

# An Architecture for Multi-Sensor Fusion in Mobile Environments

Datong Chen, Albrecht Schmidt, Hans-Werner Gellersen

TecO (Telecooperation Office), University of Karlsruhe  
Vincenz-Prießnitz-Str.1, 76131 Karlsruhe, Germany  
{albrecht, hwg}@teco.uni-karlsruhe.de

**Abstract** - This paper introduces a layered architecture for multi-sensor fusion, applied for environment awareness of personal mobile devices. The working environment of personal mobile devices changes dynamically depending on their user's activities. Equipped with sensors, mobile devices can obtain an awareness of their mobile working environment, to improve their performance with respect to usability. The mobility of the device presents two problems for building an awareness system. First, the contexts to be covered by an awareness system depend on the users, their tasks and activities, and also on the data that can be obtained from different sensors. Second, the power consumption and the size of the mobile device limit the processing capability of an awareness system. The solution presented here is to design a low cost sensor-based fusion system, which can be reconfigured by the user, to enable individualized awareness of environments. The software architecture presented in this paper is designed with four different layers, which can support reconfigurations in mobile environments.

**Keywords:** mobile environments, multisensor fusion, context-awareness, fusion architecture

## I. Introduction

Personal mobile devices, such as laptop, GSM and PDA, break the traditional desktop paradigm and bring people the powers of the computing and electronic communication anywhere and anytime. Our investigation focuses on improving the function and interface of these personal mobile devices through awareness of the user's activities and the current social environment. Different from the desktop, mobile devices are portable and accompany their users from one place to another. This kind of mobility puts the device into a changing environment, which is more complex to be processed than in fixed cases, while it also offers them more opportunities to know more about their users and their own situations with certain awareness techniques. For example, a PDA may track the locations of its user from the home to the office and adjust the items in the "to do list" from home-

related issues to the work-related issues. It may also recognize that the user starts to walk after a calmly sitting and then change its display to the large font automatically to ease reading. Many investigations have already been done on applying the desktop-based awareness to improve the interaction between human being and the computing device [1, 2]. Based on these former works, a multi-sensor fusion architecture to enable awareness for the mobile devices is presented in this paper.

To enable the awareness of mobile devices, a small multi-sensor device is developed by the European Commission funded research project Technology for Enabling Awareness (TEA, [3]). This multi-sensor device can be connected to a mobile device as an additional part and offers useful context information to the host. Aiming not to destroy the portability of the mobile device, the multi-sensor device is designed to employ only low cost sensors and rely on fusion techniques to extract useful contexts from the data obtained from these low cost sensors. "Low cost" means that:

First, the size of the sensors should be small enough to keep the multi-sensor device much smaller than the size of the host device. Second, the sensors should consume low power and the signals they produced can be processed with little processing power. Finally, the price of the sensors is also a factor that should be regarded.

Investigating how to enable awareness in mobile environments, two kinds of adaptation are necessary when the working environment of the mobile device is dynamically changing with situation and location. One is that in different situations certain sensors are more useful than others. For example, the air pressure sensor may be useful when the user is on a flying plane, but can not offer much useful information when the user is sitting in the office room. Operations to adjust sensors, such as switch on/off, affect the related fusion algorithm to produce stable results. The other

adaptation is needed because, in different environment, the mobile device is interested in different contexts. For example, at night, the mobile device may pay attention to the context about whether there are artificial lights. But in the daytime, this context may be not necessary. The fusion-based context awareness algorithms, which compute other contexts according to the context artificial light, need to be able to adapt to this modification. The multi-sensor fusion system for mobile environment should be designed robust enough to adapt the continuous reconfigurations of both sensors and contexts.

In many former works, the sensor fusion can be classified into different levels according to the input and output data types [4, 5]. The fusion may take place in the data level, feature level and decision level.

In data level fusion, the raw data from sensors is used to extract features [6]. Varieties of the methods are developed in this level, and were applied in the image processing, visual & speech recognition, data compression and intelligent control [7, 8, 9]. The feature level fusion is to fuse the features extracted from multi-sensor data into new features or the final decisions. Because most features have well-defined structures, the fusion methods in this level can be based on statistical approaches and pattern analysis approaches [10, 11]. Decision fusion is a common problem in many research areas, such as decision theory and artificial intelligence. An example of the simple decision fusion is the voting system, in which every candidate has equal or not equal right to determine the final result [12]. Artificial intelligence techniques show new trends for the solution for decision fusion, for example the neural network [13]. There are two advantages of applying neural networks to fuse the decision. One is that the neural network is noise-tolerant and can process the input features with plenty of noise. The other advantage is that neural network allows the system to be reconfigured according to the specified application instance.

