Project iWear - wearable IT solutions

Falko Buttler, Daniel Dahme, Andreas Hölke,
Christian Ober-Blöbaum, Roman Pesina, Carsten Rachuy, Jörn Reimerdes, Dennis Rohlfing,
Paul Schütte, Okhan Senkal, Tobias Warden, Xin Xing
TZI Center for Computing Technologies
University of Bremen
Universitätsallee 21-23, D-28359 Bremen, GERMANY
Contact: warden@tzi.de

Abstract - In the last few years the technology of wearable computer devices was brought forward by a considerable amount of innovation in the hardware sector, resulting in an increase of computing capability and miniaturisation. Beyond that, reseach started to consider such field as wearable interface and setup design with respect to human cognition and movement. The result can be seen in new wearable products available on the market as well as in dynamic prototypic developments which actually represent next steps towards true wearability. The project iWear participates in this process by introducing a new integrated wearable system which includes development of wearable user interfaces as well as carrying solutions for ergonomic long term use. This paper describes the efforts currently underway by introducing the iWear framework, a middleware which is developed as a basic plattform for wearable applications. Furthermore three applications and two hardware subprojects which are currently under development will be introduced and described.

I. INTRODUCTION

The iWear project started in October 2003 as a project of twenty nine students, supervised by two research assistants and one professor. The project's length is scheduled to four terms. The project concentrates on two aspects, first the research and development of mobile computing technology and second its integration into existing industrial workflows. One major goal of the project is to develop a basic system that will work as a middleware for wearable computers. The middleware system consists of four parts, three layered components which are responsible for sensor input recognition, user interface support and output data distribution, and one central system core which is responsible for input and output management, application coordination and network support.

Another goal is the development of applications which use this middleware. These applications should benefit from the underlying structure by accessing the wearable computer using the middleware system and so not having to cope with system specific low-level tasks like input/output management. A third goal of our project is the improvement of existing, as well as the development of new hardware which is to be optimised for the use with our system.

In the following the relation and relevance of our work for the topic mechatronic and robotics will be described in section II and afterwards a short description of the requirements which are met by our system will be given in section III. The presentation of our work starts with the description of our middleware system (core and layer structure) section IV followed by a description of the applications in section V, afterwards the hardware will be described in section VI. The paper finishes with a note on enhancement and support in section VII and look on the future work in section VIII.

II. RELATION TO MECHATRONICS & ROBOTICS

Within the last few years, computers have been reduced in size, making integration in small devices like pens, belt buckles, or small backpacks possible. Depending on the required by the application, even integration into clothing could be possible.

One major benefit of wearable technology for engineers is the improved possibility of viewing and accessing vital data on the fly. Using half-transluminescent displays it is possible to enhance reality with information like status reports and error messages so that, in case of an error, the engineer can access instruction manuals, reports of earlier errors with same symptoms and maybe solution proposals without the neccessity of interrupting his workflow. Once the problem is fixed, the appropriate information on how the problem has been solved can be stored and help speeding up the fixing of similliar/reoccuring errors. Another improvement of using wearable technolology is the possibility of live conferencing. This feature would allow experts who are currently off-site to participate in solving problems without the need to be physically present. There are several other possibilities where wearable computing can improve the workflow.

Before introducing the iWear middleware, acting as a foundation for our concrete wearable applications, we will first describe a set of general requirements which guide the overall design of our system.

III. Requirements Met by Our System

First of all the middleware as well as applications are consequently developed in modular fashion. This is due to the fact that, in spite of the complexity of the iWear middleware, flexibility should be maintained.Thus the development process remains flexible as the modules identified in the following sections can be developed independently from each other. This modularity is especially important for such modules which are physically connected to our system e.g. sensors or other hardware devices.

Since we expect frequent changes of our hardware setup during field tests, encapsulating appropriate code segments responsible for the control of the devices, enables quick changes without much effort. Another benefit of keeping the system architecture modular is that extending the iWear core system, the modules currently under development or the hardware modules is possible without affecting other components of our framework.

IV. iWear Middleware Structure

The iWear framework is designed to support applications for mobile computing with a comprehensive range of standard functions. Fig. 1 shows the structure model of the iWear framework. It consists of a predefined number of basic modules, providing some basic functions. If desired, additional modules enhancing the functionality can be loaded dynamically to the system as plug-ins.

A. Input

The input layer is used to support applications with different input methods. Since the user can't work with a keyboard or a mouse in all situations the input device has to be changed to one that fits into the appropriate context. While driving, for instance, the user has to keep his hands on the steering wheel, making haptic interaction not practical in this situation. Thus interaction concepts like speech should be preferred in this concrete situation. The input advisor system, given the current context and a list of available input devices, decides which input device is best to be used in this situation. To achieve this goal the system uses a naive proactive behaviour which relies on predefined situations and values. Furthermore the possibility to override this setup is always given, allowing the user to change his input device as desired.

