IEEE TRANSACTIONS ON CONSUMER ELECTRONICS

NOVEMBER 2008 VOLUME 54 NUMBER 4 ITCEDA (ISSN 0098-3063)

A Publication by the IEEE Consumer Electronics Society

(Continued on back cover)

Design and Implementation of a Universal Appliance Controller Based on Selective Interaction Modes

Hyoseok Yoon and Woontack Woo, *Member*, IEEE

Abstract — *In next generation of smart computing environment, myriads of networked resources are pervasive and embedded into everyday consumer products and appliances. Still, most controller systems provide separate control user interfaces for different uses when users inevitably face an increased number of interfaces for control and interaction tasks in such environment. To simplify burdens of user and reduce complexities in control tasks, we propose a universal appliance controller based on mobile device with two selective interaction modes. First mode is a network scanning-based indirect interaction mode which provides fast and reliable connection to the appliances with specific filters in the networked environment. Second mode is a camera-based direct interaction mode which provides an intuitive interface as an input device. These two modes are interchangeable for different tasks and according to user's preference. Moreover, control user interface is personalized with preference described in user profile to reflect the user of the system better. To verify usefulness of our approach, we present a prototype in a smart home test bed. Through an initial user study, we demonstrate usability of our prototype through a set of mobile interaction tasks. Moreover, we note usability issues, limitations of our work and analyze the participants' feedback*[1].

Index Terms — universal controller, service discovery, mobile interaction, camera-based interaction.

I. INTRODUCTION

In the future computing environment, an increased number of electronic appliances and devices provide various useful services to users as in cases of ubiquitous computing and context-aware computing. However the increasing number of interaction frequency in these environment burdens end-users who need to learn and get familiarized with different interfaces to use them. The learning burdens get heavier as we advance toward an environment like smart home with a large number of next generation consumer products and appliances. In this setting, end-users need a better and novel way to survive in the information and device-flooding era.

Recently, use of mobile device is encouraged as an interaction device [1]. Many users carry and access an increased number of mobile electronic devices in daily basis. Especially mobile phones have gained so much popularity to be a major communication device for many people around the globe. Mobile phone sales already topped 800 million in 2005 [2], and specifically camera phone sales is expected to reach 900 million by 2009 [3]. Moreover, today's phone and smart phones boast increased computational power, large touch screen and wireless connectivity as well as high resolution digital camera. As Mark Weiser envisioned, "the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it [4]." Such technology development can be found in mobile devices which have become a widely deployed and prevalent interaction device.

In this paper we present a prototype for a universal appliance controller based on mobile device. The prototype provides two interaction modes which can be operated separately or interchangeably to increase the use of the system along different domains and user preferences. Both interaction modes are implemented in mobile device platform to better use today's technology and take advantage of user's knowledge in familiarity of the deployed technology. The contributions of this paper are:

- Network scanning-based service discovery and list generation for indirect interaction mode are provided as a contextual background filtering process without taking user's attention
- Camera-based "take-a-picture" metaphor for direct interaction mode exploits intuitive and familiar technology to serve as an input method
- Personalization based on user preference reflects user-specific information onto control user interface generation and presentation

The rest of the paper is structured as follows. Section 2 describes related works in controllers for consumer electronics and appliances. Section 3 explains our proposed design principles for building an appliance controller. Section 4 presents a prototype of the proposed controller. In section 5, we evaluate the performance of the prototype and analyze the findings. Section 6 concludes the paper by discussing the future works.

II. RELATED WORK

There have been several different approaches in controlling consumer products and appliances in many literatures.

[1] This research is supported by the Foundation of UCN Projects, the MKE, and the 21C Frontier R&D Program in Korea as a result of subproject UCN 08B3-O1-20S.

H. Yoon is with the Department of Information and Communications, School of Information and Mechatronics, Gwangju Institute of Science and Technology (e-mail:hyoon@gist.ac.kr).

W. Woo is with the Department of Information and Communications, School of Information and Mechatronics, Gwangju Institute of Science and Technology (e-mail:wwoo@gist.ac.kr).

Contributed Paper
Manuscript received October 15, 2008

Most works have focused on home networking infrastructure and interoperable platforms for establishing connection between user and the environment. SOAP-based smart home system is presented in [5] where Web Service is used to control each appliance. In [6], they investigated into extending UPnP-enabled home networks to support broader ranges for media streaming. Lo and his colleagues presented a mobile device based system to control digital TV by sending commands over IP network in [7]. Cabrer et al. described a smart home controller system that bridges OSGi (Open Service Gateway Initiative) and MHP (Multimedia Home Platform) [8]. Nikolaidis et al. presented a system that automatically configures consumer electronic devices with existing Internet standards for controlling and interacting with devices [9].

The advantage of using IP networks in the system is that the complex process of the system can be hidden from users. But on the other hand, when these systems are applied to controlling tasks and controller system, users consider IP-based scanning method more technically complex due to its indirectness [10].

