Ubiquitous Presence Systems

Matthias Kranz, Paul Holleis, Albrecht Schmidt
Research Group Embedded Interaction
Amalienstrasse 17
80333 Munich, Germany
{matthias,paul,albrecht}hcilab.org

ABSTRACT

Instant messaging has become a part of our daily live. Instant communication, either with a mobile device or with a computer based application, is an increasingly used form of communication. In this paper we analyze how current instant messaging can be extended to a new and more general form of what we call 'ubiquitous presence systems'. We discuss how ubiquitous presence systems become a part of our daily live, extending the current systems, which only use states like 'online' and 'away', to location- and context-aware systems. We present the design space for ubiquitous presence systems together with a classification of those systems. We discuss information acquisition and also shortly address related privacy issues. We present an architecture for connecting arbitrary devices into the presence system, according to the vision of ubiquitous computing. We conclude the paper by presenting a fully working and deployed prototypical implementation following the developed architecture.

1. INTRODUCTION

Instant messaging and remote presence systems are commonly used by many computer users. It is a central tool for staying in touch and for being connected. Application areas range from personal and private use to enhancing formal communication at the workplace. Combining information about the availability and the means of quick communication using text, audio or video provides an important advantage for users in addition to traditional communication media such as the phone or email. Instant messaging and remote presence systems are available for different operating systems and integrate well with traditional desktop systems.

In our research we investigate how presence systems and minimal communication can be deployed beyond the desktop in our physical environment. We look at how objects and devices that offer tangible user interfaces (TUIs) can improve the user-experience and usefulness of such systems. Due to the increasing availability of ubiquitous computing technology, the extension of systems beyond traditional computer systems is feasible and can provide a new quality of interaction.

The following scenario highlights the use of presence systems beyond desktops that provide a physical user interface.

1.1 Scenario: Presence System in a Home Environment

Sally lives in a flat in London. She is working full time and likes to communicate with friends. During her work she uses her desktop based instant messenger (IM) to stay connected with her friends. At home, she has several physical presence devices. These devices are tangible artifacts that meet her aesthetics and can be found in different places in her flat, e.g., one on the kitchen table, one in the living room, one her bedside table like e.g. the Network Alarm Clock [12]. After she arrives at home, she makes herself a cup of tea in the kitchen and wonders about what to do in the evening. Waiting for the kettle to boil she puts the presence device to the state 'Any suggestions what to do?'. Her friends see her state in their IMs or their physical presence system. Later a friend calls on the phone and asks her if she likes to go to a birthday party of one of his relatives.

1.2 Motivation and Structure

The motivation for our research was twofold. First, the use of today's presence systems centers very much on the PC as one single point of interaction. Information about the state of the people does often not reflect what it should be. Second, it was the observation that people like to manipulate physical objects - play with things - while doing something else. One example are people in office environments. It is common that people move objects around the table or scribble with a pen while speaking on the phone. Thus, having a physical presence device, taking it and setting it to 'not available' for the time being on the phone and thereby reflecting the correct state may happen more often than just clicking somewhere on a small icon, searching for the desired state and changing it. Also, the amount of people using presence systems as well on the PC as on mobile devices increases steadily. Despite this development, the usage of presence information has not yet changed much.

Taking this as a staring point, we investigated opportunities and challenges in the creation of physical presence systems. In Section 2 we look at related work and compare this to our concept. We set out to map the design space for physical presence systems and instant messaging beyond the desktop system; this is described in Section 3. We experimented with various technologies to explore the

SAC'06 April 23-27, 2006, Dijon, France

potential of such systems. In particular, we looked at how a tangible device is used as a handle to explicitly change presence information. In Section 4 we show the design and implementation of a tangible location-aware user interface for the presence information in the IM software *Skype*. The results of a user study, further findings, and general recommendation for the design of physical presences systems are the presented in Section 5.

2. RELATED WORK

Ubiquitous presence comes in many flavors and with many implications. The knowledge of one's pure presence is the most basic information. The knowledge that someone is out there *and* thinking of you, like with the interactive picture frames Lumitouch [2], already adds significant personal information. This example may be taken from a private scenario, but it is surely applicable and extensible to, e.g., office scenarios as well, or more general, to collaborative scenarios.

