

A New Architecture for Home Computing

Kyoko Matsuura, Tadahiro Hara, Akashi Watanabe, Tatsuo Nakajima
Department of Information and Computer Science
Waseda University
3-4-1 Okubo Shinjuku Tokyo 169-8555 JAPAN

Abstract

Currently, many companies have been working to build home servers that manage home appliances in our houses. However, the approach is not appropriate for future ubiquitous computing environments because it is very difficult to allow us to use home appliances in a seamless way when we move at several places. We propose a new architecture called *personal home server*. In our approach, each user has a home server and customizes how to use these appliances according to his preferences and situations. The approach is very helpful to build future domestic environments.

1 Introduction

The paper proposes a *personal home server*, that is a new architecture for home computing. A personal home server is carried by each person, and configures home appliances near him in a spontaneous way. The personal home server has the following characteristics.

- Our system allows us to configure to control home appliances according to each user's preference and situation.
- Our system adopts standard protocols as much as possible to incorporate commercial products.
- Our system can be used to configure various appliances at any places.

Our system supports context-awareness to select which appliance a user likes to control. The characteristic reduces complexities in our lives because appliances that he has no interest are hidden from us. Our system has adopted SOAP to control appliances. We believe that SOAP is one of the most promising candidate for the protocol to make middleware simple. Also, the command to control an appliance and its name are encoded in a URL. Since many presentation documents allow us to include URLs, so our system can use various presentation documents such as HTML, MS Power Point, Flash and SMIL to control home appliances. This is very important because these documents will become ubiquitous to

design user interface now. The approach enables us to separate user interface design from application design. A personal home server selects a target appliance from attributes encoded in URLs, and a command is translated to a SOAP message. Moreover, in our approach, since a personal home server chooses the configuration of appliances near a user, it is easy to configure them at any places where he is.

2 Future Home Computing Scenarios and Requirements

In this section, we show three future scenarios in home computing environments. In the first scenario, Tatsuo and Rei are in their living room. Tatsuo likes to turn on a television by using his cellular phone. While they are watching a TV program, Rei likes to buy a fantastic table appeared in the program. She plays back a scene to find the table on another display near her, and orders it by navigating a control panel on the display by using her phicom(physical icon)[1]. In the scenario, two persons need to control the same appliance from their own different devices.

In the second scenario, Tatsuo visits his friend's house. In the friend's house, Tatsuo likes to record a TV program suddenly. He configures his friend's television and a video recorder in Tatsuo's personal storage in his pocket. Then, when he goes back to his home, he connects his personal storage to his television to play the recorded program in the storage. In the scenario, home appliances should be controlled in a uniform way from any devices, and he will be able to connect a television near him to his personal storage in a spontaneous way.

In the last scenario, Tatsuo likes to watch a television in his friend's car, and to change a channel. Also, in a future town, there may be many public televisions. He likes to control the televisions in the same way as in his house.

The above scenarios require to satisfy the following requirements.

- A different user has a different requirement to control home appliances.

- Each user likes to use the same way to control home appliances whose functions are similar.

The personal home server satisfies the above requirements. We believe that the requirements are a key to make home computing practical.

3 Design Issues

The section discusses how our system satisfies the requirements described in the previous section. We believe that the following issues are key to satisfy the requirements.

- The system should be simple and based on standard protocols.
- The system should be separate GUI design from application design.
- The system should be customized for each person.

The system should be very simple, and it is easy to be implemented in various appliances easily. The request is especially important to collaborate various specialized appliances. Currently, generic appliances are popular, but it is necessary to decompose the generic appliances into specialized appliances to reduce complexities caused by the genericity[4]. In the future, everyday objects may give various information through embedded sensors. Also, the objects should be integrated by middleware infrastructures with home appliances. Therefore, simple protocols are desirable for future devices. We have adopted UPnP to control any appliances because UPnP is a simple protocol, and can be implemented in a small size. Our middleware infrastructure is implemented on UPnP, and provides high level spontaneous interactions.

Currently, many services are deployed on the Internet, and are accessed by using the SOAP protocol. In the future, it is important to integrate services on the Internet and services on home appliances. If they adopt different protocols, programming to compose these services will be very complex. We believe that simplicity is the most important to make a system robust and practical. Therefore, we have adopted the SOAP protocol to access home appliances. On the other hand, a service specification defined in UPnP is not appropriate to support high level spontaneous interaction. Thus, we have defined a service specification based on RDF.

To separate GUI design from application design, we choose to use standard presentation documents such as HTML, MS Power Point, Flash and SMIL since the presentation documents are popular by interaction designers, and there are many tools to create them. However, since SOAP is not popular to be embedded in a standard presentation document. We need a simple way to allow us to use standard presentation documents to control home

appliances. Our approach allows us to control appliances from various presentation documents by embedding commands to control appliances and their attributes in URLs. The approach also allows us to specify our preferences in URLs as attributes for the appliances. We believe that our approach allows interaction designers to design various home computing applications by using their existing presentation design tools.

