Implicit Human Computer Interaction Through Context

Albrecht Schmidt
Telecooperation Office (TecO),
University of Karlsruhe
Germany
albrecht@teco.edu

1 Introduction

Analyzing the way people use ultra-mobile devices (personal digital assistants, smart mobile phones, handheld and wearable computers) it becomes apparent that the periods of interaction are much shorter than in traditional mobile settings. Notebook computers are mainly used in a temporary stationary setting, e.g. one takes a notebook to a meeting and takes note and a salesman takes a mobile computer to a customer for a presentation. In these scenarios the time the application is used temporary stationary is mainly between several minutes and hours. Whereas considering the usage of ultra-mobile devices interaction periods are often much shorter e.g. looking up an address takes only a few seconds and making a note on a PDA is often in the range of several seconds up to some minutes. This implies that the time to set up an application must be significantly smaller than traditional mobile systems. Also the fact that the applications are mainly used while doing something else or to fulfill a certain task (like tools in the real world) creates for a need reduction of the explicit human-machine interaction. This creates the need to shift from explicit towards implicit HCI.

2 Context

We propose to reagard situational context, such as location or state of the device, as additional input to the system. With situational context, the interaction process can be simplified because the system knows more about the user and their context; this concept, with respect to informational context is widely spread in standard applications (e.g. context menu). For devices that are used in different real world situations (e.g. at home, in the car, in the office, etc.) we suggest to extent the notion of context to include the information about the real world environment.

2.1 What is Context

To build application that have knowledge about their situational context it is important to gain an understanding what context is. Current research in context-awareness shows a strong focus in location [1], [6]. An architectural approach, based on a smart environment is described by Schilit et. al. [13]. Other scenarios are using GPS and RF to determine the users location, e.g. [4], [11]. But, as pointed out in [14] context is more than location. We use the term *context* in a more general way, as also suggested by [2], to describe the environment, situation, state, surroundings, task, and so on. Context is used with a number of different meanings; this is illustrated by the following definitions:

> ***Context** n 1: discourse that surrounds a language unit and helps to determine its interpretation [syn: linguistic context, context of use] 2: the set of facts or circumstances that surround a situation or event; "the historical context" (Source: WordNet ® 1.6)*
>
> ***Context**: That which surrounds, and gives meaning to, something else.*
> *(Source: The Free On-line Dictionary of Computing)*
>
> ***Synonyms Context:** Circumstance, situation, phase, position, posture, attitude, place, point; terms; regime; footing, standing, status, occasion, surroundings, environment, location, dependence. (Source: www.thesaurus.com)*

2.2 Applications in Context

Knowledge about the context is of primary interest to the application, because we consider that the application will adapt to the context. Therefore our approach is to look at the context from the point of view of the application.

The observation that an application is:
(a) running on a specific device (e.g. input system, screen size, network access, portability, etc.),
(b) at a certain time (absolute time -9:34 p.m., class of time - in the morning)
(c) used by one or more users (concurrently or sequentially),
(d) in a certain physical environment (absolute location, type of location, conditions such as light, audio, and temperature, infrastructure, etc.),
(e) social settings (people co-located and social role),
(f) to solve a particular task (single task, group of tasks, or a general goal)
holds for mobile and stationary settings alike. We consider the items (a) to (f) as the basic dimensions of location. For mobile applications especially (d) and (e) are of major interest. In mobile settings the physical environment even can changes while application is executed (e.g. making a phone call while walking from the office desk to the car park).

2.3 Specifying Context

To specify applications that use context it is inevitable to have a specification language to describe contexts linked with events/change that occur in the application in these contexts. In our recent work we found it helpful to use a notation that is human readable as well as easily to process using a computer. We decided to use a markup language that is specified in XML for this purpose. Extending the SGML based description model introduced by Brown in [2], [3] we added two further concepts - grouping context with matching attributes and trigger attributes to make the description more expressive and suitable for our projects. Depending on the platform (e.g. context sensing module in a microcontroller) we use a different implementation language.

