Mobile User Personalization with Dynamic Profiles: Time and Activity

Christoforos Panayiotou, George Samaras

Department of Computer Science, University of Cyprus, CY-1678 Nicosia, Cyprus
{panchris, cssamara}@cs.ucy.ac.cy

Abstract. Mobile clients present a new and more demanding breed of users. Solutions provided for the desktop users are often found inadequate to support this new breed of users. Personalization is such a solution. The moving user differs from the desktop user in that his handheld device is truly personal. It roams with the user and allows him access to info and services at any given time from anywhere. As the moving user is not bound to a fixed place and to a given time period, factors such as time and current experience becomes increasingly important for him. His context and preferences are now a function of time and experience and the goal of personalization is to match the local services to this time-depended preferences. In this paper we exploit the importance of time and experience in personalization for the moving user and present a system that anticipates and compensates the time-dependant shifting of user interests. A prototype system is implemented and our initial evaluation results indicate performance improvements over traditional personalization schemes that range up to 173%.

1. Introduction

Today it is understood that wireless access is not about browsing the Web on your cell phone; it is about providing personalized services that are highly sensitive to the immediate environment and needs (i.e., context) of the moving user. The most recent efforts to support the mobile user focus to the ability to access **local** and the most relevant information and services. The solution of personalization and user profiling is often used to effectively aid this task. Solutions, however, that was well studied and provided for the desktop user proved to be inadequate for the moving user as these two types of users differ in quite fundamental aspects. The one is restricted in a fixed place, for a fixed period of time and his device (i.e. the desktop PC or even laptop) is not generally used as a *personal assistant*. On the other hand the moving user is quite mobile and at any time, place or situation turns to his mobile device (PDA or mobile phone), the *truly* personal device, for access to information and services that are relevant to his current needs. In essence, the moving user is a new breed of users and his handheld device is constantly complementing his current activities. Looking a bit deeper in the mobile user's environment one can clearly see that the new factors involved are **time** and current **activity**. In reality, the **user's needs**, and thus his context, are a function of time and experience. His interests and needs change along with the time and the situation he's currently experiencing.

These are new factors to the personalization problem and are introduced mostly because the needs of the moving users are not any more limited to the time he is in front of his office PC, but around the clock, all year long, including weekends and vacations. Imagine the following scenario where a user cruises around at lunch time browsing his favorite content provider through a personalization system. Most likely, the system would provide only the local to him content. However, the provided content, while matching his interests, would not differentiate between restaurant services, bookstores or fax centers in any meaningful way. Thus, if our user was hungry he should first navigate through all the available services find the restaurant services and then invoke them to get the desired information. In this scenario the personalization system ignored a vital piece of information, namely the fact that it is "lunch time". If the system took the **time**, and what time represents into the users day cycle, into consideration it could alter the provided results to display first the restaurant services.

Another interesting, but not thus far utilized, concept is the so called "**user experience**". By "user experience" we mean the activity (or condition) the user is currently experiencing. For example, during normal working days the user experience could be described as "normal day", while when on vacation as the "vacation" experience. Obviously his needs during "normal days" are quite different than during "vacation". Even the day cycle of the user during the various experiences might be different. It is thus, imperative, for the personalization system, to take the changing of activities into consideration. We want to enable, for example, the system to effectively provide, at a specific time, a vacationing user with the nearest bar or pool, while when he's back to work with the nearest business center. Given an experience, time identifies the specific interests of the user during that activity at that particular time, e.g. during vacation at 8 PM, open bars and happy hours are of great interest to vacationers while during normal days, Pizza restaurants and rent-a-movie places might be of interest instead. The task of such a personalization system is to identify and match these dynamically changing interests to the local services.

The rest of the paper is organized as follows. Section 2 presents a general overview of the personalization problem and further elaborates on "time" and "user experience" concepts. Section 3 presents the needed design changes for a personalization system in order to incorporate time and experience in a personalization system. How to support time and experience based personalization for the moving user is discussed in section 4, while the next section presents an implemented prototype, metrics, experiments and performance analysis. Finally, some related work is presented in section 6 and section 7 concludes the paper.