Communicating not only the fact of thinking of someone or something, but actively exchanging messages is a feature of nearly every presence system, e.g. all current instant messaging systems. For announcing one's willingness to communicate, states were introduced, like e.g., depicted in Figure 4. However, people tend to not explicitly update their current communication settings. This may lead to annoyance to the communication partners, if, say, you are always shown as available, but are not. For ubiquitous presence systems, the fact of being available everywhere and all of the time may become annoying and found to be disruptive. This issue has been addressed in [3].

The combination of several pieces of information (what am I doing currently, where am I right now, am I available to talk at all and with whom, what presence device am I using for input and output) introduces privacy issues which can not easily be resolved. A way of filtering this information is urgently needed. This problem together with other related issues have been discussed by [5], but a solution is not at hand and will probably not be for a longer time to come.

But not only the means of communication and the related privacy issues change. Also the devices for input and output change. Currently, input and output are closely related, especially with PC based systems. It is only a matter of time when input and output will be split up, e.g. to use sensors in smart appliances or the infrastructure in the environment for presence detection and use displays or colored light for output and feedback. The work from Eisenstadt [5] combines presence, messaging and location-based services which could be employed.

Instant messaging, a feature of most current presence systems, is a very fast and convenient way to exchange information – if the communication is desired by both parties. Studies [8] show that communication does not even switch to more explicit and richer forms of communication (e.g. telephone where, among other things, the sound of the voice is available as additional information compared to text chat) if the communicated content is getting complex.

In contrast, our work focusses on how current systems can be extended in two ways. First, we bring the physical action of being there or not back into the physical world by introducing a tangible user interface for status setting. Second, we extend the sphere of instant messaging to other domains, e.g. home environments.

3. DESIGN SPACE FOR PHYSICAL PRESENCE SYSTEMS AND TOWARDS SHARING CONTEXT

An essential feature of a traditional presence system is to communicate the state of a person in an abstract form to others that may be interested in and allowed to access this information. When looking into physical environments, it becomes apparent, that presence functionality can be extended to objects and environments. A generic presence system can essentially be seen as a system to share context information.

3.1 Asking the Right Questions

Considering ubiquitous computing settings, the variety of information that is available and can be shared is impressive. Examples typically relate to location, co-location, and activities of people. The level of abstraction and the timeliness of the information becomes much more of an issue as these parameters offer ways for preserving privacy. The following questions provide a starting point for the design of a physical presence system.

- What information is going to be shared and on what level of abstraction?
- What time constraints with regard to acquisition and delivery are implied?
- Who has access to the shared information?

Tools commonly used in desktop computers and mobile computing devices interweave presence information sensing, presence information presentation, and instant communication. The symmetric design of sensing and presentation is very much due to the nature of the PC as the host computer. For the inclusion of instant communication, one rationale is that changes in the presence state provide opportunities for initiating communication. If persons log into a computer and becomes visible online to their friends this prompts reaction such as *'are you in the office, can we quickly have a chat?'*.

Therefore, considering the design of a presence system that is integrated with everyday environments we have to take the following into account.

- How is presence acquisition and presence information presentation related? Are they within one (logical) unit or are they separated?
- Should the device that presents presence information include means for instant communication?

In an ubiquitous computing environment a wide variety of tools and mechanisms for context acquisition are available [13], much more than on traditional desktop computers. Similarly, there are many more options for presenting information ranging from traditional displays to dedicated displays, including ambient media. This leads to the following questions.

- How can one acquire the information that is to be shared?
- How can the acquired information be represented?

All the questions above help the designer and developer of a physical presence application to explore the essential parameters of a potential system.

3.2 Explicit and Implicit Acquisition

Presence systems can be distinguished by the way they acquire information with regard to the user's involvement in this process. In general, we separate explicit from implicit acquisition. In the case of explicit acquisition, the user is actively and consciously involved in setting the context or communicating the state to the system. It is called implicit acquisition when the system detects information from sensors or applies reasoning on data available to conclude the current state or context. This discrimination is in line with the definition of implicit and explicit interaction as introduced in [11].

Most current desktop-based IM and presence systems use a combination of implicit and explicit acquisition. The basic detection of states (e.g. 'away', 'online') is done implicitly based on the network connection of the computer, the keyboard and mouse activity, and a timer. Additionally, those systems offer an interface to explicitly set a state (e.g. 'do not disturb'). In research systems beyond the desktop, implicit and explicit input are also common.

