



WalkMap: Developing an Augmented Reality Map Application for Wearable Computers

J. Lehtikoinen, R. Suomela

Nokia Research Center, Tampere, Finland

Abstract We have designed, implemented, and evaluated a map application for wearable computer users. Our application, called WalkMap, is targeted at a walking user in an urban environment, offering the user both navigational aids as well as contextual information. WalkMap uses augmented reality techniques to display a map on the surrounding area on the user's head-worn display. WalkMap is constructed by the means of software development, user interface design and evaluations, and existing knowledge on how humans use maps and navigate. The key design driver in our approach is intuitivity of use. In this paper, we present the design and implementation process of our application, considering human-map interfaces, technical implementation, and human-computer interfaces. We identify some of the key issues in these areas, and present the way they have been solved. We also present some usability evaluation results.

Keywords: Augmented reality; Context-awareness; Head-worn display; Human-computer interaction; Human-map interface; Navigational map; Wearable computing

Introduction

The development of small, personal digital devices is rapid. Advanced smart mobile phones, like Nokia Communicator, and handheld and palmtop computers are paving their ways into everyone's pockets, and even wearable computers have slowly started to emerge. With these various forms of mobile terminals and a wireless communication channel, we are able to gain access to computational resources and services regardless of the time and place.

One of these services is a positioning service. In short, a positioning service combines geographical data and position information to allow the service user to see himself/herself located on a digital map. This

development greatly enhances tasks such as *wayfinding* [1–9]. In addition to wayfinding, the location information is an essential part of *context-awareness* [10–14]. In short, context awareness can be understood as knowledge about user's current location, mental and physical state, environmental conditions, and so forth. By applying context-awareness, the computer can sense the changes in the using environment, and act accordingly.

A potential platform for a personal digital map is a wearable computer. Equipped with a high quality see-through head-worn display, in a future not so far away, a wearable computer user can have a totally personal, high-performance computer integrated into his/her clothing and available at all times. The system

can be operated while walking, it is aware of the user's current context, including position, and can assist the user in many daily tasks, including navigation in familiar or unfamiliar environments.

We chose a fresh approach to developing such a map. First, we were targeting at a system that should be wearable in each and every aspect, including user interface. Second, we studied the way people are accustomed to using conventional paper maps, and applied that knowledge to our digital maps. As a result, the user is offered an augmented reality context-aware software map application that can assist in wayfinding as well as allow access to contextual information.

Our goal was to develop an augmented reality context-aware wayfinding guide for a walking user in an urban environment. The guide should provide the user with information on the surrounding area, to assist in finding a route to a specified target location, and to prevent the user from getting lost. The user was expected to have a wearable computer equipped with a see-through head-worn display, and an input device for both discrete input (buttons) as well as continuous input. We did not expect the user to need to input any text while moving. Our application was named WalkMap.

When designing an application like ours, there are three primary issues that need to be addressed:

1. The human-map interface
2. the technical software and hardware implementation
3. the human-computer interface.

In the next section, we will present the human-map interface of WalkMap. The other two aspects are fully covered in the subsequent sections. Finally, we conclude the work after discussion.

Human-map Interface

Using a map is a complex task. Map use can be defined as 'the ability to relate a two-dimensional map to the three-dimensional world' [2], i.e. an ability to match the terrain and map models in one's mind. This ability includes the performance of several cognitive and spatial processes. Thus, not surprisingly it has been found that even adult map users often use maps inefficiently, and they do not feel confident with navigational maps [2]. Indeed, one challenge in developing a wayfinding assistant is to reduce the map reader's cognitive load and hence make the map using experience more positive.

The skills required for a successful navigation [3] can be stated as follows:

- to locate oneself on a map
- to recognise the destination
- to plan and execute the route to the destination.

To locate oneself on a map, the person must be able to perform the conversion of the 3D environment to the scaled 2D map, to match the objects in the environment with the symbols on a map, and to find the correct orientation. Only after that can the route planning and execution begin. At least this is the case with conventional paper maps.

Nowadays, more and more maps appear in electronic form. These digital maps are most often used in aircraft and cars, but are spreading into personal usage as well. A digital map consists usually of an electronic representation of the environment (the data), and a software component to interpret and display that information (the map application). The dynamic nature of the information behind the visualisation allows several various views of the same data.

Digital maps are able to free the user from some of the cognitive processes involved in navigation. They also provide the user with features that are not possible with paper maps, such as reacting to changes in the environment, or displaying the positions of other map users.

Positioning services are closely related to digital maps. A positioning service provides the map user with information on the current location. The most common positioning service is the satellite-based Global Positioning System, or GPS. Positional data can be acquired with a GPS receiver. The accuracy of a GPS service is usually around 7 m, but with differential correction it can be brought down to a few dozen centimetres. For an extensive explanation on GPS, refer, for example, to Getting [15].

The Alignment

One of the key aspects in navigation is *alignment*. Basically, this term defines how the user is holding the map, i.e. it specifies how the map is oriented with respect to the user and the environment.

A map may be *reader aligned*, in which case the orientation of the map remains constant with regard to the reader's body. A typical example of this is a reader holding the map with the print upright. In these cases, the map is usually misaligned with respect to the environment. In many maps, north is presented up on the map; therefore, this kind of map can be referred to as *north-up aligned*.