If contexts are composed of a number of components we found it very helpful to have a mechanism to bundle certain contextual variables in groups and select a matching semantic for each group description. For matching in a group we provided the following semantics: `one` (match one or more of the variables in the following group), `all` (match all variables in the following group), `none` (=match none of the variables in the following group).

```
<context>
   <group match=one>
      <touch>
      <on>
   </group>
   <group match=none>
      <alone>
      <pen_down>
   </group>
</context>
<action trigger=enter time=3s>
   <confidential>
</action>
```

Example 1: Context description

We discriminate three different triggers: enter a context, leave a context, and while in a context. The enter and leave triggers take a time value that specifies the time after[1] which the action is trigger if the context stays stable over this time. For the while in a context trigger the time indicates the interval in which the trigger is fired again.

In example 1 a description of a context and a action is shown. The context description consists of two groups of contextual variables. In the first group the match semantic is that at least one of the variables must be true, in this case either the device is touched or the state of the device is on. In the second group the match semantic is none, which means that the contextual variable `<alone>` must not be true and that the user must not have put the pen down on the screen.

[1] The parameter indicating the time after that an action is performed is often 0 (immediate context action coupling) or positive. In certain circumstances, when future situations can be predicted (e.g. you drive your car into the parking, the context walking will appear soon) a negative value does make sense, too.

If the context evaluates to true an action is triggered. Here the semantic is that if the context is entered and is stable for at least three seconds the action is performed.

The complete description means that if the device is on or in the users hand and if the user is not alone and he has not the pen on the screen then after three seconds the display should be hidden by a images as depicted in figure 2 (d).

2.4 Sensing Contexts

There are several ways to sense context. We consider the following four basic approaches as most important:

- device-databases (e.g. calendar, todo-lists, addresses, profile, etc.)
- input to the application running (notepad - taking notes, calendar - looking up a date, etc.)
- active environments (active badges [10], IR-networks, etc.)
- sensing context using sensors (TEA [5], Sensor Badges [1], GPS, etc.)

In this paper we concentrate on the last case, knowing that in most scenarios a combination of all four cases is the way of choice.

Our approach is to collect data on the situational context by using low level sensors. In this project we built a context recognition device equipped with a light sensor, acceleration sensor, a passive infrared sensor, a touch sensor, and a temperature sensor. All sensors, but the touch sensor are standard sensors and produce analog voltage level. The touch sensor recognizes the human body as a capacitor and supplies a digital value. The heard of the device is a *BASICTiger* microcontroller that reads from all the physical input channels (it offers four analog digital converters and a number of digital IOs) and statistical methods are applied to recognizes contexts. The board is depicted in figure 1.

The communication between PDA and the device is physically realized using a serial line connection running at 19200 bits/s. The communication protocol is a request-replay protocol. Each time the application likes to have an update on the contextual information (usually while the application is idle, e.g. catching the `NullEvent`) it sents a `GET`-request to the context-awareness device. The device then replies with string containing the encoded current contextual information that can be processed by the application.

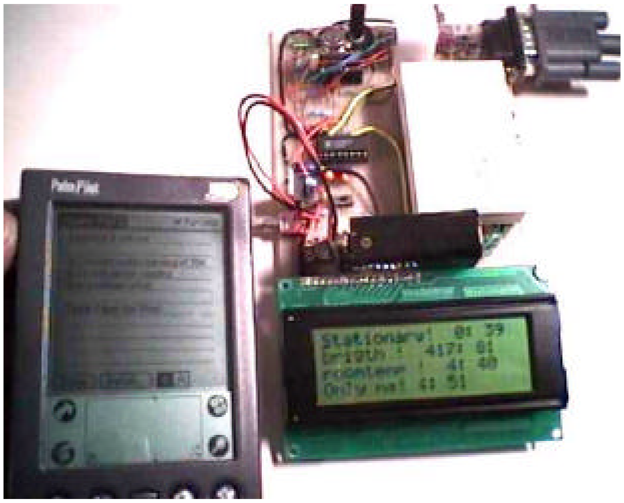


Figure 1: Context Sensing Device and PalmPilot

3 How Can HCI benefit from Context?

HCI for mobile devices is concerned with the general trade-off between devices qualities (e.g. small size, light-weight, little energy consumption, etc.) and the demand for optimal input-output capabilities.

