Minimizing Mobile Phone Disruption via Smart Profile Management

Amnon Dekel
The Selim and Rachel Benin School of Computer Science and Engineering
The Hebrew University Jerusalem
+972548138160
amnoid@cs.huji.ac.il

Dan Nacht
The Selim and Rachel Benin School of Computer Science and Engineering
The Hebrew University Jerusalem
+972547343534
dan.nacht@mail.huji.ac.il

ABSTRACT
In this paper, we describe the Smart Profile Management application that was designed to help minimize mobile phone disruptions. The system does this by making phone profile changes depending on the type and content of calendar entries it sees as well as an analysis of past usage history. A prototype was developed in Python for S60 and an informal usability test was run for a period of 7 days. Results show that a large proportion of profile suggestion changes were accepted by subjects hinting that such an application can in fact lower disruptions. The results also raise a number of issues that need to be dealt with before such an application can fulfill its intended use.

Categories and Subject Descriptors
[H.5.2 User Interfaces]: Interaction styles, User-centered design

General Terms
Algorithms, Design, Human Factors

Keywords
Mobile Phones, rule based systems, context based computing, smart interfaces

1. INTRODUCTION
The mobile phone has become the most ubiquitous user operated electronic device in history, with over 1 billion of them being sold in 2006 [6]. With the growing power of small footprint low power processors the functionality of mobile phones has grown to enable people to carry a phone, a full featured PDA, an audio recorder, a music player, a high resolution photo and video camera, a photo and video viewer, a powerful web browser and a computer gaming console in their pockets, all in one small device. Coupled with online contact and calendar management systems that the phone can interface with, people can have access to and manage most of their lives through the small window of their mobile phone. But this explosion of functionality and fusion of services comes at a price – people are finding themselves struggling with more complicated phone interfaces as well as frequent disruptions from incoming phones calls, SMS, MMS and WAP push messages, RSS feeds, email notifications and calendar alarms.

1.1 Mobile Phones as Disruptive Devices
As stated above, since the phone has become capable of being the hub of a person’s informational life, it has also become a device that too many a time disrupts a person in the middle of doing something. In the past this disruption came from incoming phone calls and preset alarms. But with all the information services that the mobile phone is interfaced to today, one can be disrupted more than ten times an hour1. If the user finds this useful then so be it - but in many cases people find these disruptions to be, well, annoying, and would be happy to minimize the phone’s potential for disruption in any way they can.

1.2 Profile Management on mobile phones
One useful feature that has developed over the years is a phone profile manager. This program allows a user to manage the behavior of the phone in specific situations. Standard profiles include general, outdoor, meeting, pager and silent. In each of these, the user can control how loud the phone will ring (from maximum loudness to silent), how many times it will ring (from none, to endless), the ring tone it will use when ringing and whether the phone will vibrate when ringing, in addition to second level characteristics. Some systems even allow a person to set a timer for a profile and have it revert to a previous profile when the time has passed. Although profile managers exist on many phones, people tend to use them infrequently or in a very simple fashion - for example manually switching to meeting or silent profiles. When manually configuring profiles, an altogether too frequent problem is forgetting to change the profile back to general or outdoor, with the subsequent problem of missing incoming phone calls.

1.3 Automatic Profile Changing:
Although we have not found any published research into such systems, there are a few commercial applications (Handy Profiles for S60 [5], Photo Contacts Pro 5 [9]) that can automatically switch profiles depending on a calendar entry, the time of day, or even the day itself. Although such programs go a long way to solving the manual profile setting problem, they rely on a simple model that matches time to profiles and cannot deal with special cases or learn from continued use by the user.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobileHCI 2008, September 2–5, 2008, Amsterdam, the Netherlands.
Copyright © 2008 ACM 978-1-59593-952-4/08/09 ...$5.00.

1 In our study a phone was synched to Google calendar which sent out SMS reminders. This was in addition to other incoming notifications from friends and email.
We identify three methods to develop such systems: Simple Middleware applications that switch profiles automatically (as used in the commercial examples above), Rule Based Systems where a system of configurable, flexible pre-defined rules changes the active profile, and Machine Learning Based Systems - the most sophisticated solution- where a machine learning algorithm is used to select the active profile. The application reported here uses a rule based system, whereas a future version we are working on will include a machine learning module.

2. THE SMART PROFILE MANAGER

Our smart profile manager application is part of the Mobile-Smarts project at Hebrew University Jerusalem. The Smart Profile Manager is a rule based application for Symbian S60 that watches the user’s calendar and offers to change profiles depending on the type of calendar entry that is starting. The system also looks at past usage history and can make profile change suggestions accordingly.

2.1 Technical Description

The program was developed with Python for S60 [8] on a Nokia N95. We decided to use Python to develop the application because it enables rapid prototyping of mobile applications. Since we are new in this area, we wanted to be able to iterate quickly and frequently. Having said that, there are some disadvantages such as a limited set of OS access commands, lack of event notification (i.e. our software cannot be triggered by OS events when a user adds appointments, changes the active profile, etc). Thus, the current version implements a polling system process that wakes up every predefined time interval and checks several parameters.

