

An Intelligent Architecture for Service Provisioning in Pervasive Environments

Anis Yazidi
Dept. of ICT

University of Agder, Grimstad, Norway

Ole-Christoffer Granmo
Dept. of ICT

University of Agder, Grimstad, Norway

B. John Oommen
School of Computer Science

Carleton University, Ottawa, Canada.

Martin Gerdes
Ericsson Research, Aachen, Germany.

Frank Reichert
Dept. of ICT
University of Agder, Grimstad, Norway

Abstract—The vision of pervasive environments is being realized more than ever with the proliferation of services and computing resources located in our surrounding environments. Identifying those services that deserve the attention of the user is becoming an increasingly-challenging task. In this paper, we present an adaptive multi-criteria decision making mechanism for recommending relevant services to the mobile user. In this context, “*Relevance*” is determined based on a user-centric approach that combines both the reputation of the service, the user’s current context, the user’s profile, as well as a record of the history of recommendations. Our decision making mechanism is adaptive in the sense that it is able to cope with users’ contexts that are changing and drifts in the users’ interests, while it simultaneously can track the reputations of services, and suppress repetitive notifications based on the history of the recommendations. The paper also includes some brief but comprehensive results concerning the task of tracking service reputations by analyzing and comprehending Word-of-Mouth communications, as well as by suppressing repetitive notifications. We believe that our architecture presents a significant contribution towards realizing intelligent and personalized service provisioning in pervasive environments.

I. INTRODUCTION

The environment in which we live in today is truly “pervasive”. The proliferation of services and computing resources, indeed, makes the very dream of computing in such a pervasive environment realizable. However, this task has numerous real-life hurdles. Most prominent among these is the task of identifying those services that deserve the attention of the user. Ironically, as the services and tools become more pervasive, this task, in itself, is becoming increasingly-challenging due to the fact that:

- 1) The increasing number of services can overwhelm the attention of even the most educated user. It is, rather, plausible that an arbitrary user is *not even aware* of the services at his disposal.

B. J. Oommen is a *Chancellor’s Professor*, a *Fellow : IEEE* and a *Fellow : IAPR*. He also holds an *Adjunct Professorship* with the Dept. of ICT, University of Agder, Norway. The first author gratefully acknowledges the financial support of the *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada.

- 2) The changes of a user’s preferences and needs over time, renders the task of predicting his current services/interests extremely difficult.
- 3) The result of the interaction of the user with any specific service is usually uncertain. It surely depends on the performance of the latter. As low performance services can provoke his dissatisfaction, it is mandatory that an expedient system must be capable of identifying reputable (and disreputable) services [15], [17].

The complexity of understanding what services could be interesting and important enough to justify disturbing the user, is the main challenge of our research. To respond to this challenge, we argue that service recommendation should rely on a multi-criteria decision maker that combines different aspects (dimensions) of the system/environment in order to decide, on behalf of the user, whether a service is relevant or not. “*Relevance*”, we propose, should be determined based on a user-centric approach that collectively combines the reputation of the service, the user’s current context, the user’s profile, as well as a record of the history of recommendations. This is precisely what we have attempted to achieve in our endeavor, and we thus believe that our architecture presents a significant contribution towards realizing intelligent personalized service provisioning in pervasive environments.

To clarify things, we shall present an instantiation of our architecture to a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services. In the scenario, the goal is to build a personalized and context-aware decision maker that delivers narrowly-targeted notifications to the user about relevant services in his environment. Nevertheless, even though this instantiation is specific, the proposed architecture is generic and can be applied to recommend a wide range of services.

Before we proceed we would like to mention that it is impossible to *comprehensively* describe the design and implementation of the entire system in a single paper. The system which we propose contains numerous modules which deal with inter-user communications, the ranking of services, inferring the dependability of other users within a social network, communication from the system to the user, discovering and

recording reputations etc. Each of these modules, in itself, is a contribution in its own right. The pertinent results describing some of these modules have already been published, and the results concerning the other modules are currently being compiled. Thus, we emphasize that while this paper contains the design and implementation details of the overall architecture, we will briefly describe the functionality of some of the component modules, and omit the details which are found in the associated citations. The reader should note that a more detailed description of all of these components and the overall system will be found in the doctoral thesis of the first author [14].

