

Design, Implementation and Standardization of Overlay Networking Protocols and Device Metadata towards Smart Home

Norihiro Ishikawa, Komazawa University *

Abstract

Many researches and standardizations have been done for controlling and managing home networks and devices during past ten years. Such activities and technologies include UPnP/DLNA, ECHONET and Continua Health Alliance and so on. However, such technologies control and manage specific home devices. There have been no technologies and standards for controlling and managing such heterogeneous home devices over heterogeneous home networks in an integrated manner. In addition, such standards fully define technologies for controlling and managing such home devices from outside home through mobile phones and cloud servers.

Compared with traditional home networking technologies, peer-to-peer/overlay networking technologies have functions to realize resource discovery, multi-hop networking in a highly distributed manner. Peer-to-peer/overlay networking is one of the most important and suitable technologies for communications among home devices over heterogeneous home networks. In addition, metadata is also an important technology for describing various home devices in a consistent manner.

We have designed overlay networking protocols and metadata for realizing various applications, which include home devices control and management. We explain our overlay networking and device metadata technologies, and describe the general middleware platform, which have implemented such technologies in more details. Our middleware platform are now running over PCs, home gateways (HGWs), mobile phones, tablet devices, cloud servers and so on. Applications towards smart home having been developed over our middleware platform are also described. Peer-to-peer Universal Computing Consortium (PUCC) was established in 2005, to deploy our overlay networking protocols and device metadata technologies toward de facto standardization.

Keyword: Overlay Network, Home Network, ECHONET, UPnP, PUCC

1. Standardization on Home Networks and Devices

Many standardization activities have been done for controlling and managing home networks and devices during past ten years. Such activities include UPnP/DLNA, ECHONET, Continua Health Alliance and so on. We briefly describe such standardization activities on home networks devices, and the current status on development and deployment of products based on such standards

1.1 UPnP/DLNA

Universal Plug and Play (UPnP) [1] is a technol-

ogy that allows the use of standard Internet technology for connections between networked devices. The Digital Living Network Alliance (DLNA) [2] has developed design guidelines for the sharing of video and other digital content by home appliances, personal computers, and mobile devices through the use of UPnP and other standard Internet technology. Digital media players (DMP), which present content for user enjoyment, use protocols such as Simple Service Discovery Protocol (SSDP) to discover digital media servers (DMS) that have contents, and retrieve a list of contents from a Content Directory Service (CDS). Then a request for the desired content is sent and the content is streamed from the

* Professor, Faculty of Global Media Studies, Komazawa University

DMS via the LAN to the DMP for viewing.

The DLNA Guidelines 1.0 and 1.5 were developed in October 2004 and March 2006 respectively, and DLNA compliant devices have been developed, focusing on PCs, TVs, Hard Disk Drives (HDD) and Blu-ray disk players, digital music players and so on. DLNA is also conducting a logo certification program to ensure interoperability between products developed by various manufacturers. To protect copyright in the transfer of content, the Link Protection Guidelines for Digital Transmission Content Protection over IP (DTCP-IP) were developed in October 2006. UPnP/DLNA-compliant AV devices have been developed and deployed, to allow sharing of downloaded digital content (e.g. Video, Audio) among AV devices, listening digital music from HDD via a digital music player and so on.

1.2 ECHONET [3]

Energy Conservation and Homecare Network (ECHONET) is a standard for control of air conditioners, lighting equipment, and other home appliances and facility devices such as power consumption monitors and other sensors.

While Version 1.0 of the ECHONET specifications was developed in March 2000, some manufacturers have developed ECHONET-compliant products, and such products are available in the market, but such products have not yet spread throughout the market as a whole. Because the recent trend is to communicate with AV devices, version 3.60 of the ECHONET specifications was published in December 2007. It includes gateway specifications that allow conversion between ECHONET and UPnP.

ECHONET Lite was published in December 2011. ECHONET Lite simplifies ECHONET specifications by removing their specifications in physical layer. The main purpose of ECHONET Lite is to apply ECHONET specifications to Home Energy Management System (HEMS). For this purpose, ECHONET Lite defines an ECHONET object for a smart meter, and ECHONET properties for energy management on home appliances such as air conditioner.