An *environment aligned* map, on the contrary, is oriented consistently with regard to the environment;

i.e. the north on the map always corresponds to the north in the environment. This implies that in order to keep the map aligned, the reader is required to turn the map in relation to himself/herself when turning in the environment. Environment aligned maps are often referred to as *forward-up* (or *track-up*) aligned.

The alignment is clearly the single most significant factor concerning spatial knowledge and hence map understanding. Fundamental research to point out this issue was conducted by Levine et al. [16]. They concluded that a map is easiest to use when it is aligned with the terrain. Similar results have been obtained in later experiments; see, for example, Warren et al. [9].

Since digital maps are usually shown on displays, they are very flexible when it comes to issues like alignment. It is the map software designer's responsibility to decide which alignment modes to implement, and how they are used. Some research has been done on these issues with digital maps.

Aretz [1] has studied the design of electronic displays. He performed a cognitive analysis on a pilot's navigation task in an aircraft in a simulated environment on a workstation. In the simulation, the aircraft remained in the middle of the screen while the map moved. He compared the forward-up and north-up approaches. The results showed that the alignment should depend on the task: tasks that primarily use the egocentred reference frame (ERF; the user's forward view) should use forward-up alignment (such as localization), whereas tasks that need a world-centred reference frame (WRF) should have north-up aligned maps (such as reconnaissance).

As discussed earlier, with digital maps the alignment mode can be changed any time; therefore, the map software designer can leave this selection to the user. However, we required that there is a proper default value for the alignment. Should this alignment be forward-up or north-up?

The default task in our case is targeted search. This task uses clearly the ERF; therefore, the forward-up is the correct choice and is on by default. However, the user can switch to any other alignment mode if he or she so wishes.

Automatic or Manual Alignment?

In WalkMap, the default value for alignment is forward-up. One question concerning the alignment remains, though. Should the alignment be automatic, or should the map remain static until the user manually updates the map? As pointed out by Aretz [1], the inconsistency of the rotating map display hinders the development of a cognitive map. In this respect, the map should not be aligned continuously. However, it is questionable

whether an average walking user in a city needs to develop a cognitive map.

Since no previous research has addressed this issue, we arranged a field experiment to find out the answer [17]. Our results showed that neither automatic nor manual mode should be preferred. This result was indicated both by the quantitative analysis (there were no statistically significant differences in performance between the two modes) and the user opinion survey: four users out of 10 preferred the manual mode, while six preferred the automatic mode. Therefore, the users should be offered both methods. The mode should default to automatic, however, since those users who performed the automatic mode section first had a tendency to simulate the automatic orientation functionality by constantly orienting the map manually. This could imply that over time, the automatic mode could turn out to be better.

The You-Are-Here Symbol

It is essential for the user to locate himself/herself on the map. With digital maps this task becomes trivial – the user can see the *You-Are-Here* (YAH) symbol on the screen on the coordinates indicated by the GPS service. However, how should the YAH symbol behave?

There are two basic ways to visualize dynamic positioning information on a digital map. Either

- the YAH symbol is drawn in a static position in the middle of the screen, and the map moves as the user moves, or
- the YAH symbol moves on the screen and the map image remains static. In this case, when the user is about to move out of the displayed area, a new image has to be created and the whole display has to be changed at once.

Clearly, the YAH symbol behaviour is tied to the alignment mode in use. In case of automatic forward-up alignment, it is not logical to rotate the map and to move the YAH symbol at the same time. Therefore, for a forward-up aligned map, the only viable choice is to move the map and keep the YAH symbol static.

It seems that the former method is better regardless of the alignment mode used. Kim et al. [4] have examined the effects on rotation and movement of the map in an automobile. They had a simulated driving environment to measure the arrival time, navigation efficiency, and map reading time. Among other things they found that the arrival time is shorter if the car is not moving on the map but the map moves and the car

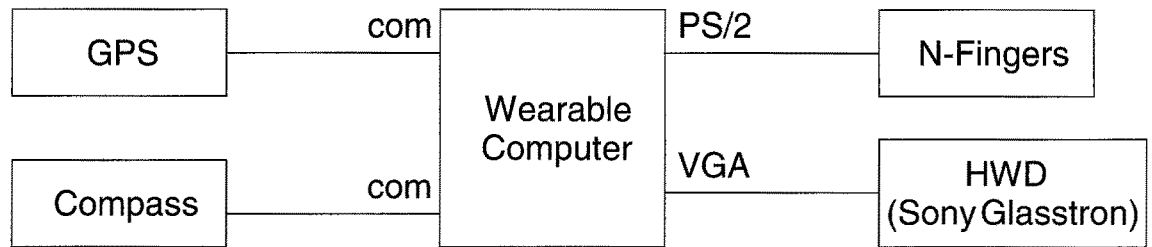


Fig. 1. The hardware components and their interfaces to the wearable computer.

symbol remains fixed in the middle of the screen. This was due to the fact that the changes in a display were constant and smooth. With a moving car, when the car reaches the edge of the screen, the whole display should be changed at once.