2. The Personalization Problem

The problem of personalization is a complex one with many aspects and issues that need to be resolved [15]. Some of these issues become even more complicated once viewed from a moving user's perspective. Such issues include, but are not limited to, the following: *What content to present to the user, how to show the content to the use, how to ensure the user's privacy* and *how to create a global personalization scheme.* They could be summarized in the following phrase: "***What, how and for everything.***" There are many approaches to personalization [1-14] and each one of them usually focuses on a specific area, whether this is profile creation, machine learning and pattern matching, data and web mining or personalized navigation. ***Time*** and ***current activity***, however, as factors affecting personalization for the moving user are widely missed. Thus, the summarizing phrase for the personalization issues for this type of users (i.e., mobile and wireless) should actually be: "***When, what, how and for any activity***," where "when" relates content and user preferences to time and current experience/activity.

2.1 Time Based Personalization

Beyond the exploitation of location "***timing***" and "***user experience***" factors seem to have significant importance. These are the. To understand the significance of these factors, one must consider the needs of the mobile user and the effect they may have if exploited. Undeniably his need for content is not limited to a specific time period. Instead the mobile user may browse for content 24 hours a day, 365 days per year. Hence it becomes a reality to have a user where his interests change in an extremely dynamic way (just consider the lunch time scenario where we see this dramatic shift in preferences). A shift that seems to be tied to the various time intervals governing the user's day-cycle becoming even greater when he switches between activities (e.g. taking a vacation after a tiresome business week).

In order to be able to tackle these shifts gracefully we need to know what the users preferences are at any given time. When using user profile for storing those preferences we can easily reach a situation where the user profile is too big to handle and thus useless. Especially when dealing with great continuum called "time". As a first step we can divide the day into several time-zones and store user's preferences per time-zone. These time zones represent the user's day cycle. Yet the problem still remains as there is too much information which must be handled, most of it just replicated. We circumvent this problem by associating each user's interest with a set of weights as they relate to a specific time-zone and experience. This allows the dynamic creation of the user profile based on the current time-zone and activity by applying the relevant weight set on his preferences. In this way time based personalization becomes possible and the benefits seem quite significant (see experimental results in section 5) despite the possibility of higher computational costs.

3. Incorporating Time Based Personalization

Having in place a personalization system that handles user's profiles, content description and application of the user's profiles on that content is the first step towards incorporating time in the personalization process. Most current personalization approaches (that are profile based) handle the following:

- Capture, maintain and adapt the user's preference profile, either implicitly or explicitly. Performed by a "user profile management" component.
- Capture the user's device profile. This could be part of the "user profile management" component.
- Describe the available content. Having a "content description" component suffices.
- Apply the user's preferences on the content description in order to select the desired one. A "content selection" component could implement this.
- Reform and deliver the selected content based on the user's device profile via a "content reform" component.

Beyond that, personalization systems that are focused on the mobile user just adapt the user's profile to the local content. Since these systems don't consider the concepts introduced by time and experience certain changes to their design are needed. *The necessary changes affect mainly the selection of the content to be displayed and the user's profile.*

The user profile must be enhanced in order to accommodate all the (newly) required/available metadata such time-zones preferences and user experiences. As a chained reaction the description of the available content may also need to be enhanced, e.g. from a simplistic keyword scheme to a sophisticated ontology scheme. We need better description of the content in order to be able to make more intelligent decisions. Another needed change, is related with the profile maintenance. We show that keeping the user's profile in sync with his interests' shifts in our case becomes more complicated and an even bigger necessity. Now there is also the need to compensate for user timing shifts (time-zone shifts) and thus, the monitoring of user's preferences mechanism should consider (and adapt) this as well.

4. Designing a Time Based Personalization System

4.1 Content Description Format

In order to have a good description of the provided content, ontologies were used. When using ontologies, in essence we define a vocabulary and a structure (using XML schema) for certain content domains. Having in place the vocabulary (which may or may not be common for all content providers) we define the structure of the content description. Two major distinctions are made: content categories/subcategories and content instances.

The content categories describe general characteristics of some content. We can categorize content on the type of the provided information e.g. restaurant content, pharmacies content etc. Of course we can continue this categorization with more levels, e.g. we can elaborate on the news content category by introducing the sports or political subcategory. Note that a content category does not describe any actual data generator (e.g. the "Paradise Hotel" content page) rather than it groups the characterizing attributes of similar content pages (e.g. all the pages that contain information on restaurants). On the other hand content instances provide actual values for content categories. This information is directly linked with a given content page.