B. Sensors

The sensor layer deals with the input that does not require any user interaction. It is devided into two parts, localisation and general input. Currrently the localisation part of the sensor layer supports WLAN, GPS, GSM and an acceleration sensor. The general input part supports sensors for temperature, brightness, shock vibration, sound intensity and identification. The layer has permenent access to all available sensors and provides their values for all applications if requested. The various sensor modules which are conected to this layer will always keep the last recorded value and will return it with its timestamp if no new data is available, which guarantees fast replies. The application itself can be configured to get more advanced or more precise data.

As instant access to a localisation hardware like GPS or WLAN is not always given, e.g. in buildings without WLAN, is is planned to compensate for this deficiency by using an acceleration sensor. The sensor will log the current direction and speed which allows a reconstruction of the position using the last valid position and the logged way of the acceleration sensor.

C. Output

The output layer is the interface for several different output devices. They are categorised as devices that are capable of displaying text, graphic, sound and simple signals. Text data can be presented to the user via different devices and ways of output. It can be displayed on

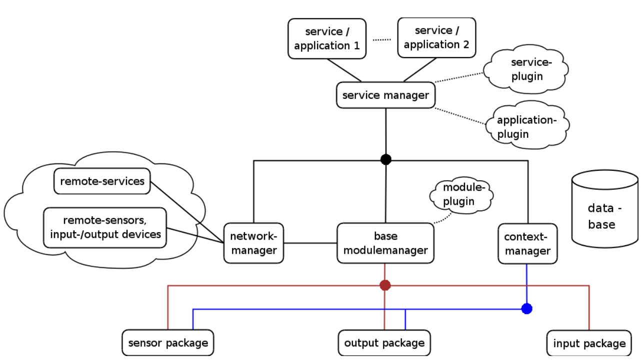

Figure 1: iWear framework structure diagram

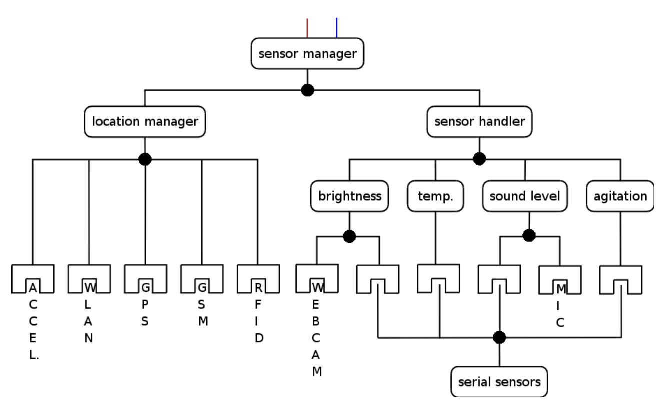

Figure 2: iWear sensor structure diagram

a screen or small LED arrays, presented as audio (text reader) and even through haptic interfaces like Braille displays or acoustic interfaces using morse code. Graphics and general audio streams are modes of information that are more complicated to transfer to other media, contrary to signals which can be used simply through any sense unless the perception is highly distracted or focused on another source of data. This leads directly to the output-advisor. Its function is to analyse the context and redirect display data from the preferred output device to a type that is more appropriate e.g. it would warn a user about an important incoming message or appointment, choosing tactile signals (vibration) or video signals if a high background noise level was detected.

D. Context

The described functionality of specifying and evaluating a context is provided by the context management system. It is a component which is able to either supply context information on demand or directly notify applications on changes. The term "context" is split into two groups: atomic contexts which are resolved by retrieving sensor data, and composite contexts which combine several atomic as well as composite contexts. The context management system will react on changes of atomic context information. Knowing the relation between atomic contexts and composite contexts, latter will be evaluated and the new context information becomes available.

The context information is provided by the context management system, a component which is, based on sensor input and/or user specific data, able to identify the context the system is currently in. This information can be accessed using the previously described context management system. Using the so provided context information, context sensitive and/or proactive system behaviour is made possible.

E. Network

The network component enables the framework to establish connections to similiar systems. To gain benefit from this ability, different devices present on different systems

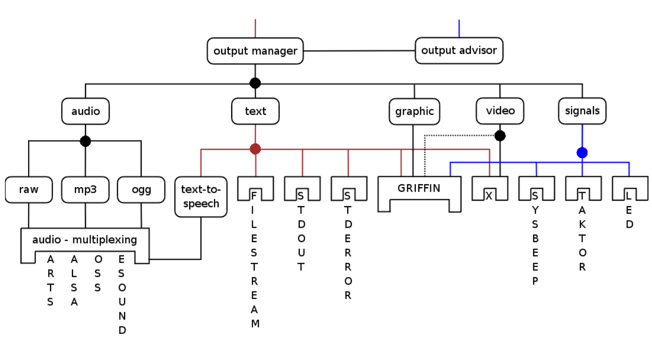

Figure 3: iWear output structure diagram

can be added to our system as remote or shared devices. For example sensors present on another system can be added to the sensor management as a remote service, enabling the context management system to use their information for context identification purpose.

