Advanced Interaction in Context

Albrecht Schmidt[⊕], Kofi Asante Aidoo[✦], Antti Takaluoma[)(],
Urpo Tuomela[)(], Kristof Van Laerhoven[✦], Walter Van de Velde[✦]

[⊕]TecO, University of Karlsruhe, Germany
[✦]Starlab Nv/Sa, Brussels, Belgium
[)(]Nokia Mobile Phones, Oulu, Finland

albrecht@teco.edu, kofi@starlab.net, Antti.Takaluoma@nokia.com,
Urpo.Tuomela@nokia.com, kvlaerho@vub.ac.be, wvdv@starlab.net

Abstract. Mobile information appliances are increasingly used in numerous different situations and locations, setting new requirements to their interaction methods. When the user's situation, place or activity changes, the functionality of the device should adapt to these changes. In this work we propose a layered real-time architecture for this kind of context-aware adaptation based on redundant collections of low-level sensors. Two kinds of sensors are distinguished: physical and logical sensors, which give cues from environment parameters and host information. A prototype board that consists of eight sensors was built for experimentation. The contexts are derived from cues using real-time recognition software, which was constructed after experiments with Kohonen's Self-Organizing Maps and its variants. A personal digital assistant (PDA) and a mobile phone were used with the prototype to demonstrate situational awareness. On the PDA font size and backlight were changed depending on the demonstrated contexts while in mobile phone the active user profile was changed. The experiments have shown that it is feasible to recognize contexts using sensors and that context information can be used to create new interaction metaphors.

1   Introduction

Current research and development in information technology is moving away from desktop based general purpose computers towards more task specific information appliances. Mobile phones and personal digital assistants (PDAs) dominate the research landscape as their dominance grows commercially. Last year alone there was a 56% increase in the use of mobile phones across Western Europe resulting in approximately 23 million users of cellular technology. The functionality of these appliances is the crucial issue.

Users of these devices are not specialists and don't accept long learning phases. Nevertheless users like advanced functionality, but it is important that this does not compromise the ease of use for these appliances. An important challenge in competition is to develop new functionality with *added value* for the user and still keep the interaction mechanism simple and straightforward. Answers and ideas can be

found in the inherent nature of mobile electronics. People take their phones and PDA's everywhere using them in various environments and situations to perform different tasks. The user's expectation towards the device also change with the situation (e.g. the user would like different ring tones for a phone in a meeting than on a noisy road). Ideally, devices that know about the situational context the devices could transparently adapt to the situation. Such devices lead to the invisible computer, discussed by Weiser [14] and is a step towards the ideal of a disappearing interface as demanded by Norman [8].

In our work we aim to prove the idea that the more the device knows about the user, the task, and the environment the better the support is for the user and the more the interface can become invisible.

To build devices that have knowledge about their situational context, it is important to gain an understanding of what context is. Current research in context-awareness shows a strong focus on location [1], [6]. An architectural approach, based on a smart environment is described by Schilit et. al. [11]. Other scenarios are using GPS and RF to determine the users location [3], [4]. The visual context in wearable computing is investigated by Starner et. al. [13]. But, as pointed out in [12] context is more than location; this is also recognized in the approach at Georgia Tech to build a context toolkit [10].

We use the term *context* in a more general way to describe the environment, situation, state, surroundings, task, etc.. To provide a view on what we understand by the term context we like to provide a definition from the dictionary as well as a number of synonyms that can be found in a thesaurus:

> **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)

As it can be seen from above context is used with a number of different meanings. In our research project Technology for Enabled Awareness (TEA, [5]) we define *Context awareness as knowledge about the user's and IT device's state, including surroundings, situation, and, to a lesser extent, location.* To describe contexts we use a three-dimensional space as depicted in fig. 1, with dimensions *Environment*, *Self*, and *Activity*.