Using an ID-Card, an electronic key or a finger print scanner to gain access to a building is an explicit action performed by the user and could be used as input into a physical presence system. Such systems offer great opportunities for implicit acquisition as well. Implicit presence acquisition by, e.g., tracking people using some (indoor) location system with for example badges [7] can be done in an unobtrusive way. However, integrating control and explicit interaction is rather difficult. Questions like 'How can a user set the presence to 'unavailable' with such a system?' remain a big issue.

In general, we think that 'smart' systems using implicit data acquisition together with context information could collect presence information with a minimum of user interaction and annoyance. It is clear that in such settings issues of privacy and control are essential and need to be addressed, see Section 3.5.

3.3 Presence Information and Context

Using personal computers, it is clear that presence information relates to the user that is using this specific machine. Even though IM system allow the use of multiple desktop computing systems, the essential idea is that presence information is related to a person using a single computer at the time.

In ubiquitous computing, systems are more diverse and become a part of the environment. In such environments, the one-to-one relationship between computer and user is not given in general. Such systems may provide many explicit points of interaction and additionally a rich sensor infrastructure to acquire additional context information.

Beyond the acquisition of presence information, we identified the following categories whose presence information is desirable in certain situations:

- person (is a colleague or team member available or not?)
- object (has the awaited item already arrived?)
- environment (is an environmental condition satisfied?)

Extending presence beyond the person can be considered as a system for sharing context. It appears interesting to extent IM systems into this direction. The hope is to build on the well understood and accepted usage model of current IM and make them into tools for sharing context.

3.4 A basic classification

To guide the design of ubiquitous presence systems it is important to understand the design space. We have identified the following six criteria as central for classifying ubiquitous an physical presence systems.

- **information acquisition**
  - implicit acquisition (e.g. timer of non-activity of input, passive infrared sensor)
  - explicit acquisition (e.g. click on a button)
- **the object of interest**
  - person-related-context (e.g. team member)
  - object-related-context (e.g. delivery good)
  - environment-related-context (e.g. rain)
- **the input and output capabilities**
  - input and output united in a device (e.g. buttons and displays)
  - input and output separated (e.g. input via badge, output via phone in the vicinity of the user, [7])
- **the supported means of communication**
  - means for explicit communication integrated (e.g. keyboard)
  - no means for explicit communication integrated
- **the extent of presence**
  - environmental presence (how does the environment react to my presence?)
  - social presence (how much do other people appear to be (virtually or physically) present (e.g. do I see them or an image of them on my presence devices) and how much am I aware of them?)
- **the communication mode**
  - asynchronous (e.g. leaving a text message to be read when the other person connects again)
  - synchronous (e.g. use of a video or voice chat)

This classification can be used to systematically design presents systems. By make conscious decisions on these six aspects it is easier to design a system that meets an anticipated set of requirements. To analyze existing presence the classification can also be useful.

3.5 Privacy

By sharing presence information with others a user is inevitably giving up some privacy. Allowing others to see the connection state (e.g. 'online', 'away', 'offline') is very common and many people think they give little away with this. But considering the possibility of accumulating this information over time, a lot of information can be inferred. Questions such as when is she usually leaving home and how long is she having her lunch break can be answered to some extent. In an ubiquitous presence system, these issues are touched as well, but with implicit acquisition using sensors and sharing of context even more issues arrive, which cannot easily be resolved.

In principle, it is essential that system designers and potential users understand that there is a trade-off between providing useful information to others about one's context and preserving one's context. When using implicit mechanisms for acquiring context and presence information, it is important to design it in a way that users can grasp the conceptual model. If they have a model of how the system works, they can make qualified decisions.

With regard to allowing protection of privacy, we propose the following basic design guidelines for ubiquitous presence systems.

- provide a conceptual model to the user that allows understanding when, where and how privacy issues are touched
- give the user easy means to override and manipulate implicitly collected context information
- provide easy means for (temporarily) switching-off the system
- enforce information symmetry within the system (e.g. when someone sets the state to 'offline', disallow him or her from seeing who is 'online') or make access to information accountable (e.g. 'I can see who checked on me and when').
- make acquisition of a history of someone else's context and presence information difficult

Providing means for opting-out (e.g. allowing the user to switch-off the system) are technically often easy but have social implications. Being connected always and everywhere and hence offering presence information may introduce a social pressure to people. If possible, a system should be designed that it is easy to not publish state information all the time.