Even though the content description format is not directly related with time-based personalization we need to have knowledge of both the category schema and category instance when we incorporate timing factors in the user's profile. We need to assign weights to both categories and instances of these categories. To understand this lets compare the keyword approach with this one. When using keywords we have a list of all possibly describing words associated with each node without making any distinction as to what they describe. Thus, for example, we cannot tell that the word "Chinese" could describe the restaurant type (content instance) while the word "Restaurant" could give the content category. On the other hand when using an ontology we know what the content category is (i.e. "Restaurant") and what the restaurant type of the described specific node is (i.e., *"Chinese"*). This is important as it allows prioritization based on both, the category and the instance of a content node. In this way timing and the user's experience can be used with both of these, thus leading to the case where they can effect the selection of an entire content category (e.g., display or not any restaurant), as well as, the selection of specific instances of a category (e.g., display ***only Chinese*** restaurants). Of course the timing and experience information will be stored in the user's profile.

4.2 User's Theme Profile Format: Time and Experience Factors

In order to be able to match the user's interests with the provided content his preferences must be captured in a similar format as the description of the content. Towards this goal we use an ontology that is very similar to the content description one. Thus, what we see as user's preferences are the same content categories and instances as the ones in the content description, and thus enabling the match between them. However, this merely enables us to know if a user is interested in something.

We need to assign a weight to each of the user's preferences to differentiate among them. Weights in effect, show how much a user likes or dislikes a specific content category and/or instance. Being able to differentiate among the user's preferences is crucial in time based personalization. To achieve time based personalization we need to know how the user's preferences change over the 24 hour day cycle.

To represent time we suggest dividing the day into different time-zones. This is possible if we study the daily routine of our users and then split it into time-zones based on the user's activities. By dividing the day in time-zones, we drastically reduce the possible combinations between time and user's preferences, keeping our design scaleable. Having the time-zones on one hand, and the preferences' weights on the other, enables us to capture the required information: ***we just need to record how the weight of each preference changes over each time-zone***.

The same information (i.e. preference weights per time zone) can be recorded for the various content instances. Fig. 4.1 shows how time affects the user's restaurant preferences in relation with the restaurant type (i.e., cuisine). We, thus, know how the user preferences for a specific cuisine change according to a time zone. Note that in fig. 4.1 we show weights for both category and instance. In essence we define how much weight the characteristic 'cuisine' has on the user's preferences (1^{st} line of fig. 4.1) and also define the weight of different values for that characteristic. Figure's 4.1 example means that finding a restaurant that has a desired cuisine is averagely important for the user (stated by the first line). However, when examining the cuisine of restaurants, our user states that between 9 and 12 ‘Kebab House’ is preferred over ‘Cypriot’ cuisine.

User experiences. Using this weighted system enables the user to declare what (and when) something is more important to him. We can repeat the weight association for each user’s experience. In that way by just adding another set of weights we can record the user’s preferences for a new experience. For example if the user goes on vacation then his experience would be vacation. What changes from his normal (or daily) experience is the composition of his day cycle. In effect, integrating the user’s experience is like building a user’s profile for each of his activities cycles. The difference, however, is that we may exploit the obtained knowledge and that we do not need to store a separate profile. Basically we replicate the weight structure of the normal daily cycle with different values, resulting a dynamically changing user profile that covers all user’s activities in a continuous fashion.

```xml
<restaurantsTypeByTime categoryWeight="50">
    <timeZone time="0-3">
        <restaurantType name="Fast Food" weight="90"/>
    </timeZone>
    <timeZone time="3-6"></timeZone>
    <timeZone time="6-9"></timeZone>
    <timeZone time="9-12">
        <restaurantType name="Fast Food" weight="90"/>
        <restaurantType name="Greek" weight="70"/>
        <restaurantType name="Cypriot" weight="60"/>
        <restaurantType name="Kebab House" weight="95"/>
    </timeZone>
    <timeZone time="12-15">
        <restaurantType name="Pizzarias" weight="85"/>
    </timeZone>
    <timeZone time="15-18"></timeZone>
    <timeZone time="18-21">
        <restaurantType name="Chinese" weight="90"/>
        <restaurantType name="Italian" weight="80"/>
        <restaurantType name="Mexican" weight="75"/>
    </timeZone>
    <timeZone time="21-24"></timeZone>
</restaurantsTypeByTime>
```

Figure 4.1: User’s restaurants preferences based on time.