Designing our next major version, which will implement a machine learning algorithm, we had to characterize the parameters needed for the learning process. For that purpose we created a module which tracks all updates applied to the phone's calendar by the user (adding, removing or updating appointments). The program saves this data to a separate file, which will help us learn about usage patterns (e.g. special words used for appointment content, times of removing old appointments, etc).

2.1.1 Application Modules:

The software contains three main modules:

- **Profile Data Collector:** This module checks the active phone profile and documents changes to a profile data file.
- **Calendar Tracker:** This module checks what updates were applied to the calendar and documents them to calendar tracker data file.
- **Profile Advisor:** This module contains the rule based system. Using the data acquired by the other modules and the set of pre-defined rules, it changes the active phone profile. In initial stages, where the user is prompted to approve profile changes, this module documents the user responses to a log file.

<table>
<thead>
<tr>
<th>Context</th>
<th>Description</th>
<th>Use case example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of the Day</td>
<td>Changes related to the time of the day</td>
<td>Reverting from Silent to General mode if the user appears to have forgotten switching back after N hours</td>
</tr>
<tr>
<td>Phone’s Database</td>
<td>Events appearing in the phones calendar, such as appointments</td>
<td>Switching to Meeting when a meeting starts, and back to the previously active profile when it ends</td>
</tr>
<tr>
<td>Usage History</td>
<td>Following user behavioral patterns</td>
<td>Switching to a certain profile which the user constantly switched over a period of time (e.g. Outdoor every Monday at 10:00AM, and Pager everyday between 4:00PM and 4:30 PM)</td>
</tr>
<tr>
<td>Location (future version)</td>
<td>Connecting the user location with the active profile</td>
<td>Switching to a certain profile when the user is in a certain vicinity, either GPS or cell based</td>
</tr>
<tr>
<td>User Patterns (future version)</td>
<td>Learning special and exceptional usage patterns</td>
<td>Allowing ringer to bypass silent mode for a specific set of contacts</td>
</tr>
</tbody>
</table>

We developed a rule-based system which changes the active phone profile according to various parameters. Table 1 summarizes them.

2.1.2 Profile Change Rules:

2.1.2.1 General Rules:

The system switches to meeting profile whenever an appointment takes place, unless the appointment contains a word belonging to a pre-defined collection of special case words (such as "lecture", "seminar", "class", "lesson", "teaching" etc) – in which case the system switches to silent profile. If the appointment contains a word belonging to the exception words collection (such as "birthday", "wedding", "party", etc) – the system ignores it.

2.1.2.2 Usage History Tracking:

The system also keeps track of the user's habits. On top of the calendar database, it monitors the profile selection patterns of the user and predicts future profile changes based on previous behavior: a user who constantly activated the "silent" profile at 7pm on two consecutive Wednesdays, for example, will be advised to activate it again on the following Wednesday. A user who selected the "meeting" profile at 10am during most of the weekdays, will be advised to do activate it again the following day. Another rule will release the phone from a "silent" profile which has been active for more than a certain number of hours, given that no meeting exists in the calendar at that moment, and that the time is not between 8pm and 8am. Note that the parameters for the decision algorithm are configurable for all rules.

2.1.3 User Interface:

Before switching profiles the application asks permission to do so. User acceptance or rejection is saved to a log file for later analysis. An additional dialog enables the user to release the system from asking for approval in the future (see figure 1) and thus not disrupt the user further.
2.1.4 Data Acquisition:
As mentioned above, the program keeps track of the user's behavior via the file system. Besides the files which monitor the changes in active profile along time and the updates applied to the calendar, a log file tracks the user's responses to the profile change suggestions. This log contains the suggestion timestamp, suggested profile, suggestion context and user's reaction. This data allows us to analyze the various rules and how they relate to the actual user experience.

3. USABILITY STUDY
We ran an informal usability study in order to gauge initial user impressions and find usability problems. We configured three Nokia N95 phones with the application and asked the subjects to continuously use it for a period of 7 days. We also asked the subjects to move all their calendar management to Google calendar [3] which was then synched to the phone calendar using the free Goosync service [4]. We did this in order to simulate the use of office based meeting management services with the mobile phone user. After the test we analyzed the usage logs and ran informal interview with the subjects.

3.1 Results
Table 1: Application Suggestion Acceptance ratios

<table>
<thead>
<tr>
<th>Subject</th>
<th>Suggestions</th>
<th>Accept</th>
<th>Reject</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29</td>
<td>18</td>
<td>11</td>
<td>62.07%</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>25</td>
<td>0</td>
<td>100.00%</td>
</tr>
<tr>
<td>Total</td>
<td>54</td>
<td>43</td>
<td>11</td>
<td>79.63%</td>
</tr>
</tbody>
</table>

It is clear that a more formal and robust usability study must be run in order to be able to feel more confident about the results and the issues that are brought up. We plan on doing so with a future more capable iteration of the application. Within these constraints, we think that the data shows that users were open to using our application and believe it to be helpful. Notwithstanding the above, the trial raised a number of issues which need to be dealt with before such an application will be really useful.