Other works focus on configuring user interface for controlling and interacting with devices. Komine et al. designed a user interface and evaluated with two major components, functionality and appearance. In their work, they found different user impressions in these components between prototype user interface and commercial one [11]. Verhoeven and Dees defined services with an abstract user interface description written in XML [12]. Sathyan and Ramakrishnan presented self-contained and brand-independent universal remote control based on IR code generator [13]. Lee and Wang's work personalized and automated channel selection for multimedia services through user preferences and feedback [14]. Chong et al. discussed user interface design requirements for mobile devices and designed a simplified single-layered touch screen based user interface [15].

From the related work on user interface, we observe that system features such as functions and presentation of the system are configured through a written description while maintaining simple and intuitive form.

The last set of work focus on alternative control modality. Several works explore speech modality in controller system, such as Yuksekkaya's work [16] for controlling home automation system and Zeng's work [17] to address speech recognizer for activating wireless home devices. However the accuracy dependency of the verbal data set and different users make it unfavorable modality for controller system which solely depends on user commands as the only input.

III. Design of Appliance Controller

To design and implement an effective and useful controller system for the future computing environment, we surveyed on typical mobile systems that interact with services and devices in the ubiquitous computing environment in our earlier work [18]. From this work and observations based on recent related works, we propose three common functional requirements (service discovery, service selection and personalization) which are essential in mobile systems. We adopt these three requirements to shape system architecture for our appliance controller.

A. Service discovery

Service discovery is a mechanism to notify users what kind of services is available and what kind of functions they can offer. Also it is a process of making invisible and logical services into visible and tangible services. Even though people have good ideas on what a consumer product is capable of, the same process of mapping for computer systems is quite complicated. There are several well-known approaches to discover services in the environment. Notable ones include methods that use service discovery protocols such as UPnP [19], Jini [20] and Bluetooth to search and scan the network in the vicinity of user. To use service discovery protocols, wireless network connection is needed for discovery service in the mobile environment.

Fig. 1 shows the contextual network service discovery process. First, the system captures network messages or service announcements in the network. Then these messages are internally processed and parsed to produce a service ID or URL where the information about the discovered service is accessed. In this step, a filtering mechanism is used to contextually filter out unnecessary or unwanted services.

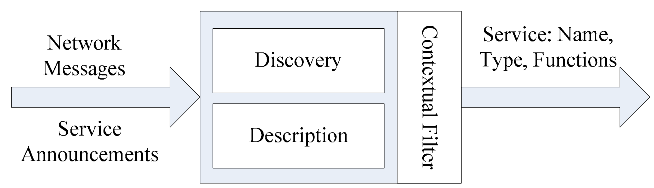


Fig. 1. Contextual service discovery process.

Another popular method is pointing or tag-based methods, which is prevalent in traditional remote controller and recent tags, barcode, visual tags, and NFC (Near-Field Communication) [21] systems respectively. This method requires readers on user's mobile device which reads data from the tagged objects. Table I summarizes three common service discovery methods. For our system prototype, we adopt service discovery methods known as network scanning and tag-based method.

TABLE I
Service Discovery Methods

Methods	Examples
Network Scanning	UPnP, Jini, Bluetooth
Pointing	Infrared Remote Controllers
Tag-based	Barcodes, Visual Tags, RFID, NFC

B. Service selection

Service selection is a process of explicitly or implicitly focusing into a target service. Service selection is a natural next step from service discovery and has a little overlapping area with service discovery. There are 3 categories of selection type by where the selection is made and by different levels of user intervention as shown in Fig. 2.

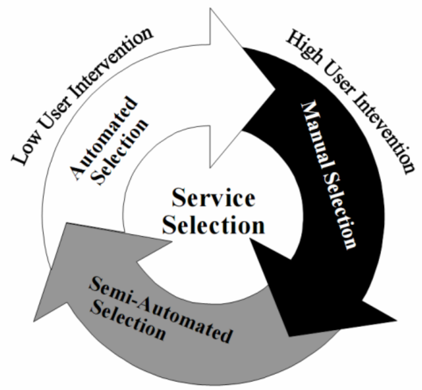


Fig. 2. Service selection methods based on user intervention levels.

First level is automated selection level where the selection is made by the system and little or no user intervention is allowed. This level is suitable for system that does not require high accuracy, since the decision by the system can be ambiguous and incorrect. Second level is semi-automated selection where the selection is initiated by the system and confirmed by the user. This level is useful since the initial decision can narrow down the possible selections for users, then the user can proof check for final decision. Fig. 3 shows semi-automated service selection where contextual information is used to make a recommendation which still needs a confirmation from users.

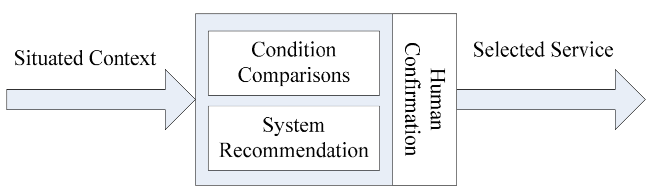


Fig. 3. Semi-automated selection process.