Context has many aspects and in the work presented in this paper we focus on physical parameters and information provided by the appliance (e.g. PDA, or mobile phone). To acquire physical parameters we use low-cost and widely available sensors. With this information we determine the user's current situational context. This approach seems complementary to the idea of smart environments as proposed by [9]. With the new generation of mobile devices having increased processing power we focus on making the devices smarter, giving them the ability to recognize and

interpret their environment. However, smart devices and smart environments are not mutually exclusive, and it is easy to imagine that a combination can be used.

After briefly introducing the concept of context and situational awareness, we propose an architecture for context recognition. The architecture is composed of four layers, namely sensor, cue, context, and scripting. In section 3 we describe a prototypical implementation that performs the complete mapping from the real environment to awareness enabled applications on a GSM-phone and on a personal digital assistant (PDA). This proves the feasibility of the approach described here. Finally we summarize our results and discuss future directions of our work.
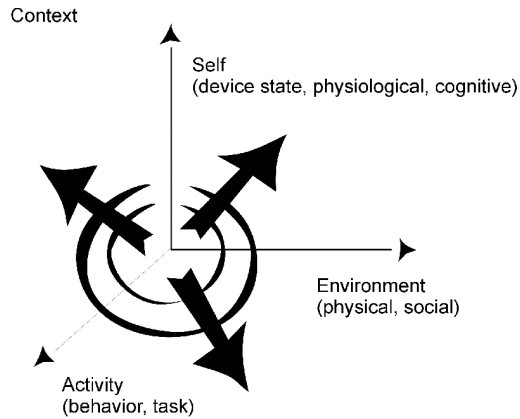
Fig. 1. 3-D Context Model

2 Architecture

To build a flexible and yet efficient system we introduce a layered architecture for the TEA-system. As depicted in fig. 2, the architecture consists of four layers, sensors, cues, contexts, and an application layer. In later development phases the part of the architecture that is implemented in hardware will move up, whereas in the early phases as much as possible is implemented in software to enhance flexibility.

2.1 Sensors

We distinguish between physical and logical sensors. *Physical sensors* are electronic hardware components that measure physical parameters in the environment. All information gathered from the host (e.g. current time, GSM cell, etc.) are considered as *logical sensors*.

Each sensor S_i is regarded as a time dependent function that returns a scalar, vector, or a symbolic value (X). A set (finite or infinite) of possible values (domain D) for each sensor is defined.

$S_i: t \rightarrow X_i$
t is the time (discrete), $X_i \in D_i$, i is the identification of the sensor
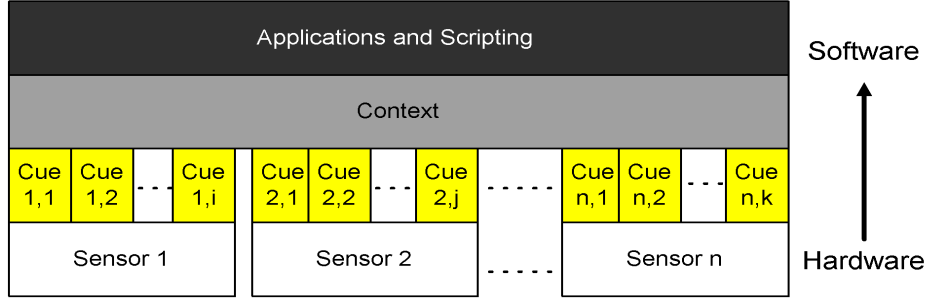
Fig. 2. Layered architecture of the TEA-system.

2.2 Cues

The concept of *Cues* provides an abstraction of physical and logical sensors. For physical sensors, introducing a layer for cues also solves the calibration problem. A cue C_{ij} is regarded as a function taking the values of a single i sensor up to a certain time t as input and providing a symbolic or sub-symbolic output Y. A set (finite or infinite) of possible values (domain E) for each cue is defined.

C_j: $S_i(t)$ x $S_i(t\text{-}1)$ x … x $S_i(t\text{-}n)$ $\rightarrow$ Y_{ij}
t is the time (discrete), $Y_{ij} \in E_{ij}$, t≥0 n≥0, j is the identifier for the cue

As seen from the definition, each cue is dependent on one single sensor but using the data of one sensor, multiple cues can be calculated.