In the previous section, we show that a home gateway is not a right approach to solve many problems in home computing. We believe that the problems can be solved easily by making a home gateway personal. This means that each person carries a personal device such as a cellular phone or a personal digital assistant that contains functionalities of a home gateway. We call the approach a *personal home server*. In the approach, home appliances near a user can be configured by using his personal home server. The approach is very attractive because the configuration of appliances can be customized by using his preferences since the configuration is determined by his personal home server. Also, private information cannot be used by other devices. Thus, the approach is desirable to protect privacy information.

Since the approach configures appliances near a user, the personal home server can be used to control a variety of appliances at any places. For example, a user can control any music players whether he is in his house, in an airport, or in an automotive space.

4 Design and Implementation

4.1 Overview of Personal Home Server

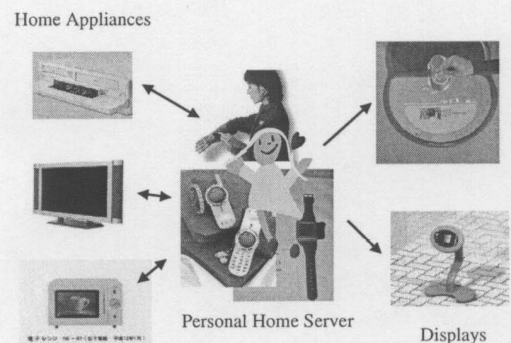


Figure 1: Overview of Personal Home Server

Figure 1 shows an overview showing how a personal home server works. A personal home server is implemented in a personal devices like a cellular phone, a wrist watch, or a jacket. Thus, the server can be carried by a

user anytime anywhere. The personal home server collects information about home appliances near a user, and creates a database storing information about these appliances. Then, it creates a presentation document containing the attributes of appliances and the commands to control them. A display near the user also detects the personal home server, and retrieves the presentation document containing the automatically generated user interface. The display shows the presentation document on the display. The document contains URLs embedding the attributes of appliances and their commands. Also, the presentation document is customized according to a user's preference. When a user touches the display, a URL containing the attributes of an appliance and its command is transmitted to his personal home server via the HTTP protocol. The server translates the URL to a SOAP command by using a database containing information about the appliance that he likes to control. Finally, the SOAP command is forwarded to the target appliance.

4.2 Structure of Personal Home Servers and Appliances

In ubiquitous computing environments, it is not easy to say which functionality is useful because the functionality should be different according to a user's situation. In our approach, a sever consists of several components, and the components should be dynamically loaded or unloaded according to the current situation.

Our system has been written in Java, and we have adopted the OSGi(Open Service Gateway initiative) Service Gateway Specification[5] as a component framework. The system is current running on Linux, and we are using the Blackdown's Java virtual machine. In the current prototype, we have adopted Oscar which is a open source implementation of the OSGi service gateway specification.

4.3 Service Database

In our system, a personal home server knows respective appliances via the SSDP protocol. Each home appliance transmits advertisement messages periodically. Also, a personal home server asks respective appliances to transmit advertisement messages by sending a SEARCH message. After receiving the advertisement messages, the personal home server transmits a HTTP GET request to get a service specification document to each appliance. Our service specification document is represented as an RDF document. For example, a service specification document for a light appliance is specified as follows.

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:core='http://www.dcl.info.waseda.ac.jp/homecomp/
    schemas/core/1.0#'
  xmlns:soap='http://www.dcl.info.waseda.ac.jp/homecomp/
    schemas/soap/1.0#'>
  <rdf:Description rdf:about="urn:homecomp:x10light1-855503ae">
```

