ISWC 2005

THE NINTH INTERNATIONAL SYMPOSIUM ON WEARABLE COMPUTERS **OCT 18 – 21, OSAKA, JAPAN**

09

IEEE COMPUTER SOCIETY · KiSS · NAIST · bplatz · IEEE

IEEE Computer Society Order Number P2419
ISBN 0-7695-2419-2
ISSN Number 1550-4816

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by Bob Werner
Cover art production by Joe Daigle/Studio Productions
Printed in the United States of America by The Printing House

IEEE
COMPUTER
SOCIETY

IEEE

IEEE Computer Society
Conference Publishing Services
http://www.computer.org/proceedings/

An Event-driven Navigation Platform for Wearable Computing Environments

Masakazu Miyamae, Tsutomu Terada,
Yasue Kishino, and Shojiro Nishio
Grad. School of Info. Sci. and Tech., Osaka University
{miyamae, tsutomu, yasue, nishio}@ist.osaka-u.ac.jp

Masahiko Tsukamoto
Faculty of Eng., Kobe University
tuka@kobe-u.ac.jp

Abstract

Wearable computing environments are slowly becoming a reality, worn and used by a user wherever he or she goes. As a result, a lot of attention is given to navigation systems that provide various information according to the users' situation. In this paper, we identify the requirements of wearable navigation systems and propose an event-driven navigation platform to fulfill them. Since the behaviors of our platform are described in a set of event-driven rules, users can adopt a variety of attached devices and customize system configurations. Moreover, we have developed a prototype and made an experimental study for its evaluation. Using our system, service providers can easily construct flexible wearable navigations.

1. Introduction

In recent years, the downsizing of portable computers has attracted much attention in the field of wearable computing. Wearable computing is a computing paradigm where a user brings and uses his/her own computer wherever he/she goes. We assume that wearable computing has three characteristics: (1) *Hands-free*: users almost always browse information without using their hands because they are wearing the computer; (2) *Always on*: since the wearable computer is always powered while worn, users can use the computer whenever they want; (3) *Supporting daily life*: users utilize wearable computers to support activities in their daily lives.

Figure 1 shows the typical appearance of today's wearable computer. In the figure, the user brings her own computer in a backpack that she wears and can inspect information using a Head Mounted Display (HMD). The features of *Always on* and *Supporting daily life* enable a wearable computer to monitor the user's situation, such as her location and direction, exploiting various sensors (e.g. a GPS, a geomagnetic sensor, a gyro sensor). As a result, a wearable computer can provide information well-suited to the user's situation. Moreover, there is no restriction on browsing information because of the *Hands-free* feature.

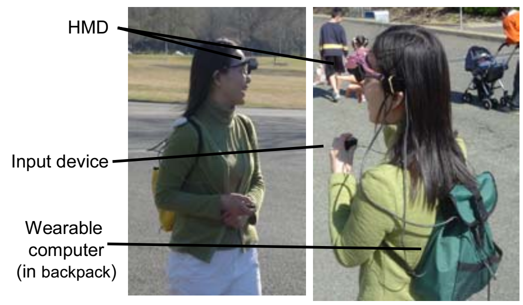


Figure 1. Style of wearable computing

Taking these characteristics into consideration, one of the most promising applications in wearable computing environments is a navigation system, such as a guidance system in a museum. Generally, a person interested in such a navigation service rents a mobile terminal specialized to the service [6]. However, in wearable computing environments, since the user provides her own wearable computer, she can also use it for navigation services by receiving navigation contents from the service provider. To realize such navigation services, the navigation content format has to be standardized. In addition, the device configurations of users' computers differ from one person to another because these computers are strongly personalized to the users. Therefore, the navigation platform needs a mechanism to flexibly accommodate various device configurations.

Keeping these requirements in mind, we propose a new navigation platform for wearable computing environments that enables service providers to construct and manage flexible navigation contents easily. Moreover, we fully implemented the navigation platform and verified its effectiveness by field tests.