2.3 Contexts

A context is a description of the current situation on an abstract level. The context is derived from the available cues. The context T is described by a set of two-dimensional vectors. Each vector h consists of a symbolic value v describing the situations and a number p indicating the certainty that the user (or the device) is currently in this situation. The finite set V of the symbolic values is defined.

T: $C_0(S_0, t)$ x $C_1(S_0, t)$ x … x $C_k(S_0, t)$… $C_0(S_i, t)$ x $C_1(S_i, t)$ x … x $C_m(S_i, t)$ $\rightarrow$ h
$h = \{(v_1, p_1), …, (v_j, p_j)\}$
t is the time (discrete), $v \in V$, k≥0, i≥0, m≥0, j≥0

2.4 Applications and Scripting

To provide a mechanism to include context information in applications we offer three different semantics. Basic actions can be performed when entering a context, when

leaving a context, and while in a certain context. In our approach we offer the following scripting primitives[1]:

Entering a context:
```
// if the context is: T=h={(v,p)}
// if the situation v is indicated with a certainty of
// p or higher than p the action(i) is performed after
// n milliseconds, v is a situation, p is a number,
if enter(v, p, n) then perform action(i)
```

Leaving a context:
```
// if the context is: T=h={(v,p)}
// if the situation v is indicated with a certainty
// below p the action(i) is performed after n
// milliseconds, v is a situation, p is a number,
if leave(v, p, n) then perform action(i)
```

While in a context:
```
// if the context is: T=h={(v,p)}
// if the situation v is indicated with a certainty of
// p or higher than p the action(i) is performed every
// m milliseconds, v is a situation, p is a number,
if in(v, p, m) then perform action(i)
```

Beyond the defined scripting primitives the application programmer is free to use context knowledge in any part of the applications where it seems appropriate. A different approach is described by Brown et. al [2].

3 Feasibility Demonstration

The demonstrator described in this section was implemented to prove feasibility, gaining contextual knowledge using low level sensors. A main requirement for the prototype was flexibility to enable experiments with different sensors as well as a variety of recognition technologies.

The prototypical system was used in two phases. In the first phase data in several situational contexts were collected and than analyzed off-line. In the second phase the prototype was used for real-time context recognition.

3.1 Hardware

The individual sensors have been chosen to mimic typical human senses, as well as more subtle environmental parameters. An outline of the schematic is given in fig. 3.

---

[1] The parameter n 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 into the garage, the context 'walking' should appear soon) a negative value does make sense, too.

- The photodiode yields both a nominal light level (as experienced by humans) and any oscillations from artificial sources (not a human sense). It is sampled at a rate of approximately once per millisecond, but only for a few hundred milliseconds at a time, allowing other signals to be multiplexed in.
- The two accelerometers provide tilt and vibration measurements in two axes. Due to the limited sampling power of the current board, the signal is filtered down to 200 Hz, though the sensors are able to supply up to 1 kHz of signal.
- The passive IR sensor detects the proximity of humans or other heat-generating objects. The sensor provides a signal corresponding to the amount of IR received, possibly filtered for sensitivity to the human IR signal. This sensor is sampled at the same rate as the photodiodes.
- The temperature and pressure sensors each provide a conditioned signal between 0 and +5 volts directly, and need no amplification. These sensors are sampled a few times a second.
- Sampled at the same rate as the temperature and pressure sensors is a CO gas sensor. The PIC controls the heating and reading of this sensor.
- For sound, there is an omni-directional microphone that is directly connected to the computer.

Each of the sensors provides an analog signal between 0 and 5 volts which is read by the 8 bit, 8 channel analog-to-digital converter. The signals to the A/D converter are routed through switches that allow for off-board sensors also to be sampled. This makes the board expandable and gives the flexibility for testing new combinations of sensors.

Our first module needed to have flexibility, both in the sensors to be included on it, and the processing to be performed on the data collected. The architecture developed, whether for the first prototype or the final design, would have to follow a set protocol of development and communication that will become the standard for the duration of the project. Being a project based on sensor information the hardware architecture has to follow the basic principles of sensor based systems. The sensor information is sent as a signal to an input function. The input function prepares the signal for processing. The processing and amplification block translates the signals according to the individual sensor need and also to have uniform output signals for all sensors.