4.1.1 Improve Application Dialog Interaction
One usability issue came up almost immediately had to do with the program’s dialog window that came up to ask users if they agreed to change the profile for the upcoming meeting. If another application came to the foreground before the user had time to react - for example the calendar meeting notification was presented- then the smart profile window disappeared into the background. This caused the program to wait without changing the profile, and to the user it seemed as if it had stopped working. We saw this as a program bug. A new version of the application fixes this problem by giving users a 3 second time window to answer the dialog, after which the default behavior will be to change the profile according to the application suggestion. Users will always be able to manually change the profile if they do not agree with the application generated profile.

4.1.2 Create a richer set of meeting categories
Although we used a list of special case words in order to enrich the sensitivity of the system to the different types of calendar entry types, we found that they were not enough for the real life use cases involved. In one case, a subject missed a series of calls because his phone went into meeting mode when he took his daughter to the dentist. Table 2 shows the calendar categories and their matching profiles. Category groups 1 through 3 were implemented in the version tested. Interviews with subjects suggest that we should add category groups 4 though 6 which enable more freedom to capture people’s real life behavior. Note that although we think that the user should be able to manually configure additional categories and match them to specific profiles, we believe that the application logic should be as complete as possible out of the box so as to necessitate a minimum amount of user management and setup effort.

<table>
<thead>
<tr>
<th>Meeting Category</th>
<th>Profile</th>
<th>In the Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Meeting</td>
<td>Meeting</td>
<td>Yes</td>
</tr>
<tr>
<td>2 Teaching, Recitation, Class, Seminar, Lesson</td>
<td>Silent</td>
<td>Yes</td>
</tr>
<tr>
<td>3 Birthday, Party, Wedding, Pub, Drinks</td>
<td>Do not change profile</td>
<td>Yes</td>
</tr>
<tr>
<td>4 Appointment (General with a weaker ring + vibration on)</td>
<td>Appointment</td>
<td>No</td>
</tr>
<tr>
<td>5 Errand, Store, Supermarket, Shopping</td>
<td>Outdoor</td>
<td>No</td>
</tr>
<tr>
<td>6 Movie</td>
<td>Silent</td>
<td>No</td>
</tr>
</tbody>
</table>
4.1.3 Dealing with special cases

The problem of missing phone calls because of automatic profile changes is significant and has led to special cases. For example, the system should be able to learn that if a user accepts a call from someone consistently, no matter what the regular profile was, then subsequent phone calls from that contact should cause the phone to “put them through” even if it is in meeting or silent mode. This will not solve all problems relating to missed calls when in these profiles, but will help to ensure that the ones seen as important will raise attention. Note that such special case treatment does not only have to occur through learning- a user can manually add a special case flag to a contact which will do the same thing. This is important since learning takes time.

4.1.4 Improve program integration with the Host operating system.

Even though the application was run as a system application, and was thus transparent to the user, there was one architectural problem which caused the application to perform less well that it might have. Because the application did not implement a system event watcher, the only way to run it pragmatically was to put it to sleep and have the OS wake it up every time period T. In order to minimize system resource allocation and battery usage we set T to 4 minutes. The problem with this is that in some cases the application awoke after a meeting had started. Although the application immediately offered to change the profile according to the rule base used, the user still got a feeling that the application lagged behind and was not robust enough. The new version of the program will use a smaller value for T (we will experiment with values starting at 30 seconds). In parallel we will implement an event watcher version which will awaken the program only when a specific event occurs, thus saving a lot of activity.

5. SUMMARY

The results hint that our program can help in reducing mobile phone disruptions since the suggestions it made were accepted in a wide majority of cases. A number of problems were identified. The major problem seems to be that a richer set of profile categories must be implemented in order to allow the application to serve its user’s real world needs and behaviors. The calendar is to be an important database of profile information, the system will learn to handle exceptional scenarios, such as bypassing silent mode when a certain contact calls.

Dealing with special cases:

• Enhancing the training process: a more sophisticated weight-based system will prompt the user to confirm profile changes until a certain confidence threshold is reached. From that point on, the profile changes will occur automatically.

• Fine tuning the system rules: testing the software with additional users will enable us to refine the existing rules. This will allow us, for example, to distinguish between a casual meeting and an appointment with the doctor.

• Location based Profiles: The system will be able to make profile changes relative to a user’s physical location.

7. ACKNOWLEDGMENTS

We would like to thank Paul Wisner and Nokia Research for providing us with a generous equipment grant.

8. REFERENCES