As a result, it is obvious to choose the former method: in WalkMap, the YAH symbol remains static while the map pans as the user moves. This is true regardless of the alignment mode.

Implementation

In this section, we will take a look at the way WalkMap is implemented. As a general principle, we apply iterative design in all our work. In other words, as soon as the requirement analysis has been completed, we design and implement the first working prototype. We then evaluate the prototype with an appropriate method, identify the required changes, and go on with the next revision. Even the requirements may be modified if the results so indicate.

Our Wearable System: An Overview

In order to develop the WalkMap application, we needed to have a working prototype of an entire wearable computing system, including hardware components, and software architecture. In this section, we will briefly describe these essential elements.

Hardware

Since our research focus was not on hardware, we used only COTS (commercial off the shelf) equipment. Our wearable computer was a Plug'n'Run NetCARD PC by Cell Computing with a 333 MHz Pentium II processor, 128 MB RAM, a compressed 340 MB IBM Microdrive, two serial ports, two USB ports, one parallel port, and two PCMCIA slots. In addition, the

computer was equipped with a Trimble Nav-GUIDE GPS receiver (accuracy about 7–10 m), and an electronic compass, Leica DMC-SXZ. The compass and the GPS receiver were connected to the serial ports of the computer. Our discrete input device, called N-Fingers, is connected to the PS/2 port. It is described in more detail in section 4.3.

A Sony Glasstron PLM-S700E see-through head-worn display (HWD) was used. We ran the display in 640×480×16 resolution at 60 Hz refresh rate. A schematic of the hardware configuration and interfaces is shown in Fig. 1. In Fig. 2, a user wearing the computer is shown. The components are installed on a belt with a shoulder string. The CPU unit and the battery are attached in front on the belt, while the display control unit and the GPS receiver are in the belt case behind



Fig. 2. A user wearing the computer.

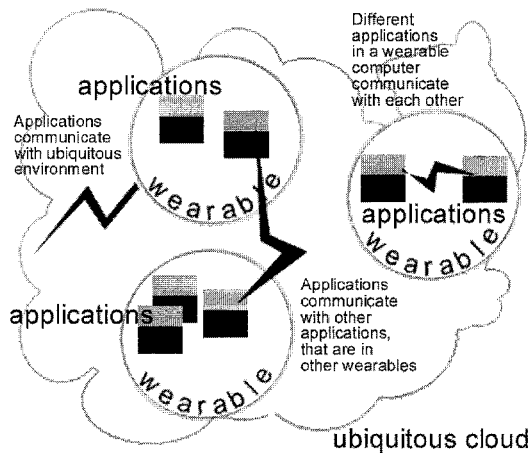


Fig. 3. MEX message delivery system.

the user's back. The compass is in the visor of the cap, and the GPS antenna is located in the shoulder string.

Software

We have designed and implemented a dedicated software architecture for wearable computers, called Message EXchanger, or MEX [18]. MEX is a low-footprint message delivering system. It allows easy connections between various software components in one or more computers. Each software component is a MEX service capable of sending and receiving messages through MEX (Fig. 3).

The MEX message delivery system allows easy addition and removal of components even when the system is running. The applications in MEX are composed of several different software components that together form an application. A typical application is composed of sensors, application logic, and a user interface. All the components are separate pieces of software. This approach allows us to customise the applications and reuse the components. The same component can be used in several different applications that might be running simultaneously. The distribution of the software components also allows us to experiment with different combinations of components.

WalkMap Software Implementation

WalkMap is built on top of the MEX system. The WalkMap software takes advantage of the service distribution provided by MEX. The WalkMap functionality is

divided into several separate software components that communicate through MEX. Some of these are presented in this section.

Virtual Objects

The main functionality that the user gains from WalkMap is the help in real world navigation and access to surrounding data. WalkMap creates a user interface that represents the surroundings to the user. The user interface is composed of a presentation of the environment and location dependent information.

A lot of research has been done on the contextual information. For instance, Schilit [14] studied dynamic environment objects. These objects belong to the current context and they can represent real or non-physical objects. Pascoe [13] studied the creation and manipulation of virtual objects. Pascoe's virtual objects, referred to as *stick-e notes*, contain GPS information and they can be created and accessed at the location.

We use *virtual objects* to present the information around the user. The virtual objects are a presentation of entities around the user. The entities can be real world objects, objects that only exist in the virtual space, or other users.

Virtual objects are stored in a network database. The user accesses this database at regular intervals or by manually requesting this information. The virtual objects have several properties that are required in order to be presentable to the user. The required properties are location coordinates, visibility range and object name. Important, but not necessary properties are a URL presenting the image of the object, a link to the content of the virtual object, and object type.

The location coordinates and the visibility range are used to determine when the user can see the object. The location of the object is presented or transformed to the same coordinate system as the user is using, such as GPS. If the user is inside the circle defined by the virtual object location and its visibility range, the object can be shown to the user.

The object type is used for filtering. The information space around the user can be composed of a vast number of virtual objects and not all of them need to be shown to the user. The user should see only the objects that he or she finds interesting. The virtual objects form an important part of the user context.

The name and the image are used when visualising the object to the user. The link is used if the user wants to access the information behind the object.