## 2. Layered architecture

The adaptation of the reconfiguration of the sensors and contexts in the mobile environment is the important factor in designing the architecture of the fusion software system. When the sensor is modified (being switched on/off or adjusted its sampling rate) in the system, there should be a feasible mechanism to let the related fusion processes know this change and make correct responds. On the other hand, when the user reconfigures a context in the system, the feedback of this adjustment should also activate the correct

adjustment of the related processes and sensors. To develop a common and feasible reconfiguration fusion system, one method is that we define the whole fusion system with several independent layers. Each layer consists of certain structures and data processes, and keeps contact with next layers through defined interfaces. In this way, the reconfiguration in one layer can be controlled by the predefined function in this layer and the effect of the modification can be limited by the interface to the next layers. In other words, the result of the reconfiguration in one layer can be regarded as a kind of normalizing the input of the other layer, so that the adaptive fusion algorithms can be developed in different layers separately. In this paper, we describe a fusion architecture with four layers, see figure 1.

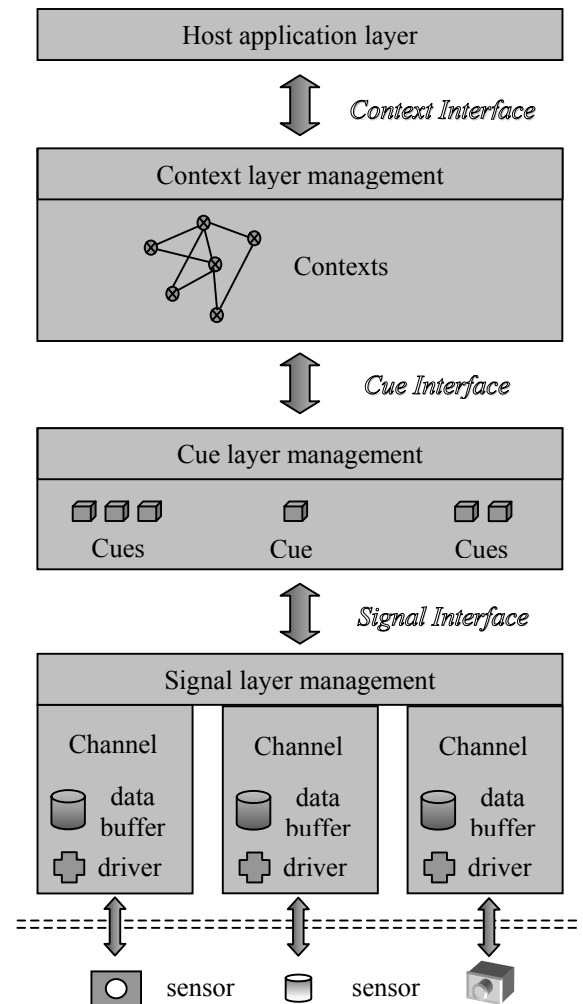


Figure 1. Four layers fusion architecture

## 2.1 Signal layer

The lowest layer is called signal layer, which connected with the sensors directly. The function of the signal layer is to control the data collection of the sensors and write the data into a uniform structure. A special kind of software channel is employed in this layer to adapt the reconfigure of the sensors. For each sensor, there is a channel with corresponding driver, data buffer and other attributes to manage it temporally. Three attributes of one channel are the logical name of the signal read from the sensor, which is used to identify the corresponding driver of the sensor; a time stamp system to manage the data stored in the buffer; a sampling frequency system, which is used to respond to the current available sampling statuses. When the hardware of the system is modified, for example a sensor is added, a sensor is removed, a sensor is switched on/off and so on, the sampling frequency system of the related channels will detect the change automatically and adjust the value of the sampling frequency. This sampling frequency value can also be set by the system through software directly.