The remainder of this paper is organized as follows. In Section II, we describe some reported studies which are closely related to our approach. Then, in Section III, we present the details of the architecture by explaining the functionality of each of the components and their mutual interactions. Section IV reports the results of simulations conducted. These results demonstrate the efficiency of our design in reducing the unobtrusiveness that might be caused by traditional service recommendation systems. Concluding remarks and future lines of research are outlined in the conclusion.

II. RELATED WORK

The rich availability of services in pervasive environments has the effect of over-burdening the system's service selection task. According to the vision of pervasive computing promoted by Mark Weiser, the intention of incorporating more advanced technology should be that it provides the user the possibility of operating in a *calm* frame of mind [12]. He writes: "Increasingly, the bottleneck in computing is not its disk capacity, processor speed, or communication bandwidth, but rather the limited resource of human attention" [4]. Filtering out irrelevant information has been a focal concern in a number of studies. The main issue has been to reduce the cognitive load on the user when it comes to selecting services. It is well known that "pushing" (or downloading) notifications messages to users can cause interruptions and distractions. Users who receive irrelevant notifications may become dissatisfied with their recommendation service. According to the I-centric paradigm proposed by Wireless World Research Forum (WWRF), the service provision should be tailored to the actual needs of the user [3]. The I-centric vision promotes personalization, ambient awareness and adaptability as the core requirements of future services.

A number of studies have been performed to realize this user-centric vision. A pioneering recent work was performed by Hossain *et al.* [5]. In this work, the authors proposed a gain-based media selection mechanism. In this regard, the gains obtained by ambient media services were estimated by combining the media's reputation, the user's context and the user's profile. As a result of such a modeling process, the service selection problem was formulated as a gain maximization problem. Thereafter, a combination of a dynamic and a greedy approach was used to solve the problem. There are some

fundamental differences between the study of [5] and the approach that we have proposed in this paper. From an architectural point of view, our work is based on a Publish/Subscribe paradigm in order to realize matchmaking between available services and the user's preferences¹. Moreover, the authors of [5] did not present mechanisms to compute the reputation of the media services, thus, in effect, assuming that it is merely static. We argue that this assumption is not always valid, and that it is of paramount importance that the system tracks the variations in the reputation of the services since they, almost certainly, change over time.

A pertinent study that falls in the same class of our current work is the *Dynamos* project [10]. The *Dynamos* approach is an example of a context-aware mobile application that can be used for recommending relevant services to the user. In [10], the authors designed a hybrid recommender system for notifying users about relevant services in a context-aware manner. The model is based on a peer-to-peer social functionality model, where the users can generate contextual notes and ratings, and attach them to services, or to the environments. They are also permitted to share these with their peers. The attached notes to the environment are delivered to other users whenever they are in the spatial vicinity of the entities associated with the notes. A main difference between their work and what we propose is the way by which preferences are described. Their work assumed that the user was expected to explicitly describe his preferences by manually entering them. In this sense, the profile is defined by the user by explicitly specifying the types of activities and associating multiple interests to them. Such an approach can be considered to be a more "primitive" approach – it is not viable in pervasive environments where preferences change over time. Moreover, the issue of suppressing repetitive notifications was not addressed in [10].

A comprehensive study for personalized service provision has been performed by Naudet *et al.* from Bell Labs [8]. In [8], Naudet *et al.* designed an application for filtering the TV content provided to users' mobiles based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [1], which leveraged Machine Learning (ML) techniques for user profiling.

The motivations behind our work are the following:

- 1) First of all, most of the reported context-aware recommendation systems do not consider the reputation of the services when issuing recommendations. In order to ensure that our hybrid recommender systems is unobtrusive, we need to locate reputable services. The success of reputation systems (such as *Ebay*) suggests that there are significant latent benefits in the convergence of these ideas in pervasive environments.
- 2) Secondly, in order to ensure minimal user distraction, the system should be able to track the changes in a user's

¹Adopting a "Push" based approach does not limit the applicability of our approach. In fact, the paradigm is still valid and can function in a "Pull" based manner, as in the *Dynamos* project [10].

interests, over time. In fact, static approaches, where the user manually defines his interest’s domains, are usually not expedient as the user’s needs and interests change over time. Therefore, appropriate ML techniques are needed for adapting to changing interests by *inconspicuously* monitoring service usage.