The remainder of this paper is organized as follows. We explain navigation on wearable computing environments in Section 2, and Section 3 describes the design of our system. Section 4 explains the implementation and gives examples

Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05)
0-7695-2419-2/05 $20.00 © 2005 IEEE

of navigation systems developed using our platform. Section 5 describes considerations, and Section 6 presents our conclusions and future work.

2. Navigation for Wearable Computing Environments

In this section, we define and outline *wearable navigation* that implements a navigation service for wearable computing environments and a *navigation engine* that implements a system for processing wearable navigation.

2.1. Wearable Navigation

We call a navigation service for wearable computing environments *wearable navigation* and define it as a set of services that provide navigation contents when a user enters a specific *scene*. A scene exists at every place where we want to provide navigation contents, such as an intersection or a signboard. For example, wearable navigation includes a service to guide users by voices when she faces a signboard, a service to show an arrow for directions on her HMD when she enters an intersection, and a service to play an instruction video on her mobile display when she pushes a button while standing in front of an exhibit in a museum. Wearable navigation is expressed as a set of such scenes that corresponds to navigation contents.

2.2. Navigation Engine

We define *navigation engine* as a system that provides wearable navigation services and has the following three characteristics:

- **Autonomy**: it automatically provides information according to the situation of the user.
- **Device Flexibility**: it can adopt to a variety of attached devices.
- **Customizability**: users can customize how contents are displayed.

Since a wearable computer is always powered, a user can watch contents using her HMD and headphones whenever she wants. Thus, the navigation engine needs autonomy to handle not only information requested from users but also automatic indications according to user location and situation. Moreover, a wearable computer has the diversity of connected devices because it is specialized to one user. For example, restricted to position tracking devices, there are many types of devices, such as GPS, RFID, and integrated sensors including a GPS and a geomagnetic sensor. Therefore, a navigation engine needs device flexibility to adopt the diversity of connected devices. Furthermore, it needs customizability that enables the user to define the relation between output devices and type of contents. For example, when the navigation engine provides audio content to a user who brings a speaker and headphones, the user should be able to select which device is used for playing the content.

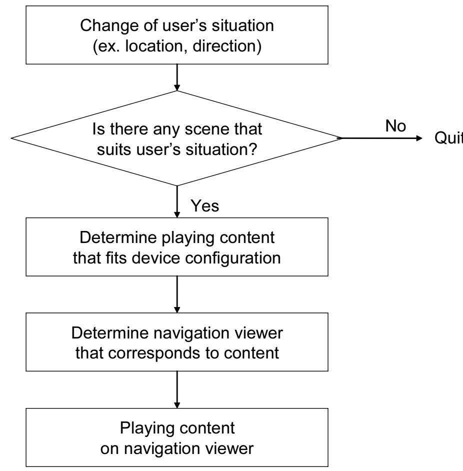


Figure 2. Flow diagram of wearable navigation

2.3. Navigation Procedure

Figure 2 shows a flow diagram of processing wearable navigation services by navigation engine. First, the engine extracts a corresponding scene to a situation when the user's situation changes. If there is an extracted scene, the navigation engine selects the optimum content according to the device configuration of the user's computer. The relationship between the device configurations and contents is defined for every scene by contents creators. For example, a creator prepares three kinds of contents: audio navigation, animation with arrows, and navigation video with sound. The creator can configure the relationship, for instance, making the engine select video with sound when the user wears a display device *and* an audio device, or it selects a corresponding content when she wears only one of these devices. After the content is determined, the engine selects which *navigation viewer* is used for playing the content. The navigation viewer includes software and hardware to play navigation contents, such as a WWW browser, a Windows Media Player, and a map viewer. For example, when navigation content is represented as an image, the navigation engine enables the user to select a navigation viewer depending on the type of contents, such as displaying the image on a WWW browser. In this way, we achieve device flexibility and customizability by separating the decision of play-