The output data of the signal layer is the raw signal data with a structured description. The description involves the information about the current data, such as the time stamp, the sampling frequency, the number of dimensions, and the size of the each dimension. Most of the signals employed in TEA project have one dimension, for example, light signals, audio signals, temperature, etc. There is also two or three-dimensional signal such as the acceleration signals.

## 2.2 Cue layer

The processes in the cue layer mainly focus on the time independent features extracting from each single channel data. The time independent features extractions transform the time-varied data space into time independent feature space. From our point of view, the information fusion can be regarded as a data compression process. The raw data from several sensors will be compress into the result space. The fusion across different sensors is to reduce the redundancy among the data of these sensors. The reduction of the redundancy among the data of on sensor is also a kind of information fusion. Except for the time independent features extractions, the data from multi-dimension sensors is transformed into independent feature space in the cue layer. The time-varied analysis in the cue layer is limited within only a short period of sample data. Long term analysis will be done in the higher layer.

We call these kinds of the self-independent features from single sensor channel as cue, in order to show their differences with the common concept of feature. The cue layer keeps a specified period of history of cues, which serves as a history description of the changing environment.

## 2.3 Context layer

The perceptible events in the environment are treated as the contexts of the activities of the host device in this layer. The current contexts can be derived from several cues, deduced from former or other current contexts, or combine the two approaches together. The system employs semantic nets to represent the former and current contexts. This semantic nets are designed with a limited verb set and probability description, for example, the current contexts can be represented like that "At 10:32, with 85% probability, (it) starts to walk, in the office". Each context keeps a value of its own respond frequency, which can be adjusted by the user according to his needs. More deep reconfigurations of the context, such as add a new context or training the context layer to recognize your new office room, need the cognition and deduce fusion approaches in context layer are self-adaptive or can be trained manually. Artificial neural networks are good tools to support the deep reconfigurations of the context, because they can be trained through the examples automatically. The decision tree is another possible method to reconfigure the deduce algorithms. The context layer keeps the history of the contexts, which can be rewritten into the nodes in semantic nets to perform certain deduce algorithms.

## 2.4 Application layer

The application layer is developed within the operation system of the host and uses the result of the fusion system to improve the services of the host devices.

## 2.5 Interfaces

The communications between different layers rely on the fixed interfaces defined in the architecture. The interface between signal layer and cue layer is called signal interface. Through signal interface, the cue layer can read the data from each available channel and set the sampling frequencies of it. On the other hand, the signal layer can sent messages to activate the cue layer whenever the data is updated or the sensors are switched on/off. The cue interface is designed to keep contact between the cue layer and the context layer. By using this interface, the context layer can not

only access the current cues, but also has access to the stored history cues. The information about the updating of the respond frequency of the context can be sent to cue layer and further extended to the signal layer. Similar as in the signal interface, the cue interface also supports to send the cue-updating message from the cue layer to the context layer. The interface between the context layer and the application layer is the context interface. In order to apply the multi-sensor awareness device to different mobile devices, the context interface is designed as a one way interface, which offers the access only from the application layer to context layer. It offers a rich set of functions to the host applications, including reading current and history contexts, setting the respond frequencies of the contexts, setting the attributes of the contexts, recording the samples and training the algorithms in the context layer, adding a new context or deleting an old one, and so on.