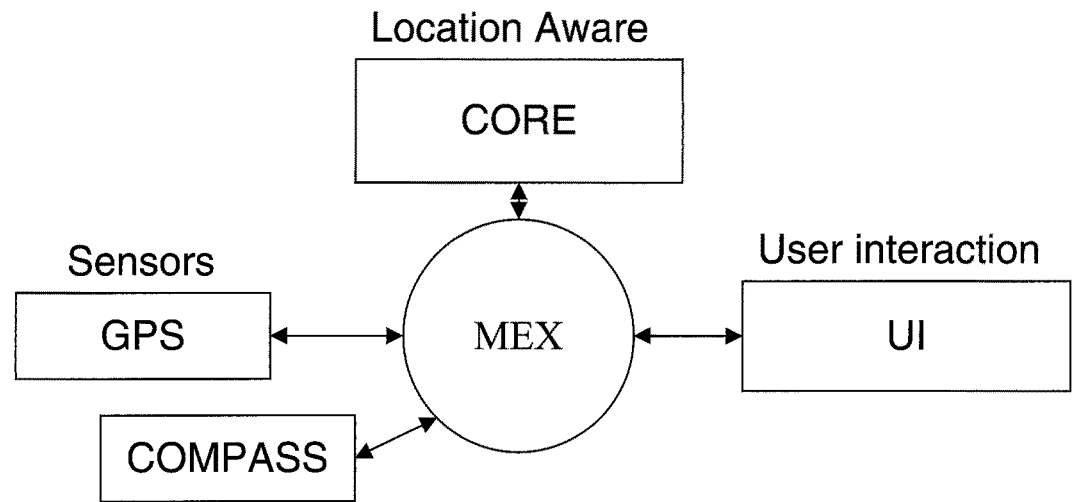


Fig. 4. The WalkMap software architecture.

Context Awareness: The Core Component

There are many definitions for the term 'context'. For our purposes, a good one has been presented by Dey [12]: 'Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications themselves'.

However, in this particular work, we do not take as wide an approach to context. The context in WalkMap is composed of the user's location and the virtual objects around him or her. As stated above, the objects exist in the real or virtual world, or they are other users. Therefore, as far as WalkMap is concerned, the context can be understood as 'A collection of entities in the limited virtual space around the user'.

WalkMap has a context-aware component that takes care of dealing with the context. As we are currently using only a limited definition of context, we thought there could be a need to extend our definition in the future. Thus, the context-awareness was built into one separate component. This component can be rewritten to take into account a wider definition of context including additional factors, such as temperature and weather.

The context-aware component constantly monitors the user's current location. It is also connected to the virtual object database. As the user moves, the context aware component examines the visibility ranges of the current virtual objects and updates the user's context space accordingly. The context-aware component is called the Core, as it contains all the core functionality of the application.

In addition to the virtual object database, the Core also accesses the map database containing geographi-

cal information. In our prototype system, the geographical data is stored locally on the user's terminal. However, it is straightforward to connect the Core into any existing remotely stored Geographical Information System (GIS) database.

Software Components for the Sensors

WalkMap exploits several sensors attached to the system. Each hardware sensor has a respective software component to feed this data to the MEX message system. The necessary sensors include compass and GPS receiver. These sensors constantly monitor the user's location and orientation. They both have threshold values that require a big enough change in order to send information to MEX. This threshold value is dependent on the hardware used. The more accurate and the less lag the component has, the more data we can feed to the system.

Summary

The complete software system can be seen in Fig. 4. In addition to the components described above, the user interface is also a separate software component. It receives the context space from the context aware component and sensor data from the different sensors attached to the system. Based on these, the user interface is created.

User Interface

WalkMap is supposed to be used while walking. Hence, the user interface should be as simple and

Table 1. Comparison of the different visualisations in WalkMap

Name/Property	Virtual Objects	Map	See through	Map Orientation
Blank	No	No	Yes	N/A
ContextCompass	Yes	No	Most	N/A
VectorMap	Yes	Yes	Some	North/forward up
BitmapMap	Yes	Yes	No	North up

intuitive as possible. Further, it should not require complicated actions that are hard to perform, or that may draw attention to the user on the street. All user interface components are required to follow these guidelines.

There are several user interface components in WalkMap. These include

- map's visual representation on the screen
- object selection on the map
- controlling the map behaviour.

These components are presented in this section. In addition, some issues concerning usability evaluation are discussed.

Visual Representation

The user has several choices on how the user interface is visualised. The general idea is to give the user a set of different user interfaces that have different see through capabilities. As the target user should be able to use the system while walking, there should be at least a non-blocking view of the real world. Since a (nearly) blank screen would probably not provide enough information on the surroundings, there should also be visualizations that provide more geographic information, but block more of the user's real world view.

The WalkMap user interface has several different levels of visualisations. Their key properties are compared in Table 1.

The different visualisations have different properties. On the one hand, the first two visualisations have less overall information available, but on the other hand they have much better see-through capabilities. These user interfaces should be used when the user is walking and knows where he or she is going. In this case the user requires much less navigational information and can just concentrate on walking.

