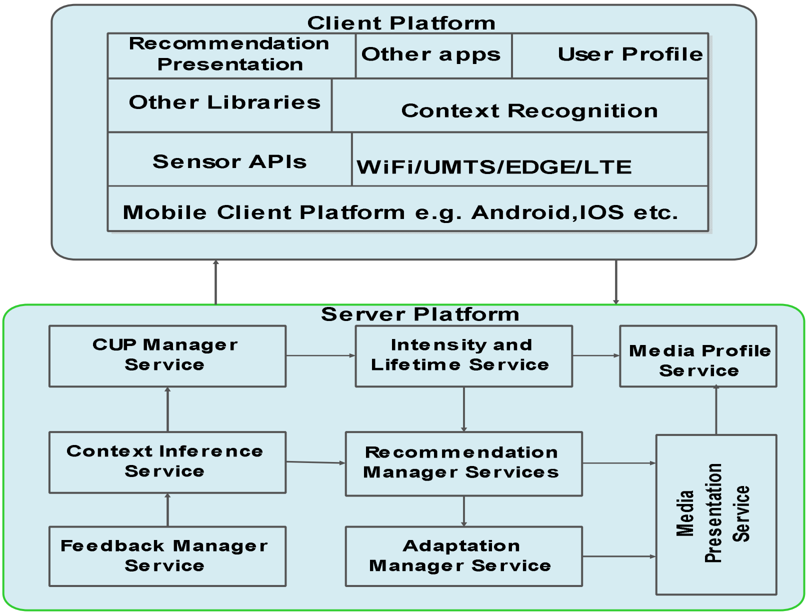

Fig. 6 Implementation Architecture

information is fed into the context knowledge base, which infers higher level situational context, such as "*It is weekend night, user is sitting at home*". This is then fed into the contextual user-profiling model, which is stored in a context history repository. The process for deriving high-level context information is based on context recognition using context classification algorithms described in section 4. The figure also shows that the user is indoors, which has been obtained using the device built-in GPS. Figure 8 (a&b) shows the recommendation interfaces of the context-aware mobile recommendation application, showing the recommended movie content and all options available to user to visualize the recommendation such as playing the movie trailers, or connecting to online sources for additional information on the recommended items.

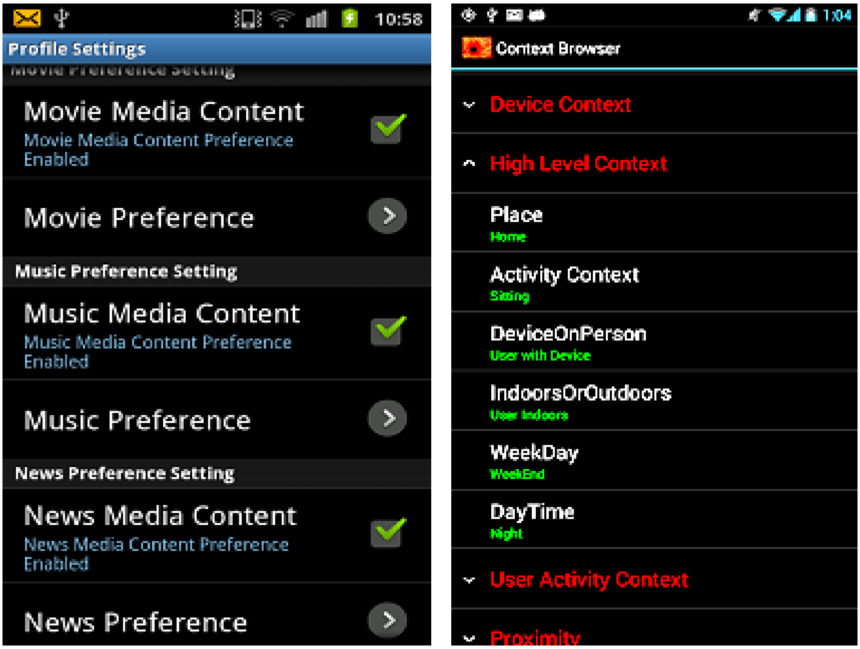

Fig. 7 (a) content preferences. (b) context browser.