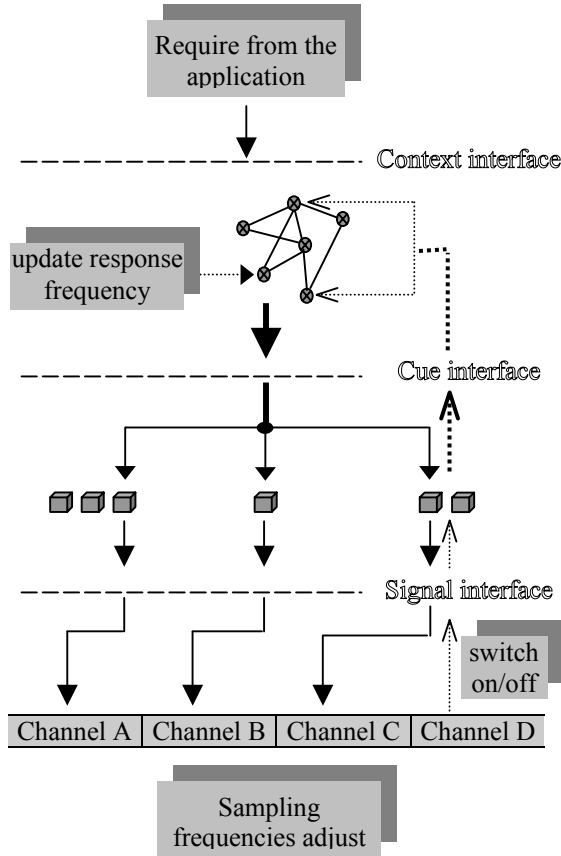


Figure 2. Reconfigure information feedback

## 2.6 Reconfigure information feedback

The information to reconfigure the system can be transmitted both ways: from the signal layer to the context layer and from the application layer to the signal layer.

The both feedback processes are depicted in figure 2. When the host application wants to modify the response frequency of a certain context, it sends a command to the context layer through context interface. In the context layer, first, the respond frequency of the specified context will be updated to the new value according to the command, if this new value is valid. And then, the new value will be transmitted to the related cues in the cue layer. Because one context may be the fusion result of several cues, and one cue may also be employed by different contexts, in the cue layer, the related cues decides whether they should adjust themselves to adapt the change of this context while do not affect other related contexts. If the cue chooses to change its respond frequency to the new value, this value will be transmitted to the corresponding channel in the signal layer. The channel, which receives this information, may adjust its sampling frequency after checking all the cues extracted from this channel.

When a sensor is switched off, the corresponding software channel should detect it and informs all the cues that based on this channel. This channel will be disabled under the signal layer management, but the related cues are still enabled because the history of these cues can be used for the future awareness. If a sensor is switched on, the signal layer will detect its signal, enable the channel and recover to send the updating message to the related cues. The context layer will check the time stamp of the cues before using them. A cue, which has not been updated for a long time according to its own respond frequency, will be regarded as unavailable resource. If this happens, the related algorithms in the context layer will be reconfigured with predefined methods.

## 3. Evaluation

In the experiment described in this section, we deployed the prototypical tea-device [14], a sensor-board that reads environmental parameters using a number of low cost sensors.

### 3.1 Hardware

The board consists of four major blocks: the sensors, the analog-to-digital converter, the microcontroller, and the serial line. The sensors measure the 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 microcontroller. The microcontroller 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 serial line. Finally, the serial line connects to the higher layer, see Figure 3. In terms of the architecture described earlier, the hardware incorporates sensor and parts of the sensor dependent drivers (signal layer) implemented in a microcontroller. The communication between the sensor board and the mobile device is using a serial-line in a multiplex mode. In this prototype, the higher layers are emulated with a laptop, which connected between tea-device and the host device to control the experiment easily.

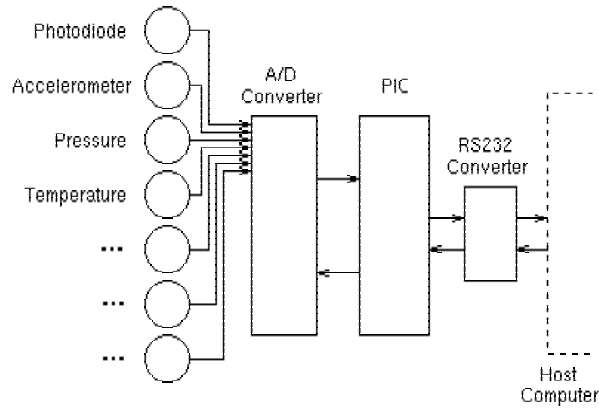


Figure 3. Schematic

### 3.2 Software and Interfaces

The context, cue, and signal interface are offered as C++ methods to the next higher layer. The context and cue layers are implemented entirely in C++, too. For the host application layer we used different host dependent implementations. The signal layer is partly implemented in C on the microcontroller and partly in C++.

### 3.3 Experiments and Results

In the experiment, we collected data of all sensors in different contexts cycle by cycle, as described in Table 1. Within each cycle, the sensors were activated and read according to their sampling frequency to feed the

environmental parameters. The data for each context was collected over a time of about 100 seconds, or about 120 records. Selected parts of the data are depicted in the following figures.