4.3 Matching User Interests with the Content

Since the user profile and the content description are based on similar ontology we achieve improved effectiveness during the content selection. At matching, we compare each content information node (content service) description with the user’s profile. The comparison is made on the characteristics between matching categories of the content description and the user’s profile for the current time zone. We get the preference weight for each characteristic, from the user profile along with the weight for the given category. The weight set associated with the current user’s experience is loaded. In essence we have a dynamic user profile which changes, literally, by the minute and current experience. The final step in the content selection is the averaging of the retrieved weights and assignment of it as a selection weight to each content node. The results are presented to the user sorted, based on this weight. Nodes with lower weight than a user selected threshold are completely omitted. The high level algorithm of this procedure would look like this:

1. Find the interesting content categories using the “How much I like a given category – e.g. Restaurants” weight.
2. Evaluate each characteristic of each content node in the given category that is present in the user’s profile using the following formula: ("How important is this factor to me - e.g. Cuisine type" weight * "How much I like this value for this factor - e.g. Italian" weight) / 100.
3. Assign the average of the prev. evaluation as the selection rank of each node.

4.4 Manipulating the time based profile

Initially the user is assigned a default (predetermined by experts) profile based on his social group. Social group can be used as a clustering factor as users tend to form their interests according to their group. However since this is not absolute, collaborative filtering (which may also include history analysis), along with direct user input, can be used to fine tune these default profiles. There might still be a difference between the actual user interests and his profile in terms of interests and weights. Through the use of the system, the user's profile is constantly adapted based on his selections thus leading to a convergence with his actual interests.

The system accepts weights in the range -1 to 100. Value -1 enables the system to completely disregard certain content categories and instances (the system can present disregarded content if the user asks for it explicitly). Updating of the user profiles is based on user selections. Each time a user selects to view a given content service the system tracks it and updates his profile. We need to consider (and thus adapt) all of the characteristics as we cannot single out the ones that led to the selection of the given content. We reduce the weight of unmatched characteristics in order to phase out of the profile preferences that might not be valid in the future. This result in a large number of updates, however, since all the updates are made on the already loaded profile the costs is negligible. The high level algorithm of this procedure would look like this: For the category of the user selected content node do:

1. For all characteristics that are present in both node description and the user's profile increase their weight in the user's profile by 1.
2. For all characteristics that are present only in the user's profile decrease their weight in the user's profile by 1.
3. Add all characteristics that are present only in the node description to the user's profile.

4.5 Keeping in Sync the Profile and User's Timing

As already stated, it is imperative to have the correct timing information in the user's profile. Thus the monitoring mechanism that watches over user's preferences must be extended to incorporate "time" as well. In our case, we monitor the click stream of the user in order to know when and what the user selected. In detail, we timestamp each user click to a service and save it in a log, much like a web log. Later on we analyze this log (when the user is offline or on a different machine) to detect anomalies in the timing. Such anomalies include the user consistently requesting a specific service during a time period that he is not expected (based on his current profile) to do so. A minimum consistency of the anomalies found should be met before considering them to be timing shifts in order to avoid having the casual fluke influencing the profile. One such fluke could be the user anticipating his lunch and thus making a reservation early in the morning. If that happens once in a while we do not want to shift his lunch time zone in the morning. However if it is common practice then we might add a new time zone in the morning that reflects on this specific behavior.