Third level is manual selection where all selection is made by users explicitly. Second and third levels of selection are useful for critical systems that produce irreversible or high cost results. Nonetheless, more user intervention is required for these levels which can be burden and cumbersome for end-users. In our controller system prototype, we make it highly user-interactive by providing manual selection as well as situated semi-automated selection.

C. Personalization

Personalization is a specialized subset of user to service interaction. User to service interaction is characterized by a typical scenario of user controlling or accessing a service. Personalization for user to service interaction plays a role in making the target service personal to each user by gathering user-information during interaction phase [22]. There are two facets in personalization, user interface level and functional level. In user interface level, the appearance and structure of user interface are changed according to user preference described in each user profile. On one hand, user interface personalization personalizes the presentation displayed to user. On the other hand, functional personalization selects highly relevant functions to user in given situation. In our prototype, we provide both levels of personalization by using user profile information.

IV. Implementation

In this section, we describe implementation of our appliance controller prototype and our approach toward three aforementioned design requirements.

A. Platform and Basic Framework

We use a mobile device as a basic platform and computational device. Recent mobile devices such as mobile computers come with wireless network connectivity which makes it a good networking application platform for mobile applications. A modest computational power is also required for the mobile device to support user interface generation, image processing and networking. We especially use a built-in camera of mobile device as an additional sensor to capture visual information from the environment. Even though wireless network provides general connection to appliances, we combine camera into our system to support various scenarios and patterns for extended uses. A small mobile device can be hand-held and light enough so that users can hold the device with one hand. The screen of many mobile devices is a touch screen which is an intuitive and easily accessible input device for general users.

Our approach builds upon a mobile device that satisfies computational power, network connectivity, weight, and input mode requirements. On top of this device, Unified Context-aware Application Model for ubiquitous computing (ubi-UCAM) [23] and Universal Plug and Play (UPnP) are used as middleware to define smart objects and appliances. ubi-UCAM supports communication with context in each application such as smart TV, smart light and smart table where UPnP protocols are used for service discovery.

For appliances in the environment, these appliances are implemented with ubi-UCAM and have its own conditions and actions called service conditional context (SCC). Manual control of these appliances overrides its predefined conditions and reacts upon user's explicit commands. Service discovery block contains modules for discovering networked services and interpreting its data description.

Service selection block contains two separate sub blocks for two interaction modes, network scanning mode and camera-based mode. Network scanning mode performs its operation in the background and filters out unnecessary services. Camera-based mode is executed with a user's explicit intention for selection and input operation.

Personalization block contains two sub modules for user interface presentation and functional presentation. Also the framework holds three dedicated databases. User profile is stored and used in personalization. Service and marker databases are used for network scanning based mode and camera-based mode for recognition respectively. Fig. 4 shows the basic framework of our prototype.

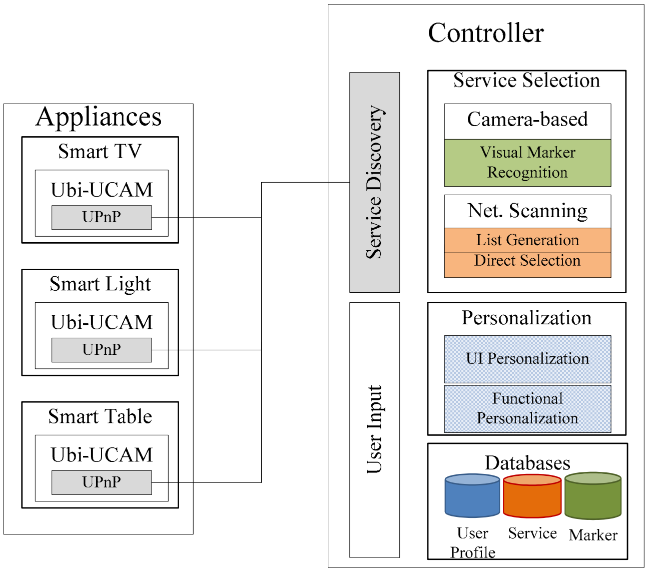


Fig. 4. Basic framework for controller prototype.

Algorithm 1. Categorical Service Discovery

Input:
A discovered UPnPDevice D;

Outcome:
If the name of D has prefix "User", User_List is searched for filtering otherwise Service_List filtering is applied;

```
1:  if ( NameHasPrefix ( D ) == TRUE ) then
2:      if ( CheckUserList ( D, User_List ) == TRUE ) then
3:          ListAddUser ( D ); // user is added
4:          Num_of_User++; // increase # of discovered users
5:      else
6:          Ignore D; // discarded
7:      end if
8:  else
9:      if ( NameHasPrefix ( D ) == FALSE ) then
10:         if ( CheckServiceList ( D, Service_List ) == TRUE ) then
11:             ListAddService( D ); // service is added
12:             Num_of_Service++; // increase # of services
13:             GenerateTabMenu( D ); // tab menu is generated
14:         else
15:             Ignore D; // discarded
14:         end if
15:     end if
16: end if
```

B. Service Discovery

In our prototype, we use service discovery protocol UPnP where user's mobile device acts as an UPnP control point and appliances in the environment are implemented as UPnP services. However, UPnP itself does not support the use of context; therefore we concurrently use it with our context-aware application framework, ubi-UCAM to utilize contextual information in both UPnP service and UPnP control point.