The processed signal then goes to an output function, which primes it for output reading or display. More accurately, the sensors measure conditions in the environment and translate them into analog voltage signals on a fixed scale. These analog signals are then converted to digital signals and passed to the micro-controller (PIC). The micro-controller oversees the timing of the analog-to-digital converter and the sensors as well as manipulating the data from the analog-to-digital converter's bus to the RS-232 serial line. Finally, the serial line connects to the data-gathering computer(Host).

The PIC acts as the brains of the board and executes a loop, polling the sensors through the analog-to-digital converter, and moving the data onto the RS-232 serial line. Higher bandwidth signals like the accelerometers and photodiodes are polled often, on the order of every millisecond, while slower signals like temperature are only polled once a second
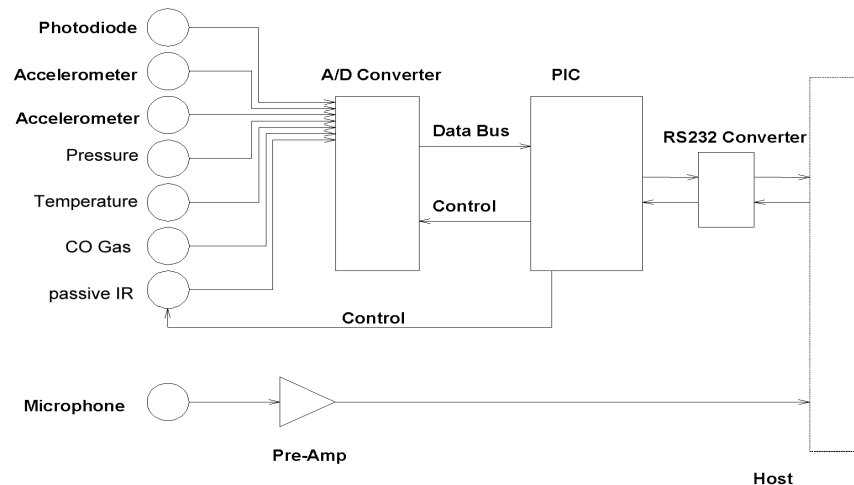
Fig. 3. Schematic of the sensor board.

Another requirement of the system is mobility. In order to simulate mobile devices the board also has to meet certain size constraints. The board is the standard Eurocard PCB size of 100mmx170 mm. At that size the board is less than half the size of a laptop. This allows for ease of movement when it comes to experimentation. The board can easily be connected, via serial cable, to a laptop for data gathering. The second phase of the project will produce an even more portable board with direct connection to the device.

3.2 Off-line Data Analysis

As described before, the TEA sensor board sends periodically a large block of data, represent the digitized sensor outputs, through its serial port. In the experiments a portable computer is connected to this port, which makes it possible to receive and store the data. A piece of software was written for this purpose.

After this TEA reader software has made a datafile, the datafile can be analyzed to predict how a learning system could map the raw data to a context. One of the easiest and fastest methods to obtain this is by plotting the output of all sensors directly on a time scale in parallel (see fig. 4a).

This time series plot only shows the sensor values of the acceleration sensors and the light sensor in three different contexts. Initially, the TEA sensor board was placed on a table and remained there for about 100 seconds. After this period, the device was taken along and stayed in the hands of its user for another 100 seconds. Finally the TEA board was put in a suitcase for yet another 100 seconds.