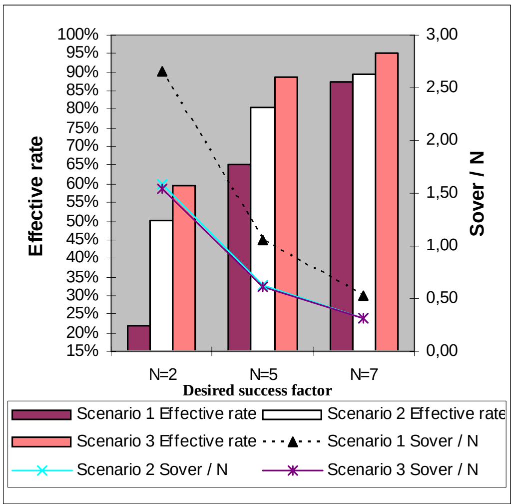
Having flagged all the anomalies in this log we can adapt (or even create new) the time zones that regard the service where the anomaly was noticed. The log analysis is clearly a costly operation and thus should be optimized as much as possible. One such optimization is to identify potential anomalies as they happen. Another optimization is that we keep two logs instead of one; the first holds all the click stream of each user and the second keeps only the identified potential anomalies. The first log can be used to discover previously unknown user preferences and their related time zones. Both logs are analyzed in order to adapt (or create new) the time zones of the user's preferences. However the analysis, which is periodic, for each log has a different frequency. The first log can be analyzed on a much slower pace than the second. In fact, the first log is mainly used to add new time zones and preferences in the user profile, while the second log is (mainly) used to urgently detect shifts in the user's timing. This is the reason that the analysis of the second log has a much higher frequency. The high level algorithm of this procedure would look like this:

1. Monitor and log the user's click stream.
2. Flag potential anomalies with various degrees of priorities and log them.
3. Analyze the anomalies log for timing shift and adapt the user profile. This is periodic with high frequency.
4. Analyze the click stream log to identify new time zones as well as timing shifts missed by the previous analysis. This is periodic with low frequency.

5. Prototype and Experimentation

Our prototype is derived from the mPERSONA [15] system. In our current version, we focused and experimented on the new factors (time and experience). Thus we modified only the content description, selection and profile management components.

***Metrics*:** In order to show that time and user experience are important factors in the personalization process we need to define a way to measure its effect. We do so by comparing the results of personalization without using timing or user experience versus using them. To do this comparison, we need to measure the quality and quantity of the effect of personalization. Towards this end we define the "effective rate" as a quantitative metric and the "overall success factor" as the qualitative metric. The "effective rate" is the percentage of the times the system was successful in providing what the user wanted. A provided result is considered successful if the user finds what he wants within the first N provided choices. That is if the user chooses the n^{th} element of a given result, then the provided result is considered successful if, and only if $n \leq N$. We denote n as the "actual success factor" and N as the "desired success factor". Having a high effective rate while keeping the desired success factor low indicates that the personalization process works well and that the users profile accurately presents his interests. The "overall success factor" S_{over} is denoted as the average of the "actual success factor" for all provided results. Lower values mean that the quality of the personalization results is high. The ratio between overall success factor and desired success factor (S_{over} / N) provides an indication if a personalization system meets the given quality restrictions. Furthermore, keeping N value fixed enables the comparison of the result quality of two (or more) personalization systems.



Graph 5.1: Comparison of test scenarios:
Scen.1: Ignoring time and experience, Scen.2: Exploiting time/Ignoring experience, Scen.3: Exploiting time & experience

***Experiments*:** Three different scenarios were tested. Under the 1st scenario, a set of users request a predefined set of services without exploiting time and experience. The service set is defined by random selection. Which services are selectable is determined according to the high-level user's interests (e.g. likes Chinese restaurants). The users' requests are made during the whole period of a week at 4 different time-zones per day. Scenario 2 is similar to the 1st with the difference that time is used (user's experience is still ignored). In the 3rd scenario we repeat the previous scenario but this time we include user's experience as well. The test bed used utilized 10 service categories with thirty instances each.

Graph 5.1 is a direct comparison of the scenarios. We repeated our measurements for the values 2, 5 and 7 as the desired success factor. Since the lower the desired success factor the higher the expectations from the systems our results look very promising. Looking at the graph one can see that the higher the expectations from the systems the more benefit we have from incorporating time and experience. Even though our initial results show only 8% to 173% gain (for N=2 scenario 3 is by 173% better than scenario 1 while for N=7 only 8%) we expect that having more services the gain will be significantly higher due to the number of available choices and the degree of differentiation. Considering the second measurement that shows ratio between the "overall success factor" (S_{over}) and the "desired success factor" (N) we can see that the 3rd scenario also performs better. Recall that lower values mean better quality results (scenario 3 gives 42% lower values than scenario 1). Clearly as the value of N is increasing the effect of time and experience decreases.