V. IWEAR APPLICATIONS

Building the iWear framework some basic applications are developed as well. They will support the development of mobile IT solutions providing interfaces to a graphic user interface, digital maps and voice transmission.

A. GRIFFIN

GRIFFIN (which stands for GRaphical Interface For Fast Intuitive Navigation) is a graphical user interface (GUI) especially designed for wearable computers. The basic idea is to develop an interaction concept which is not relying on the well-known desktop metaphor of classic PC systems. Since GRIFFIN will be used on ultra-mobile wearable computers the interaction concept is restricted to only one degree of freedom for switching between options and the additional possibility to select/deselect/activate objects. We claim that, in spite of these restrictions, a highly optimised navigation concept based on hierarchical menu structures can still be effective in everyday mobile use. The navigation concept should only require a low cognitive effort from the users so that most navigational tasks can be performed intuitively. Due to the layout, the system is not as susceptible to noise caused by the users' normal movement as other pointing devices like a mouse or trackball. The interface will be mutable, so that various layout metaphors can be individually chosen either to the likes of the current user or the requirements of the current situation. The latter means that GRIFFIN is developed from the first sketches on as a context aware GUI-application. The context itself is provided through other parts of the iWear framework such as the context management system. This should enable a robust mapping from context to appropriate means of representation.

B. Map View

This module will be used to indicate the location of various objects (e.g. persons or places) on a simple map. There will be different kinds of styles for viewing the maps, depending on the users actual position. The maps are chosen from a database where various maps for both inside and outside of buildings are stored. Regarding the users' position the context manager decides which map has to be displayed on screen. Besides the user can override the context-managed choice by manually switching between maps. Additionally, the user has the option to define points of interest. These points will be associated to different categories, like production machinery, storage or similiar points of interest. The user is able to apply them for his own private use or to announce them for public access, so other users of the map module can access them. The visibility of these markings can be toggled on/off either manually or using the context management system. This module should help the user to find a target but will not provide the path to it. Instead it will just mark the appropriate location on the displayed map. Nevertheless route-planning software might also be an interesting add-on for future versions. If a target is situated outside the map's scope, the direction towards this target can be indicated by a coloured point at the map border. Several targets, static as well as moving, can be shown simultaneously.

C. Voice Transmission

The voice transmission system implements a communication platform which provides audio communication between several users. Dialogue partners are added to a system-wide communication pool using unique addresses, provided by the voice communication system. Every user has the ability to take part in dialogues either in an active or passive way. To ensure a fast voice transmission from one user to the other the voice data will be encoded using a free and open source codec system.

VI. IWEAR HARDWARE

A. Sensors

To enably connection of external sensors, either commercial or self developed, this subproject was initiated. The specially developed iWear sensor module which is shown in Fig. 4 is based on a 10 bit a/d modifier attached to the serial port (RS232). At the module, up to six sensors in arbitrary combination can be attached. The associated software recognises automatically which sensors are available. Arbitrary sensors can be used, as long as they supply resistance values appropriating to measured values. Currently sensors are available for the measurement of brightness and temperature.

The software for the iWear sensor module supplies measuring data, which are determined by sensors attached to the sensor module. The software automatically recognises sensors, which are attached in arbitrary combination. If several sensors of the same type are available, a measured average value is computed by sensor fusion. To supply measured values constantly and fast, these are buffered, so that in case of an request for information, no additional communication with the sensor module is required

B. SHADOW

SHADOW (which stands for Simple HAndbased interaction Device Optimised for Wearables) is a hardware project which aims at creating an alternative to the classic pointing devices currently common in wearable computer systems such as trackballs and twiddlers. The work