```
<core:friendlyName>Light Controller Device</core:friendlyName>
<core:manufacturer>X10</core:manufacturer>
<core:function>light</core:function>
<core:modelDescription>This device allows us to control
  the brightness of a light appliance via the X10 protocol.
</core:modelDescription>
<core:modelName>Light Controller</core:modelName>
<core:modelNumber>1.0</core:modelNumber>
<core:modelURL>http://www.x10.org</core:modelURL>
<soap:WSDL>light.wsdl</soap:WSDL>
<soap:controlURL>/control/light1</soap:controlURL>
<core:type>Ceiling</core:type>
<core:location>Living Room</core:location>
</rdf:Description>
</rdf:RDF>
```

The service specification document contains a link to a WSDL document identifying commands that can be accepted. If an appliance contains several functionalities, its specification document may contain several links to WSDL documents. Also, attributes of the document are used to identify a target appliance. For example, if a URL contains `?function=light&?type=ceiling`, the function tag and type tag in the document are examined.

The service database in a personal home server contains all service specification documents detected currently. The personal home server transmits the SEARCH message periodically, and the service specification document of an appliance is removed if there is no advertisement message within a fixed duration.

Our service database offers two functions to access information in the database. The first function is used to transmit a command. When a personal home server receives a command, it extracts the attributes of function for a target appliance like `?func=television` in the URL. The service database returns an IP address of the appliance. Thus, the command is converted to SOAP, and it is transmitted to the target appliance specified by the IP address. The second function is used for generating a presentation document. Currently, we support a HTML based presentation document. The function returns an array containing information of respective appliances. The information of the appliances includes the names, command names, and the attributes of the appliances. These information is converted to URLs, and they are embedded in a HTML document.

In our scheme, each appliance can specify various attributes in a service specification document. The attribute can be used to identify a target appliance. For example, a light appliance can contain a type tag to specify the type of a light such as a ceiling light. Also, some attributes can be filled by sensors attached to a home appliance. For example, if an appliance can know its location through an attached location sensor, it fills its location attribute in its service specification document. On the other hand, a service specification document for an appliance may contain a URL of a Web server. In this case, the Web server

is connected to sensors, and they return the context information of the appliance. The server returns a RDF document containing the context information. The information is tagged by a serial ID. Thus, the server merges the information to its service specification document, and stores the merged document in the service database. Since the context information is stored as attributes of a service specification document, the information can be used in a presentation document to identify each appliance to take into account a user's preference.

4.4 How a Personal Home Server Works?

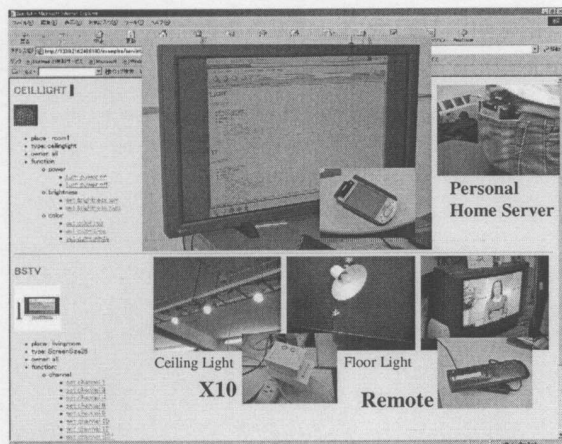


Figure 2: Current Personal Home Server Prototype

In this section, we describe how our systems works by using an example. In the example, we consider that a television and two lights are in a room. We assume that one is a ceiling light and another is a floor light. When a user enters a room, his personal server in his pocket detects appliances by receiving SSDP advertisement messages containing IP addresses from the television and the lights. The personal home server accesses Web servers of both appliances, and retrieves their service description documents. Then, the documents are stored in the service database of a personal home server.

On the other hand, a display device near the user detects an advertisement message from the personal home server, and retrieves an automatically generated HTML document. The document contains URLs for controlling the television and a ceiling light. The user specifies that he likes to use only ceiling lights in a rule stored in the context database. Thus, the ContextInference component removes information about a floor light.

The document is shown by a browser of the display device as shown in Figure 2. In this example, a user can navigate the browser by using

a touch panel¹. When the user clicks a URL, `http://102.10.2.2/?func=light&?type=ceiling&!power=on/`, a GET command containing the URL is transmitted to the personal home server. The server searches a light appliance that matches to specified attributes in the service database, and finds that its IP address is 102.10.2.10. Also, a command encoded in the URL is converted to a SOAP request, `power("ON")`. Finally, the commands is transmitted to a target light appliance whose IP address is 102.10.2.10, then the appliance turns on its power. In our current prototype, light appliances are controlled via an X10 device, and an analog television is controlled through an intelligent remote control device.

5 Conclusions

In this paper, we have described design and implementation of a personal home server. The personal home server configures home appliances near a user in a spontaneous way. Since each user has his own personal home server, the configuration is determined according to his preference. In traditional approach based on a home gateway, the configuration is determined according to a place where a user is, but our approach determines the configuration according to a person. We believe that our approach is more natural and simple since the configuration policy can be defined for each person. In the near future, each person has several personal devices anytime anywhere, and it is not realistic to configure appliances by using a server that resides in every place.

References

- [1] H. Ishii, B.Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", In Proceedings of Conference on Human Factors in Computing Systems,1997.
- [2] T.Nakajima, D.Ueno, I.Satoh, H.Aizu, "A Virtual Overlay Network for Integrating Home Appliances", In the Proceedings of the 2nd International Symposium on Applications and the Internet, 2002.
- [3] T.Nakajima, A.Hasegawa, T.Akutagawa, A.Ibe, K.Yamamoto, "Universal Interaction: Making Existing Interactive Applications Context-Aware", Technical Report, Waseda University, 2002.
- [4] D.A. Norman, "The Invisible Computer: Why good products can fail, the personal computer is so complex, and information appliance are the solution", MIT Press, 1998.
- [5] Open Server Gateway Initiative, "OSGi Service Platform Specification", <http://www.osgi.org/>.

¹The demonstration uses our Universal Interaction system[3] to use a PDA's touch panel for navigating the browser.