

Wearable Multi-modal Remote Control

Dongwoo Lee, John Sunwoo and Ilyeon Cho
Next Generation PC Group, ETRI, Daejeon, Korea

ABSTRACT

Remote controls are often used to control many electronic appliances, however, stacks of remote controls result in an inconvenience. We introduce a wearable system that can be used as Universal Remote Control (URC) and it operates through various modalities such as voice and gesture commands. Then we suggest how the wearable user can control diverse electronic devices in the same way.

I. INTRODUCTION

As the number of electronic appliances (including personal computers) increases, many companies or consortiums such as Universal Remote Console (URC) are developing solutions to control various home appliances using one remote control [1]. In the future ubiquitous environment, the demands for controlling such devices using user friendly interfaces will grow as there will be more electronic appliances.

This paper introduces a wearable system that can be used in controlling various types of devices within proximity of the user. This wearable system utilizes a user-friendly interface such as voice and gesture.

II. DESIGN OF MULTI-MODAL INTERFACE

Conventional remote control recognizes the button pushes and uses IrDA to control electronic appliances. However, if we can use more user-friendly and natural modalities such as speech and gesture commands, it will be more convenient[2]. This paper defines *ActionXML* and it contains formatted rules to arrange and combine various modalities for controlling appliances.

TABLE 1 *ACTIONXML* DTD

```
<?xml version="1.0" ?>
<!DOCTYPE device [
<!ELEMENT device (action)*>
<!ELEMENT action (input?, integration)>
<!ELEMENT input (modality)+ >
<!ELEMENT modality (command)+ >
<!ELEMENT command (#PCDATA) >
<!ELEMENT integration ((or*|and*)set*+)>
<!ELEMENT or (modname*|or*|and*+)>
<!ELEMENT and (modname*|or*|and*+)>
<!ELEMENT modname EMPTY>
<!ELEMENT set sequence+>
<!ELEMENT sequence actionname*>
<!ATTLIST device id ID #REQUIRED>
<!ATTLIST device model CDATA #IMPLIED>
```

```
<!ATTLIST action name ID #REQUIRED>
<!ATTLIST action type (multi|single) "single" >
<!ATTLIST modality mode CDATA #REQUIRED>
<!ATTLIST modality name CDATA #REQUIRED>
<!ATTLIST or weight CDATA #IMPLIED >
<!ATTLIST and weight CDATA #IMPLIED >
<!ATTLIST modname weight CDATA #IMPLIED >
<!ATTLIST modname value CDATA #REQUIRED>
<!ATTLIST sequence value CDATA #REQUIRED>
]>
```

Table 1 shows the Document Type Definition (DTD) of *ActionXML*. Our meaning of *action* is a final command that is sent to the target appliances as defined in Table 1. *Action* can be classified as a single action or as multiple actions. Multiple actions consist of one or more single action or multi-modal device interaction. One action is defined using ‘input’ element and ‘integration’ element- The ‘input’ element defines user input from various modalities and these input modalities are combined according to the rule defined in the ‘integration’ element. Rule of single action can be defined using ‘and’ and ‘or’ element, whereas multiple actions can be defined using ‘sequence’ element so that it can guess new action by using 2~3 consequent actions.

For instance, using *ActionXML*, action of changing TV channel can be defined as Table 2.

TABLE 2 *ACTIONXML* EXAMPLE OF CHANNEL UP

```
<?xml version="1.0" encoding="ksc5601" ?>
<adxml version="1.0">
<device id="0xFFFFFFFF" model="SS501TV" >
<action name="CHANNELUP" type="single">
<input>
<modality mode="voice" name="voice1">
<command>channel up</command>
</modality>
<modality mode="voice" name="voice2">
<command>channel</command>
</modality>
<modality mode="gesture" name="gesture1">
<command>Up</command>
<command>Right</command>
</modality>
</input>
<integration>
<or>
<modname weight="1.0" value="voice1"/>
<and weight="1.0">
<modname value="voice1"/>
<modname value="gesture1"/>
</and>
```

```

<and weight="0.8">
  <modname value="voice2"/>
  <modname value="gesture1"/>
</and>
</or>
</integration>
</action>
</device>
</adxml>