The interesting thing about this plot is that the different contexts are immediately visible, while other types of plots experience problems with the high dimensionality of the data and are less clear. A phase space plot for instance, which is limited to three
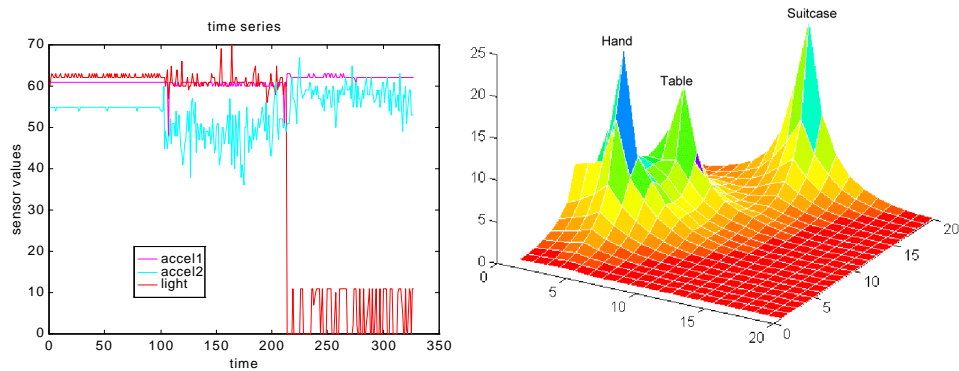
Fig. 4. a) Time series of sensor data,   b) Kohonen Clustering of sensor readings

dimensions, is unable to visually represent the eight or more sensor values at every timestep.

The high number of sensors doesn't just cause a problem for the analysis of the sensor-outputs, it is also making the mapping phase difficult. For this reason, it is crucial to cluster the raw sensor data first, preferably with an adaptive clustering algorithm. Experiments with the Kohonen Self-Organizing Map and some of its variants show promising results. The Kohonen Map is an unsupervised neural network that is known to perform very well under noisy conditions. It clusters the values coming from the sensors onto a two-dimensional grid in an adaptive way (the cells in the grid – the neurons - 'learn' to respond better for a certain input). After the data is clustered into a low-dimensional discrete space, it has become significantly easier to process this data with symbolic AI techniques, such as predictive Markov Chains. Fig. 4b shows the clustering of a data set from our experiments is shown: the 20x20 grid depicts the Kohonen Map, while the z-axis represents the frequency of activation for every cell in the map. This way, the organization of the cells (or neurons) is visualized: three activity bubbles emerge representing the map responding to three contexts.

3.3 Online Recognition Software

Based on the experiments, methods have been developed and selected for the real-time recognition system. A set of functions to calculate cues as well as logical rules to determine contexts have been implemented.

Sensor to Cues Mapping. In the prototype the concept of cues proved to be very useful to make changes of the hardware transparent for the context recognition layer. When including new sensors with different characteristics only changes in the corresponding cues must be adapted. In our current implementation the cues are calculated on the notebook computer from the actual data read by physical sensors included in the hardware.

Cues are one way to reduce the amount of the data provided by the sensor board. In the current implementation we focus mainly on statistical functions. They can either provide a summary of the values over time or they help to extract features from the raw data that characterize the data over the last period of time. The following functions are used:

- **Average**. The average of the data items provided by a single sensor over a time frame of about one second is calculated. This is applied to data from the light, acceleration, temperature, and pressure sensor. For the acceleration sensor this value gives also the one angle of the device to the gravity vector.
- **Standard derivation**. Standard derivation for data read from the light, passive IR, and acceleration sensor is calculated over about one second.
- **Quartile distance**. For light, passive IR, and acceleration we sorted the data and calculate the distance between values at one quarter and three-quarter. This proved more reliable as using the range.
- **Base frequency**. For light and acceleration we calculate the base frequency of the signal. This provides useful information on the types of lights (flickering) and on activities such as walking (certain acceleration pattern).
- **First derivative**. For passive IR and acceleration we approximated the first derivative of data to gain an understanding of changes that happen.

In our prototypical implementation we calculate these cues in real time and provide the results in the context layer.

Cue to Context Mapping. A context is calculated on the notebook computer from the information delivered by cues. In our experiments we work with a number of context sets, all of them using exclusive contexts. Examples of these sets are:

(a) holding phone in hand vs. phone in a suitcase vs. phone on a table
(b) walking while using the device vs. stationary usage
(c) using the device inside vs. using the device outside
(d) in car vs. on a bus vs. on a train.
(e) having a phone in a stationary car vs. in a moving car

Working with exclusive contexts makes the development of the recognition algorithm easier and also simplifies the development of applications. In GSM-application we used two of the sets of exclusive contexts - (a) and (c) .