Once the controller system is executed, it searches for new devices and services in the environment. When a new device or service is discovered, its information is saved onto the service database which is accessed for service control. Depending on the prefix of the discovered device name, the device either represents user's mobile device or appliances in the environment. This information is later used to visualize discovered services as well as users in the vicinity. First conditional check in Algorithm 1 manages and calls appropriate function for user and service categories.

As inputs for network service discovery, network messages and service announcements are received by the mobile device. These messages are essentially description of device and service which are interpreted to identify a service in the message. To reduce a number of unwanted discovered services, a service filter is used to check against the discovered services. A service filter is a set of allowed UPnP device and service names which is compared in real time when a new service is detected. Starting from line 10 in Algorithm 1 describes filtering mechanism in the prototype. Then as a filtered output, an identifier or service data structure is generated. The service discovery process is executed in the background and does not require user's participation. Instead, user will find discovered services as icon buttons or tab menu items generated automatically. Additional menus are created and removed when the services are disabled or stopped.

C. Service Selection

In our controller prototype, we implemented semi-automated selection and manual selection. We avoided automated selection, since automated service execution has a high cost to revert its effect. Rather we implemented semi-automated selection. Context-aware services such as ubiTV [24] provide recommendation based on user's location, profile and preference. The system generated recommendation is displayed in a public display as well as user's mobile device through personalization. Then the selection is confirmed by users through their mobile devices. The detailed interaction mechanism and flows are described in [25].

In manual selection, users can use two interaction modes. First mode is network scanning based selection. In this mode, user selects an item from the generated menu in Fig. 5. When the menu is selected, the associated command is transferred to the target appliance. However, this method is problematic with a large number of services. Since new tab menus are generated for additional services, the tab menu gets long and the user needs to scroll sideways to find the target service. We remedy this problem by introduction the second interaction mode.

Fig. 5. List generated from network scanning.

Second mode is direct and camera-based "take-a-picture" approach. When using camera-based approach, a user explicitly takes a picture of a target appliance or a representative visual marker as shown in Fig. 6. As depicted, the user is taking a picture of switch of the light as an entry point visual marker for accessing and acquiring control user interface for the light appliance.

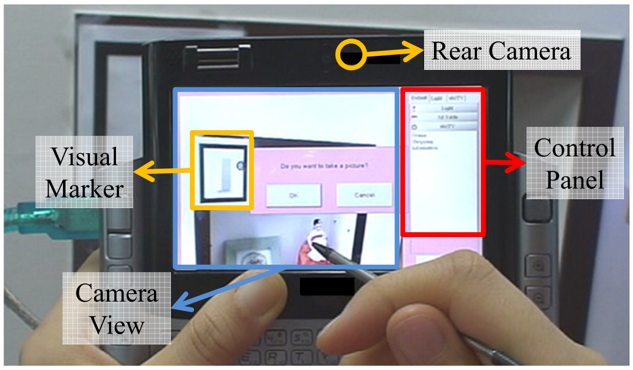


Fig. 6. Taking picture with the controller system.

Then the captured image is converted to binary image for labeling regions and detecting the visual marker in the input image with the patterns in the database. When the correct pattern is found from the image, the associated service or appliance ID is returned. From this appliance ID, appropriate control user interface is generated.

D. Personalization

As discussed earlier in section 3, we implemented two levels of personalization, user interface personalization and functional personalization. For both levels of personalization, we use personal profile stored on user's mobile device. User's profile is stored locally on mobile device which is read when the controller application is launched. The profile is named “userdata.ini” and follows the schema description,

Schema: User ID|UI Preferences|Functional Preferences
Ex) Father|Blue|News|Sports|Information|Leisure (1)

User ID is specified at the beginning of the profile, followed by a set of user interface preference and functional preferences. The specified example describes user preference on blue colored user interface and favored TV service channels. The personalization process checks user ID then matches user interface preferences with color, fonts, and menu style values. Also functional preferences of each appliance, which is specified by a single line and delimiters, are checked first when user interface is generated. Examples of functional personalization include filtering functions and contents or ordering them according to user preference.

The presentation of user interface layout is divided into two areas as depicted in Fig. 6. Camera view area and control panel area are statically fixed regardless of selected services for keeping consistency in the controller. User interface personalization changes colors used in the layout, fonts in text labels and menu styles. Functional personalization changes list items in the control panel to reflect user's preference described in user profile. The profile used for personalization is updated as user history is accumulated over duration of time and reflected on the appearance of the generated user interface.