- 3) Thirdly, repetitively reissuing the same notification regarding the same service is usually regarded as a nuisance to the user’s attention. In [15], we addressed the issue of suppressing repetitive notifications in a social mobile application. With regards to recommender systems, to the best of our knowledge, the question of suppressing repetitive notifications has not been addressed before in the literature.

Stemming from these observations, we construct a hybrid recommender system that minimizes the distraction to a user’s attention while, simultaneously, maximizing the hit ratio of the service notifications. In accordance with the multiple dimensions that affect the decision making process, we have also defined a set of enabler components. The synergy between these enabler components is ensured through a Publish/Subscribe architecture.

All of these issues will be crystallized in the next section where we describe the architecture of our proposed system.

III. ARCHITECTURE

The main goal of our our multi-criteria decision maker is to pro-actively notify the user about relevant services. In this section, we present the different components which articulates the architecture of our system.

A. Context-Dependent Service Category Activation Rules

Within our framework, the “context” includes any information that can be used to characterize the situation of a mobile user requesting a service. It could include numerous pieces of information including the user’s location (where), the time of presence (when), his current activity, his “mood” etc.

We should emphasize that in general, a user’s interests are *context-dependent*. For example, recommendations about restaurants might be of interest to a certain user during weekends, when he is both close to the restaurant in question and when he is not busy. Therefore, a viable approach is to provide the user with the ability to specify that certain kinds of services (those of interest) are active in a particular context. This is the approach that we have adopted in the current study. The idea is relatively novel and has been recently applied in the *Dynamos* framework [10]. In [10], a user is permitted to specify several types of activities and their associated status, and to associate multiple interests to each of them.

With regard to specifics, in this paper, we will adopt a *two-level filtering approach* in order to support efficient matching between the available services and the user’s profile. The first level of filtering is based on the user’s context and is called *Context-Dependent Service Category Based Filtering*. The concepts here are akin to those found in [10], where for *each service category* (for example, restaurants, shopping,

tourist attractions etc.) the user specifies the context attributes needed to make this category valid. Note that this sort of filtering is static, and can be implemented using fixed rules or stereotypes. Consequently, since the rules are static, they can be entered by the user or can be given by a template, while the names of the interests can be predefined based on a service taxonomy. From this perspective, this filtering is coarse, since we retain the *service category* such as restaurants, but do not consider refining the service recommendation by considering sub-categories of restaurants, such as Italian Pizza restaurants, Japanese restaurants, etc. In order to realize a more diversified service category matching, we integrate a wider range of pieces of contextual information, and not only location. The context attributes are mainly:

- Where: The location of the user.
- When: The time context.
- What: The activity of the user.
- What Mood: The mood of the user.

We define a function F , that statically maps a set of context attributes to a service Category as:

$$F : C_{location} \times C_{time} \times C_{activity} \times C_{mood} \mapsto ServiceCategories$$

An example of a Context-Dependent Service Categories Activation Rule, based on the inferred context attributes is:

$$F(location = *, time = weekend evening, user activity = walking, mood = *) = Restaurants$$

B. Learning Preferences Manager

In the previous subsection, we explained our approach to filter the available services based on their categories using Context-Dependent Service Category Activation Rules. Obviously, the category-based filtering will reduce the number of eventual services that might be of interest to the user. Nevertheless, such a filtering is coarse and needs further refinement. Therefore, we propose to carry out a second level service filtering which performs an even closer match. In this sense, the second level filtering re-filters the services *via* a finer granularity, based on the learned interests in the sub-categories. In fact, it is important that we want to model not only a *general user’s interests* such as restaurants, shops, movies etc., but also the *sub-categories* of these interests that are relevant to a given user. In [16], we had presented a novel, personalized *Learning Preferences Manager* that is able to adapt to changes brought about by variations in the distribution of the user’s interests, using the principles of weak estimation. This module is a fundamental component of our architecture. In the quest to learn the user’s dynamic profile, the Learning Preferences Manager is guided by so-called *Relevance Feedback* (RF) [7]. In this paper, we rely on the *Service Usage History* maintained by the authors of [5], [6] as the main source of the RF. A Service Usage History (also known as the *Interaction History*), contains the history of the services used by the user over time. For example, when the user has used a certain service at a certain time instant, the Learning Preferences Manager refines and revises the user’s profile based on the current instance

of the usage history, which, in turn, is automatically and unobtrusively observed in the background. To now quantify this, we have recommended the use of a Weak Estimator (devised by Oommen *et al.* [9]) so as to update the score of the data-item based on the usage history.