For the real-time recognition system used in the described prototype we used logical rules defined for each set of contexts to determine the current situation. In Table 1, a simplified rule set for the discrimination of situations in (a) is given. The recognition in this example is based on only three sensors: light, and acceleration in two directions (X and Y). The rules are built on observation of usage in certain contexts as well as from an analysis of the data collected in test scenarios. This data was also used to determine the constants used in the example (Dx, Dy, L, Xnormal, Ynormal, D, and Q).

```
Hand(t):-      standard_ deviation(accelX,t) > Dx,
               standard_ deviation(accelY,t) > Dy,
               % device is slightly moving in X and Y
               average(light,t)>L. % not totally dark

Table(t):-     abs(average(accelX,t)-Xnormal)<D,
               abs(average(accelY,t)-Ynormal)<D,
               % the device is level in X and Y
               quartile(accelX,t)<Q, quartile(accelY,t)<Q
               % the device is stationary
               average(light,t)>L. % not totally dark

Suitcase(t):-   average(light,t)<L. % it is totally dark
```

Table 1. Simplified recognition rules for (a).

3.4   Initial Experimentation

We implemented experimental setups for a Palm Pilot PDA and for a mobile phone. In both cases the recognition software was running on a PC, to which both the TEA hardware and host device (GSM or PDA) are connected. Thus, the GSM setup consists of four objects: TEA Hardware, TEA Software (running on the PC), and the GSM Demo Software (running on Laptop) and the Nokia 6110 mobile phone. We performed two classes of experiments.

Experiment 1: adaptive context recognition. In the first experiment we wanted to test the robustness of the context classification when using the adaptive cue to context mapping (based on the Kohonen net). In a training phase data was collected while taking the board through a progression of contexts, related to being in the hand, being on the table, in light or in dark. The system succesfully clustered sensor reading into a two-dimensional grid in which areas of high-activation were seen to correspond to these contexts. It should be noted that parameter settings in the clustering algorithms (e.g. proximity) sometimes lead to the creation of spurious clusters. The labeling of these clusters was added by hand.

   Then we tested the recognition of these contexts, when taking the board through similar (but not identical) context changes. In informal evaluation we observe a robust classification, estimated close to 90% when attempting to follow similar contexts as the ones in the training set.

Experiment 2: context-enhanced applications. In the second type of experiment we tested the full cascade of mappings from sensors to scripting, using the model-based cue to context mapping instead of the adaptive one. The context enhanced applications running on GSM or PDA received the information about the current situational context, from the recognition software. The application then decides how to adapt to this context. In the PDA scenario a notepad application was changed to adapt the fontsize to the users activity (when the user was walking a large font, small font when stationary) as well as to environmental conditions (e.g. light level). In the remainder of this section we will focus on the applications implemented for the mobile phone. Fig. 5 shows parts of the demo setup.

In the GSM scenario the profiles of the mobile phone were selected automatically based on the recognized context. Currently Profiles exist in most of Nokia mobile phones. These offer a way to personalize the behavior of mobile phone. For example, on some occasions the phone must ring as loud as possible (e.g. at Bar where is lot of noise), while in others the phone must be silent (e.g. library or movie theater). There are even cases when the phone must filter calls (e.g. a meeting where urgent home calls must get through), which can be done by using so called Caller groups, a feature that is supported when the operator provides CLI -functionality (Caller Line Identification). Profiles offer a direct and easy way to define these situations and the corresponding behaviors (what we have called scripts).

Contexts, as described in this paper, can be thought of much similar to the Profiles on Nokia phones. Profiles are currently hand selected from a menu. However, by using a device that is context sensitive and connecting that to a Nokia phone, the Profile (or Context) change can be automated. Automatic Profiles are one new type of applications that context sensitivity introduces.