Table 1. Contexts samples

Context	Description
Inside-1	office, artificial light, stationary
Inside-2	office, artificial light, walking
Outside-1	outdoors, daytime, cloudy, stationary
Outside-2	outdoors, daytime, cloudy, walking

Looking at the light data sample in Figure 4, it shows the values of brightness at cloudy outside and inside with artificial light. It is obvious to find the difference between inside and outside on the level of light as well as on the oscillation of the light. Comparing the acceleration data for a stationary device in figure 5. with the one for a moving device in figure 6, it can be seen that they differ significantly.

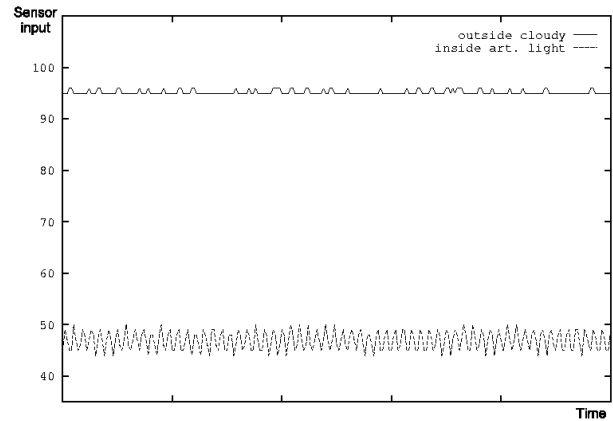


Figure 4. Light sensor data

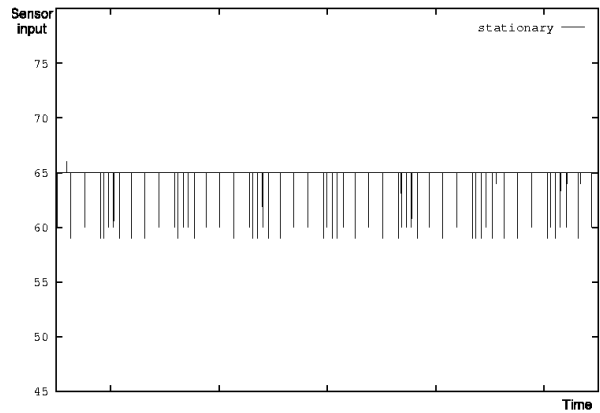


Figure 5. Acceleration sensor for stationary device.

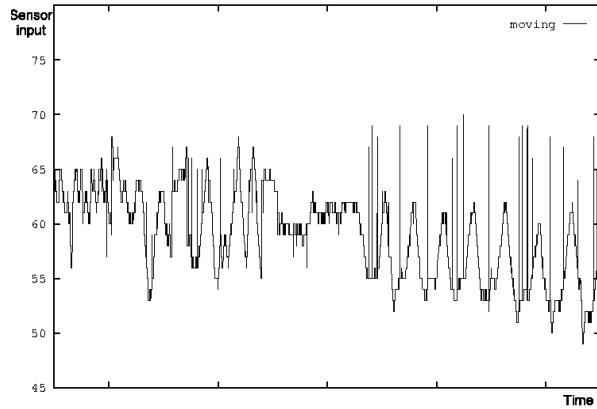
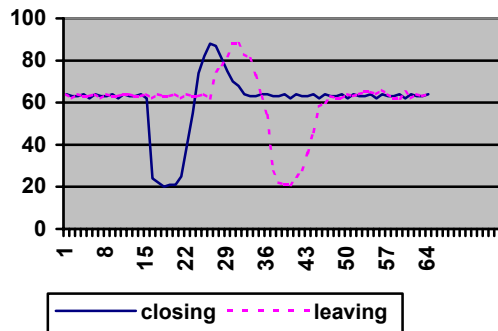


Figure 6. Acceleration sensor for moving device.

### 3.3.1 Cue extraction & context awareness

There also other sensors on the sensor board, such as the sensors of the temperature, the air pressure, the passive infrared and so on. Each cue is extracted from the data of one corresponding sensor with proper algorithm. In the figure 7, we can see a typical period data from passive infrared sensor when the user moves the device in hand (the X-axis represents the time & the Y-axis represent the value of the passive infra data). Using the sequence analysis algorithm, the cues leaving and closing can be recognized within one



sampling cycle.

Figure 7. Passive infrared sensor for moving in hand

The data from some sensors, especially from light sensor, involves some random noises that usually occur with no more than two sequential values in one sampling cycle. Before analyzing the data from this kind of sensors, we suggest to use a mid value filter with 5-value-size window to do the preprocess.

