Towards Alternative User Interfaces for Capturing and Managing Tasks with Mobile Devices

Harald Radi
Catalysts GmbH
Prager Straße 6
4040 Linz, Austria
harald.radi@catalysts.cc

Rene Mayrhofer
Vienna University
Institute for Distributed and Multimedia Systems
Universitätsstraße 10
1090 Wien, Austria
rene.mayrhofer@univie.ac.at

ABSTRACT

Mobile devices, per definition, are supposed to assist in organizing all kinds of things, also tasks of course, because usually such devices are always at hands. But due to the very limited and time consuming possibilities to interact with such devices many fall back to other means to organize their life, like a simple pencil and paper.

We developed a collaborative task repository that facilitates collaboration and teamwork, but on the other hand demands that all tasks have to be entered into that system. Therefore a smart and userfriendly interface to that repository is mandatory.

This work presents concepts on how to improve the user interface of mobile devices so that capturing tasks on-the-go becomes feasible. We propose to move away from display driven user interfaces to more sophisticated interfaces that utilize all the sensors and actors of current mobile devices.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems*; H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

General Terms

Mobile Computing

Keywords

Mobile UIs, Context, Task Repository.

1. INTRODUCTION

The amount of tasks, whether those are business or private tasks, one has to cope with in his daily life is ever increasing. Hence Mobile Digital Assistants (MDAs) and almost all available cell phones these days offer calendar and task management applications that aim to help the user to keep track of his activities. Those devices most often offer means to synchronize that information with either a desktop or a laptop computer, or a global repository that can be synced with other devices on its own. Nevertheless, it is still common practice to capture such information on an ordinary computer and use the synchronization capabilities to just push it to mobile devices in a one-way manner, the synchronization capabilities are by far not leveraged.

The reasons for this are rather obvious, due to the User Interface (UI) limitations such devices are currently just usable as a viewer. Entering or managing tasks is hardly possible with available UI technologies. Taking textual notes with a numeric key pad is laborious and those on-screen keyboards on MDAs with touch capabilities use up half of the screen and are still to small to type efficiently. Creating an appointment with all its details, which is a matter of seconds when sitting in front of a desktop computer, easily lasts several minutes on such a device. On the other hand, devices with a built-in keyboard are rare and still too big and heavy, so that it becomes questionable if they will be used in everyday situations.

But even without the UI barrier such devices can hardly fulfill an average users needs. Appointments are postponed, new tasks arrive, priorities are changing, meetings might be canceled and so forth. Such information is mostly kept locally on the device and usually not synchronized often enough to reflect the latest changes. Therefore the status on the mobile device is more often a proposed plan failing to be carried out than a dynamic representation of the current reality.

It is hard to cope with that for single individuals, but it becomes even worse when whole teams or projects need to be managed. The dynamics in such situations are much higher, it is not just the own information that potentially changes, but the information of all team members with all the possible impact and side effects on own tasks due to possible dependencies. The effect that is caused by not synchronizing often enough is reinforced by the fact that users do not enter their tasks immediately because of the above mentioned UI barrier. As a result even synchronizing more often would not increase the accuracy of the task list.

In systems theory, a system is a set of interacting or interdependent entities forming an integrated whole, and where the coupling adds additional functionality to that of the individual entities. That definition can be rephrased to: A system is limited by the limitations of the connections and dependencies of its entities if the performance is not restricted

MoMM 2008, November 24–26, 2008, Linz, Austria.

by individual entities.

If we look at a distributed team as a system the limits lie clearly in the communication, or the lack thereof to be more precise. The bigger the team the more effort is put into coordination which is an overhead which reduces the performance of the individual team members. When it is not transparent who is picking up which task such communicational overhead is produced to clarify that. Even worse, team members have to be interrupted for that clarification. To reduce or avoid such interrupts and to have an efficiently working team it is mandatory that everybody has immediate access to relevant information.

This work has the aim to identify such barriers and propose mobile UIs for a collaborative task repository.

A way to address those UI insufficiencies can be the use of recommender systems, that take proven knowledge derived from earlier tasks and propose reasonable parameters as defaults to the user. By correlating the output of the recommender system with further context information (e.g. location, time, current activity, etc.) the result should be a rather small amount of suggested options to choose from. Such support can drastically reduce the amount of time needed to capture specific information. This can be compared to the Amazon online store where the UI is just able to show a very small extract of the millions of articles that can be ordered, but most of the time the impression is that the shop better knows what one is interested in than oneself.