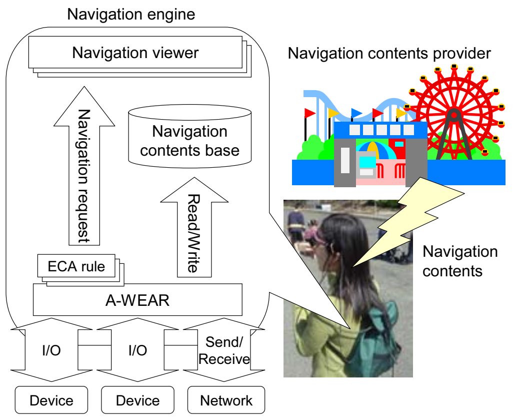


Figure 3. Structure of navigation engine

```
DEFINE Rule-ID
WHEN Event-type [ (Target-of-event)]
IF       Conditions
THEN DO Actions
```

Figure 4. Syntax of ECA rules

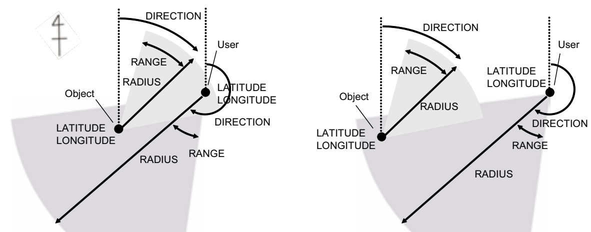


Figure 5. Criteria for entering a scene

ing contents into two phases: the selection of playing contents and the decision of a navigation viewer.

3. Design of Navigation Engine

In this section, we describe the navigation engine design.

We constructed the navigation engine based on A-WEAR [7], which is a rule processing engine for wearable computing environments proposed in earlier research. Using A-WEAR, since system behavior is described in a set of event-driven rules, we can construct systems that satisfy autonomy. Moreover, users can customize system behavior by adding/deleting rules and extend system functions to handle various devices by a plug-in mechanism. Because of these advantages, we can construct a navigation engine that fulfills device flexibility and customizability.

Figure 3 shows the system structure of the navigation engine. The engine receives contents from the navigation contents provider and stores them in the *navigation contents base*. The engine extracts appropriate contents according to inputs from such devices as a GPS, and requests that the navigation viewer shows them.

In the next part of this section, we detail A-WEAR, which is the basis of our system, and describe the criteria for a user to exist in a scene. After that, we explain the method of selecting a suitable navigation viewer. Finally, we describe a scene editor that helps creators to construct navigation contents.

3.1. A-WEAR

We proposed a rule processing engine for wearable computing environments called A-WEAR whose behavior is represented as a set of ECA rules. Each ECA rule consists of three parts: *Event*, *Condition*, and *Action*. Event describes the occurring event in the system, condition describes the conditions for executing actions, and action describes the operations to be carried out. Figure 4 shows the syntax of ECA rules for A-WEAR. In the figure, *Rule-ID* describes the name of the ECA rule, and *Event-type* describes the name of the event that triggers this rule. *Conditions* specifies conditions for executing the following actions, and we can use AND and OR operators in *Conditions* for describing complicated conditions. *Actions* specifies executing operations and their arguments. Events and actions that can be used are defined by plug-ins, which are extension modules for the system. In other words, we can use new events and actions in ECA rules by adding plug-ins. By employing such a plug-in mechanism, we can enhance such functions as the adaptation of a new device by adding a plug-in for the device.

Table 1 shows the functions of several plug-ins that we have already implemented. In the table, EVENT and ACTION describe events and actions that the plug-in provides, respectively.

In the navigation engine, every device for tracking user activity generates the same event per device category. For example, all direction tracking devices (e.g. a geomagnetic sensor, a gyro sensor) generate *ROTATE* events when they detect change of directions. Moreover, when we use an RFID reader for position tracking, the RFID plug-in converts the tag ID to a pair of latitude and longitude coordinates and generates the *MOVE* event. As a result, the system can handle user activity in a common format.