In environments where information about the user is acquired by different means, like using sensors in the environment, it may not be obvious what state is communicated to others. It is therefore important to provide the user with the information what others see of him or her. This is an additional issue related to information symmetry. Such a system is proposed in [9].

4. CASE STUDY

In this section we describe the conducted case study in detail. We introduce the technology used and depict the realized system architecture. We close this chapter by presenting the results of an initial user study on the use of current personal computer based instant messaging systems.

4.1 Tangible User Interface

We developed a tangible user interface (TUI) for conducting the case study. The TUI is depicted in Figure 1. It supports up to six different stable states and comprises a display for giving feedback. The overall size of the device is 6.5 cm x 4.5 cm x 4.0 cm (width x height x depth). It can easily be handled with one hand. It is supposed to be placed on the desktop near the user. The display has a good contrast and is easily readable in different light conditions.

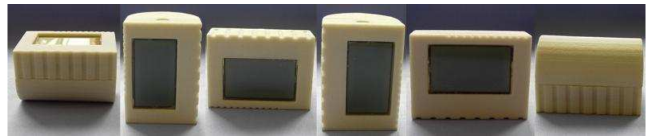


Figure 1: All stable states the case supports by its affordances. There are at maximum six stable states. Some can additionally be prevented by placing the battery at the bottom of the case (see leftmost state) or directly under the display.

4.2 Technology - Particle Computer System

Particle Computers [4], [6] are small wireless sensor nodes. The node's hardware comprises a communication board integrating a microcontroller, a radio transceiver (125 kbit/s, with a range of up to 50 meters), a real-time clock, additional Flash memory and LEDs a speaker for basic notification functionality. It can run of a single 1.2V battery which consumes on average 40mA. The particle computer can be extended by additional boards. Current sensor boards integrate X-Y-Z acceleration sensor, a light sensor for two different wavelengths, a temperature sensor and a microphone. Other sensor can be connected very easily using prototyping boards providing pin-heads for analog and digital input as well as digital outputs.

Particles especially address scenarios of high node mobility and issues of small size. The AwareCon [1] communication stack implements a slotted TDMA scheme which is established and maintained by the nodes in a self-organized manner without any support of a master node. Synchronization with a new network in range takes typically around 12ms. The mean delay for the synchronization with another single partner is around 40ms. The Particle communication board together with one of the sensor boards measures 15x48 mm. This is equal to the size of an AAA battery and allows unobtrusive embedding of wireless sensor technology.

Two two-axes accelerometers (ADXL311JE from Analog Devices) which are orthogonally mounted inside the TUI are used for determining its current orientation. The acceleration sensors detect acceleration as well as gravity which is exploited for state detection. The display is a Barton BT94060 display with 96x40 pixels resolution. The display is connected over the I2C bus.

Both the acceleration sensors as well as the display are situated on the add-on board [10] which is connected to the base Particle using the standard Particle Conan adapter.

4.3 Application Architecture

To connect the great variety of inputs needed for an ubiquitous presence system to an even larger number of applications as output, we developed a generic architecture capable of dealing with this. The architecture depicted in Figure 2 and the prototype application in Figure 3.

The general idea is to connect the various input devices like tangible user interfaces or location systems using their respective communication capabilities (e.g. RF with Particles) and the existing infrastructure to a standardized network. Via this network (e.g. IP network), data is transmitted to eventually necessary intermediate or proxy applications. These translate, if necessary, the data format and meaning to a specific format, e.g. a Windows window mes-