As mentioned above, a second way to improve current UIs is to incorporate context information. Unlike desktop computers, mobile devices have a much broader knowledge of their surroundings and as such the users context. Having that information available to a task management application allows a variety of new functionality. Certainly, tasks are related to the context they originate in. The more information the capturing device has available about that context the less the user needs to fill in. The application might already know time, location and participates once the action is taken to enter a new task during an ongoing meeting.

Context information can not just facilitate capturing of tasks but also improve the way the information is presented. Most of us also have to manage their private life which has to be balanced with the business schedule. Usually tasks with high priorities are conflicting, since they use up each others time. Private tasks can be important as well, nevertheless they do not conflict with business tasks, e.g. if they are concerning the recreation program for the weekend. An intelligent device could recognize the context it is in and switch the priorities on the fly. A business task can be very important the whole day during working hours, but once you line up at the groceries cash desk at 7:30 pm it is way more important to not forget the cheese.

Often enough appointment details of a task become relevant only short before it is due. It is sufficient to know that there is a meeting with a person at a certain company, the exact location details, like floor and room numbers, are not relevant till once you are there. If you look them up beforehand, they will most likely be forgotten again when actually needed. The MDA can prefetch such information based on the context and have it at hands once you need it.

The challenge is to add intelligent (ambient) devices to our lives in a way that supports our activities, complements our skills, and adds to our pleasure, convenience, and accomplishments, but not to our stress [7].

2. TASK MANAGEMENT

In our work a task represents as a small unit of work which should be described as short as possible. A task has different attributes such as its creator, its owner, a status, its due date, its priority, and obviously a description of the action that has to be performed to accomplish that task. A task can be part of a workflow, hence those attributes can be interdependent if the workflow requires that, e.g. changing the owner of a task will have an impact on its status. A task can have perquisites, which themselves are formulated as tasks, hence a task can have multiple dependencies even to other peoples tasks.

In order to maintain those dependencies we developed a collaborative task repository that is responsible for that coordination. Unlike conventional project management tools that concentrate on the planning phase without offering enough functionality for task tracking and later adjustment, our task repository aims to support the execution of project plans. By providing that support, a global repository has several advantages over individual task lists:

- During meetings and discussions usually everyone takes his own notes. When splitting up the outcome of the meeting into several tasks for the team, nobody can be sure if the others are aware of the same important details. Most often this results into further roundup meetings or discussions in smaller groups where again not all are attending. A task repository that allows a collaborative definition of tasks removes this shortcoming as everyone is involved in the final definition of a task.
- During the realization of a project all team members have immediate access to the status of all relevant tasks. Problems and bottlenecks are made transparent for the team and hence can be identified earlier. Team members with lower utilization can help out in other areas to reduce bottlenecks.
- A global repository has a much broader knowledge of the ongoings than it would have if it were just managing an individual task list. Hence it can leverage that knowledge to act as a support system by providing meaningful hints and suggestions for possible shortenings.

On the other hand, a global repository requests much more discipline from the users:

1. Information must be entered as soon as it becomes available. If important information, like a changed priority, is not entered immediately, others cannot benefit from it and continue their work based on the old facts.
2. All available information must be entered. If the user leaves out essential information, because of privacy concerns or for other reasons, neither the support system nor other users will be able to support the regarding user well.
3. Information entered must be accurate and self contained. If it is not accurate others might make wrong assumptions based on the available information, and if the task description is not self contained others, who might not see all the prerequisite tasks, will not be able to know what to do.

4. Individual tasks have to be as small as possible. Small tasks reduce the time needed for accomplishment and hence improve the response time. Repriorizing and scheduling small tasks is much easier than organizing big chunks of work, that has a positive impact on ones throughput. It is also vital for the support system to have a detailed picture of the tasks and their dependencies, which is not the case if only work packages are captured, that contain several individual tasks.