C. Service Reputation Manager

In this section, we introduce the Service Reputation Manager [15], [17], which is a cornerstone component of our architecture. Reputation is a particularly important criterion for filtering services.

With the abundance of services available in a pervasive environment, identifying those of high quality is a crucial task. When services are pervasive, in order to maximize the usefulness of the services accessed, the user needs to build his opinion about these services in the absence of direct experience, and as a consequence, must rely on the experiences of his acquaintances. In fact, through leveraging the power of Word-of-Mouth communications, our hybrid recommender system permits us to identify reliable services possibly deserving the user’s attention. Traditional reputation systems, usually compute the reputation of a service as the average of all provided ratings. This corresponds, for instance, with the percentage of positive ratings in the *eBay* feedback form [11]. Such a simplistic approach of just blindly aggregating users’ experiences may mislead the reputation system if some of the user’s acquaintances are misinformed/deceptive users. Misinformed/deceptive users attempt to collectively subvert the system by providing either unfair positive ratings about a service, or by unfairly submitting negative ratings. Since an alternate way to interpret unfair ratings is to consider the unreliable referrals as coming from people with different tastes, such “deceptive” agents may even submit their inaccurate ratings innocently – due to differences in tastes. Our system can easily become intrusive and ultimately become unusable if the “trust component” (or equivalently, the Service Reputation Manager) does not deal with unfair ratings of this sort. The risk of attacks from malicious users is a crucial issue that we have incorporated in our system, which is especially pertinent in a competitive marketplace.

It is reasonable to assume that the acquaintances of the user can be divided into two classes: trustworthy acquaintances that provide accurate ratings, and unreliable acquaintances that provide unfair ratings. It follows that a good reputation manager component would seek to classify the acquaintances in one of these two classes so as to counter the detrimental effect of unfair ratings. In [15], [17], some of the authors of this present paper developed a Service Reputation Manager which is based on a concept analogous to collaborative filtering in order to separate between these two classes. The premise of the scheme in that paper was to separate the users’ types by observing how they rate the same services. The latter scheme was designed in such a way that these users would be in the same group by maximizing the “within-group” similarities and minimizing the “between-group” similarities.

D. Notification Novelty Checker

The last phase of our decision maker is a module whose task is to identify if triggering a service notification will be perceived by the user as being “repetitive” information. In [18], we have argued that the user’s activities can be modeled to follow some “noisy” periodic pattern, and so we can, in turn, affect the services notifications to be periodic as well. This argument will be true unless we suppress repetitive information. Consequently, any notification about a service that provides redundant information to the user can be regarded as being an unnecessary distraction. Arguing along the same vein, in this paper, we propose that we can incorporate here the same approach that we have used for suppressing repetitions in the friendly reminder application of [18]. In [18], we introduced a new scheme for discovering and tracking noisy spatio-temporal event patterns, with the purpose of suppressing re-occurring patterns, while discerning novel events. Our scheme is based on maintaining a collection of hypotheses, each one conjecturing a specific spatio-temporal event pattern. A dedicated Learning Automaton (LA) – the *Spatio-Temporal Pattern LA* (STPLA) – is associated with each hypothesis. Whenever a user receives a service notification related to a given service, a STPLA is instantiated, and this is attached to the latter service notification in order to learn the periodicity of the context in which the service is available to the user. By processing events as they unfold, we attempt to infer the correctness of each hypothesis through a real-time guided so-called random Walk/Jump process.

E. Service Notification Based on a Publish/Subscribe Paradigm

Now that the individual modules have been explained, we state that the overall architecture of our system would be as described pictorially in in Figure 1.

