Project Development and Administration 2010/2011

The comparison of the explicit and the implicit way of Human-Computer-Interaction on a Smartphone

January 05, 2011

Christoph Pohl
Dresden University of Technology

Abstract
Today nearly every User Interface is created to interact explicit with the user. However, there are some research projects and different approaches trying to integrate implicit Human-Computer-Interaction into systems like smartphones. This project will compare the modern, explicit way of Human-Computer-Interaction with the upcoming implicit way under certain criteria, finding the solution that it is not the time right now to do the shift towards implicit User Interfaces. Therefore, necessary sensors like GPS or the gyroscope, which are used to describe the context a user is in, will be defined. Afterwards, it will be discussed with which pros and cons each way of interaction can come up with. Finally, a concrete recommendation is given based on the findings of the previous discussion.

Table of Contents

The comparison of the explicit and the implicit way of Human-Computer-Interaction on a Smartphone

1. Introduction

The way how people interacted with computers changed in the past and we are now near another shift in Human-Computer-Interaction (HCI). In former times there was a shift from punch cards to interactive text terminals and also a shift from the command line to graphical user interfaces (GUI). With upcoming and even current technologies – providing more processing power and the availability of many sensors – we are facing a new shift from explicit HCI towards a more implicit interaction with smart devices. With an implicit Human-Computer-Interaction, which not only presents the raw data of the built-in sensors, but also extracts and combines that information, there will rise a whole new way in mobile computing.

1.1 Background

Nowadays, the interaction between the human and the computer is explicit. That means the user tells the computer in a certain level of abstraction what he expects the computer to do. This usually happens by entering a command in the command-line or by direct manipulation using a GUI or speech input for example. In contrast to this, a human-to-human communication is based on body language, voice, and gestures. Furthermore, the surroundings and the behaviour of the participants is a determining factor for understanding the messages. In a conversation between two computer-scientists pointing at a computer the word "mouse" has a different meaning than the same term being used by two exterminators.

Implicit Human-Computer-Interaction only works if the computer has a certain understanding of how a human being behaves in certain situations. This knowledge can be seen as an additional input while doing a certain task. As seen from the example of the word "mouse" we can affiliate the two main concepts of implicit interaction and their challenge:

- The ability to have *perception* of the use, environment and circumstances
- A mechanism for the *interpretation* what the sensors see, feel and hear

In addition to those two concepts there is the need of an application that can make use of this information [1].

This information about the location, surrounding environment or state of the device is called "context". To build applications that have knowledge about their context it is important to gain an understanding what this exactly means. As already mentioned, there is a strong focus on location because this concept is well understood. Furthermore the benefit of location-awareness is clear: particular services are more important than others at a certain location or in a certain situation.

Analysing the way how people use their ultra-mobile devices (like phones or PDAs), it becomes clear that the periods of interaction are much shorter than in the traditional mobile environment. Notebooks for example – which are considered to be mobile computers – are mainly used in a stationary setting like a meeting, which could last several hours. In contrast to that, looking up an address in your smartphone only takes a few seconds. In addition these ultra-mobile devices are often used while doing something else. All these facts call for a reduction of the explicit HCI and create the need to shift towards the implicit way.

The following two figures will draw a concrete line between the explicit and implicit way of Human-Computer-Interaction:

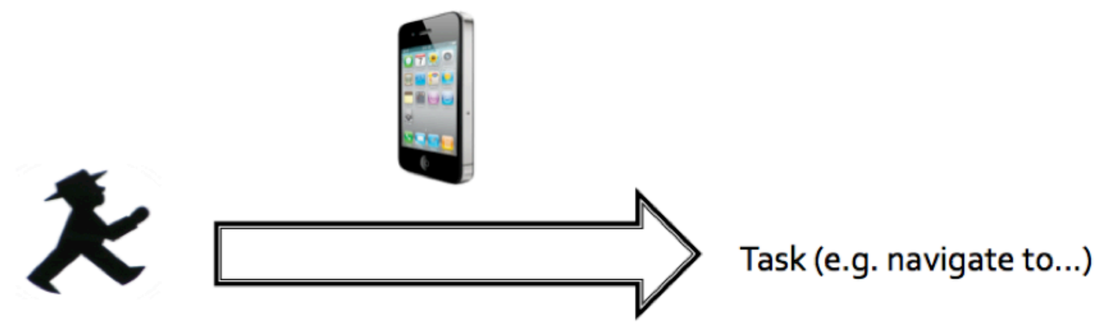

Figure 1

Figure 1 shows the normal, explicit way how people interact with their smartphones. As seen in this picture the user gives his smartphone a concrete command that is directly executed by his device. In contrast to the implicit way – which is denoted in Figure 2 – the smartphone only does what the user wants it to do.

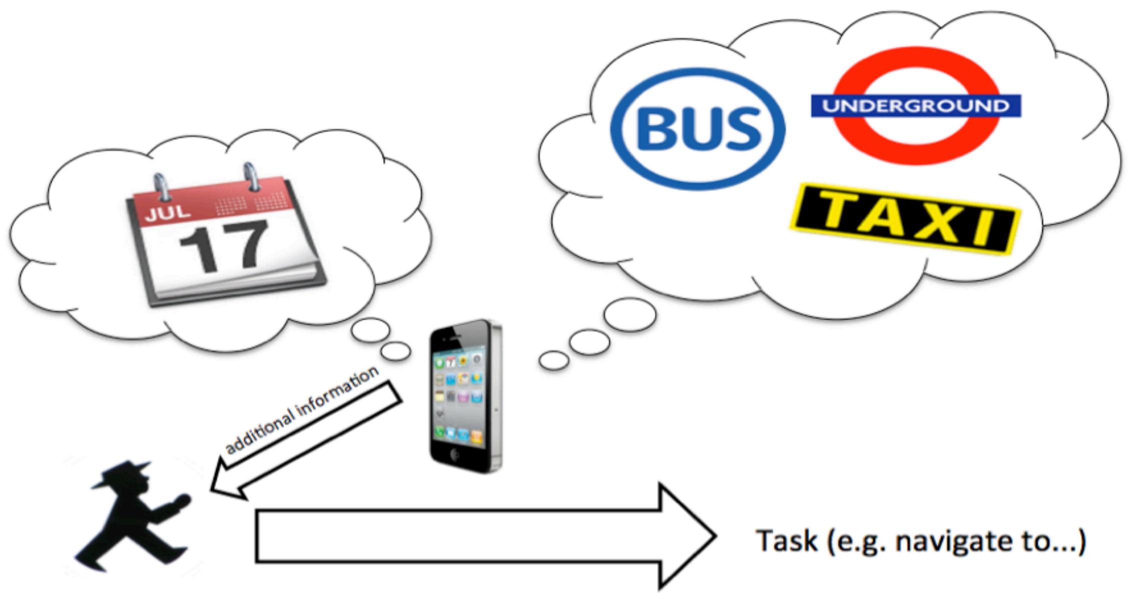

Figure 2

Figure 2 demonstrates the iHCI (implicit Human-Computer-Interaction). The basic idea remains the same but in addition the smartphone starts “thinking” why it gets this specific task (e.g. navigate to…). With the help of many sensors and the ubiquitous world-wide-web the device has the ability to decide whether it is necessary to take the bus or underground for example, or it could provide the user with additional information about the sights he navigates to, if he wants to do sightseeing; again this information could be read from the calendar.

1.2 Purpose

The global market of smart devices (especially smartphones) is in a decisive change since 2007. The iPhone 2G was the first phone which was completely controlled via multi-touch and without any digital pen. From this moment on the fusion of normal phones with PDAs (Personal Digital Assistants) began and the so-called smartphones should came within reach every moment. Until now there were released three more generations of the iPhone. The iPhone 4 has very high computing power and a huge amount of built-in sensors like GPS or a gyroscope. It is possible to gather much information out of the raw data of each sensor with the help of many *different* applications and the explicit way of Human-Computer-Interaction (eHCI). And even because of the fact that there are so many information-fragments, which are provided by all those sensors, it becomes more and more necessary to find a way to present all the information in a well-arranged way. This is exactly the point where iHCI comes in. Due to the fact, that smartphones constantly get more market share and functionalities without becoming bigger they will assist nearly every person in the near future. It is obvious that no one wants to be overwhelmed by a confusing and overloaded mass of functions; that is why it is necessary to analyse the implicit way of Human-Computer-Interaction. The aim of this project is to compare the two ways of HCI and to find a solution whether or not it is time right now to do the shift from explicit to implicit Human-Computer-Interaction.