We propose that 80%[1] of all tasks originate on-the-go, that is during driving, business lunchs, sports, telephone calls, or even at night. Another 18% originate in meetings and discussions and only the remaining 2% are actually created in situations where conventional input devices, such as a mouse and a keyboard, are available.

Since only few people are such dutiful to satisfy the above mentioned requirements, the process of acquiring that information has to be as barrier free as possible. Consequently we need to be able to capture tasks in mobile environments where existing UIs are not sufficient.

3. MOBILE INPUT UIs

As stated above, there is a huge demand for improving the UI capabilities of current mobile devices. The iPhone would not be such a hype if there were no demand for that, nevertheless it does not accelerate the operation of basic cellphone functionality. The opposite seems to be the case, due to the trend to make such devices as small as possible it becomes even harder to use them.

In current applications tasks are not captured when and where they originate, but are mostly forwarded by mail or a phone call, written on meeting protocols or even post-its. Later on those tasks are re-entered into task management systems on a computer with a keyboard. This barrier more or less ensures that ad-hoc tasks never find their way into such a system since the time overhead is way too big and the user thinks twice before entering the task. Unfortunately, that fact is the reason why most users violate all four of the above mentioned requirements for our global task repository.

1. Information is not entered as soon as it becomes available. It is easier to do it on the desktop computer when back in the office, hence people do so.

2. Seemingly unimportant facts might not be entered. Since it takes minutes to enter a task only the really important facts are entered.

3. Information entered later is never as accurate. People tend to forget facts if they do not write them down immediately.

4. The information will be entered as work packages. As people enter their tasks later, they automatically start to aggregate some of them into bigger chunks of work.

These barriers will cause the idea of a collaborative task repository to fail, therefore this work concentrates on eliminating the UI barriers that cause users not acting as intended. From a users perspective, the constraint is the input speed that is limited by the size of the device and the means to enter textual information. As we do not expect any significant improvements in that area the only way to durable increase efficiency is to get away from the display driven approach of current UIs [4].

[1]This data is based on a survey of Catalysts GmbH

Besides a display, a keypad or a touch screen, state of the art mobile devices provide a lot of other options to interact with their environment:

- Microphone
- Speaker
- Camera
- Tilt sensors
- Accelerometer
- Vibrotactile output
- Bluetooth
- Wireless Local Area Network (IEEE 802.11) (WLAN)
- Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS)
- Mobile network access, High Speed Downlink Packet Access (HSDPA)
- Global Positioning System (GPS)
- Less obtrusive feedback via LEDs

In our previous work we described a framework for context recognition [6, 10] that utilizes those sensors to learn and distinguish different contexts. Since tasks always belong to a certain context, we can pick up on that work and use that context information to enrich the task definition.

For example, if one defines a task during a meeting the device knows, that one is in the meeting room. The device can be able to identify the participants by scanning for other devices in that room. It knows the purpose of the meeting as well by looking at the meetings appointment definition, if available. With that information the device is able to derive and suggest reasonable defaults for that newly created task, such as the project it belongs to, that it is not a private task in this particular case. It can also suggest a list of people (the attendees) as potential owners for that task.

As already expressed in our previous work the device must not make any assumptions or trigger any actions based on that information. It may only make suggestions and at best, it can set reasonable defaults and reorder option lists so that more likely suitable values are listed first (e.g. in a list of people, those who can be identified in your surrounding can be on top of the list, just like the online/offline contacts list in an instant messenger).

Smart machines of the future should not try to read the minds of the people with whom they interact, either to infer their motives or to predict their next actions. The problem with doing this is twofold: first, they probably will be wrong; second, doing this makes the machine's actions unpredictable. [...] it is far better for the machine to behave predictably and let the person respond appropriately [7].

Oakley et al. [8] presented navigation concepts that incorporate the tilt of the device. The combination of pressing buttons and tilting enormously increases the possibilities of selection and navigation.

In meeting situations it will be common that not just a single person interacts with the device, but the device might be positioned at the conference desk or lunch table and thus needs to interact with all participants. In such situations the device needs to be able to distinguish between the participants by identifying them based on their voice or eventually by certain gestures. Hazen et al. [2, 9, 1] proposed ways for person and language identification on mobile devices. By being able to distinguish between different persons tasks can be defined collaboratively with a single device, but automatically assigned to the identified creator.