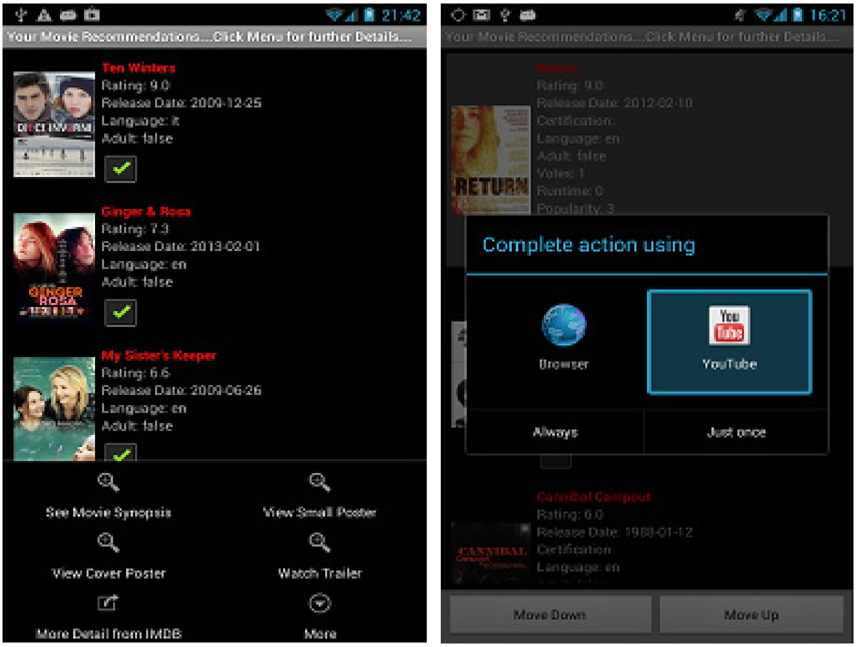

Fig. 8 (a) recommendation list (b) recommendation presentation

5.3 *Experimentation*

To experiment and to evaluate the feasibility of the proposed CAMR, we have implemented the framework, and based on it developed a context-aware movie recommendation shown in Figures 8(a&b). In this section, we present the evaluation of the framework.

The Experimental Data

To evaluate the feasibility of the proposed framework, preliminary experiments have been conducted 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.com. This metadata set contains 23 different movie genres, and each record contains on average three different genre labels. Language, cast, country, duration, and release date characterize the genres. These terms, thus constituted the media item's context in our user profile model (as illustrated in Figure2). Second, we solicited real world user profile data from 200 online users, each having 19 different entries in the genre level (the entry in the category level was the same for all users – movies). High-level contexts such as Location = "*Home*", DaysOfWeek ="*WeekEnd*", TimeOfDay = "*Night*", Activity ="*Sitting*" were associated with these entries.

The Evaluation Metric

We need to define a way to measure the impact of contextual information on the recommendation quality. We do this by comparing the recommendations with or without activity contexts. We define a recommendation quality metric (also known as precision), which is the percentage of the number of times CAMR successfully provides mobile users with preferred media items. A provided recommendation is relevant if the user finds what he wants within the first N provided recommendation list. For example, if the user finds n^{th} media item of a given recommendation list, then the recommendation is successful if and only if $n \leq N$. Having a higher accuracy with lower value of N is low means that the recommendation framework works well and that its contextual user profile accurately presents the user's preferences.

Experiments

This section presents the evaluation of the framework; we evaluate the accuracy of the context recognition service, the quality of the recommendation and the energy consumption of the framework.