E. Applications

First application is controlling navigation system. Typically, a user interacts with services by means of triggering, invoking, querying and controlling. Fig. 7 shows a user using our prototype as a directional input device to navigate the virtual world by physically moving and tilting the mobile device respect to the displayed grids. Through camera-based selection mechanism, the visual marker on the display is identified. Then the user can place the visual marker onto grids in mobile device screen which are marked with four directions.

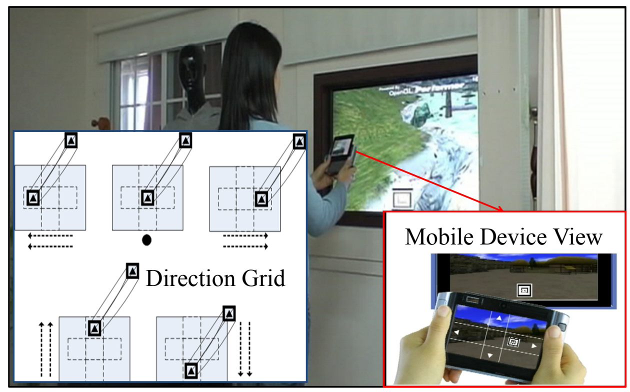


Fig. 7. Navigation input device.

Second application supports multi-user scenario when there is conflict between users. In multi-users scenario, our controller is used to notify users for preference difference in the same service area. If a user's interest conflicts with another user, a context-aware system starts mediation to reach a common service to both users. In a conflict situation, personal mobile device is used as a display for each user to notify conflicts and as an input device to express each user's selection [25]. Another use for our controller in multi-user scenario is that it can be used to hand-over control access [26]. Fig. 8 shows two users in mediation where TV control is being handed over from user A to user B.

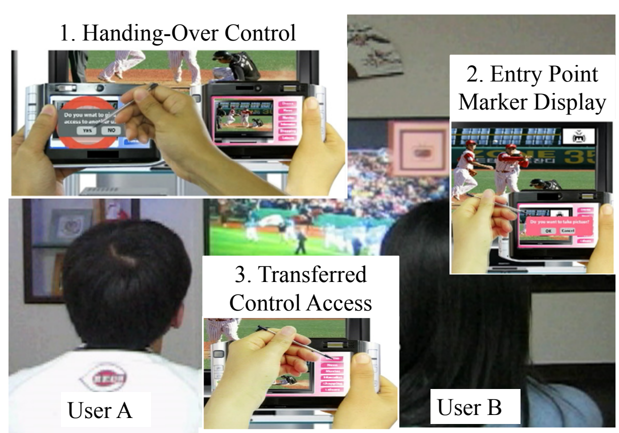


Fig. 8. Control hand-over.

Through user's mobile screen, TV channels that interest both users are recommended. Then the user with a higher priority or control right can keep the control access or yield and hand-over control access to other user involved in the conflict situation. When the control is handed over to user B, a visual cue representing an entry point to the service is displayed on TV which can be captured by the user to gain the appropriate control interface.

V. Experiment Results and Analysis

In this section, we show quantitative and qualitative experiment results of our prototype in ubiHome [24], a smart home test bed.

A. Experiment Setting

For the main controller system, mobile device with 1.2GHz CPU and 512MB memory is used. Also its internal camera and wireless network are used to take a picture and scan the network for service discovery.

For testing and evaluation purposes, we have implemented several appliances based on ubi-UCAM and UPnP which include smart TV, smart table, smart window and smart light. Smart appliances are connected by wired network in the same room. Fig. 9 shows implemented appliances in our smart home test bed. Smart appliances are categorized into appliances with screen and appliances without screen. Appliances with screen have screen-embedded visual markers [27]. These markers are hidden when the appliances are off or in stand-by mode, but made available when the user is in the vicinity of the appliance by verifying the location of user. Appliances without screen have physically embedded markers where a visual cue is embedded into a physical form factor of the appliance.

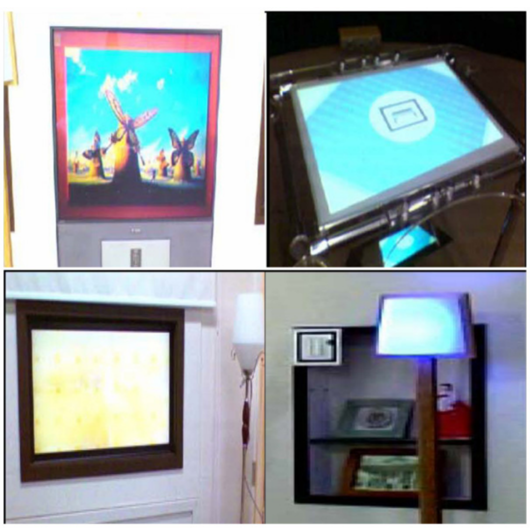


Fig. 9. Implemented smart appliances, smart TV and smart table are pictured on top, smart windows and smart light are shown in bottom.