Database Plug-in

Type	Name	Content
EVENT	SELECT	Acquire the data
	INSERT	Insert the tuple
	DELETE	Delete the tuple
	UPDATE	Update the tuple
ACTION	QUERY	Database operation

Current Position (GPS) Plug-in

Type	Name	Content
EVENT	MOVE	Change of position

Common Plug-in

Type	Name	Content
EVENT	CMN_START	System initialize
	CMN_TIMER	Timer expires
ACTION	COMM_EVENT	Generate specific event
	CMN_SET_TIMER	Set a timer
	CMN_KILL_TIMER	Kill a timer
	CMN_ADD_RULE	Insert rules
	CMN_LOAD_PLUGIN	Load a plug-in

Direction (motion sensor) Plug-in

Type	Name	Content
EVENT	ROTATE	Change of direction

Network Plug-in

Type	Name	Content
EVENT	NET_RECEIVE	Receive the data
ACTION	NET_UNICAST_SEND	Send the data
	NET_BROADCAST_SEND	Broadcast the data

RFID Plug-in

Type	Name	Content
EVENT	MOVE	Change of position
	DETECT	Detection of an RFID

Table 1. Details of plug-ins

Location table

attribute	content
ID	Location ID
LATITUDE	Location latitude
LONGITUDE	Location longitude
DIRECTION	Area direction
RADIUS	Area radius
RANGE	Angle range

Scene table

attribute	content
ID	Scene ID
POS	Location table ID
XML	Scene content

Viewer table

attribute	contents
TYPE	Content type
VIEWER	Action to viewer
PARAM	Paramaters for viewer

Table 2. Database schemata

3.2. Criterion for Presence

In wearable navigation, we stipulated that the trigger for presenting contents is user participation in a scene. In this section, we define the criteria for participating in a scene. Since we want to handle such directional objects as a signboard, we define the area of a user's vision and the area of scene in sector form. Figure 5 shows an example of the criteria, which is an area of a user's vision and a scene consisting of a set of latitude, longitude, direction, range, and radius. If the origin of the user exists in the scene's area and if the scene exists in the user's field of vision, the navigation engine determines that the user belongs in the scene. In Figure 5(a), the engine determines that the user is in the scene area. On the other hand, in Figure 5(b), since the origin of the user is not in the scene area, the engine determines that the user does not belong in the scene.

The navigation engine provides services by referring to the scene information stored in the navigation contents base where location and scene information are stored in the location and scene tables, respectively. Table 2 shows the schemata of these tables. The POS attribute in the scene table stores the corresponding location ID in the location table. The XML attribute in the scene table stores some pairs of device configuration and content in XML format. Details of the XML attributes are described in Section 3.3.

Figure 6 shows rules for scene extraction. When the direction of the user changes more than $10°$, the *OnRotate* rule retrieves scenes near the user. The *OnMove* rule also retrieves scenes when the position of the user changes more than $0.0001°$ in latitude or longitude. The *CheckPosition* rule decides an appropriate scene from retrieved scenes. The *DoNavigation* rule displays the scene ID.

```
DEFINE OnRotate
WHEN ROTATE
IF !CMN.IS_IN_BOUND[NEW.DIR, OLD.DIR, 10]
THEN DO DB_QUERY('SELECT * FROM PositionTable WHERE
  LATITUDE > %GPS.LATITUDE% - RADIUS
  AND LATITUDE < %GPS.LATITUDE% + RADIUS AND
  LONGITUDE > %GPS.LONGITUDE% - RADIUS
  AND LONGITUDE < %GPS.LONGITUDE% + RADIUS')

DEFINE OnMove
WHEN MOVE
IF !CMN.IS_IN_BOUND[NEW.LONGITUDE,OLD.LONGITUDE, 0.001]
  OR !CMN.IS_IN_BOUND[NEW.LATITUDE,OLD.LATITUDE, 0.001]
THEN DO DB_QUERY('SELECT * FROM PositionTable WHERE
  LATITUDE > %NEW.LATITUDE% - RADIUS
  AND LATITUDE < %NEW.LATITUDE% + RADIUS AND
  LONGITUDE > %NEW.LONGITUDE% - RADIUS
  AND LONGITUDE < %NEW.LONGITUDE% + RADIUS')

DEFINE CheckPosition
WHEN DB_SELECT(PositionTable)
IF ?CMN.IS_OBJECT_VISIBLE(GPS.LONGITUDE,GPS.LATITUDE,
  0.05, DIR.ALPHA_NORTH, 60, NEW.LONGITUDE, NEW.LATITUDE,
  NEW.RADIUS, NEW.DIRECTION, NEW.RANGE)
THEN DO DB_QUERY('SELECT * FROM SceneTable
  WHERE POS = %NEW.ID%')

DEFINE DoNavigation
WHEN DB_SELECT(SceneTable)
THEN DO CMN_ALERT('You are in the scene of ID %NEW.ID%.')
```