7. Conclusions

Mobile users are a new and more demanding breed of users and they differ significantly from the traditional desktop users. In this paper we have identified factors that until this day were overlooked in the design of personalization systems for this type of users. These factors are related to time. We presented two major factors, time and user's experience, showing their importance. We showed that exploiting time enables us to capture the shifts of user's interests based on the time of the day and adapt his preferences accordingly. User experience, takes the concepts introduced by timing one step further. It provides a means to effectively merge many different instance of the user's profile (one instance for each state of mind, e.g. vacation, work etc.) into one dynamic profile. This dynamic profile can accurately cover the preferences of a user at all times and situations. We have identified which parts of a personalization system is affected by these factors (i.e. profile). We devised appropriate metrics, implemented a prototype and via experimentation demonstrated the viability of our proposal. Our first results, considering the small size of services used, are quite encouraging indicating performance improvements up to 173%.

8. References

1. Eugene Volokh, *Personalization and Privacy,* Com. of the ACM, Vol. 43(8), pp. 84-88, August 2000.
2. Corin R. Anderson, Pedro Domingos, and Daniel S. Weld. "*Adaptive Web Navigation for Wireless Devices*." In Proceedings of the 17th Int. Joint Conference on Artificial Intelligence (IJCAI-01). 2001.
3. Corin R. Anderson, Pedro Domingos, Daniel S. Weld. "*Personalizing Web Sites for Mobile Users."* In Proceedings of the 10th Conference on the WWW, 2001.
4. Paul Maglio and Rob Barrett, *Intermediaries Personalize Information Streams,* Communications of the ACM, Vol. 43(8), pages 96-101, August 2000.
5. C. Thomas and G. Fischer. *Using agents to personalize the web*. In Proc. ACM IUI'97, pages 53--60, Orlando, Florida, USA, 1997.
6. Rucker, J., Marcos, J.P., Siteseer: Personalized Navigation for the Web, Communications of the ACM, pp. 73-75, Vol. 40, no. 3, March 1997.
7. Adomavicius, G. and Tuzhilin, A. "User profiling in personalization applications through rule discovery and validation." KDD-99, 1999.
8. E. Adar, D. Karger, and L. Stein. Haystack: Per-user information environments. In Proceedings of the 1999 Conf. on Inf. and Knowledge Management, CIKM, 1999.
9. Haym Hirsh, Chumki Basu, and Brian D. Davison. Learning to personalize. Communications of the ACM, 43(8), Aug 2000.
10. Kurt Bollacker, Steve Lawrence, and C. Lee Giles. A system for automatic personalized tracking of scientific literature on the web. In Digital Libraries 99 - The Fourth ACM Conference on Digital Libraries, pages 105--113, New York, 1999.
11. Sung Myaeng and Robert Korfhage. Towards an intelligent and personalized information retrieval system. Technical Report 86--CSE--10, Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, Texas, March 1986.
12. M.M. Lankhorst, H. van Kranenburg, A. Salden, A.J.H. Peddemors, "Enabling technology for personalizing mobile services". Proceedings of the 35th Annual Hawaii Int. Conf. on System Sciences HICSS, p.1107 -1114, 7-10 January 2002
13. N. Bogonikolos, D. Fragoudis, S. Likothanassis, "ARCHIMIDES": an intelligent agent for adaptive-personalized navigation within a WEB server. Proc. of the 32nd Annual Hawaii Int. Conf. on System Sciences, HICSS-32. Vol:Track5, 5-8 Jan. 1999.
14. Doron Cohen, Michael Herscovici, Yael Petruschka, Yoëlle S. Maarek, Aya Soffer, "Personalized pocket directories for mobile devices". Proceedings of the eleventh international conference on World Wide Web, p. 627-638, isbn 1-58113-449-5, Honolulu, Hawaii, USA, 2002.
15. Christoforos Panayiotou, George Samaras, "mPERSONA: Personalized Portals for the Wireless User: An Agent Approach". Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on "Mobile and Pervasive Commerce".