1.3 OSGi [4]

Open Services Gateway Initiative (OSGi) was founded in March 1999 and has since been working on standards for mechanisms for the remote and dynamic addition and changing of home gateway (HGW) functions. OSGi middleware is installed on HGWs that execute Java Virtual Machine (VM). OSGi middleware enables a HGW to add various functions to be executed on it, by downloading specific Java components called bundles from cloud servers. OSGi release 4, published in 2005, extended the OSGi specifications for mobile devices. OSGi middleware has been implemented and used on various devices such as mobile phones and in-vehicle devices, in addition to HGWs.

1.4 Continua Health Alliance [5]

Continua Health Alliance was founded in June, 2006. Continua Healthcare design guidelines version 1 was developed in June 2009. The guidelines define specifications, which are based on ISO/IEC 11073 Personal Health Data (PHD) standards [6], between healthcare devices and gateways (e.g. PC) via USB and Bluetooth. Such healthcare devices include blood pressure monitor, thermometer and weighing scale and so on. Many healthcare devices are available in the market, which are compliant with Continua Healthcare design guidelines. PC software that implements protocol stacks and device drivers, which are compliant with the above design guidelines, are available from some software vendors.

2. Problems on Standardization Regarding Home Networks and Devices

Various standards have been developed for different classes of home devices as described in section 1. There is no compatibility among those standards. In addition, there are no standards for security devices such as security camera and electronic key. Such devices support proprietary specification developed by vendors. Two main problems posed by this situation are described below (Figure 1).

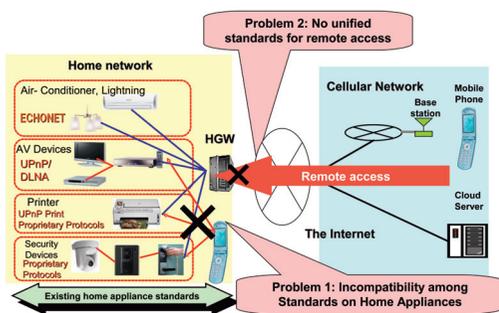


Figure 1 Problems regarding remote access to home networks

Problem 1: Different classes of home devices support different standards

The current situation is that there is no compatibility among the various standards. A mobile phone to control home devices compliant with different standards must support multiple standards. However, current mobile phones cannot support many standards for home devices, due to limited processing capability and memory capacity. In addition, Home Gateways (HGWs) must also support many standards to control and manage home devices which comply with different standards.

Problem 2: No standards for controlling and managing home devices from outside home

From the mobile phone point of view, it is very important for a mobile phone to be able to remotely control home devices via a cellular network. From the cloud server point of view, it is very important for a cloud server to be able to remotely manage home devices via the Internet. However, many standards on home devices only define protocols for communications within a home network. In current standards on home devices, much attention has not been paid to communications for remote control and management of home devices from outside home.

To solve those problems, we have conducted joint researches with Ericsson, Kyoto University and Keio University to develop peer-to-peer/overlay networking and metadata technologies as a solution for unified control and management of various home devices connected to home networks, from mobile phones and cloud servers. Based on the results of the above joint researches, we have also established Peer-to-peer Universal Computing Consortium

(PUCC) in 2005, to deploy our overlay networking and metadata technologies toward de facto standardization. Such activities include the development of PUC specifications, development of prototype systems using PUC technologies, demonstrations on PUC technologies at world-famous exhibitions (e.g. CES and MWC) and the efforts toward International Standardization [7].

We describe our peer-to-peer/overlay networking and metadata technologies, and the general middleware platform, which have implemented such technologies in more details. Our middleware platform is now running over PCs, HGWs, mobile phones including feature phones and smart phones, tablet devices, web servers and so on. Some applications that we have developed over our middleware platform are also described.

3. PUC Peer-to-peer/overlay Networking Protocols and Device Metadata Technologies

Compared with traditional home networking technologies described in Section 2, peer-to-peer/overlay networking technologies have functions to realize resource discovery, multi-hop networking in a highly distributed manner. Peer-to-peer/overlay networking is one of the most important and suitable technologies for communications among home devices over heterogeneous home networks. In addition, metadata is also an important technology for describing various home devices in a consistent manner.