1.3 Scope
At first there will be given an overview of the built-in sensors of the iPhone 4 which can be used to locate the user and provide additional context-aware information. These sensors are:

- GPS
- Gyroscope
- Bluetooth
- Wi-Fi
- 3G / UMTS

After that the two forms of interaction (explicit and implicit) between a computer and a human are compared under the criteria usability, costs of implementation (e.g. what is needed to provide this way of interaction) and additional benefits for the user.

This project does not contain any specific statistical-data like “the percentage of implicit graphical user interfaces (GUIs) on the market”, because there are no systems *on the market* yet that implemented the implicit way of HCI. However, some case scenarios with associated programs will show what could be possible at the moment. At the end a recommendation is given which form of interaction will be the best in the near future in order to provide the accurate start-up for further research and development.

2. Materials & Methods
The following paragraphs describe location-based sensors of common smartphones like the iPhone 4. Furthermore a short overview of additional modules, which are necessary to gain access to information about the context of the user, is given.

2.1 GPS-Sensor
The Global Positioning System (GPS) is a space-based global navigation satellite system that provides reliable location and time information. The BCM4750 GPS-Receiver is a single-chip solution (<35 mm² PCB area) and the most common built-in sensor for smart devices. In cooperation with other modules like Wi-Fi or UMTS it is possible to gather meteorological data or traffic reports. Therefore GPS plays a major role at dynamical contexts like navigation or weather.

2.2 Gyroscope
Beside GPS-Sensors, which are very common in the current smartphone generation, there are some exceptional modules like the L3G4200D-Motion Sensor. With the help of this digital gyroscope it is possible to detect movements like rotation or inclination. This sensor was created to provide a whole new gaming experience. Beyond that it could be used to recognize

whether the user is walking, standing, running or bicycling which helps to interpret the context the user is in.

2.3 Bluetooth
Bluetooth is a proprietary open wireless technology standard for exchanging data over short distances using short wavelength radio transmissions. It is a packet-based protocol with a master-slave[1] structure. Until now three versions were released and the newest one (Version 3.0 + HS) comes up with a data rate up to 24 Mbit/s. Apart from the simple data exchange feature, Bluetooth in smartphones is often used to provide a headset feature for the user.

2.4 WLAN and UMTS
A wireless local area network (WLAN) links two or more devices using some wireless distribution method providing a connection through an access point to the wider Internet. In contrast to that, UMTS (Universal Mobile Telecommunications System) is directly connected to the world-wide-web and mainly used for verbal communication. The third-generation (3G) of the mobile telecommunications technologies supports maximum theoretical data transfer rates of 45 Mbit/s, whereas WLAN supports transfer rates up to 120 Mbit/s (IEEE 802.11n standard assumed).

3. Results & Discussion
The following section presents the findings of the concrete comparison of the explicit way of Human-Computer-Interaction with the implicit way. The following three criteria were chosen to facilitate the decision which way of interaction will be the best in the future for smartphones:

- Usability
- Costs of Implementation
- Benefits for the users

In order to remain on a practical viewpoint two concrete projects (one for the usability- and one for the implementation part) will be named and explained to support the idea behind the iHCI. In every subsection first eHCI will be reviewed followed by iHCI.

3.1 Usability
According to Stefan Posland the explicit Human-Computer-Interaction puts the user at the centre of the interactive systems, so that the control of the system is driven externally by the user [3]. Poorly designed UIs (User Interfaces) can lead to customer dissatisfaction because they are more or less passive and do not adapt to the user abilities or physiological limits of them. Most developers are trying to avoid this effect by developing flexible applications.
An explicit application on a smartphone works deterministic. That means that one action performed on a device will cause the exact effect every time.
A disadvantage can be discovered when the user wants to get different information that may rely on the same problem e.g. driving abroad. Therefore he may wants to check the weather and the actual traffic on the streets. An intelligent User Interface would provide all the related information in one application, but using explicit HCI the user have to look in many different applications.