Blank screen has no information presented on the screen. The user sees the real world unblocked, but the access to the context space is limited to audio only.

ContextCompass is a compass presentation of the surrounding contextual data. It requires very little space in the HWD and thus allows a good see-through properties. No geographical information is shown. The user only sees the virtual object locations in relation to him or her. The ContextCompass is described in more details in Object Selection below.

There are two full-screen map presentations in WalkMap. The Bitmap presentation shows a full-screen bitmap of the environment and the virtual objects are overlaid on it. The bitmap has no see-through capabilities and thus it cannot be used while the user is walking. One example of a Bitmap map can be seen in Fig. 5. There can be any number of 'skins' to the bitmap presentation, allowing specialized maps such as tourist maps, or maps for children. The number of skins depends on the GIS information available.

The Vector map uses a vector presentation of the surroundings. This map also requires full screen, but it is still somewhat semitransparent. The user can see through the geographical data as seen in Fig. 6. The virtual objects are also overlaid on top of this map and zooming is also allowed in this mode. The Vectormap can have different skins as well.

Vectormap presents streets in white and the buildings in yellow. The YAH symbol and the virtual objects are drawn on top of this.



Fig. 5. The Bitmap view. The virtual objects are laid on top of the map data.



Fig. 6. Vector map. The black parts are see-through in the head worn display.



Fig. 7. ContextCompass. The user has most of the view unblocked. Virtual objects are shown on the compass. This is a synthesised image.

Object Selection

The current context may consist of dozens of objects. Furthermore, there can be numerous objects visible on a map at any time. From these, the user should be able to select any object effortlessly. We have two solutions for this task.

The first is called ContextCompass [19]. The visual appearance is based on a linear compass shown on top on the head-worn display (Fig. 7). This compass covers a 110 ° area in front of the user (55 ° on both left and right sides).

Each object that belongs to the user's current context is visualised on the ContextCompass in their respective horizontal positions (the vertical position is equal for all objects shown on the compass, regardless of their real vertical positions). Hence, the object that is located straight ahead is shown in the middle of the compass, whereas other objects are shown on the respective side (left or right). In case an object is not in the 110 ° range, it is drawn on either side of the compass.

The object that is currently in the middle is highlighted and can be selected. This implies that any context object can be selected by simply turning one's head towards it. As the user turns his/her head, the objects on the compass are continuously updated.

Some key information on the selected object is shown to the right of the ContextCompass. This information includes the distance to the object, and a short description. This information, along with the objects' geographical coordinates and icons, is provided by the context-aware software component discussed earlier.

The user may also need to select an object while viewing a map. Therefore, in addition to ContextCompass, we also needed a method for selecting an object on a 2D plane. While this is straightforward to perform with any current pointing device (like the mouse), we wanted to develop a more simple way of performing the selection. As a result, the 1D selection technique was implemented.

In 1D selection, the user controls a visual circle on the screen. Once the selection tool has been activated, the circle radius can be manipulated; the middle point remains in the middle of the screen. The object that is currently on the perimeter of the circle is highlighted and can be activated. This way, only one-dimensional movement is required to select any object on the screen.

Controlling the Map

In addition to controlling contextual objects, the user will also need to control the map behaviour (i.e., to change the alignment settings, to change the visualisation modes, and so forth). He or she may need to initiate commands directly, or to browse simple menu structures. We have developed a dedicated interaction technique for such discrete commands.

Our interaction technique is called N-Fingers [20]. N-Fingers relies on the natural movements of fingers towards each other. There are four 'buttons' in the joints of the index finger, middle finger, and ring finger; these buttons are 'clicked' by thumb. This way, the user can have up to eight discrete buttons when both hands are used. Even though the current prototype is implemented by using physical snap fasteners and a somewhat clumsy copper plate on the thumb (Fig. 8), our extensive user evaluations [20] showed that for cursor controlling tasks, N-Fingers is as fast the keyboard cursor keys.

The hand icon visible in the Figs 5–7 is one form of displaying the available commands to the user. In Fig. 7, for example, the 'Open' command can be activated by

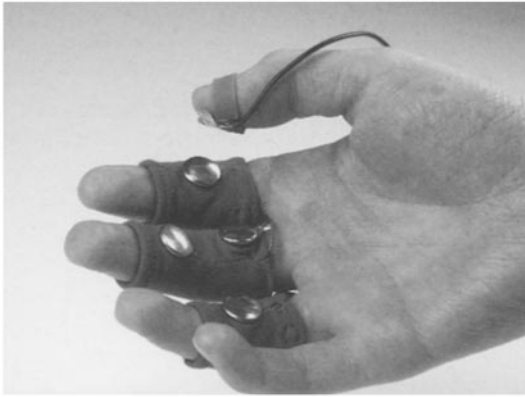


Fig. 8. N-Fingers.

clicking the button in the middle finger. As a result, the currently selected object is opened. The contents of the hand icon then change, reflecting actions that can be performed on the selected object.

Usability Testing

In order to achieve meaningful results from usability studies concerning wearable computing and augmented reality, at least some of the experiments should be arranged in real use conditions. To support the outdoor experiments, we have developed a dedicated usability evaluation software system [21], which takes full advantage of the customisation provided by MEX and WalkMap. Since all the key software components are separate services, we can have different combinations to be tested and to reuse the software components.