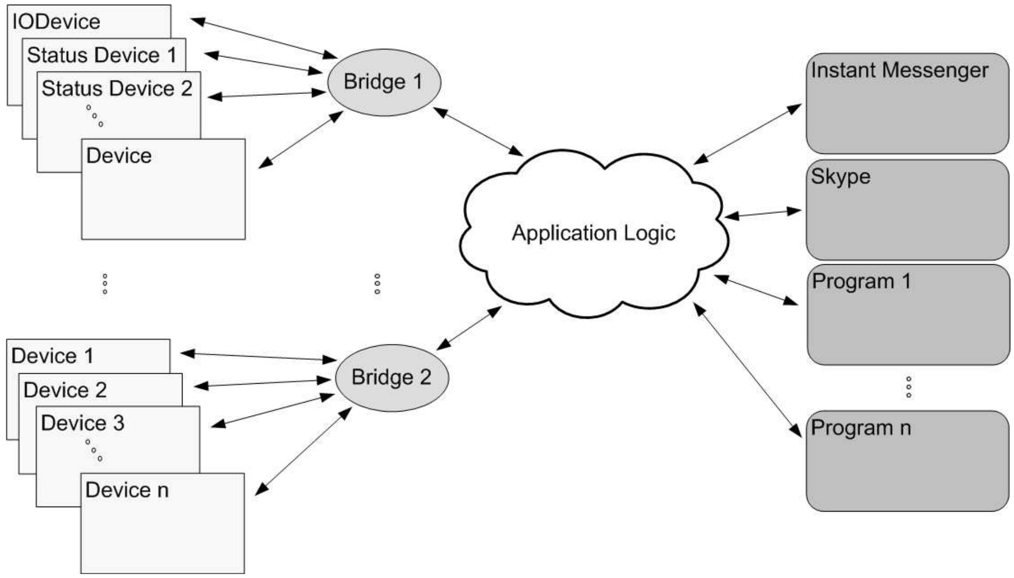


Figure 2: Architecture overview. Various different input devices, like the status device described later on, are connected via network bridges to the application logic. The data from the input devices is processed and made available to the applications, e.g. *Skype*.

sage, that is accessible for the final output, e.g. the *Skype* application or a location system, or, in general any data sink interested in the data.

Prototypical Implementation

The application architecture, depicted in Figure 3, is a concrete instance of the generic architecture described in Section 4.3. The fully working prototype, which has been deployed in our office environment and currently is subject to long-term evaluation, is described below.

The input device is the *Skype* state tool depicted in Figure 1. It calculates data about its orientation, transmits it via RF, and via an infrastructure element to the local network. An intermediate application gets the data, analyzes sender and content information and delivers this in a format understandable by the output application. The current orientation is calculated based on the sensor values of the two acceleration sensors. The application runs on a Particle Computer system as described in Section 4.2 above. The Particle is programmed in C using the CCS C compiler.

The data is, whenever the orientation has changed, transmitted via RF. If no change in the orientation occurred, the Particle is put to power save mode. Depending on the history of calculated orientations, the sleep time is enlarged to a maximum sleep time of about two seconds. The maximum sleep time was carefully chosen to not delay the state change in the *Skype* application and thereby make the system seem to react too slow. If the device is held in a non-stable position, e.g. if the user plays with it, the orientation is set to 'undefined'. This state is also transmitted to signal the activity. So, if the user is playing with the TUI, no state changes occur.

The data packet with the current orientation value is received by one or more XBridges in the infrastructure. The XBridges translate the RF packet into UDP packets which are then broadcast according to the XBridge configuration on the current subnet.

The intermediate application is responsible for

- analyzing the packet origin to only accept packets which are targeted to current user's *Skype* and therefore support an unlimited number of such presence devices
- dealing with the problem of duplicate packets which can occur if the original packet is received by multiple XBridges (each Xbridge forwards all packets it receives).
- setting the state in the user's *Skype* application

If the state is 'undefined', the intermediate application does not change the current state within *Skype*. The intermediate application is programmed in C++ using *Microsoft Visual Studio .Net*.

The communication between the intermediate application and *Skype* happens over the *Skype* API. The messages exchanged are limited to state settings which are in the API since the initial API release. Those are unlikely to change in future. The *Skype* API basically sends messages to Windows windows. The *Skype* window is identified and then communicated with. The message format is textbased and human-readable, a message to *Skype* is e.g.

```
SET USERSTATUS OFFLINE
```

The overall time from changing the orientation of the device to the change of the state within the application is less than half a second.