Figure 6. ECA rules for scene extraction

```
<CONTENTS DEVICE="BBBAABCCCC" TYPE="SOUND"
  PARAM="FILE=http://192.168.0.1/navi01.wav"/>
<CONTENTS DEVICE="ACCCCCCCCC" TYPE="WEBPAGE"
  PARAM="URL=http://192.168.0.1/navi01.html"/>
```

Figure 7. Example of contents description

3.3. Decision to Present Content

The contents creator can specify relations between the device configurations of a user and present contents by describing these relations in the XML attributes of the scene table. Device configuration is specified by a set of states for each output device and category in the following three values: (A) must have, (B) not allowed to have, (C) does not matter. In our system, we suppose three categories and seven practical devices: image output devices (a HMD and a small display such as a wrist watch display), audio output devices (a headphone and a speaker), and actuators (a vibrator, a buzzer, and a light). Figure 7 shows an example of a scene description including device configurations and contents descriptions. The CONTENTS tag describes content,

```
DEFINE DecideViewer
WHEN DISPLAY_NAVIGATION
THEN DO CALL_VIEWER(
 DB.ViewerTable.VIEWER TYPE=%NEW.TYPE%,
 DB.ViewerTable.PARAM TYPE=%NEW.TYPE%, %NEW.PARAM%)
```

Figure 8. ECA rule for viewer decision

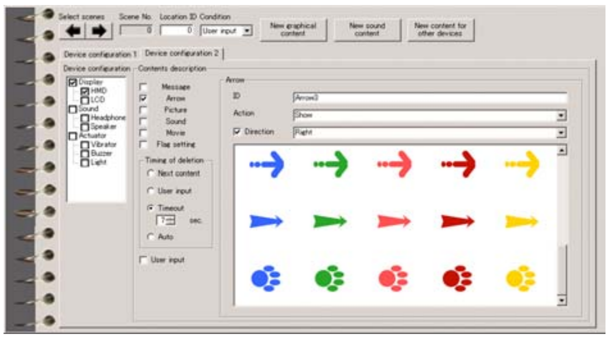


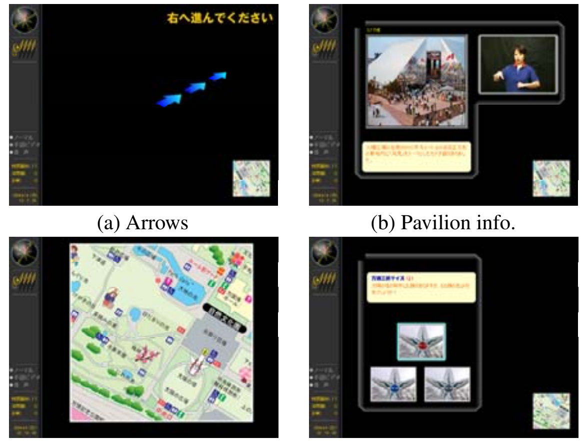
Figure 9. Screenshot of scene editor

and the DEVICE attribute describes device configuration. The former content has higher priority. As shown in the figure, description “BBBAABCCCC” means device configuration where no device in the image output category is attached (**B**BB), headphones are attached (A**A**B), and actuators are not selected (**C**CCC). The TYPE attribute describes the type of content, and PARAM attributes explain the content.