For the experiment the following Profiles were defined on the Phone:

- **Hand**. When the user holds the phone in her/his hand, the audio alarm is not needed. The phone rings by vibrating.
- **Table**. Here we assume a meeting situation. The phone is almost silent. Incoming call is indicated by a very gentle sound.
- **Silent (pager)**. Phone is silent. Here we assume that the phone is put a way in a box or suitcase, and must be silent. The phone still receives calls, so that the callers' numbers can be recalled later.
- **Outside**. Here the ring volume goes as high as possible and vibra-alarm is on. All possible ways to get the users attention are used.
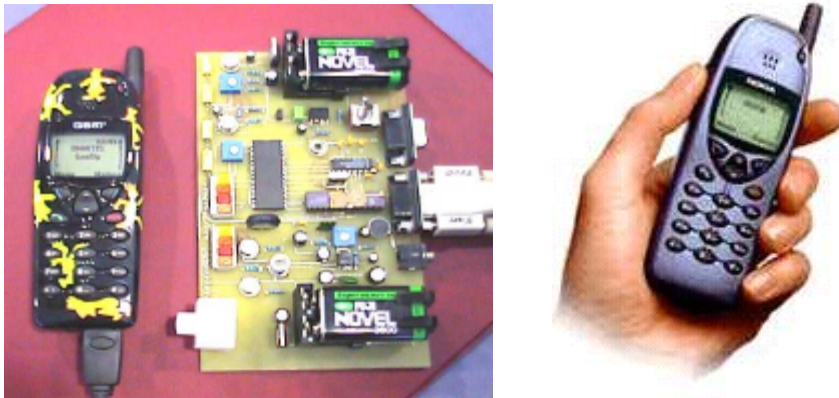- **General**. General mode is used when none of the above apply.

Fig. 5. Phone on a table together with the first prototypical sensor board. Phone in hand

The purpose of the demonstration was to automate the Nokia 6110 [7] Profile change, based on the cue-to-context mapping as defined in the Architecture section. The experiment was performed (in public) in a regular meeting room, by taking the TEA component from the table, in the hand, switching off the light, taking the board outside, and then in a suitcase. As expected the phone Profile followed the context of the TEA Hardware. For instance, when TEA Hardware was taken to hand the phone went to profile *Hand*. When left on the table the phone profile was *Table*.

Rule based recognition is limited by the validity of the model that it is based upon. Thus, there are at least three possible ways that recognition can fail:

1. Ambiguity. For instance, when it is very dark outside, the suitcase context may be erroneously recognized.
2. Operating close to boundary conditions. For instance, the model definition rely on threshold settings that make system behavior sensitive when operating close to them.
3. Undefined context model.

We will to tackle some of these problems in future work by combining adaptive and symbolic techniques, and by adding training and learning features to the system.

4.   Conclusion and Further Work

We proposed a layered architecture for a context-aware system based on sensors. The concepts of cue, context, and scripting are introduced. Implementing a prototype we demonstrated that it is feasible to recognize contexts with low-level sensors and that this information can be used to create new interaction metaphors. Taking the examples of a mobile phone and PDA, we showed that applications can be enhanced by context recognition.

In the next phase we will extend the recognition to more complex contexts, to context sets with a larger number of situations, and to non-exclusive contexts. We also will investigate efficient methods for the recognition process to make it feasible to include the TEA-component in mobile appliances such as PDAs and mobile phones. Currently we are developing a component that can be plugged into a phone or PDA to provide contextual information. Having a device that it is not an additional burden for the user we plan to make a usability study.

We would also like to introduce the concept of learning in the TEA-component. This will give the user the ability to induce appliances to learn individual and user specific situations.

Sensing User contexts (e.g. by bio sensors) will open doors for applications that are trying to guess user purposes. These applications may either help the user or provide additional information.

Acknowledgment

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

[10] Salber, D., Dey, A.K., Abowd, G.D. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In the Proceedings of CHI '99. May 1999.

[11] 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.

[12] 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.

[13] Starner, T., Schiele, B., Pentland, A. Visual Contextual Awareness in Wearable Computing. Proceeding of the Second Int. Symposium on Wearable Computing. Pittsburgh, October 1998.

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