In Figure 6 the tangible user interface is depicted. It visualizes the current state set in *Skype* on its display. If turned, the state is updated. The 'offline' state is connected with the

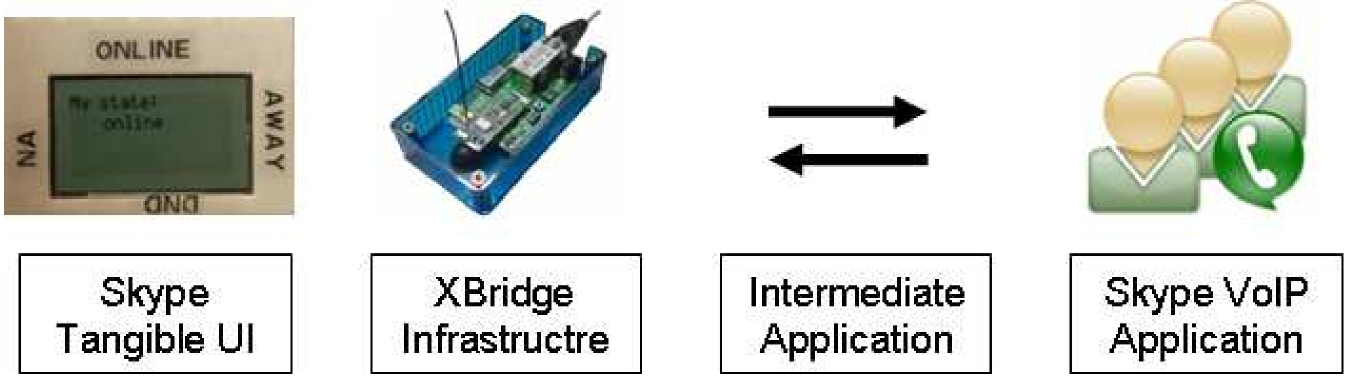


Figure 3: Prototype overview: The tangible user interface sends the current state, based on the orientation of the device, via RF (radio frequency) to the XBridge infrastructure. The RF packets are translated into UDP packets. These are processed by an intermediate application which then sends the corresponding state to the local *Skype* application.

device lying on its display. This is a quite natural mapping: I am offline, I cannot be seen and see no other information about my contacts.

A potential problem occurs if the user manually changes his or her state in *Skype* - the tangible user interface is unchanged and therefore could display inconsistent state information. If the TUI does not possess output functionality at all, there is no way of communicating back the current state. If there is some output functionality, e.g. like the display in our prototype, this can be communicated to the user. But, as the user is the one who changed the state him or herself, we do not consider this inconsistency as a problem.

4.4 User Study

We conducted an initial user study to test the need and acceptance of the proposed system. We chose a group of largely unrelated people from whom we knew that they are using computers at work or at home. There were 16 participants including 5 women. All participants were in the age between 23 and 35. Most of them have a background in computer science, although their use of computers in their everyday lives largely varies.

Two of the participants do not use IM systems at all. However, they are aware that those programs exist and that these offer the possibility to switch between states like 'idle' or 'busy'. Obviously, the following data concerned with using IM systems does not take these two submissions into account.

Most people use a subset of the available free systems like *Microsoft Messenger*, *AIM*, *Skype*, *Miranda* (*ICQ*), *Yahoo Messenger* and others. More than 70 percent of the participants state that they are using such a system 7 or more times a day, meaning to communicate seven or more times with one or more contacts, and feel comfortable with its usage.

All participants know that these applications offer several states to be chosen. However, several of them did not know by heart how many or exactly which states they could choose from. The states 'online' and 'away' where known by all persons, the state 'do not disturb' by all but two and 'offline' by all but one. In Figure 4 the different states of current instant messaging systems is depicted.

These observations motivated our choice to include 'online', 'offline', 'away', and 'do not disturb' in our set of states on the device. We added 'not available' as fifth state as it was mentioned as the fifth most often stated item.

The second set of questions are concerned with if, under what circumstances and how often people normally change their states in their IM applications during the day. There was one question for each of the following activities: Being busy, leaving the workplace, not wanting to be disturbed, having a break (e.g. for lunch), leaving the computer to something different.

Only 2 people state that they manually change their state according to all these activities. On the other hand, 35 percent of the participants say that they normally don't change their state in these instances at all. One reason that has been mentioned three times is, that several IM programs can detect when the user stops working or disconnects and automatically sets an appropriate state. However, these automatisms mainly operate by monitoring the input devices mouse and keyboard. They can hardly judge whether someone is really not sitting in front of the computer or just reflecting upon a larger portion of text. Neither can it judge whether the user is prepared to be disturbed or not in a particular moment.

More important, however, is that more than 80 percent of the people affirm that they would reflect their activities with their state if they didn't forget to do it or if it were less tedious to do.