Undertaking usability tests of a wearable computer is a very difficult task. Wearable computers are very personal by nature and this poses problems for experiments. On the one hand, we wanted the user to use the system alone, as in a real-use situation. On the other hand, the test supervisor should control the test and know what is happening in the wearable computer. The solution to this problem is achieved by connecting the laptop of the test supervisor to the wearable computer. The test supervisor gets full access to the personal information on the wearable computer and by controlling the context-awareness, the test supervisor can control the entire test. This functionality was achieved by replacing the Core component (in Fig. 4) with a test control component. This component runs on the test supervisor's laptop and controls the test subject's user interface in such a way that the user will not notice any difference compared to normal usage.

The wearable computer and the test supervisor's laptop are connected via a wireless local area network (WLAN).

The user interface of the wearable computer is duplicated and the new situation is shown on the test supervisor's laptop. The supervisor has access to exactly the same user interface as the test subject and the supervisor can now observe the user both in the real and virtual worlds. The test supervisor also has a general information component that informs the supervisor about events occurring with the use of the wearable computer. This approach tackles the general problem of wearable computer usability testing by opening the personal virtual world of the test subject to others.

The software architecture of the test system is seen in Fig. 9. The basic system is the same as seen in Fig. 4 with a few minor changes. The core component is changed to the controller, there are two instances of the same user interface, and the INFO component is added to the test supervisor's laptop. The sensor data is stored into logfiles during the test. The data is synchronized and can be used after the test to replay the user interface as it was seen by the test personnel during the test.

The evaluation system described above has been successfully applied to WalkMap research [17]. We will now take a closer look at this experiment. It should be noted that the evaluation system is general and can be used to evaluate many different applications in addition to the one described here.

The purpose of the experiment was to find out which alignment method (automatic or manual) is better for a WalkMap user, i.e., to study human-map interaction. The experiment was carried out on a university campus area that consisted of several bigger buildings, roads, sidewalks, and parking areas. The users were asked to find virtual targets shown on the map, located around the campus area. The vector version of the map was used.

We had 10 subjects in the experiment, eight male and two female, whose ages varied between 18 and 30; the average age was 26. All users were familiar with, and had used, different types of map, including city street maps, road maps, and terrain maps. Each user also said that he or she knew how to use a compass. One user ranked himself a totally inexperienced map user, whereas the others thought that they had intermediate map-using skills. No one considered himself or herself a professional in map use. The environment was totally unfamiliar for four out of 10 users. The rest knew the area in more or less detail.

The equipment used in the experiment was our standard wearable computer, as described in the

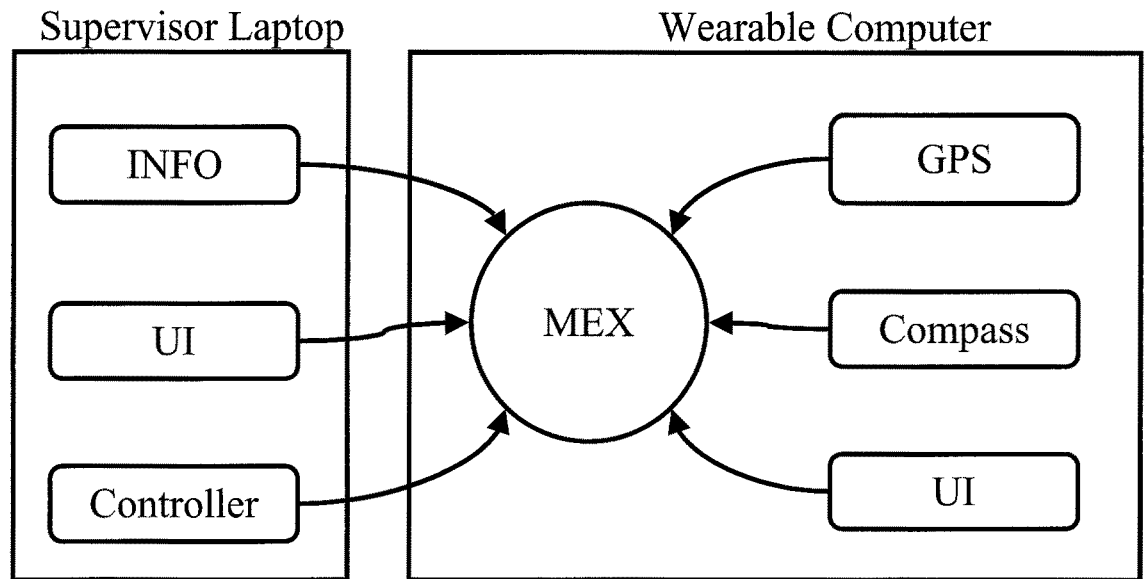


Fig. 9. The usability test software architecture.

section on our wearable system above. The test supervisor had a laptop computer. The usability evaluation system allowed the supervisor to see the subject's user interface, as well as to control the course of the experiment, without distracting the user's task execution. The supervisor followed the subject as he or she performed the tasks.