Besides the input, also the presentation of information can be optimized with a certain context knowledge. The way how feedback is given to the user may vary from unobtrusive to rather noisy depending on the current context. During meetings something important can be signaled by a tiny flashing light while it might trigger a loud ring if the user is outdoors. In any way, the user must be able to define which kind of feedback is appropriate in which situation and which kind of interruptions are acceptable.

By the use of positioning services, such as GPS or location fingerprinting [3, 5], it becomes feasible to define notifications and priority changes based on spatial data. That way, ones shopping list can automatically become due once the mall is entered. The support system can assist in rescheduling tasks based on the current location.

Further more the accelerometer together with the tilt sensor can give information whether the user is actively using the device or not. Presenting information on the screen is useless when the device is in the users pocket or lying on the table upside down. On the other hand heavy vibrotactile feedback might be annoying if the user tries to write something with the stylus.

Unfortunately, mobile devices have very limited battery capacity, hence they have advanced power saving strategies. That means that we cannot access all the sensors all the time, either because the user completely turned off some, or because not all sensors are available on all kinds of devices, or at least because the device would run out of battery in minutes if we use all of them all the time. One one hand, that means that we carefully have to decide which context information is needed and only then access the respective sensor. On the other hand, that means that not all sensors will be available all the time and that we need fallback strategies to guarantee a minimal set of functionality.

4. CONCLUSION AND FUTURE WORK

We developed a collaborative task repository that aims to facilitate communication and collaboration in teams. Assignments in that repository are formulated in form of tasks with interdependent attributes and possible dependencies on other tasks.

This work focuses on the improvement of UIs on mobile devices so that tasks can be captured immediately when they are identified, on-the-go. If we manage to bypass the shortcomings of current mobile UIs, such devices even offer benefits over desktop or laptop computers. With the use of a context recognition framework, tilt sensor and accelerometer, location services and other sensors most of the issues with current UIs can be eliminated.

In future works we will incorporate those concepts into our existing task management application and we will concentrate on user studies that measure the time benefit gained by those new UI concepts compared to conventional approaches. Instead of investigating further methods to recognize context we will use the framework from our previous work and concentrate mainly on the user acceptance and the actual gain in real world applications.

We are also working on concepts how to determine the relevant context information for certain decisions and how to address the respective sensors then. That is needed to maintain the devices power management capabilities. Power consumption and hence battery lifetime is the most limiting factor in our current prototypes.

Further more, we are also aware that such a collaborative task repository brings up privacy issues and bears some risks. We therefore have to analyze how a collaborative task repository interferes with a users perception of privacy. If misused, such a repository might depict sensitive information of users or allow conclusions based on the available information. On the other hand, if users tend to leave out their private details the support system cannot work as expected and might do more harm than good.

5. ACKNOWLEDGMENTS

We want to thank all at Catalysts GmbH for their thorough support and the funding of this work.

6. REFERENCES

[1] T. Hazen, E. Weinstein, and A. Park. Towards robust person recognition on handheld devices using face and speaker identification technologies, 2003.

[2] T. J. Hazen and V. Zue. Segment-based automatic language identification. *Journal of the Acoustical Society of America*, 18(4):2323–2331, 1997.

[3] J. Hightower and G. Borriella. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.

[4] K. Hinckley, J. S. Pierce, M. Sinclair, and E. Horvitz. Sensing techniques for mobile interaction. In *UIST*, pages 91–100, 2000.

[5] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting.

[6] R. Mayrhofer, H. Radi, and A. Ferscha. Recognizing and predicting context by learning from user behavior. *Radiomatics: Journal of Communication Engineering, special issue on Advances in Mobile Multimedia*, 1(1), May 2004.

[7] D. A. Norman. *The Design of Future Things*. Basic Books, December 2007.

[8] I. Oakley, J. Angesleva, S. Hughes, and S. O'Modhrain. Tilt and feel: Scrolling with vibrotactile display, 2004.

[9] A. Park and T. Hazen. Asr dependent techniques for speaker identification, 2002.

[10] H. Radi. Adding smartness to mobile devices – recognizing context by learning from user habits. Master's thesis, Institut für Pervasive Computing, Johannes Kepler Universität, A-4040 Linz, Altenbergerstraße 69, 2005.