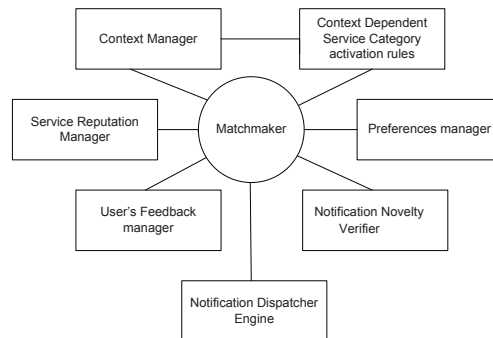


Fig. 1. The high-level architecture of our system.

Our requirement of offering highly pertinent information through a push-based approach to the user can be well supported through the Publish/Subscribe paradigm [2]. The system can be deployed using a Publish/Subscribe System, which puts all the pieces of this puzzle together. A Publish/Subscribe model consists of information providers, who publish events to the system, and of information consumers, who subscribe to events of interest within the system. A

Publish/Subscribe architecture ensures the timely notification of events to the interested subscribers. Note too that the use of a Publish/Subscribe server will enhance privacy, since no user-sensitive private information need be transmitted to the service providers.

A notification is issued whenever the service matches the user’s subscription. In other words, this occurs whenever the following conditions should be met:

- A spatial filter reports that the service is in the user’s vicinity. We agree that in the case of location-based services, the knowledge of the user’s context is the most differentiating information within this context.
- The Service Reputation module returns the truth value of whether the service is reputable, as per the approach defined in [15], [17], and briefly explained in Section III-C.
- The service description matches the user’s profile according to the above-mentioned two-level filtering approach. To identify service items of interest, the matching process consists of two steps. First, for each service, its associated category is matched with the set of active service categories. These service categories, generally, specify the business branches of the service (e.g., Restaurants, Shops). After applying the context-dependent category activation rules, only the services belonging to the active categories are maintained. Moreover, the service subcategory should match the second filter characterized by a finer granularity, namely the Learning Preferences Manager.
- The Novelty Detection module reports that the eventual service notification would not be repetitive by checking whether the notification is a part of a spatio-temporal pattern.

IV. EXPERIMENTAL RESULTS

To demonstrate the proof of these concepts, in this section, we present results of simulations that we have conducted, that puts into a nutshell all the components of the proposed architecture. To do this, we have adopted a Discrete Event Simulation methodology. The performance metric to assess our architecture is the hit ratio, denoted $\alpha(t_n)$, and defined at any given time instance, t_n , as the ratio of the relevant notifications delivered to the user at time t_n . We define a relevant (or equivalently, non-distractive) notification as one where:

- The service matches the user’s profile
- The notification is not repetitive
- The user’s interaction with the service leads to the user’s satisfaction.

By virtue of the above, we consequently regard a distractive notification as one that is either repetitive, or if the interaction with the service does not lead to the user’s satisfaction due to its low performance value [15], [17], or if the recommended service does not match the current user’s interests.

In the same vein, we define the “Distraction” ratio (denoted $\beta(t_n)$), at any given time instance t_n , as the ratio of distractive

notifications delivered to the user at time t_n . Clearly $\alpha(t_n) + \beta(t_n) = 1$.

To demonstrate the power of our architecture, we compare our approach to an *Unguided Recommendation System*, that delivers to the user notifications regarding services that match only his contextual preferences based on the Context Based Service Category static filtering. In other words, we suppose that the *Unguided Recommendation System* performs only coarse contextual filtering. For example, in the case of the notification of location-based services, the *Unguided Recommendation System* sends restaurant suggestions every time the user is close to a restaurant without learning his profile, without suppressing repetitive alerts and without checking the reputation of the service. In our simulation, we considered delivering only a single notification per location².

We assume that the user’s mobility follows a given noisy periodic spatio-temporal pattern. As explained previously in Section III-D, the location and time primitives are combined from their cross-product spaces to produce spatio-temporal patterns. Let us suppose that the mobile user in question, u , visits a given location R according to a weekly spatio-temporal pattern characterized by an omission noise $q = 0.1$. We suppose that a pool of services S is available in the visited area, for eventual access by the user.