A) CAMR recommendation quality

Experimental Procedure: To evaluate the quality of the recommendations, given that most of the 200 users were anonymous and thus were not available to provide continuous on-device feedback, we devised an approach to allow marking recommended items as *relevant* or as *irrelevant*. This allowed us to simulate the acceptance or rejection of suggested content by the users as shown in Figure 8 (a). In the evaluation process, an item is marked as relevant provided that at least 2/3 of the terms that appear in its metadata record (but not less than two terms), also appear in the user profile with an intensity larger than the average intensity of all terms in the user profile. The position (n) of the item in the recommendation list of N items must fulfill the condition that $n \leq N$. For example, a suggested movie item with a metadata record presenting three terms is marked as relevant if two out of those three terms matching the user profile, with intensities larger than the average intensity of all terms. Otherwise, it is marked as an irrelevant media item. We adopted this approach because we observed in our experiments that the classification of candidate items is also influenced by the number of terms contained in their metadata records, particularly in the *genre-property* nodes. We defined two scenarios to experiment and to evaluate the quality of the recommendation service.

1) The first situation involves generating recommendations based on *category-genre-property* nodes in the user profile. This scenario is realized by using content-based recommendation and collaborative based filtering as the recommendation algorithms, without contextual information. *(ii)* The second scenario is similar to the first. However, in this experiment the recommendation is based on the *category-genre-property-context* nodes of the user profile tree. In other words, this is a situation where contexts play very significant role in the user's preferences i.e. contextual recommendation where the system generates recommendations using

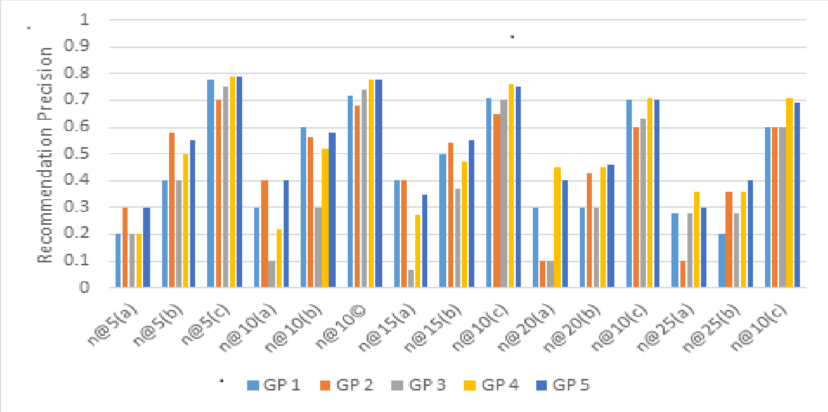

Fig. 9 Contextual recommendation prediction (Precision)

user's contextual information, especially user's activities. The 200 user profiles are divided into 5 by grouping them according to their similarities. We then issued recommendations based on the context-aware hybrid algorithm and the traditional algorithms to a user, representative of each group. The recommendation is repeated for *n = 5,10,15,20,* and *25*. Figure 9 is the precision plot showing the recommendation quality for each group, where a, b, and c represents the precisions for CF, CBF, and the CACBR.

B) Energy consumption evaluation: Since the proposed solution involves sensor monitoring to recognize user contexts and activities, battery drain becomes a concern for the smartphone users. GPS, WiFi or cellular network, accelerometer, orientation, rotation, and other sensors have a significant impact on the battery lifetime. Therefore, to evaluate the impact of the context recognition service on the battery lifetime, the context recognition service was deployed on a Samsung Galaxy S I9000 smartphone as the only active application. We then evaluated the system in eight situations, using the power tutor application [23]. These situations and the percentages of power consumed are shown in table 1. The table shows that the smartphone screen consumes a lot of energy in the active state.