The map front-end running on the test subject's computer had two modes: manual orientation, and automatic orientation. In the manual orientation mode, the user had to align the map to forward-up manually, whereas in the automatic orientation mode the map was continuously aligned to forward-up. No other alignments were possible.

The user's computer was equipped with a Track-point-like pointing device. No pointing was required in this test, only two buttons were needed. With the first mouse button, the user could orientate the map in the manual orientation mode, and with the second mouse button, the map could be toggled between hidden and shown in both modes.

Each test session began with an interview. Along with basic data (age, gender, education) the user was asked about previous map and compass experience. We also wanted to know whether the user knew the environment in advance. After the interview, the rough course of the test was explained.

The test consisted of two parts. In each part, the user had to find 10 targets that existed in the virtual world only. He or she was shown one target on the map at a time. Once the user had walked to the target and indicated that it was found, the target was removed

from the map, and the next target was shown. The next target was always visible on the map without zooming (actually, zooming was disabled in this experiment). The user was allowed to make comments during the walk.

The actual experiment consisted of two sections: a manual orientation section, and an automatic orient section. Every other user performed the manual orientation first. Regardless of the order of the sections, the route was always travelled in the same order.

In the manual orientation section, the user was able to align the map forward-up by pressing one of the mouse buttons; the map remained in this orientation until it was oriented again. In the automatic section, on the contrary, the map was rotated continuously so that it was always kept aligned forward-up.

After the first section, the map behaviour changed. At this point, we also asked for comments and opinions such as whether the user used the map during the route (in contrast to checking the target location in the beginning and just trying to find the target without the map), how long it took to learn to use the map, how he or she used the orientation (in the manual orient mode), and whether the rotation of the map made the usage more difficult (in the automatic orient mode). After both sections, the user was asked to compare the two maps to each other. One test session took about one hour, including the interview, the introduction, and both test sections. The route was about 2 km long.

We measured the time between displaying a target and reaching a target. The number of incorrect target locations was counted. In the manual orientation

mode, we counted the number of orientations made.

The quantitative data analysis showed that there were no statistically significant performance or accuracy differences between the two alignment modes. Also the user preferences were roughly equal for both modes. Hence, the map should be equipped with both manual orient and automatic orient methods. However, since the users that performed the automatic mode test first used the orientation function in the manual mode significantly more than the others, one might conclude that over time, the automatic orientation might be preferred. Therefore, the automatic alignment was made default in WalkMap.

Discussion and Related Work

Several research projects on personal electronic maps have emerged. Some of these are based on wearable computers, while others rely on hand-held devices. There are context-aware guides, navigational assistants, and combinations of both. In this section, we present some of the most relevant previous research. We also discuss the differences to WalkMap.

A fine example of an electronic map is the Cyberguide by Abowd et al. [10]. Cyberguide is a mobile, context-aware tour guide. It is aware of the user's current position, and can assist the user in several ways. It is focused on context-awareness, and provides the user with the means to get additional information on the surroundings.

When it comes to functionality, Cyberguide is the closest equivalent to WalkMap. It provides the user with both a map and context-aware information. However, Cyberguide is implemented on a hand-held device, limiting severely the augmented reality approach. Furthermore, Cyberguide does not consider the overall system, including human-map interfaces.

The GUIDE [22] is somewhat similar to Cyberguide. It is based on a modified web browser running on a hand-held PDA. GUIDE is targeted at tourists, providing the user with navigational and context-aware information. Furthermore, it supports guided tours around the city.

Thomas et al. [8] have used a wearable computer and a head-worn display for terrestrial navigation. Their system is called 'map-in-the-hat'. Map-in-the-hat assists the user in navigation from point A to point B via several waypoints. No actual map is shown, but visual cues are displayed including current date and time, the

compass bearing, the current latitude and longitude, and the next waypoint.

Map-in-a-hat is an example of a pure wayfinding approach. No map is used, since the user only needs to know into which direction to go and how far it is to the next waypoint. In WalkMap, similar functionality can be achieved by adding waypoints along the route as virtual objects, and displaying these objects on the ContextCompass.

Touring Machine [23] provides information about the surroundings to the user, creating a touring machine. The system augments data onto the real world, supports a large space for users, and combines multiple display and interaction techniques. The system consists of a backpack computer and a hand-held computer. A head-worn display is connected to the backpack computer.

Touring Machine has two applications on separate computers. The 'tour application' runs on the backpack computer and takes care of the augmentation. The hand-held computer has a web browser that is used to show additional information on the surroundings.

Again, Touring Machine is similar to WalkMap. Our system shares similar goals, but we wanted several different visualizations on a single display, interaction capabilities, and an easy distribution model.

In general, the design approach is unique in WalkMap. We have considered the application domain from several different point of views, and developed a complete, integrated, and easily scalable map system for wearable computers.

Conclusions

We have developed an augmented reality navigation application for wearable computers, called WalkMap. WalkMap combines a navigation assistant and a context-aware information centre into one, highly integrated yet dynamic system.

WalkMap is targeted at a walking user in an urban environment. The human-map interface has been developed by applying results gained in previous research concerning maps and navigation, as well as arranging field studies. As a result, the map is forward-up aligned by default (although the alignment mode can be switched to north-up), and by default the map is aligned automatically (auto alignment can be turned off should the user prefer manual alignment). Furthermore, the user symbol is in the middle of the screen and the map moves as the user moves.