3.1 Output in Context

Over recent years the output systems for mobile devices became much better; features such as stereo audio output, high resolution color screens even PDAs and upcoming on mobile phones as well as display systems for wearable computers. Also unobtrusive notification mechanisms (e.g. vibration) have become widely used in phones and PDAs. Situational context can help to:

→ adapt the output to the current situation (fontsize, volume, brightness, privacy settings, etc)

→ to find good time for interruption [12].

→ reduce the need for interruptions (e.g. you don't need to remind someone to go to a meeting if he is already there.)

3.2 Input in Context

Considering very small appliances the space for a keyboard is very limited what results in bad usability. Other input systems, such as graffiti and handwriting recognition have been developed further but still lack in speed and accuracy [9]. Advances in voice recognition have been made in recent years, but for non office settings (e.g. in a car, in a crowded place, sharing rooms with other, and in industry workplaces), the recognition performances is still poor. Also privacy and acceptance issues are a major concern. Context does not solve these problems in general but can help to:

→ adapt the input system to the current situation (e.g. audio filter, recognition algorithms, etc)

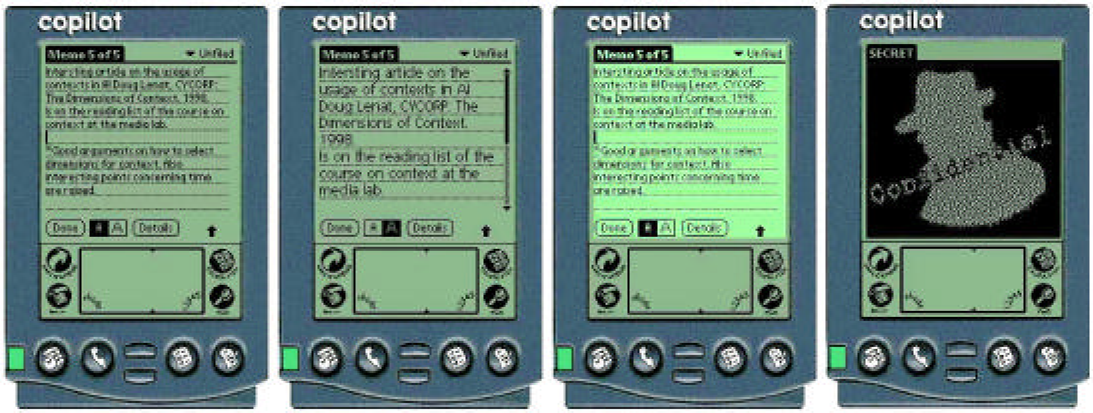
→ limit need for input (e.g. information is already provided by the context an can be captured)

→ reduce selection space (e.g. only offer appropriate options in current context)

4 ContextNotePad on a PalmPilot

To explore ways of implicit communication between the user and their environment with mobile devices we built a context aware NotePad application. This application provides very much the same functionality as the built-in notepad application on the PalmPilot. Addititionally the application can adapt to the current situational context and react in this way also to the implicit interaction. The application changes its behavior according to the situation. The following context adaptations have been implemented.

- **On/Off.** The user has the device in her hand. In this case the application is switched on, if the user is putting the device out of her hand it is switched off after a certain time. It assumes that if the user takes the device in her hand she wants to work with the device.
- **Fontsize**. If the device is move (e.g. while walking or on a bumpy road) the font size is increased to ease reading. Whereas while having the device in a stable position (e.g. device stationary in your hand or on the table) the font is made smaller to display more text at the same screen, see figure[2] 2 (a) and(b).



Interesting article on the usage of contexts in AI Doug Lenat, CYCORP: The Dimensions of Context, 1998. is on the reading list of the course on context at the media lab.

*Good arguments on how to select dimensions for context. Also interesting points concerning time are raised.

Figure 2: Adaptation to Context a) small font, b) large font, c) backlight, d) privacy

[2] The screenshots where made on the pilot simulator on a PC because it is easier to get good quality images this way.