The navigation engine decides the best content by comparing the device configuration of the computer and the description in the DEVICE attribute.

3.4. Decision of Navigation Viewer

After the navigation engine decides which content to exhibit, it selects a navigation viewer to provide the content by referring to the viewer table shown in Table 2. TYPE attribute represents the type of content, VIEWER attribute describes the actions of the ECA rules to call the navigation viewer, and PARAM attribute describes parameters used when calling the navigation viewer. Figure 8 shows a rule that selects a viewer by extracting information from the viewer table. When we want to add a new navigation viewer to the navigation engine, we only have to add a tuple to the viewer table. Moreover, users can change the relation between types of contents and navigation viewer selected automatically by modifying the viewer table.



(a) Arrows (b) Pavilion info. (c) Map (d) Quiz

Figure 10. Screenshots of navigation system for Expo’ 70 Commemoration Park

3.5. Scene Editor

When we create contents for wearable navigation, we have to set many parameters to provide flexible services. We have therefore developed a scene editor to facilitate the creation of contents. Figure 9 shows a screenshot of the scene editor. We can easily specify a scene with a GUI, and the editor automatically creates tuples for location and scene tables. Moreover, since the editor can obtain data from the GPS and a geomagnetic sensor in cooperation with A-WEAR, we can create contents with such information to decide scene location. Therefore, we can construct navigation contents in situ.

4. Implementation and Practical Use

In this section, we describe the implementation of our system and explain the actual navigation contents created using the implemented prototype.

4.1. Implementation

We implemented the system based on the design described in Section 3. We used A-WEAR with a database plug-in, a network plug-in, a current position plug-in, and a direction plug-in previously implemented. Moreover, we enhanced the common plug-in to add presence criterion functions and implemented them using Microsoft Visual C++ .NET 2003 Enterprise Architect.

4.2. Examples of Navigation Contents

We developed three navigation systems using the implemented navigation platform in the following steps.

1. A navigation system programmer extracts several plug-ins according to such requirements as devices that he/she wants to use.
2. He/she may implement new plug-ins if he/she needs functions that do not exist (e.g. functions to display 3D CG).
3. He/she describes ECA rules if he/she needs specialized behaviors for the navigation system (e.g. interactive functions).
4. A contents creator produces contents such as web pages or sounds.

We offer details of the developed navigation systems.

4.2.1. Expo Park Navigation System We created navigation contents for the Expo' 70 Commemoration Park in Osaka, Japan. Figure 10 shows screenshots of the displayed contents. In this system, a user wears a GPS and a geomagnetic sensor to reveal location and direction. She walks around in the park according to the navigation. When approaching an intersection, arrows are displayed to guide her (Figure 10(a)). When she approaches an area where a pavilion used to exist for the EXPO (now, a memorial stone rests there), the system automatically plays an introduction movie for the pavilion (Figure 10(b)). Moreover, we provide many quizzes about the park, which are displayed as she approaches specific areas (Figure 10(d)). We created 60 scenes (19 arrows, 18 quizzes, 23 videos).Each scene has three types of contents: normal contents including pictures, texts, and sounds; sound content for blind persons; movie contents in sign language for the deaf. The presented contents are selected automatically according to the device configuration of the wearable computer. The system plays normal contents if both an image output device and an audio output device are attached to the wearable computer, sound contents if only an audio output device is attached, and movie contents in sign language if only an image output device is attached.

We implemented a navigation viewer specialized to this navigation by using Macromedia Flash MX. The viewer has functions that not only display navigation contents but also a map with user's location when she makes a request (Figure 10(c)) and processes input from buttons for answering a quiz.