At this juncture, it is important to remind the reader that, for the sake of clarity, we use two time granularities (or two time scales) for different events in our Discrete Event Simulation model. In fact, at the granularity of a week, namely at time instances t_n (n denotes the week index), the user visits the location R , and therefore, it is likely that service notifications can take place. On the other hand, at the lower time scale (or equivalently, at the finer time granularity) of a day, we assume that other possible events can take place, such as the generation of Relevance Feedback that serves as input to the Learning Preferences Manager, or the submission of a service rating by user in \mathcal{U} that serves as input to the Service Reputation Manager.

We further assume that the mobile user u possesses a set of acquaintances \mathcal{U} that communicate their experiences regarding the performance of the available pool of services, S . We assume that at discrete time instances, the acquaintances in \mathcal{U} communicate their ratings to u . In the absence of direct experience from the user, the feedback provided by the acquaintances serves as input to the Service Reputation Manager, referred to in Section III-C.

For the sake of clarity and simplicity, we assume that the user preferences fall into two categories C_1 and C_2 . We further assume that the underlying distribution of the weights of the preferences that reflect the affinity of user’s interest in each of the preferences categories C_1 and C_2 follows a binomial distribution [16]. Therefore, the problem of estimating the user’s interests in this particular case is modeled as the estimation of the parameters for binomial random variables.

²It is possible to adopt a top- N recommendation approach in order to not overwhelm the user with a long list of services and thus limit the size of the list.

The Relevance Feedback concerning the preferences categories C_1 and C_2 is generated according to the true underlying value of s_1 and s_2 . The intention of the Learning Preferences Manager is to estimate S , i.e., s_i for $i = 1, 2$. We achieve this by maintaining a running estimate $P(n) = [p_1(n), p_2(n)]^T$ of S , where $p_i(n)$ is the estimate of s_i at time granularity ‘ n ’, where n denotes the day index. Note that we assume that the Relevance Feedback is available at the finer time granularity of a day.

If $s_i > s_j$, we say that category C_i represents the user’s preferred interest category, and thus assume that only services that belong to category C_i are of interest to the user. All the services belonging to category C_j will not be of interest to the user, and notifying him about these services will result in a distraction. Consequently, the matchmaking of the preferences will rely on the same simple mechanism, and recommend the services whose estimated category weight is larger between the two categories. The reader should observe that this simple rule is similar to decision rules in classifiers, where the decision maker has to decide on a hypothesis on the state of nature between two exclusive hypotheses. However, a more sophisticated preferences matchmaking approach, analogous to the one in [5], [6] that is based on assessing a linear combination of the weights, can be easily adopted in combination with our Learning Preferences Manager [16].

An important parameter that must be specified is the rate at which the Relevance Feedback occurs. We suppose that at the finer time granularity of a day, a Relevance Feedback is generated according the underlying distribution, S . Therefore, the estimated weights of C_1 and C_2 are tracked and updated at the granularity of a day.

We further model the performances of services as either being High Performance or Low Performance as reported in [15], [17]. We also assume that the services either belong to C_1 or C_2 . Therefore, we will have a combination of 4 exclusive classes of services in the current experiments:

- 25 High performance services that belong to C_1
- 25 High performance services that belong to C_2
- 25 Low performance services that belong to C_1
- 25 Low performance services that belong to C_2 .

If, for example, C_1 represents the current preferred interest category, the Recommendation System will recommend services to the user that are both of high performance, and that belong to category C_1 . In the simulation settings, we assume that the user possesses 40 acquaintances in his social network – 20 of which are deceptive and the remaining 20 are trustworthy [15], [17]. Furthermore, the trustworthy user’s acquaintances are characterized with $p = 0.8$, while the deceptive ones have $p = 0.2$. In all the experiments, we configure the STPLA with $N_1 = 5$ and $N_2 = 5$. The high performance services have an performance probability of 0.8, while the low performance services have are characterized by the performance probability of 0.2 [15], [17].

As alluded to previously, we suppose that the user’s mobility follows a weekly periodic noisy pattern, and thus we conducted the simulations for a period of 40 week instances.

We report now the results obtained by testing our proposed architecture in a variety of settings³.