B. Quantitative Results

First we measured the service discovery performance of our prototype which needs to be operated in real-time for controller application. To measure service discovery time, we implemented a logging feature to mark the start of the first discovered service and also mark the end of the last service added. The total time is then measured. Fig. 10 shows the code used for this experiment with a log that shows the time it took to discover each service. In the smart home test bed, we ran 3 separate servers for simulating and hosting the implemented appliances. We varied the number of deployed services from 1 to 10 which is reasonable number of services in the room-level networked home environment. We measured 20 times each and calculated the average performance.

```
#include <ctime>

clock_t begin;
void Initialization()
{
  begin = clock();
}

void ListAddDevice(UPnPDevice *device)
{
  // device discovered
  while (device->Services != NULL) {
  // service discovered
   clock_t end = clock();
  double msec = 1000.0 * (end - begin)/ CLOCKS_PER_SEC;
  // log on file
  }
}
```

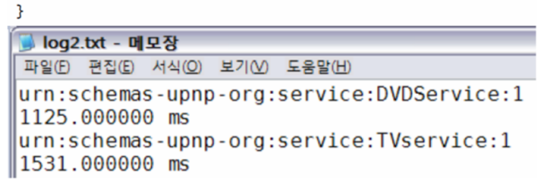


Fig. 10. Code snippet and a log for service discovery measurement.

Fig. 11 shows the service discovery performance of our prototype measured in milliseconds. As the number of deployed services in the smart home increases, the total time to discover all services also increases. But from our work, all 10 services are discovered in under 2 seconds, which is reasonable real time. The discovery time difference between 5 services and 10 services is minimal. Therefore we did not test with a service number greater than 10, this pattern holds as more services are incrementally added and removed in the environment. Practically, this background process can be operated in real time with greater number of services as well.

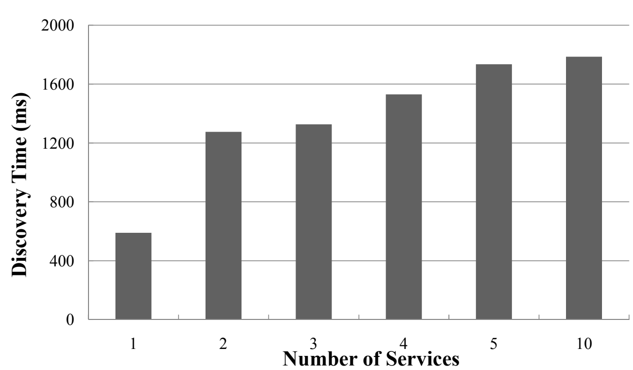


Fig. 11. Service discovery performance

C. Qualitative Results

For qualitative measure and user perception, we ran a small user study to gain insights in selection and interaction with our system. 6 participants between the age of 25 and 28 were recruited (2 females, 4 males). We gave participants four tasks to complete using the controller system. The tasks were designed to see how each participant would perform using both network scanning-based and camera-based "take-a-picture" method. In each task, users were asked to control TV and lights service for a series of commands. Two rounds of four tasks were given to participants, to see how they performed after they have used the controller system. After two rounds of tasks were finished, we gave out participants a short questionnaire. To compare how fast our participants completed given tasks, we first measured time for a trained person to complete each task. This was labeled as baseline in our data. Completion time was measured by the experimenter with a stopwatch who announced the start of task and confirmed the completion of the task.

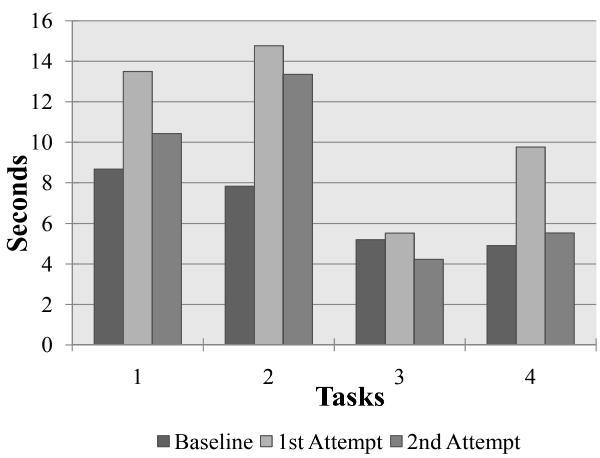


Fig. 12. Average task completion time.

Fig. 12 shows average task completion times for 4 tasks. We observed that participants completed the tasks as fast as the trained expert in their second attempt. For task 1, the average of 5.59 seconds is reduced in 2nd attempts, which is about 41% faster than the first attempt. Similarly, the average of 5.17 seconds (35%) for task 2, the average of 2.28 seconds (34%) for task 3, and the average of 3.32 seconds (40%) for task 4 are reduced in 2nd attempts. This indicates that our prototype was quite easy and intuitive to use, because participants could learn the user interface in a very short time.

