30th Annual International IEEE EMBS Conference
Vancouver, British Columbia, Canada, August 20-24, 2008

Personalized Healthcare through Intelligent Gadgets

Hyeju Jang, Sanghyun Kim, Changseok Bae, *IEEE*

Abstract—**An intelligent gadget is a wearable platform which is reconfigurable, scalable, and component-based and which can be equipped, carried as a personal accessory, or in a certain case, implanted internally into a body. Various kinds of personal information can be gathered with intelligent gadgets, and that information is used to provide specially personalized services to people in the ubiquitous computing environment. In this paper, we show a personalized healthcare service through intelligent gadgets. A service based on intelligent gadgets can be built intuitively and easily with a context representation language, called the intelligent gadget markup language (IGML) based on the event-condition-action (ECA) rule. The inherent nature of extensibility, not only environmental information but also physiological information can be specified as a context in IGML and can be dealt with an intelligent gadget with ease. It enables intelligent gadgets to be adopted to many different kinds of personalized healthcare services.**

I. Introduction

Services using everyday items have a great number of demands in the ubiquitous era when computers are more hidden and smarter. An intelligent gadget is a smart object which has the capabilities to process information, cooperate with other similarly equipped gadgets. They can be carried as a personal accessory in most cases and can be used in various context-aware use scenarios including that of healthcare services based on automatically captured physiological data [1].

The services through intelligent gadgets are mostly event-driven because they should be carried into effect when something, that is an event, happens. Moreover, what kinds of sensors an intelligent gadget has can determine what kinds of services can be run on the intelligent gadget. Hence, if physiological sensors are attached to an intelligent gadget with others, event-based healthcare services are possible with that intelligent gadget.

In this paper, we show a personalized healthcare service built with an intelligent gadget markup language (IGML). IGML is a language that describes event-condition-action (ECA) rules in XML for intelligent gadgets to do event-based services. Service developers are supposed to write rules in IGML about what the intelligent gadget to do. It helps service developers define events they want to deal with, conditions based on the events, and actions to perform after events satisfying the conditions. It makes service components built readily and conveniently. It is also extensible because a service component can call another service component as its action.

Intelligent gadgets allow services to monitor the status of a user continuously because they are daily objects people usually keep. Also, it is differentiated from others in its features that it regards all the data as events and it has a user-defined action which service developers can define using middleware-level APIs. Since every data is dealt with as an event, it can be very useful especially in emergency such as a sudden change of a patient's status.

In the following sections, we present what is an intelligent gadget and the structure of intelligent gadget. Then, we describe how to make a service component. Lastly, a prototype service is shown as what kind of personalized healthcare services are possible with intelligent gadgets.

II. References

As the personal service areas, recently there has been much research related especially to the "Life Log" to record everyday experience information. They mainly focus on how to sense personal experience information, how to log such sensed information, and how to query or retrieve the necessary information from the logged data.

The life log video system led by Prof. Aizazwa from Tokyo university used brain wave, movement, recognizing face, position, time, internet, log of using application programs as the factors managing memories [2]. As memory information of a person, not only video and audio but also daily life information recorded from various wearable sensors is used.

The representative example of research using AN INTELLIGENT GADGETs is TTT(Things That Think) project in MIT [3]. In this project, many researches are being done for the development of thinking objects. For instance, attaching information processing device to objects, attracting people's eyes, recognizing people's voice, and providing information users want is possible in "Invisible Media". MIT also developed bYOB(Build Your Own Bag) which can grab outside environment and service intellectually, a smart bag [4].

Nokia developed and provided Lifeblog connecting to their cellular phones, which is a PC based software [5]. Users save contents made with Nokia cellular phones, or video they take a

Manuscript received April 16, 2008. This work was supported by the IT R&D program of MIC/IITA. [2006-S032-02, Development of an Intelligent Service technology based on the Personal Life Log]

Hyeju Jang is with Electronics and Telecommunications Research Institute, 161 Gajeong Dong, Yuseong Gu, Daejeon, Korea. (corresponding author to provide phone: 82-42-860-1664; fax: 82-42-860-5545; e-mail: hyejujang@gmail.com).

Sanghyun Kim is with University of Science and Technology, Daejeon, Korea. (e-mail: ksh41@etri.re.kr).