In this experiment, we compared the distraction ratio as well as the hit ratio obtained by our approach with the respective ratios obtained by utilizing an Unguided Recommendation System. The results were obtained from an ensemble of 100 simulations, and we report $\alpha(t_n)$, where n denotes the week index. In Figure 2(a), we report the hit ratio, and in Figure 2(b), we report the distraction ratio. The preferences were assumed static, and thus, in other words, we employed the same underlying distribution for the weights of the preferences. We also assumed that the current preferred services category was C_1 . We supposed that at a finer time granularity, namely, at a daily basis, each of the acquaintances submitted a rating for a randomly chosen service among the pool of available services. We observe from Figure 2(b) that the distraction ratio asymptotically approaches the value 0.2 and that the hit ratio approaches the value 0.8. These values can be explained by the fact that our architecture tends to recommend only the 25 High performance services that belong to C_1 as time advances.

Furthermore, we remark from Figure 2(a) and its counterpart Figure 2(b) that the performances achieved by utilizing our proposed architecture improves almost uniformly over time, and that it outperforms the Unguided Recommendation System.

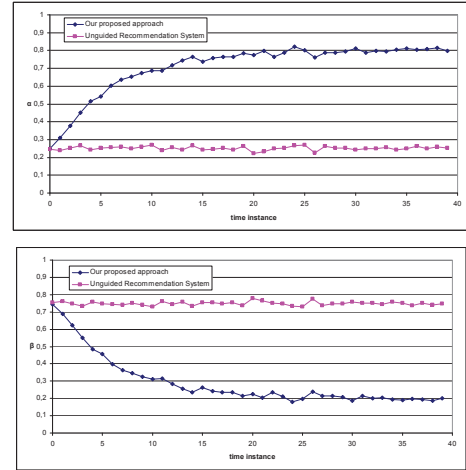


Fig. 2. (a) The evolution of the hit ratio in the case of our proposed approach and the Unguided Recommendation System, when the performance probabilities of the high and low performance services are 0.8 and 0.2 respectively. (b) The evolution of the distraction ratio in the case of our proposed approach and the Unguided Recommendation System for the same settings.

The results for another set of experiments are shown in Figure 3(a) and Figure 3(b). In these experiments, we changed the settings by assuming that the high performance services had an performance probability of 0.7, while the low performance services were characterized by the performance probability of

³We have done experiments for numerous settings and scenarios. For the sake of brevity, in this paper, we report only a few of them. More detailed simulation results are found in [13] and [14].

0.3. Whereas in Figure 3(a), we report the hit ratio, in Figure 3(b), we report the distraction ratio. As in the case of the previous figures, the convergence of the graphs to their optimal levels is clear from these figures too.

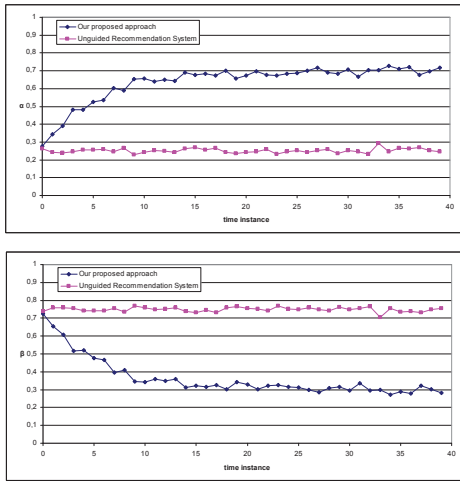


Fig. 3. (a) The evolution of the hit ratio in the case of our proposed approach and the Unguided Recommendation System, when the performance probabilities of the high and low performance services are 0.7 and 0.3 respectively. (b) The evolution of the distraction ratio in the case of our proposed approach and the Unguided Recommendation System for the same settings.

V. CONCLUSION

In this paper we have considered the problem of computing in pervasive environments, and in particular, in identifying those services that deserve the attention of the user. We have presented an adaptive multi-criteria decision making mechanism for recommending relevant services to the mobile user, where “*Relevance*” is determined based on a user-centric approach that combines both the reputation of the service, the user’s current context, the user’s profile, as well as a record of the history of recommendations. We have proposed the architecture of a system that builds a personalized and context-aware application that delivers narrowly targeted information to the user, while being unobtrusive. The design avoids flooding the user with irrelevant information. We have conducted simulations and reported results that suggest that our architecture can significantly reduce unobtrusiveness. To gain more insights into the acceptance of the system by an end user, in the future, we propose that the system be deployed into a real-life application domain, which also incorporates a user study.