From the collected post-test questionnaire, we observed that participants understood how our two interaction modes work intuitively. For measuring intuitiveness, network scanning based approach scored higher with a point of 3.83 from a 5 level Likert scale, followed by camera-based "take-a-picture" approach with comparable 3.67. In the ease of use aspect, more participants preferred network-based approach (4.17) than "take-a-picture" approach (3.17). The results are shown in Fig. 13.

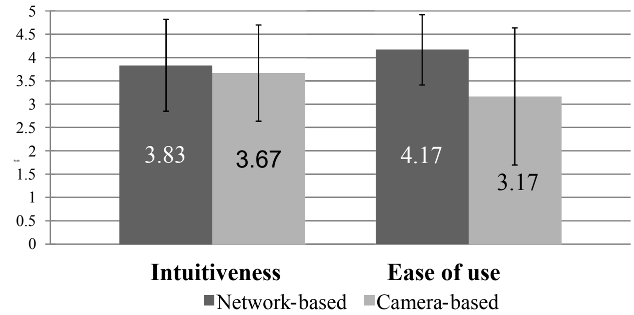


Fig. 13. Intuitiveness and ease of use scores for two interaction modes.

In the user study, 4 out of 6 participants preferred to use network-based approach with generated menu. This could be explained by the fact that end-users are used to traditional buttons and menu interface. Also discussed in section 2, manual selection requires users to explicitly intervene in selection steps whereas network-based approach is run as a background process. Still, camera-based approach is novel and applicable to a wide range of applications. Other feedbacks reported a number of usability issues. For example GUI buttons in our button were small that the participants had trouble using it. Such comments indicate the needs of carefully designing user interface for our interaction methods and the system.

VI. CONCLUSIONS

In this paper, we introduced mobile device based universal appliance controller system. In designing phase of the prototype, we reviewed various techniques and methods in designing controller system. Then we categorized most common requirements into the three main functions, service discovery, service selection and personalization. We presented an extended architecture of two interchangeable interaction modes with network scanning and camera-based approaches as well as user interface and functional level of personalization.

Through prototype implementations and experiments in smart home test bed, we presented that performance of our system can support real time operation. We also observed from a user study where participants performed given tasks in a comparable time. Also positive responses toward intuitiveness and different preference of two interaction modes are explored. Un-trained participants could complete given tasks as fast as a trained expert in their second attempts by showing more than 30% of reduction in task completion time. Both interaction modes had advantages and disadvantages. By combining them and making users to selectively choose interaction mode we extended the use of our controller system.

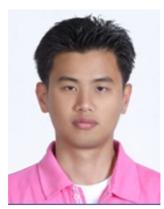
Our work has some limitations as comments from our user study participants reveals. First, we found several usability issues and rooms for improvement in GUI design of user interface. Many people complained about the size, and location of buttons which need a professional touch from user interface designers. This would not affect the system performance directly, but it surely had an impact on how users feel in ease of use and intuitiveness. Therefore we are starting to develop natural view based user interface to tackle this issue [28]. Also current prototype has no structured protective mechanism other than being a passive mobile system. This is one future work area where our prototype can improve. Moreover we will look further into what kinds of interaction modes are effective for different tasks in multiple devices and complex environment.

ACKNOWLEDGMENT

We would like to thank Hyejin Kim and Choonsung Shin for their contribution and continuous support in developing featured applications on this project.