Except for the cues extracted in time domain, the cue can also be the feature in frequency domain, for example the cue - base frequency. Base frequency

represents the main frequency of oscillation of the light. The data from light sensor was transformed into frequency domain through FFT, and then used a linear window to find out the base frequency of in the date. This base frequency should be a stable value when there is artificial light near the light sensor.

Most of the awareness of the contexts is based on more than one cue and even other contexts. The cues and contexts are regarded as different dimensions of input vector of the fusion algorithm. Artificial neural network and decision tree are investigated to fuse the input vectors into contexts. To describe the position of the mobile device, we employed three contexts: the device is in hand, the device is on the table, and the device is in a suitcase. The input vector has 15 dimensions, which corresponds with 15 cues from the sensor of gas (CO), temperature, pressure, light, passive infrared, and 2-dimensions acceleration. Automating the recognition, we used 297 samples (three classes, hand, table, suitcase; 99 vectors each) to train a neural network on them in a supervised mode. The other 297 samples were then used to test the recognition performance. With a standard back-propagation neural network we achieved a recognition rate of about 90 percent. Using a modular neural network, as described in [15], consisting of two input modules and on decision network we achieved a recognition rate of more than 97 percent.

### 3.2.2 Reconfiguration

The context “inside/outside” is used to describe the rough location of the host device is out door, inside of a building or a vehicle. The distinction of the inside and outside depends on the fusion result from the cues and contexts related with the light sensor and temperature sensor. The output data of the light sensor and temperature sensor are showed in the figure 3. Many cues are derived from the light sensor data in a standard period, such as the average brightness, standard deviation, base frequency, and so on. From the temperature sensor data, we get the cues: maximal and minimal temperature, average temperature. As showed in figure 8, two kinds of context are also useful to decide the context inside/outside.

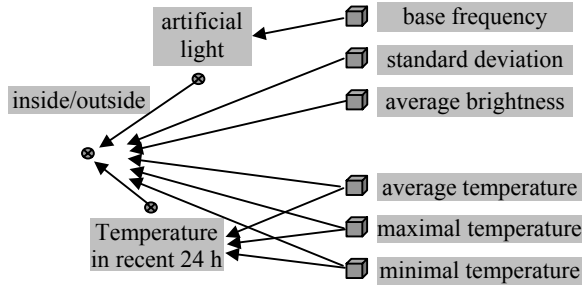


Figure 8. Deriving context inside/outside

The context “artificial light” indicates whether there are artificial lights in the current environment. The contexts “temperature in recent 24 h” describe the long-term statistic result of the temperatures in the past. We will simplify the decision process of “inside/outside” to show the reconfiguration of the awareness system.

In a normal situation, the decision tree of “inside/outside” is optimized by using the stored samples with all the attributes. In this decision tree, both the context artificial light and temperature related cues and contexts play important rolls (see figure 9).

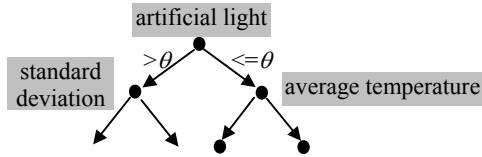


Figure 9. Decision tree for inside/outside

We discuss two reconfigure situations activated by disabling the context “artificial light” and switching off the temperature sensor. If the context “artificial light” is disabled by the host application, the decision tree has to be rebuilt according to the same stored samples but without the attribute “artificial light”. The similar reconfigure process will also be done when the temperature sensor is switched off. The decision trees in these three situations can produce the recognition results, which are described in table 2.

Table 2. Recognition results

context	Total number of test samples	Recognition rate-normal	Artificial light disable	Without temperature
inside	512	93.0%	81.7%	91.2%
outside	512	98.0%	89.4%	87.6%

#### 4. Conclusions & future work

The architecture presented in this paper is designed with a four-layer structure for multi-sensor fusion in mobile environments. The layered structure of the architecture allows the algorithms of the fusion system to be developed independently with sensors, data and the application demands. Through the interface defined between layers, the fusion algorithm face inputs with similar structure no matter whether they are real sensor data or the results of the other algorithms. The design of the layered architecture aims not only to develop the model to fuse the data from multi-sensor, but also to investigate the model to fuse the methods and techniques developed in the area of information fusion and other research area. Moreover, the layered structure makes it feasible to reconfigure the algorithms in each layer, which is important to enable awareness in mobile environments. The algorithms in the fusion system can be reconfigured properly to adapt the environment changes caused by the “movement” of the mobile devices, and produce more robust awareness results. Finally, the architecture keeps the interactions of host applications through different layers, which gives the opportunity for the host application to adjust the functions of the awareness device while also gives the chance for the fusion system to learn from the host.