Table 1 Context recognition power consumption

	Sensors	Percentage of Power Consumed
1	All sensors+ Screen	40.66
2	All sensors - Screen	20.84
3	Periodic 30 s + GPS+WiFi+all	8.90
4	Periodic 60 s + GPS+WiFi+all	7.35
5	GPS + WiFi+Accelerometer	5.60
6	GPS+WiFi+Orientation+other	10.78
7	GPS+WiFi+rotation+other	10.40
8	Wi-Fi +other sensors-GPS	7.55
9	GPS +other sensors -WiFi	8.5

Therefore, the context recognition service has been implemented as a background service running at predetermined intervals. This is an important energy optimization consideration in our implementation. In addition, the context recognition service does not have to monitor user contexts continuously since users do not naturally change activities and context every second. Therefore, we implemented another energy optimization. It activates the service at every 30 and 60 seconds. Without the screen, and with all sensors, the energy consumption percentage dropped from 40.66% to 20.84%. In addition, with the periodic execution of the service, we were able to reduce the energy consumption by almost 12% for 30s and about 13% for 60s. We also optimized by disabling some sensors such as Wi-Fi and GPS to know which combination of sensors consumes less power. We observed that leaving the GPS in active mode results in a power consumption surge.

6 Discussion

We have evaluated the traditional algorithms to allow their comparisons with the context-based hybrid algorithm. Generally, the context-based hybrid algorithm performed better than the traditional algorithms [Figure 9]. The traditional algorithms show situations where the recommendation without contexts would not give better performance and vice versa. The reason is perhaps that in the process of filtering, user preferences are not filtered according to the contexts where the users have expressed such preferences. Therefore, content that are irrelevant have been included in the recommendations. For example, precisions in Figure 9 e.g. n@5 (a) for target user groups1& 3 are as low as 0.2. The hybrid recommendation approach that combines contextual content and collaborative algorithms produced the overall best results, the same precision in those algorithms increased to 0.7, whilst achieving highest precision of 0.79 @n5 for groups (4&5) and 0.2, 0.3, 0.5 and 0.55 respectively for CBF and CF. All that is required is to identify the user's current contexts and his initial profile to generate recommendations. Additionally, the user satisfaction obtained from users shows that the context-aware hybrid recommendation is feasible in practice.

In terms of power consumption, the context recognition service performed relatively well, though there is a need for improvement through additional optimization.

7 Conclusion

Accessing enormous media items online via smartphones is a significantly different scenario from using desktops to access the same media items. In this article, we have presented CAMR, a framework that recognizes in smartphone user's contexts and physical activities based on smartphone embedded sensors. The framework first monitors and processes the low-level sensor data to derive high-level contexts or user's physical activities. These contexts and activities are then analyzed to infer situational context, which is at a higher semantic level than the low-level contexts. The framework integrates a contextual user profile service, which relates user's present and past contexts or activities with his preferences to filter online-based media contents for recommendations. Besides, the framework has the capability to track user's interactions with their smartphones, to update his contextual preferences to improve its subsequent recommendations.

Additionally, the evaluation of the recommendation quality of the framework proved that using recognized context and activity information could satisfactorily provide relevant media contents to smartphone users. A user study conducted for 20 users shows that the framework is effective and helpful in discovering interesting online media items based on the user's context and activity information.

Finally, since power consumption by application on smartphone is a very crucial issue, we have evaluated the energy consumption of the framework to understand how we can optimize its energy utilization. This evaluation shows that energy can be significantly conserved if the framework context and the activity recognition service runs at intermittent intervals such as every 30s or 60 seconds. Intermittent switching between GPS or Wi-Fi can also help to conserve power consumption, such as when the user is indoors and with available WiFi connection. In the future, we plan to implement more power consumption optimization techniques to extend the device's battery lifetime. In addition, we would like to evaluate context-aware content based and context aware collaborative filtering recommendations, and compare their performances with that of context-aware hybrid recommendation.