REFERENCES

- [1] A. Aghasaryan, S. Betgé-Brezetz, C. Senot, and Y. Toms, “A profiling engine for converged service delivery platforms,” *Bell Lab. Tech. J.*, vol. 13, no. 2, pp. 93–103, 2008.
- [2] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, “Matching events in a content-based subscription system,” in *PODC '99: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*. New York, NY, USA: ACM, 1999, pp. 53–61.

- [3] S. Arbanowski, P. Ballon, K. David, O. Droegehorn, H. Eertink, W. Kellerer, H. van Kranenburg, K. Raatikainen, and R. Popescu-Zeletin, “I-centric communications: Personalization, ambient awareness, and adaptability for future mobile services,” *IEEE Communications Magazine*, vol. 42, no. 9, pp. 63–69, 2004.
- [4] D. Garlan, D. Siewiorek, A. Smaligic, and P. Steenkiste, “Project aura: Toward distraction-free pervasive computing,” *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, 2002.
- [5] M. A. Hossain, P. K. Atrey, and A. El Saddik, “Gain-based selection of ambient media services in pervasive environments,” *Mob. Netw. Appl.*, vol. 13, no. 6, pp. 599–613, 2008.
- [6] M. A. Hossain, J. Parra, P. K. Atrey, and A. El Saddik, “A framework for human-centered provisioning of ambient media services,” *Multimedia Tools and Applications*, vol. 44, pp. 407–431, 2009.
- [7] M. Montaner, B. Lpez, and J. L. de la Rosa, “A taxonomy of recommender agents on the internet,” *Artificial Intelligence Review*, vol. 19, pp. 285–330, 2003.
- [8] Y. Naudet, A. Aghasaryanb, S. Mignon, Y. Toms, and C. Senot, “Ontology-based profiling and recommendations for mobile tv,” in *Semantics in Adaptive and Personalized Services*, ser. Studies in Computational Intelligence, M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bielikova, Eds. Springer Berlin / Heidelberg, 2010, vol. 279, pp. 23–48.
- [9] B. J. Oommen and L. Rueda, “Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments,” *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.
- [10] O. Riva and S. Toivonen, “The dynamos approach to support context-aware service provisioning in mobile environments,” *J. Syst. Softw.*, vol. 80, no. 12, pp. 1956–1972, 2007.
- [11] A. Schlosser, M. Voss, and L. Brckner, “On the simulation of global reputation systems,” *Journal of Artificial Societies and Social Simulation*, vol. 9, 2005.
- [12] M. Weiser, “The computer for the twenty-first century,” *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.
- [13] A. Yazidi, O.-C. Granmo, B. J. Oommen, M. Gerdes, and F. Reichert, “A user-centric approach for personalized service provisioning in pervasive environments,” *Unabridged version of this paper. Submitted for Publication*.
- [14] A. Yazidi, “Intelligent learning automata-based strategies applied to personalized service provisioning in pervasive environments,” Ph.D. dissertation, Department of ICT, University of Agder, Grimstad, Norway, 2011.
- [15] A. Yazidi, O.-C. Granmo, M. Lin, X. Wen, B. J. Oommen, M. Gerdes, and F. Reichert, “Learning automaton based on-line discovery and tracking of spatio-temporal event patterns,” in *PRICAI 2010: Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, B.-T. Zhang and M. Orgun, Eds. Springer Berlin / Heidelberg, 2010, vol. 6230, pp. 327–338.
- [16] A. Yazidi, O.-C. Granmo, and B. J. Oommen, “An adaptive approach to learning user’s preferences using weak estimators,” *Submitted for publication*.
- [17] —, “Service selection in stochastic environments: A learning-automaton based solution,” *To Appear in Applied Intelligence*.
- [18] —, “A learning automata based solution to service selection in stochastic environments,” in *Proceedings of the Twenty Second International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA-AIE 2010)*, ser. Lecture Notes in Artificial Intelligence, 2010, pp. 209–218.