Compared to the usability of the explicit way of HCI, the usability of iHCI can differ in a certain nerving way. The following paragraph will give a short overview of the "Car Accident Detection Hosted Service" [5] followed by a usability weak point:

[1] Master-slave is a model of communication where one device or process has unidirectional control over one or more other devices.

In Spain a prototype was developed detecting potential episodes or accidents and making emergency calls automatically. Therefore Bluetooth was used to declare whether a user is in a car or not using his handsfree-set-feature of his smartphone. GPS and the gyroscope, which acted as a speed-indicator, were working together to detect the following scenario to support the user:
Every time a sudden and dramatic decrease of speed occurs the driver will be asked whether everything is all right or not. If he is answering 'no' or do not answer for a certain period of time the smartphone will make an emergency call instantly. But what happens if the user is in a traffic jam and has to accelerate and decelerate the whole time? Every time the user would apply the brakes he would be asked whether he made an accident. This problem is a huge lack of usability.

3.2 Costs of implementation

Before a whole GUI (Graphical User Interface) exists the developers have to program it and this progress is meant by the term "costs of implementation". As mentioned before the UI has to have a certain standard according to the usability, otherwise the average user is not able to interact with the device in a comfortable way.

S. Poland says, that modern explicit interfaces have to follow three main principles [3]:

- generalizability,
- flexibility and
- robustness

1. *Generalizability* refers to the interface's capability to allow simple interactions in a new application based on the knowledge of similar situations in other applications (e.g. the action to 'save' something).
2. *Flexibility* can be understood as the ability to adjust or adapt the interface to the user's needs or abilities. Some applications on smartphones for example ask the user at the first start-up whether he is familiar with the program or not. And according to his decision the application provides him all buttons and activities or not. This could lead into a problem: If a user searches for a feature that should be included according to a friend, but he cannot find it because the application hides it, the user will become dissatisfied sooner or later.
3. The user should be able to recover from errors he made on his smart device when he detects them. This feature is called *robustness*. It allows the user to come back from an erroneous interaction path and to go another way to complete his task.

Implicit User Interfaces have to face a completely other challenge: *context*. As said before context can be understood as location, surrounding environment or state of the device. This paragraph will concentrate on a more exact definition of what context is, in order to show how complex it is to implement a program that could interpret this context in whole.
Context can be split in five parts:

1. Who. Every computer system should focus the interaction on a particular user regardless of others in the same environment. According to a smartphone this should not be a big problem and could be managed by using a password. Furthermore, it is obvious that just one person is using his/her smartphone.
2. What. A smart UI should know what a user wants to do. Because of the fact that a user just uses a particular application of his smartphone, it should be obvious what he wants to do.
3. Where. At this point it becomes a little bit more difficult but this question can be answered with the help of different sensors a smartphone is allocating. The GPS-Sensor for example knows the location of the user. The Bluetooth module could have the information whether or not the user is in a car (using his hands-free set). If the

Wi-Fi module is activated and connected to an access point the smartphone knows that the user resides inside a building for example. All those sensors help to understand where the user is at the moment and can give a rough draft about the environment the user is in.

4. When. This information can easily be read from the intern system clock. However, a more interesting fact on time is '*how long*'. As an example, if a user spends very little time on a particular picture, maybe he/she is not interested in what is displayed according to the slide show he/she looks at in that moment.
5. Why. This question is the most difficult in the field of implicit Human-Computer-Interaction. Understanding why people perform actions is more and more difficult than understanding what an action means. One starting point is the use of other context information like body temperature or heartbeat to obtain information about the emotional status of the user. Due to the fact, that there are no sensors for smartphones at the moment, another way has to be found to answer this question. This problem is really complex and could maybe solved by creating a whole new operating system that is build to act implicit. The point is that every application on the phone has to work with each other. To come back to the example from the beginning of this report (implicit navigation to a certain point), the smartphone navigation application would have to interact with the calendar to get to know whether the user wants to do sightseeing, if he navigates to a special building.