We demonstrated these navigation contents at Cyber Communication 2004 [3] held at Expo' 70 Commemoration Park on March 25th, 2004. Cyber Communication 2004 was a symposium on barrier-free navigation systems for the park. We evaluated our navigation system in conjunction

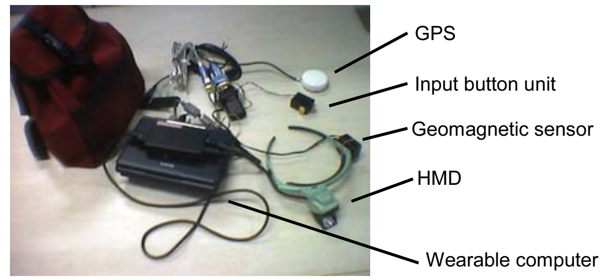


Figure 11. Wearable computer in practical use

Figure 12. Snapshot of navigation system for our laboratory

with the symposium. Evaluation participants included 20 invited handicapped persons and 60 other persons including symposium participants and visitors. They experienced our system for about an hour, and we evaluated the system for six hours.

Figure 11 shows the wearable computer used in this evaluation. We used four small customized computers, four SONY PCG-U3s as wearable computers, and a SHIMADZU Data Glass 2 as an HMD of the wearable computer. We also used RIGHT STUFF GPS-USB-RA as a GPS and Sensation VectorCube (VC-03) as a geomagnetic sensor. Moreover, we made small input devices consisting of two buttons for user operation. The user placed a wearable computer in her backpack and a GPS on her shoulders, and a geomagnetic sensor was fixed to her HMD.

4.2.2. Laboratory Navigation System We created navigation contents that provide information about our laboratory and its members. Figure 12 and 13 show a screenshot and a snapshot of this system respectively. Since it is difficult to use a GPS indoors, we used RFID tags and an RFID reader. A visitor attaches an RFID reader to her wearable computer and experiences the navigation contents. We used a WWW browser as the navigation viewer. When the RFID

Figure 13. Screenshot using a navigation system for our laboratory

Figure 14. Snapshot of using a navigation system for a museum

reader detects a RFID tag, the system acquires location information from the ID and displays the layout of our laboratory when the ID is the entrance. If the ID is the desk of a member, the system displays a web page about his research and information on his location.

4.2.3. Museum Navigation System We created navigation contents that provide information about exhibits for a museum and a gallery. Figure 14 shows an image of this system. We use a barcode for position tracking in navigation and lent barcode readers to visitors at the entrance. When users scan a barcode near an exhibit, the system plays sound contents that provide detailed information about the exhibit. In the museum, since the autonomous information provision based on user motion may prevent him from viewing, the system provides information only when he wants it by reading the barcode. Generally, since users concentrate on watching exhibits in a museum, we provide information only by sound using a MP3 player as navigation viewer. However, if the user wears an HMD, the system displays movies using a media player.

	Question	Average
1.	I usually use a computer.	4.1
2.	It was comfortable to wear a computer.	3.4
3.	Cables were not obstacles.	2.2
4.	I was able to clearly watch contents on HMD.	2.3
5.	It was easy to operate the system.	3.0
6.	Displaying information on HMD is helpful.	3.9
7.	The system helps visitors to learn the details of the park.	3.5
8.	The system is more helpful than guidebooks.	3.5
9.	The system is more interesting than guidebooks.	4.0

Table 3. Aggregate statistics of questionnaire

5. Discussions

5.1. Evaluation

In practical uses of the Expo park navigation system described in Section 4.2.1, we handed out questionnaires and got results from 39 people. Each participant was encouraged to rank (1: worst, 5: best) every question and to freely write comments. Table 3 shows the aggregate statistics of the questionnaires.

Questions 2–5 are about the hardware of the wearable computer, and questions 6–9 are about the effectiveness of wearable navigation. From the results, although the hardware weight is not a big problem, we need to make the hardware more comfortable. Since our navigation engine has the flexibility to accommodate changes of device configuration or the addition of new devices, it can easily adapt to future extensions. On the other hand, we got high marks for the effectiveness of wearable navigation. We consider that this result indicates that automatic contents provision works effectively. As there were many positive comments on the helpfulness and novelty of our system, we strongly feel the mechanism of wearable navigation has enough expressive power for navigations. Moreover, there were various suggestions from people: using the system in a shopping center, controlling the system by voice, chatting with other people in the system, and sharing information with their friends. The requests for additional features can be straightforward to incorporated in our navigation engine, and in future versions, we plan to improve it on these very points.