WalkMap was built on top of MEX architecture. MEX is a distributed message delivery system designed

especially for wearable computers. WalkMap is a collection of different software components that communicate via the MEX system. The components include sensors, user interface component and the context-aware core component. This distribution of functionality makes the system extensible, which was used when creating the usability testing functionality.

The user interface has multiple different visualisations to give the user's different amounts of geographical information and spatial data. The less screen space we used, the less geographical information was shown and vice versa. We managed to create an interface to spatial data with minimal screen space in ContextCompass. It only takes up little screen space yet provides the user with full access to the data in the current context.

In addition to the visual user interface appearance, we have also designed new interaction techniques for controlling the user interface and for object selection. All our interaction techniques rely on minimal design.

Acknowledgements

We thank Mika Røykkee, Aino Ahtinen, and Ilkka Salminen from Nokia Research Center for their contribution. The first author is associated with TaUCHI (The CHI group at The University of Tampere, Finland) and wishes to thank Kari-Jouko Rähkä, the head of the group, for comments and support. This work is partly funded by Tekes Technology Development Centre, Finland.

References

1. Aretz AJ (1991) The design of electronic map displays. *Human Factors* 33:85–101
2. Blades M, Spencer C (1987) How do people use maps to navigate through the world? *Cartographica* 24(3): 64–75
3. Board C (1978) Map reading tasks appropriate in experimental studies in cartographic communication. *The Canadian Cartographer* 15: 1–12
4. Kim B, Han SH, Nam K, Park J, Han S (1997) Presenting map information on a global positioning system. *Computers & Industrial Engineering* 33: 529–532
5. MacEachren AM (1995) *How maps work*. New York: The Guildford Press
6. Ottosson T (1988) What does it take to read a map? *Scientific Journal of Orienteering* 4: 97–106
7. Passini R. (1984) *Wayfinding in architecture*. New York: Van Nostrand Reinhold
8. Thomas B, Demczuk V, Piekarski W, Hepworth D, Gunther B (1998) A wearable computer system with augmented reality to support terrestrial navigation. *Proceedings of the Second International Symposium on Wearable Computers (ISWC)* 168–171
9. Warren DH, Rossano MJ, Wear TD (1990) Perception of map-environment correspondence: the roles of features and alignment. *Ecological Psychology* 2: 131–150
10. Abowd GD, Atkeson CG, Hong J, Long S, Kooper R, Pinkerton M (1997) *Cyberguide: A mobile context-aware tour guide*. *Wireless Networks* 33: 421–433
11. Abowd GD, Dey AK, Orr R, Brotherton J (1998) Context-awareness in wearable and ubiquitous computing. *Virtual Reality* 3: 200–211
12. Dey AK. *Providing architectural support for building context-aware applications*. PhD thesis, College of Computing, Georgia Institute of Technology, December 2000. Available online at <http://www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf> Last checked 16 October 2001
13. Pascoe J (1998) Adding generic contextual capabilities to wearable computers. *Proceedings of the Second International Symposium on Wearable Computers (ISWC)* 92–99
14. Schilit BN. *Context-aware system architecture for mobile distributed computing*. PhD Thesis, Columbia University, May 1995
15. Getting IA (1993) The global positioning system. *IEEE Spectrum* 30(12): 36–38, 43–47
16. Levine M, Jankovic IN, Palij M (1982) Principles of spatial problem solving. *Journal of Experimental Psychology: General* 111: 157–175
17. Lehtikoinen J (2001) An evaluation of augmented-reality navigational maps in head-worn displays. *Proceedings of the Human-Computer Interaction conference INTERACT01* 224–231
18. Lehtikoinen J, Holopainen J, Salmimaa M, Aldrovandi A (1999) MEX: a distributed software architecture for wearable computers. *Proceedings of The Third International Symposium on Wearable Computers (ISWC)* 52–57
19. Suomela R, Lehtikoinen J (2000) Context compass. *Proceedings of the Fourth International Symposium on Wearable Computers (ISWC)* 147–154
20. Lehtikoinen J, Røykkee M (2001) N-Fingers: a finger-based interaction technique for wearable computers. *Interacting with Computers* 13: 601–625
21. Suomela R, Lehtikoinen J, Salminen I (2001). A system for evaluating augmented reality user interfaces in wearable computers. *Proceedings of the Fifth International Symposium on Wearable Computers (ISWC)* 77–84
22. Cheverst K, Davies N, Mitchell K, Friday A, Efstratiou C (2000) Developing a context-aware electronic tourist guide: some issues and experiences. *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* 17–24
23. Feiner S, MacIntyre B, Hollerer T, Webster A (1997) A touring machine: prototyping 3D mobile augmented reality systems for exploring the urban environment. *Proceedings of the First International Symposium on Wearable Computers (ISWC)* 74–81

Correspondence and offprint requests to: Juha Lehtikoinen, Nokia Research Center, P.O. Box 100, FIN-33720 Tampere, Finland. Email: juha.lehtikoinen@nokia.com