The following paragraph will give a concrete example of how these 5 parts could be put together in one application, which supports a special group of users in their everyday life:
The UVa Bus.NET test bed [2], which was developed at the University of Virginia, enables to determine, accommodate, and predict context in mobile applications. The general idea is that students (or professors) often have difficulty meeting their next classes when they rely on the unpunctual campus bus system. This system automatically alerts the users of appropriate arriving buses for their upcoming appointments. The context is drawn from the user's current location via GPS, the user's appointments via a calendar application, the local bus system via GPS enabled buses, and local road information with the help of digital maps. Everything is managed by web services. That is why the user does not have to know the local bus system nor perform any additional decision-making besides using the application.

3.3 Benefits for the users

With this knowledge about the usability and the key aspects of the explicit and implicit way of Human-Computer-Interaction some benefits for each way of HCI can be found.
As mentioned before, the aspect of adaptation is a key feature of the eHCI. The user can benefit from it because such a User Interface has the ability to learn how the user wants to interact with the device and can add or remove features according to the users needs or abilities.
In contrast to this, implicit UI can provide the user with additional context-aware information in order to support the user. Furthermore, it is more invisible – or better to say integrated – in the everyday life, due to the fact that such devices "are aware" of the context in which the user is acting.

4. Conclusion

In conclusion, it can be said that both ways of Human-Computer-Interaction do have their pros and cons. However, iHCI is harder to implement in a system than eHCI. This results from the problem of understanding the context. There can be answered a lot of questions related to the location or time at the moment with the help of different sensors, but the key questions *why* a user performs this specific action is still difficult to answer for smart devices like a smartphone. Nevertheless, there are many research projects and different approaches to face this problem.

5. Recommendations

A clear recommendation to implement a solid, explicit User Interface with specific selected implicit aspects of HCI can be given. It is definitely not the time right now to do the hard shift from explicit Human-Computer-Interaction to an implicit one. There is still too much work to do, especially on the research of making the device understand what context is. Finding all these aspects related to the problem context, it can be concluded, that creating just one application that interacts completely implicit with the user is almost impossible. Therefore it would be necessary to create a whole new operating system that is very expensive and time-consuming.

On the other hand a stable explicit User Interface should be created which is able to learn from the users needs. It should also be able to adapt certain graphical elements according to the users abilities. In addition, some easy to implement but very useful features based on iHCI should be embedded.

6. References

[1] A. Schmidt (25 June 2000). Implicit Human Computer Interaction Through Context. Telecooperation Office (TecO), University of Karlsruhe. Springer-Verlag London Ltd.

[2] David Chu, Clement Song, Bei Zhang, Marty Humphey (2004). UVa Bus.NET: Enhancing User Experiences on Smart Devices through Context-Aware Computing. Department of Computer Science. University of Virginia.

[3] Poslad, Stefan. Ubiquitous Computing: Smart Devices, Environments and Interactions. Chichester, U.K.: Wiley, 2009. Print.

[4] Genco, A., and S. Sorce. Pervasive Systems and Ubiquitous Computing. 1st ed. Vol. 1. WIT, 2010. Print.

[5] Alejandro Cadenas, José María González, O. M. S.: Context-Aware Processing Of Mobile Device Sensor Information: Car Accident Detection Hosted Service.

Collocation List

1. to interact with computers
2. Human-Computer-Interaction
3. current technologies
4. explicit and implicit
5. processing power
6. data of built-in sensors
7. mobile / ubiquitous computing
8. to combine information
9. to understand the context
10. to provide context-aware information
11. to use ultra-mobile devices
12. to play a major role
13. a flexible application
14. to face a challenge
15. to support the user in his everyday life
16. to adapt graphical elements
17. key aspects of usability
18. to implement a feature
19. to detect a scenario
20. to gain knowledge
21. data rate
22. to act as a speed-indicator