- **Backlight**. This adaptation is straightforward but still not build in in current PDAs. By monitoring the light condition the application switches on the backlight if a certain light level is below a certain threshold. Accordingly if it becomes brighter the light is switched off again, see figure 2 (c).
- **Privacy settings**. If you are not alone and you are not writing (or touching the screen) the content on the display is hidden by a images, see figure (d). To sense if someone is walking the passive infrared sensor is deployed.

Currently we decrease the size of the context-awareness device to make it feasible to plug it into the pilot to allow proper user studies.

5 Conclusion and Further Work

In this paper we motivate a broad view of context. We suggest an application centric approach for the description of context. From current projects we learned that there is a need for a simple specification language context. We propose an XML-based markup language that supports three different trigger semantics. Basic mechanisms to acquire context knowledge are discussed, and a sensor-based context-awareness device is introduced. An analysis of potential benefits to HCI when using context is given. In an example implementation we demonstrate the feasibility of the concepts introduced earlier.

In the next phase we will extend the recognition to more complex contexts, especially by including simple time-domain audio processing. Currently we are developing a component that can be plugged into a small mobile device that provides contextual information. Sensing user contexts (e.g. by bio sensors) will open doors for Applications that are trying to guess user purposes and are given more information on the implicit communication hints.

References

[1] Beadle, P., Harper, B., Maguire, G.Q. and Judge, J. Location Aware Mobile Computing. Proc. of IEEE Intl. Conference on Telecommunications, Melbourne, Australia, April 1997.

[2] Brown, P. J., Bovey, J. D., Chen, X. Context-Aware Applications: From the Laboratory to the Marketplace. IEEE Personal Communications, October 1997.

[3] Brown, P.J. The stick-e Dokument: A Frameowrk for creating context-aware Applications. Proc. EP´96, Palo Alto, CA. (published in EP-odds, vol 8. No 2, pp. 259-72) 1996.

[4] Cheverst K, Blair G, Davies N, and Friday A. Supporting Collaboration in Mobile-aware Groupware. Personal Technologies, Vol 3, No 1, March 1999.

[5] Esprit Project 26900. Technology for enabling Awareness (TEA). www.omega.it/tea/, 1998

[6] Leonhard, U., Magee, J., Dias, P. Location Service in Mobile Computing Environments. Computer & Graphics. Special Issue on Mobile Computing. Volume 20, Numer 5, September/October 1196.

[7] Nokia Mobile Phones. 6110 Mobile phone,
http://www.nokia.com/phones/6110/index.html, 1998

[8] Norman, D. A. Why Interfaces Don't Work. The Art of Human-Computer Interface Design. Brenda Laurel (editor). Addision-Wesley. 1992.

[9] Goldstein, M., Book, R. Alsiö, G., Tessa, S. Non-Keyboard QWERTY Touch Typing: A Portable Input Interface For The Mobile User. Proceedings of the CHI 99, Pittsburg, USA 1999.

[10] Harter, A. and Hopper, A. A Distributed Location System for the Active Office. IEEE Network, Vol. 8, No. 1, 1994.

[11] Pascoe, J., Ryan, N. S., and Morse D. R., "Human Computer Giraffe Interaction: HCI in the Field", Workshop on Human Computer Interaction with Mobile Devices, University of Glasgow, United Kingdom, 21-23 May 1998, GIST Technical Report G98-1.

[12] Sawhney, N., and S., Chris. "Nomadic Radio: A Spatialized Audio Environment for Wearable Computing." Proceedings of the International Symposium on Wearable Computing, Cambridge, MA, October 13-14, 1997.

[13] Schilit, B.N., Adams, N.L., Want, R. Context-Aware Computing Applications. Proc. of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. IEEE Computer Society.

[14] Schmidt, A., Beigl, M., Gellersen, H.-W. There is more to context than location. Proc. of the Intl. Workshop on Interactive Applications of Mobile Computing (IMC98), Rostock, Germany, November 1998.

[15] Weiser, M. Some Computer Science Problems in Ubiquitous Computing, Communications of the ACM, July 1993.