REFERENCES

[1] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: a ubiquitous input device," *IEEE Pervasive Computing*, vol.5, no.1, pp. 70-77, Jan-March. 2006.
[2] http://wireless.itworld.com/4267/060127mobsales/
[3] http://www.capv.com/home/Multiclient/MobileImaging.html
[4] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev,* vol. 3, no. 3, pp.3-11, Jul. 1999.
[5] T. Perumal, A. Ramli, and C. Leong, "Design and implementation of SOAP-based residential management for smart home systems," *IEEE Trans. Consum. Electron.,* vol. 54, no. 2, pp. 453-459, May. 2008.
[6] T. Hwang, H. Park, and J. Chung, "Personal Mobile A/V Control Point for Home-to-Home Media Streaming," *IEEE Trans. Consum. Electron.,* vol. 54, no. 1., pp. 87-92, February. 2008.
[7] S.-H. Lo, C.-C. Lin, and M.-S. Chen, "Controlling digital TV set-top box with mobile devices via an IP network," *IEEE Trans. Consum. Electron*., vol. 52, no. 3, pp. 935-942, Aug. 2006.
[8] M. R. Cabrer, R. P. D. Redondo, A. F. Vilas, J. J. P. Arias, and J. G. Duque, "Controlling the smart home from TV," *IEEE Trans. Consum. Electron.,* vol. 52, no. 2, pp. 421-429, May. 2006.
[9] A. E. Nikolaidis, S. S. Papastefanos, G. I. Stassinopoulos, M.-P. Drakos, and G. A. Doumenis, "Automating remote configuration mechanisms for home devices," *IEEE Trans. Consum. Electron*., vol. 52. No. 2, pp. 407-413, May. 2006.
[10] E. Rukzio, K. Leichtenstern, V. Callaghan, A. Schmidt, P. Holleis, and J. Chin, "An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning," Proceedings of 8th International Conference on Ubiquitous Computing (UbiComp2006), pp. 87-104, 2006.
[11] K. Komine, Y. Sawahata, N. Uratani, Y. Yoshida, and T. Inoue, "Evaluation of a prototype remote control for digital broadcasting receivers by using semantic differential method," *IEEE Trans. Consum. Electron*., vol. 53, no. 2, pp. 561-568, May. 2007.
[12] R. Verhoeven and W. Dees, "Defining services for mobile terminals using remote user interfaces," *IEEE Trans. Consum. Electron.,* vol. 50, no. 2, pp. 535-542, May. 2004.
[13] J. Sathyan and R. Ramakrishnan, "A unique self-contained universal remote control," *IEEE Trans. Consum. Electron.,* vol. 50, no. 4, pp. 1151-1155, November, 2004.
[14] W. Lee and J. Wang, "A user-centered remote control system for personalized multimedia channel selection," *IEEE Trans. Consum. Electron.,* vol. 50, no. 4, pp. 1009-1015, November. 2004.
[15] P. Chong, P. So, P. Shum, X. Li, and D. Goyal, "Design and implementation of user interface for mobile devices," *IEEE Trans. Consum. Electron.,* vol. 50, no. 4, pp. 1156-1161, November, 2004.
[16] B. Yuksekkaya, A. A. Kayalar, M. B. Tosun, M. K. Ozcan, and A. Z. Alkar, "A GSM, internet and speech controlled wireless interactive home automation system," *IEEE Trans. Consum. Electron*., vol. 52, no. 3, pp. 837-843, Aug. 2006.
[17] X. Zeng, A. O. Fapojuwo, and R. J. Davies, "Design and performance evaluation of voice activated wireless home devices," *IEEE Trans. Consum. Electron.,* vol. 52, no. 3, pp. 983-989, Aug. 2006.
[18] H. Yoon, S. Lee, and W. Woo, "Discovery, Selection and Interaction as Requirements for Personal Mobile Systems," ISUVR 2007, pp. 23-24, 2007.
[19] UPnP Forum, http://www.upnp.org
[20] Jini, http://www.jini.org/wiki/Main_Page
[21] NFC Forum, http://www.nfc-forum.org/home
[22] M. Bonett, "Personalization of web services: opportunities and challenges," Ariadne Issue 28, June, 2001, Available: http://www.ariadne.ac.uk/issue28/personalization/intro.html
[23] S. Jang and W. Woo, "Ubi-UCAM: A Unified Context-Aware Application Model," Modeling and Using Context (LNCS 2680), pp. 178-189, June. 2003.
[24] Y. Oh and W. Woo, "A Unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness ", UCS 2004 (LNCS 3598), pp. 192-202, 2004.
[25] C. Shin, H. Yoon, and W. Woo, "User-Centric Conflict Management for Media Services Using Personal Companions," *ETRI Journal*, vol. 29, no. 3, pp.311-321, June. 2007.
[26] H. Yoon, H. Kim, C. Shin, W. Jung, and W. Woo, "Camera-based Context-aware Smart Object Controller," Advances in Pervasive Computing, pp. 117-120, 2007.
[27] H. Yoon, H. Kim, and W. Woo, "Personal Companion: Personalized User Interface for U-Service Discovery, Selection and Interaction," Proceedings of 12th HCI International 2007 (LNCS 4555), pp. 1052-1066, 2007.
[28] H. Yoon, W. Woo, and S. Lee, "NVUI Framework: Natural View User Interface Framework for Consistent and Intuitive Appliance Control," Proceedings of International Symposium on Ubiquitous VR 2008, pp. 63-66, 2008.



Hyoseok Yoon received his B.S. in Computer Science from Soongsil University, Seoul, Korea in 2005 and M.S. in the Department of Information and Communications (DIC), School of Information and Mechatronics (SIM) from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea in 2007. He now is a Ph.D candidate in DIC, SIM, GIST since 2007. His research interests include multi-modal mobile interaction for ubiquitous computing, context-aware computing and HCI.

Woontack Woo received his B.S. in Electronics Engineering from Kyungpook National University in 1989 and his M.S. in Electronics and Electrical Engineering from POSTECH in 1991. In 1998, he received his Ph.D. in Electrical Engineering Systems from University of Southern California (USC). In 1999, as an invited researcher, he joined Advanced Telecommunications Research (ATR), Kyoto, Japan. Since Feb. 2001, he has been with the Gwangju Institute of Science and Technology (GIST), where he is an Associate Professor in the Department of Information and Communications (DIC) and Director of Culture Technology Institute (CTI). His research interests include 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitous computing, etc.