Aggregate messenger systems like Trillian can help since one can set the state of several IM systems in one go. However, only half of the people that took part in the study said they use such systems. And all but one of those are using applications like *Skype* that cannot be integrated into such a system, which makes it again a tedious task to change the state in all of them. No matter how many different system can be included in one frontend, there surely will always be another program that is not (yet) integrated. And still the manual task of changing ones state remains.

The third part of the questionnaire asked questions about the tangible prototype at hand. The device was given to the users without any additional explanations. All of them were quickly able to guess the use and mode of operation. The size of the device was judged appropriate by 64 percent with a small tendency of being too large (for the rest it was

Figure 4: Screen shot of the available states in different instant messaging systems. From left to right: *Trillian*, *ICQ*, *MSN Messenger* and *Skype*

'slightly too large'). The weight seemed not to be an issue and for 80 percent it was just appropriate. Since the plastic casing is, dependent on the method of creation, rather rugged and coarse, most people do not think it is very pleasant to touch. The prototype depicted in the figures in this paper was created using a 3D print service. However, the number and type of states that can be set using the device was rated very good and only 2 people asked for one or two more. The size and readability of the display was evaluated diversely, but in general not very good: Only 2 people felt it was good to see and read. We expected something like that and had added the question whether or not feedback is necessary at all and, if yes, what type of feedback would be best. Rather surprisingly, 75 percent said that feedback was essential. However, more than half of the participants stated they would already be satisfied if the device had a few LEDs (or one light that can change its color and thus the appearance of the device) to indicate the current state and provide confirmation about changes. A low beep was also suggested as a means of acknowledgment.

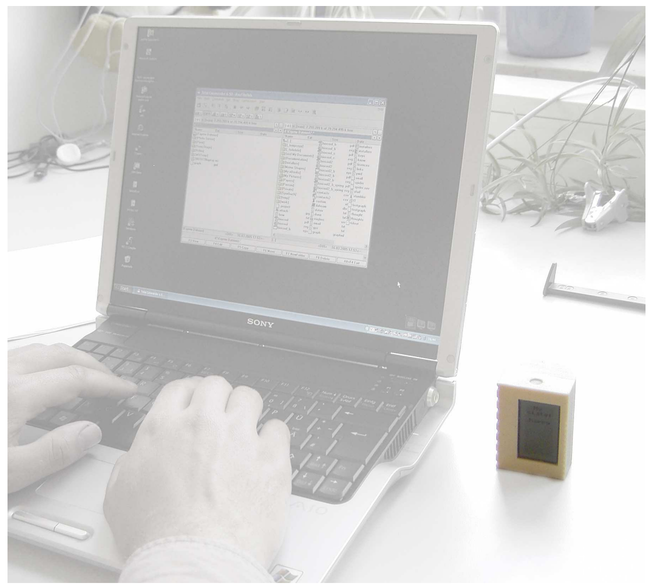


Figure 5: The status device in use, next to mobile computer. The user has selected his state to be 'happy' though he is working. This also shows that the states do not have to be limited to a working scenario.

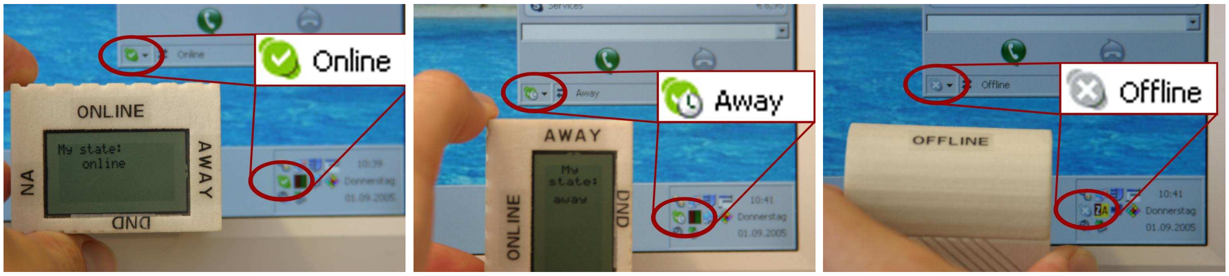
This user study consolidated our opinion that there is demand for such a system as we propose it. The prototype has been quickly accepted by the participants and they all showed interest in using it. Therefore, we are currently reacting to the criticism of the display used in the device and prepare several devices to be used in long-term studies that will show more exactly how much the added value of the system is.