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

References

1. Adomavicius, G., Tuzhilin, A.: 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 17(6), 734–749 (2005)
2. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-Aware Recommender Systems. AI Magazine 32(3), 67–80 (2011)
3. De Pessemier, T., Deryckere, T., Martens, L.: Context-Aware Recommendations for User-Generated Content on a Social Network Site. In: Proceedings of the EuroITV 2009 Conference, New York, USA, pp. 133–136 (2009)

4. Meehan, K., Lunney, T., Curran, K., McCaughey, A.: Context-aware intelligent recommendation system for tourism. In: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), March 18-22, pp. 328–331 (2013)
5. Chen, A.: Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment. In: International Workshop on Location- and Context-Awareness, Oberpfaffenhofen, Germany, pp. 244–253 (2005)
6. Otebolaku, A.M., Andrade, M.T.: 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)
7. Dong, L., Xiang-wu, M., Jun, L.C.: A Framework for Context-Aware Service Recommendation. In: 10th International Conference on Advanced Communication Technology, ICACT 2008, February 17-20, vol. 3, pp. 2131–2134 (2008)
8. Wenping, Z., Lau, R., Xiaohui, T.: Mining Contextual Knowledge for Context-Aware Recommender Systems. In: 2012 IEEE Ninth International Conference on e-Business Engineering (ICEBE), September 9-11, pp. 356–360 (2012)
9. Otebolaku, A.M., Andrade, M.T.: Recognizing High-Level Contexts from Smartphone Built-In Sensors for Mobile Media Content Recommendation. In: 2013 IEEE 14th International Conference on Mobile Data Management (MDM), June 3-6, vol. 2, pp. 142–147 (2013)
10. Yu, Z., Zhou, X., Zhang, D., Chin, C.Y., Wang, X., Men, J.: Supporting Context-Aware Media Recommendations for Smart Phones. IEEE Pervasive Computing 5(3), 68–75 (2006)
11. Wang, X., Rosenblum, D., Wang, Y.: Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20th ACM International Conference on Multimedia, Nara, Japan, October 29-November 02 (2012)
12. Ostuni, V.C., Gentile, G., Di Noia, T., Mirizzi, R., Romito, D., Di Sciascio, E.: Mobile Movie Recommendation with Linked Data. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES 2013. LNCS, vol. 8127, pp. 400–415. Springer, Heidelberg (2013)
13. Kwapisz, J., Weiss, G., Moor, S.: Activity Recognition using Cell Phone Accelerometers. ACM SIGKDD Explorations Newsletter 12(2), 74–82 (2010)
14. Mobasher, B.: Contextual user modeling for Recommendation. In: Keynote at the 2nd Workshop on Context-Aware Recommender Systems (2010)
15. Bobadilla, J., Ortega, F., Hernando, A., Gutirrez, A.: Recommender systems survey. Knowledge-Based Systems 46, 109–132 (2013)
16. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 331–370, doi:10.1023/A: 1021240730564
17. Resnick, P., Neophytos, I., Mitesh, S., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of ACM CSCW 1994 Conference on Computer Supported Cooperative Work, Sharing Information and Creating Meaning, pp. 175–186 (1994)
18. Lara, O.D., Labrador, M.A.: A Survey on Human Activity Recognition using Wearable Sensors. IEEE Communications Surveys & Tutorials 15(3), 1192–1209 (Third Quarter 2013), doi:10.1109/SURV.2012.110112.00192.

19. Java EEE, http://www.oracle.com/technetwork/java/javaee/overview/index.html (accessed in October, 2013)
20. Benitez, A.B., Zhong, D., Chang, S.-F., Smith, J.R.: MPEG-7 MDS Content Description Tools and Applications. In: Skarbek, W. (ed.) CAIP 2001. LNCS, vol. 2124, pp. 41–52. Springer, Heidelberg (2001)
21. Zhang, Y., Wang, L.: Some challenges for context-aware recommender systems. In: 2010 5th International Conference on Computer Science and Education Computer Science and Education (ICCSE), pp. 24–27 (August 2010)
22. Java RESTful Web Services http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html (accessed in October, 2013)
23. PowerTutor, https://play.google.com/store/apps/details?id=edu.umich.PowerTutor (accessed in October 2013)