5.2. Related Work

There has been much research on wearable systems: MARS [4], TOWNWEAR [8], VizWear [5], and a system by Tenmoku [9]. These systems have achieved autonomy and provide high-quality services such as displaying virtual objects and annotations that overlap the real world on a HMD. On the other hand, since the services of these systems are specialized to one purpose and device configuration is fixed, they lack flexibility and customizability. Moreover, they did not consider a designer-friendly way of creating contents.

There is also a lot of research on user navigation using portable devices such as LoL@ [2] and Cyberguide [1]. These systems guide a user by relying on his location. LoL@ was developed for outdoor use, and it acquires location using GPS and signal power levels between the device and a mobile station. Cyberguide was developed for both indoor and outdoor use, and it can accommodate various location-aware devices by extending a *positioning component*. However, the output devices of these systems are fixed because these systems use specific portable computers such as PDAs. Since there is a variety of device configurations in wearable computing environments, the system should accommodate not only various input devices but also various output devices.

6. Conclusion

In this paper, we proposed an event-driven navigation platform for wearable computing environments whose construction is based on an event-driven system called A-WEAR that presents various information autonomously. Using our system, we can construct flexible navigation services according to the device configuration of a user's computer. Moreover, we constructed a scene editor with which we can create contents by exploiting the data from sensors. The editor enables the contents creator to easily create flexible navigation contents.

We used the prototype system for several navigation contents including experimental evaluation at Expo' 70 Commemoration Park and clarified the effectiveness of our system. In the future, we will propose a new language and a new editor that can create more complicated navigation scenarios.

Acknowledgment

This research was supported in part by "The 21st Century Center of Excellence Program", Grant-in-Aids for Scientific Research number 13780331 from the Japan Society for the Promotion of Science and Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R. and Pinkerton, M.: Cyberguide: A mobile context-aware tour guide, *Wireless Networks*, Vol. 3, No. 5, pp. 421–433 (1996).

[2] Anegg, H., Kunczier, H., Michlmayr, E., Pospischil, G. and Umlauft, M.: LoL@: Designing a Location Based UMTS Application, *OVE-Verbandszeitschrift e&i, Springer, Heidelberg, Germany*, pp. 48–51 (Feb. 2002).

[3] *Cyber Communication 2004 Web Page (in Japanese)*, http://www.teamtsukamoto.com/cyber2004/.

[4] Hollerer, T., Feiner, S., Terauchi, T., Rashid, G. and Hallaway, D.: Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System, *Computers and Graphics*, Vol. 23, No. 6, pp. 779–785 (1999).

[5] Kurata, T., Okuma, T., Kourogi, M., Kato, T. and Sakaue, K.: VizWear: Toward Human-Centered Interaction through Wearable Vision and Visualization, *PCM2001 in Beijing, China*, pp. 40–47 (2001).

[6] *Minpaku Digital Guide Web Page*, http://www.minpaku.ac.jp/english/exhibitions/permanent/digiguide/.

[7] Miyamae, M., Terada, T., Tsukamoto, M. and Nishio, S.: Design and Implementation of an Extensible Rule Processing System for Wearable Computing, *The First Annual Int'l Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004)*, pp. 392–400 (Aug. 2004).

[8] Satoh, K., Hara, K., Anabuki, M., H.Yamamoto and H.Tamura: TOWNWEAR: An Outdoor Wearable MR System with High-precision Registration, *2nd Int. Symp. on Mixed Reality*, pp. 210–211 (2001).

[9] Tenmoku, R., Kanbara, M., Yokoya, N. and Takemura, H.: Annotation overlay with a wearable computer using augmented reality, *1st CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pp. 27–32 (Mar. 2002).