Changseok Bae is with Electronics and Telecommunications Research Institute, 161 Gajeong Dong, Yuseong Gu, Daejeon, Korea. (e-mail: csbae@etri.re.kr).

photograph with their interests using Lifeblog through wireless network. Saved information can be shared with other users if they want. The target of Lifeblog is mainly video and text data, it's still difficult to support other information like environment, the reaction of a living body, and movement.

Microsoft is developing a lifelog service using SenceCam and MyLifeBits Viewer. Users bring SenseCam and collect experience information. Collected information is saved and managed in MyLifeBits which is a PC based application [6].

Canada Queens University's eyeBlog is the system which saves and shows user's video information automatically [7]. EyeBlog measures people's gaze information using a wearable wireless gaze sensor, ECSGlasses(Eye-Contect Sensing Glasses). Using this, it records 1:1 conversation and recorded information can be shown through web. Video is automatically recorded based on user's interests. For example, if it catches somebody looking at the user with ECSGlasses, it thinks that's 1:1 conversation and records the conversation with video. Moreover, it's also possible for users to record the conversation manually with buttons. Newly recorded videos are saved in a certain directory of a web server, web contents are automatically manipulated and uploaded to the blog. Then, users can watch video contents built using preview function. eyeBlog is also an example of using Web 2.0.

In this paper, we sense and log physiological data as life log data with intelligent gadgets, and apply this to the personalized healthcare service. It opens out the possibilities of personalized healthcare services with life log applications.

III. INTELLIGENT GADGET

An intelligent gadget is a wearable platform which is reconfigurable, scalable, and component-based. It can be equipped, carried as a personal accessory, or in a certain case, implanted internally into a body. It is used to gather various kinds of personal information, and that can process those gathered information to provide specially personalized services for the ubiquitous computing environment. It also can be installed on diverse ubiquitous environments to gather various environmental information to provide more adequate personal services by analyzing and combining with those information.

For the hardware point of view, an intelligent gadget consists of a base block and device blocks. The base block is equipped with a processor, a memory, a network, a power and a stack-able interface. The device block consists of a sensor and a stack-able interface. The function of an intelligent gadget depends on the type of device blocks, and the function of a device block depends on the sensor equipped. an intelligent gadget can be only a base block itself, or one base block stacked with multiple device blocks. Fig. 1and Fig. 2 shows a base block and a base block with a device block, respectively.

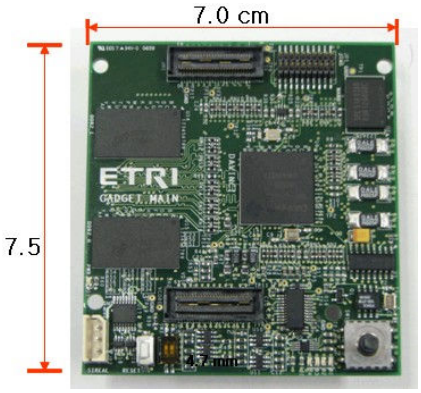

Fig. 1. The base block of an intelligent gadget

Fig. 2. The base block with a device block of an intelligent gadget

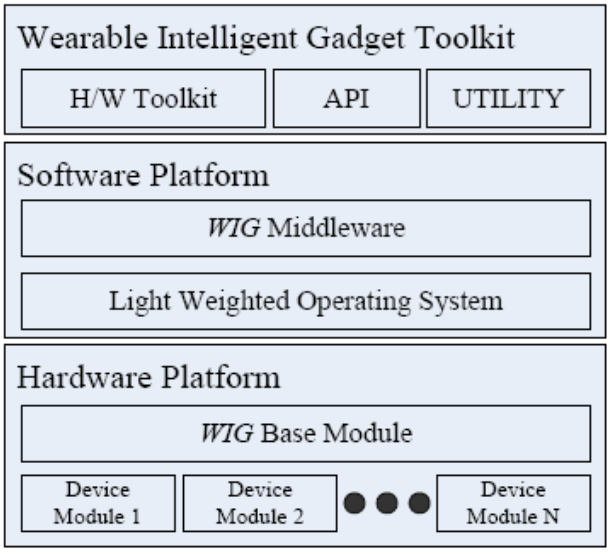

Fig. 3. The architecture of an intelligent gadget

For the software point of view, an intelligent gadget consists of a light-weighted operating system and an intelligent gadget middleware. The intelligent gadget middleware has a component-based modular architecture for the reconfigurability and the scalability. It also allows the logical and physical grouping. The physical grouping is a set

of an intelligent gadgets that compose the physical PAN or BAN, and the logical grouping is a set of intelligent gadgets that are grouped by a common attribute.