5. CONCLUSION AND FUTURE WORK

We presented a tangible user interface for communicating one's physical state to a virtual communication system. We moved the virtual action of setting the state into the real world where it belongs – the user is part of this real world and so is his or her current state. Setting it virtually by a mouse click is therefore awkward in our opinion. So far, we observed that the physical input device fosters engagement and therefore people tend to forget changing their status less often and enjoy playing with the device during e.g. phone calls.

After having conducted the initial user study described in Section 4.4, we are currently enhancing our prototype. We are extending its functionality by adding capabilities to sense context information like location of the devices (can be provided by the infrastructure we are using). This will enable a device to act more implicitly by automatically changing states and other information according to this context.

Another project will be dealing with toolkit support to facilitate integrating (input) devices with available applications and to help creating new applications using these devices. This includes an ongoing effort to use feedback and special output capabilities of the devices. One example is to use a display on the device to show the descriptions of the possible states. When a user manually changes his or her state by directly altering the instant messenger's own state, the descriptions on the device could dynamically be altered and updated. Controlling several applications with one device (e.g. several instant messenger systems with the one tangible device described above) is already easily possible by adding appropriate calls to the APIs of each IM application. We are, however, also aiming to incorporate several devices that can control or influence the same application.



(a) online (b) away (c) offline

Figure 6: The tangible user interface is used to set the user state. In the background you can see the state set accordingly in the *Skype* PC application.

It will tremendously help in judging what state the user currently wants to be in if there are several devices placed in different locations, sensing different actions. There could for example be one device that recognizes when the person is currently using his or her mobile phone and another being placed directly besides the fixed network phone (or attached to the telephone receiver).

Although our user study suggests that there is a demand for the proposed system, we are going to undertake a long term user study. Several users with different patterns of behavior with relation to computers in general and use of instant messenger systems will each be given one of our enhanced prototypes. We will be monitoring the constant use of the device and keep track of people's views on the system for periods between 3 and 6 months.

6. ACKNOWLEDGMENTS

This work has been conducted in the context of the research project Embedded Interaction (Eingebettete Interaktion) and was funded by the DFG (Deutsche Forschungsgemeinschaft).

7. REFERENCES

[1] M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson. AwareCon: Situation Aware Context Communication. In *Proceedings of Ubicomp 2003*, Seattle, USA, October 2003.

[2] A. Chang, B. Resner, B. Koerner, X. Wang, and H. Ishii. Lumitouch: an emotional communication device. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 313–314, New York, NY, USA, 2001. ACM Press.

[3] L. Dabbish and R. E. Kraut. Controlling interruptions: awareness displays and social motivation for coordination. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 182–191, New York, NY, USA, 2004. ACM Press.

[4] C. Decker, A. Krohn, M. Beigl, and T. Zimmer. The particle computer system. In *roceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 443–448, April 2005.

[5] M. Eisenstadt and M. Dzbor. BuddySpace: Enhanced Presence Management for Collaborative Learning, Working, Gaming and Beyond. In *JabberConf Europe*, June 2002.

[6] H.-W. Gellersen, G. Kortuem, M. Beigl, and A. Schmidt. Physical Prototyping with Smart-Its. *IEEE Pervasive Computing Magazine*, 3(3):74–82, July–September 2004.

[7] A. Hopper, A. Harter, and T. Blackie. The active badge system (abstract). In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 533–534, New York, NY, USA, 1993. ACM Press.

[8] E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano, and C. Kamm. The character, functions, and styles of instant messaging in the workplace. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 11–20, New York, NY, USA, 2002. ACM Press.

[9] D. H. Nguyen and E. D. Mynatt. Privacy mirrors: Understanding and shaping socio-technical ubiquitous computing systems, 2002.

[10] Research Group Embedded Interaction. Particle Display Add-On Board Website. `http://www.hcilab.org/projects/particles/particles-display-add-on.htm`, 2005.

[11] A. Schmidt. Implicit human computer interaction through context. In *Personal Technologies, Vol 4(2)*, June 2000.

[12] A. Schmidt. Network alarm clock. In *3AD: Third International Conference on Appliance Design, Design Competition Entry*, pages pp. 142–115, June 2005.

[13] A. Schmidt and K. V. Laerhoven. How to build smart appliances. *IEEE Personal Communications*, pages 66 – 71, 2001.