```

III. SYSTEM CONFIGURATION AND OPERATION

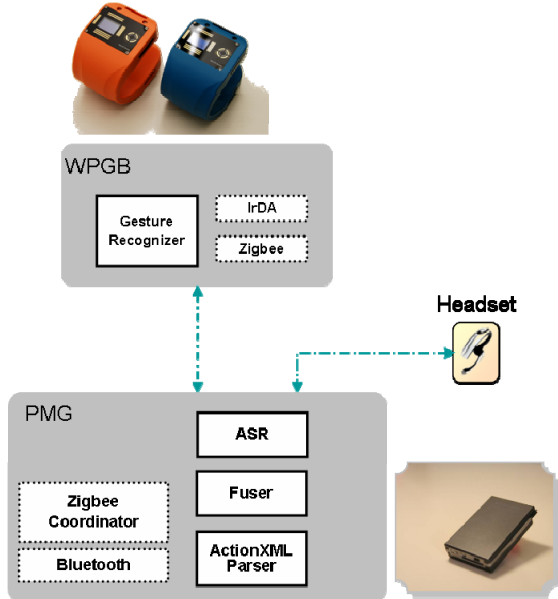


Fig. 1. Wearable Remote System Configuration

For the wearable system, we use Wearable Pointing and Gesture Band (WPGB) and Personal Mobile Gateway (PMG) as shown in Fig. 1 [3]. WPGB runs on a small-sized low-power Real-Time OS [4]. It contains forearm gesture recognizer, IrDA transceiver, Zigbee module. PMG consists of Bluetooth, Zigbee module, *ActionXML* parser, and multi-modal fuser[5] which guesses appropriate actions. Also, Text-To-Speech (TTS) engine is on PMG to support user feedback [3].

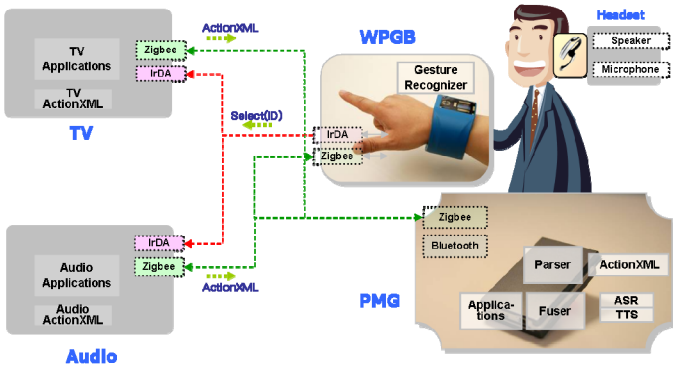


Fig. 2. System Operation Diagram

User can select a target appliance by pointing the target with a natural gesture, and the IrDA signal will be generated from the user's wristband-type remote control. The targeted appliance detects the incoming IrDA signal and extracts wearable system ID to find out who is requesting the connection. When the target appliance tries to establish communication channel with the received wearable ID, the wearable system can get the ID of target appliance. If the wearable system does not have the appropriate multi-modal commands (described in *ActionXML*) that are used to control the target appliance, it can request and receive the *ActionXML* through Zigbee network.

User's wearable system parses *ActionXML* to extract multimodal commands and fusion rules. As fuser gets user's speech and gesture command, the fuser guesses the final action based on the extracted rule and transmits the guessed action to the selected appliance. When the target appliance receives action, it executes the action and generates relevant user feedback.

IV. CONCLUSION

In this paper, we introduce *ActionXML* that describes multi-modal control commands to control electronic appliances. We also suggest that each appliance implements an appropriate *ActionXML*. When user selects an appliance, the targeted appliance sends its control commands to the user's wearable remote control. Thus, by wearing a system, the user can control any appliance through one preferred modality, or through multiple modalities as defined in the *ActionXML*. Also unlike the conventional remote control, this system lets user to move services or content from one device to another.

This system may have limitations when there are too many different appliances to control which results in many different functions. In this case, the user may have to remember a lot of commands. However, many home appliances are frequently used with only a few frequently used functions; which may solve the problem.

REFERENCES

- [1] Universal Remote Console Consortium, <http://myurc.org>
- [2] R. A. Bolt, "Put-That-There":Voice and gesture at the graphics interface, *Computer Graphics* 14.3, pp. 262-270, 1980.
- [3] I-Y. Cho, J. Sunwoo, et al., "A Distributed Wearable System Based on Multimodal fusion," *ICESS2007, LNCS 4523*, pp. 717-726, 2007
- [4] H-J. Ahn, et al., "UbiFOS: A Small Real-Time Operating System for Embedded Systems," *ETRI Journal*, 29(3), 2007
- [5] D-W. Lee, et al., "Multimodal User Interface for Wearable Computer Based On Forearm Gesture and Speech," *proc. Conference on Next Generation Computing*, pp.45-50, 2007
- [6] Johnston M., P.R. Cohen, D. McGee, S.L. Oviatt, J.A. Pittman, and I. Smith, "Unification-based Multimodal Integration" in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.