As shown in Fig. 3, an intelligent gadget consists with the hardware platform, the software platform, as well as the intelligent gadget toolkit.

The intelligent gadget toolkit is a set of tools consisted with the hardware toolkit, API, and various utilities that support the intelligent gadget hardware platform manufacturers, service developers, and service providers, respectively.

IV. How to Build a Service Component

A service component of an intelligent gadget can be built with an easy and simple way, IGML. IGML is a markup language for representing events, conditions and linking with actions [7]. It consists of four parts: Rule, Event, Condition, and Action.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="IGML">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="rule" type="rule"
maxOccurs="unbounded"/>
            <xs:element name="event" type="event"
maxOccurs="unbounded"/>
            <xs:element name="condition" type="condition"
minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="action" type="action"
maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
      <xs:complexType name="rule">
         <xs:sequence>
            <xs:element name="event-ref" type="nameType"
maxOccurs="unbounded"/>
            <xs:element name="condition-ref" type="nameType"
minOccurs="0"/>
            <xs:element name="action-ref" type="nameType"/>
         </xs:sequence>
         <xs:attribute name="name" type="nameType"
use="required"/>
         <xs:attributename="limit" use="optional"
type="nonNegativeInteger" default="unbounded"/>
      </xs:complexType>
                  •
                  •
                  •
```

Fig. 4. The rule schema of IGML

A. Rule

The rule is the actual part which is performed in an intelligent gadgett. An intelligent gadget does a service according to the rule part of an IGML document. The rule part consists of references to the event part, the condition part, and the action part. The XML schema for the rule is shown in Fig. 4. Several rules for a service can be specified inside the <IGML> element and each ECA rule is described with a <rule> element. This formation is similar to the one of [8].

B. Event

All the data coming from outside and inside are defined as events in IGML from the view point of an intelligent gadget. Table 1 shows the primitive event type supported in the intelligent gadget. Events are classified into 3 categories: raw data from sensors, manipulated data in the intelligent gadget such as activity, and appearance/disappearance of an intelligent gadget. Sensor data events are from various kinds of sensors attached to intelligent gadgets. Service component developers can also make new events modifying sensor data like activity events. Appearance events mean appearance and disappearance of intelligent gadgets in the intelligent gadget network. These events are described in the XML format according to the characteristics of each type [9].

C. Condition

The condition part of IGML is stated in terms of expressions. IGML supports some of built-in operators such as Boolean operators "and" and "or", and comparison operators "=", "!=", "<", ">", "<=", ">=" . Because of the resource restriction of intelligent gadgets, IGML limits the recursiveness of expression. Detailed description of conditions in IGML can be found on [9].

TABLE I
PRIMITIVE EVENT TYPE

Event type	Event sub-type
Raw Data	Bogy heat
	Heart rate
	Blood pressure
	ECG
	EEG
	EMG
	PPG
	GPS
	Temperature
	Acceleration
	Illumination
Activity	N/A

D. Action

When the specified events occur and satisfy the condition can be specified by IGML, the action part is triggered. IGML supports the representation of two kinds of actions; the invocation of other service components, and the call of the user-defined APIs.

The both kinds of action part show a extensible feature in IGML. Service developers can define their own functions using APIs provided by Intelligent Gadget Middleware and specify a function name, parameters and a return type in the IGML document. Then, it is performed as an action in the document. a service component with invocation of other service components makes a service component reusable. The XML schema of the action part is also in [9].

V. PROTOTYPE SERVICE

Here, we will show a prototype service using intelligent gadgets. It is a simple and tiny service which can be implemented on intelligent gadgets focusing on what kind of personalized healthcare services can be possible with this research and how it is easy to develop a service with intelligent gadgets.

The implemented service model is an emergency beep service, which sends a SMS message to a certain phone number when the heart-rate goes abnormal. It is a real-time application that exploits intelligent gadgets, which collects photo-plethysmography(PPG) signal and judge if the heart-rate goes normal or not.

In the process, intelligent gadgets collect PPG signal and yield heart-rate data from PPG signal on the service component extracting heart-rate data. However, because of the current limitation of the device block hardware, we simulated the heart-rate data in the intelligent gadget and used the simulated data in the experiment [10]. Hence, the emergency beep service-component gets the heart-rate data as an event and compares it with the condition. The condition can be specified as a case by the service component developer. In this prototype service, the heart-rate which is greater than 200 and less than 40 means the abnormal case.

Fig. 5 shows the service environment of the service. Intelligent gadgets with service components can compose gadget network as a PAN (Personal Area Network) or a BAN (Body Area Network) with ZigBee. This PAN or BAN is connected with WLAN network to communicate with the viewer and the life log server. The viewer has the CDMA module.

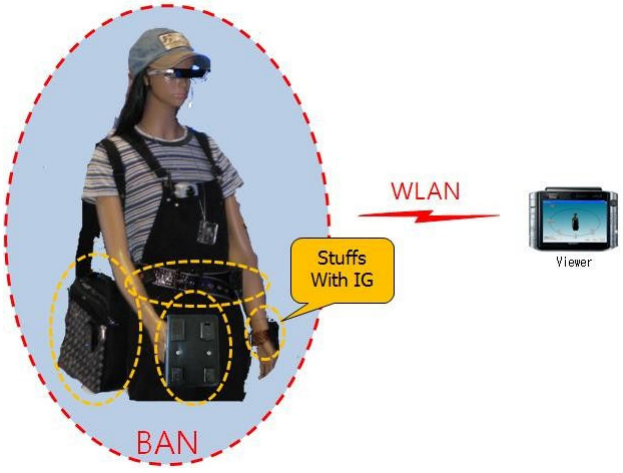

Fig. 5. Service environment

The service component runs on the intelligent gadget. They instruct gadgets to gather data from sensors attached to the gadget and to call the API which sends SMS message to the specified phone number when the data satisfy the condition. This instruction is described with IGML.

Fig. 6 shows the IGML description for an emergency beep service. This description says that "beep" api call is executed as an action when the heart-rate data appear in the gadget network and it is below 40 or above 200. As a result, "beep" api call executes the CDMA module on the viewer and the specified phone number will get the SMS message for notifying emergency. In a similar way, other kinds of service components can be described in the form of IGML. Also, more complicated service components are enabled to build with the help of the function calling another service component.