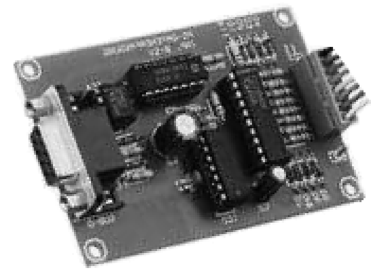

Figure 4: iWear sensor hardware-module

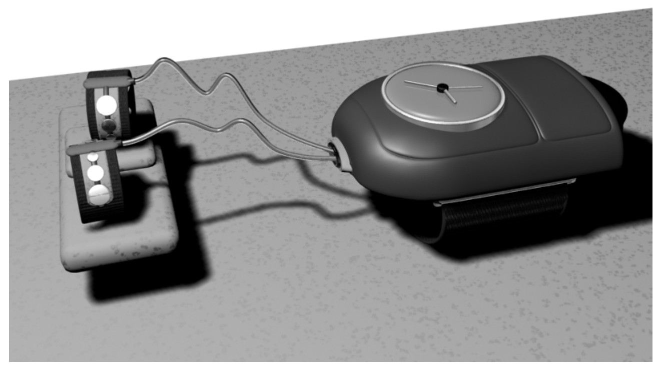

Figure 5: shadow hardware-module, cgi of final setup

was founded on the Winspect[1] system designed by the TZI[4] and is designed to be the hardware extension of the GRIFFIN-GUI since it refers to the ideas of a very simple, reduced command concept based on only one degree of freedom and one further switch for selection/activation. After building first mouse-based prototypes we decided to stick to this reduced functionality in order to evaluate whether such a minimal hardware interface can still be effective when backed up with an especially optimised software pendant.

Our vision is to build a device which is nearly invisible when worn by a user and doesn't limit the users' freedom of movement or touch in any way. While integrating the hardware in a watch-like device the "button" will be realised by two flexible rings worn on thumb and index finger. We also plan for a wireless connection with the wearable computer. A computer generated image of the final setup is shown in Fig. 5.

C. Backpack System

The aim of this subproject is the realisation of everyday wearability concepts for wearable computers. The basic motivation is the fact that commercially available

Figure 6: backpack hardware-module

wearable solutions as provided by such companies like Xybernaut[5] show disadvantages in putting on or off these systems as well as in wearing them over longer periods of time. We aim at reducing such inconveniences in wearability through a development process which firmly relies on ergonomy studies for the moving human form and aesthetics. Since the financial resources available are rather limited we decided to turn the fact into an advantage by coming up with low price wearable solutions which we can completely realise on our own.

The first prototype which currently is in development and can be seen on Fig. 6, is a backpack-based system which allows easy plug&play integration of Xybernaut wearable systems. The backpack is also designed to integrate as much as possible wearable periphery such as sensors and the battery. For such peripherals which can't be integrated, like input devices (SHADOW, pointing devices, keyboard) we plan to provide convenient plugin-interfaces reducing visible cable connections as much as possible. The backpack should be constructed in a robust fashion so that the user isn't restricted in his/her movements especially when it comes to fast (running) or abrupt (jumping) movements. The compliance with these requirements will be tested in the field and evaluated to generate further optimised concepts for future prototype development.

VII. ENHANCEMENT AND SUPPORT

Using the GNU General public License the project iWear takes advantage of the freedom to reuse sophisticetd GNU software. That fact has two implications which strongly influenced our decision for this choice of license during the first project term. Due to the restricted project length of just two years we decided it is not realistic to develop our middleware and applications completely from scratch. Rather it makes sense to rely on tools which are developed by volunteers. Using the GPL also makes sure that all code created in the project is also available for free and can be reused either by the participants of the project beyond its defined end or student groups from other universities. We also believe that working under the GPL can be an advantage in the cooperation with our project partners since customization of GPL software is easy due to the freely available source code.

VIII. FUTURE WORK

At the current state of our project, our main target is the completion of our core system and the described modules which is due to be finished at the end of September 2004.

After completion of the basic system components, a period of testing and evaluating the framework is scheduled in which both the software as well as the hardware modules have to be thoroughly tested. During this time modifications and changes of the system are to be expected.

The aim of the second stage of our project, which will be launched after this evaluation period is to develop applications which are based on our previously developed framework. One major goal is to develop applications which are not only suitable for personal, but also for industrial use. Early planning concepts include systems for identification, keydata management, maintenance and picking. In addition we plan the development of a wearable personal data assistant and a software which provides personal location based annotation services.

IX. REFERENCES

[1] Ingrid Rügge, Tom Nicolai, Michael Boronowski, “Computer im Blauman / Blue Collar Computing”, *Technologiezentrum Informatik, Universität Bremen*, (2002/2003)

[2] Ingrid Rügge, “Studie - Technologische und anwendungsorientierte Potentiale mobiler, tragbarer Computersysteme” / *TZI-Bericht Nr.24: Technologiezentrum Informatik, Universität Bremen*, (Januar 2002)

[3] Free Software Foundation Inc., “GNU's Not Unix!” http://www.gnu.org (viewed 2004.04.26)

[4] TZI Center for Computing Technologies, University of Bremen, http://www.tzi.de (viewed 2004.06.28)

[5] Xybernaut GmbH, http://www.xybernaut.de (viewed 2004.06.25)