Experimental results show that the awareness system we developed in this layer architecture performs robustly if all the possible situations of the mobile environment are known. If unknown situations occur in the environment, it is difficult for the system to produce the right and stable awareness results. The reason is that the awareness system can not find the new useful contexts in the environment by itself. Our future research will focus on application of data mining techniques in building the multi-sensor fusion system, which can adapt to unknown situations automatically. Furthermore, because the communication plays an increasingly important role in the application area of mobile devices, techniques for fusing the information from sensors with the information from communication channels will be investigated in our future work.

#### 5. Acknowledgement

The research described in this paper is supported by the EC under the ESPRIT program, project TEA. We would like to thank people at TecO, Starlab Nv/Sv, Nokia mobile phone, and Omega Generation for many discussions surrounding this work.

## References

- [1] G. Reynard, S. Benford, C. Greenhalgh, & C. Heath, "Awareness driven video quality of service in collaborative virtual environments", in Proceeding of ACM CHI'98, pp. 464-472, 1998.
- [2] S. Bly, S. Harrison, and S. Irvin, "Media spaces: Bringing people together in a video, audio, and computing environment", Communications of the ACM 36(1), pp. 28-46, 1993.
- [3] Esprit Project 26900, Technology for enabling Awareness (TEA), [www.omga.it/tea/](http://www.omga.it/tea/), 1998.
- [4] B. V. Dasarathy, "Information/Decision fusion – principles and paradigms", in Proceeding of the workshop on Foundations of Information /Decision Fusion, pp. 46-60, 1996.
- [5] B. V. Dasarathy, "Sensor fusion potential exploitation – innovative architectures and illustrative applications", in Proceedings of the IEEE, pp. 24-38, January 1997.
- [6] R. Luo & M. Kay, "Data fusion and sensor intergration: state-of-the-art 1990's", Data Fusion in Robotics and Machine Intelligence, pp. 7-136, 1992.
- [7] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, and M. Abe, "On sensor image compression for high pixel rate imaging", in Proceeding of IEEE international conference on Multisensor Fusion and Integration for Intelligent System, pp. 201-207, 1996.
- [8] H. Kabre, "Performance and competence models for audio-visual data fusion", SPIE international symposium on intelligent systems and advanced manufacturing, vol. 2589, pp. 100-107, 1995.
- [9] S. G. Goodridge, and M. G. Kay, "Multimedia sensor fusion for intelligent camera control", in Proceeding of IEEE international conference on Multisensor Fusion and Integration for Intelligent System, pp. 655-661, 1996.
- [10] R. Bajcsy, G. Kamberova, R. Mandelbaum, and M. Mintz, "Robust fusion of position data", in Proceeding of the workshop on Foundations of Information/Decision Fusion, pp. 1-7, 1996.
- [11] B. E. F. MacLeod, & A. Q. Summerfield, "Quantifying the contribution of vision to speech perception in noise", British J. of Audiology, Vol. 21, pp. 131-141, 1987.
- [12] R. R. Brooks, and S. S. Iyengar, "Multi-Sensor Fusion - fundamentals and applications with software", ISBN 0-13-901653-8, Prentice Hall PTR, 1998.
- [13] N. S. V. Rao, "Nadraya-Watson estimator for sensor fusion", Optical Engineering, vol. 36, pp. 642-647, 1997.
- [14] A. Schmidt, J. Forbess, "What GPS Doesn't Tell You: Determining One's Context With Low-Level Sensors", the 6<sup>th</sup> IEEE International Conference on Electronics, Circuits and Systems, September 5 - 8, 1999.
- [15] A. Schmidt, Z. Bandar, "A modular neural network architecture with additional generalization abilities for large input vectors", in Proceeding of international conference. on Artificial Neural Networks and Genetic Algorithms, pp. 35-39, 1997.