```
<IGML>
  <rule name="R1">
    <event-ref>E1</event-ref>
    <condition-ref>C1</condition-ref>
    <action-ref>A1</action-ref>
  </rule>
  <event name="E1">
    <type>Heart rate</type>
  </event>
  <condition name="C1">
    <expression>
      <![CDATA["(E1/data<40)||(E1/data>200)"]]>
    </expression>
  </condition>
  <action name="A1">
    <api-call>
      <service-component>
        EmergencyBeepService
      </service-component>
      <api-name>Beep</api-name>
      <return-type>void</return-type>
        <param-num>1</param-num>
        <param>
        <event-ref>E1</event-ref>
        <element>data</element>
        <data-type>int</data-type>
      </param>
    </api-call>
  </action>
</IGML>
```

Fig. 6. IGML description for an emergency beep service component

REFERENCES

[1] J. H. Won, K. H. Lee, and C. S. Bae, "Wearable Intelligent Gadgets for Personalized Services," Dubai, ICUT 2007, pp.1201-1210, Feb. 2007.

[2] D. Tancharoen, T. Yamasaki, and K. Aizawa, Practical Experience Recording and Indexing of Lfe Log Video, In Proc. of CARPE 2005, Singapore, Nov. 2006, pp.61-66.

[3] MIT's Things That Think Home Page, http://ttt.media.mit.edu/

[4] G. Nanda, Accessorizing with Networks: The Possibilities of Building with Computational Textilies, Master Thesis, MIT, 2005

[5] S.Cherry, Total Recall: A Microsoft Researcher is Determined to Record Everything About His Lfe, IEEE Spectrum, Nov. 2005, pp.24-30.

[6] Jim Gemmel, Gordon Bell, and Roger Lueder, My Life Bits: A Personal Database for Everything., Communications of ACM 2006, Vol.49, No. 1.

[7] H. J. Jang, et all., "Context Representation of Intelligent Gadgets for Event-driven Services", In Proc. of ICCE 2008, Las Vegas, USA.

[8] C. Dickie, R. Vertegaal, et al., Augmenting and Sharing Memory with eyeBlog, In Proc. of Carpe 2004, New York, USA, Oct. 2004, pp.105-109.

[9] K. H. Lee, "Intelligent gadget functional specification version 1.1," ETRI technical document [07MH2110-DE-FN-000-v1.1], 2007.

[10] S. H. Kim, "Implementation of Smart Headband for the Wearable Healthcare", In Proc. of ICCE 2008, Las Vegas, USA.