To this purpose, we have developed peer-to-peer/overlay networking protocols and metadata technologies, initially developed by joint researches among NTT DoCoMo, Ericsson, Kyoto University. We have also established the PUC in 2005 to deploy our overlay networking protocols and device metadata technologies toward de facto standardizations. The latest PUC specifications release 3, which were developed in March 2012, defines 11 specifications as shown in Figure 2. PUC activities are described in details in the article [7]. PUC peer-to-peer/overlay networking protocols and device metadata technologies are described below [8].

- (1) PUCG Architecture Ver. 2
- (2) PUCG Basic Protocol Ver. 3
- (3) PUCG Basic Protocol - Light Profile Ver. 2
- (4) PUCG Device Discovery Service Invocation Protocol Ver. 3
- (5) PUCG Printing Protocol Ver. 3
- (6) PUCG Imaging Protocol Ver. 2
- (7) PUCG Streaming Protocol Ver. 3
- (8) PUCG Device and Service Metadata Template Ver. 3
- (9) PUCG Metadata Specification –
 - Part1: IEEE 1394 Devices Ver. 1
 - Part2: UPnP Devices Ver. 2
 - Part3: ECHONET Devices Ver. 1
 - Part4: Other Devices Ver. 1
 - Part5: Camera Device Ver. 1
 - Part6: RFID Reader Device Ver. 1
 - Part7: Felica Reader Device Ver. 1
 - Part8: Barcode Reader Device Ver. 1
 - Part9: Location Information Ver. 1
 - Part10: IEEE 11073 Device Ver. 1
 - Part11: OSGI-based Device Ver. 1

Figure 2 PUCG Specifications Release 3

3.1 PUCG Architecture

In the PUCG architecture, bidirectional communication entities, called PUCG nodes, construct a PUCG network by establishing a PUCG session between them. The PUCG nodes communicate one another using the PUCG sessions. Each PUCG node has a unique ID. The key elements of the PUCG architecture are defined as follows.

PUCG node: PUCG node is an independent, bidirectional communication entity. In the PUCG architecture, it can be PCs, HGWs, home appliances, mobile phones, tablet devices, web servers, or any of a variety of devices. Each node has a unique ID and communicates using the ID independent from underlying networks (e.g. the Internet).

PUCG network: The term “PUCG network” means a logical collection of PUCG nodes that have a common interest and obey a common set of policies. A PUCG session between PUCG nodes is established on mutual trust. Each PUCG node can enter or depart the PUCG network at its convenience. PUCG messages are sent from one PUCG node to another directly or via some intermediary PUCG nodes. Routing information is discovered by broadcasting an inquiry message to the PUCG network.

PUCG message: This is data object which is sent and received between PUCG nodes. A PUCG message is a basic unit of exchanging data and has a unique ID.

PUCG session: This is a communication channel established between PUCG nodes. PUCG messages are transmitted along the PUCG sessions.

The PUCG architecture is shown in Figure 3.

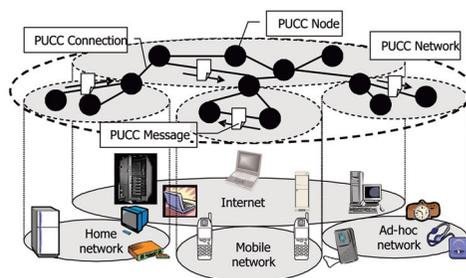


Figure 3 PUCG Architecture

In the PUCG communication model, the role of each communicating entity is not always clearly distinguishable. In order to design effective PUCG communication protocols, the “role” concept is introduced. We have established the PUCG communication model by looking into existing peer-to-peer applications. For example, we found three kinds of nodes in peer-to-peer distributed search applications [9] [10] [11]. The first provides content information (e.g. its location) in response to search requests. The second requests the content search. The third relays the search request and their replies. Those applications suggest that PUCG nodes can have three roles.

Producer role: A node acts as a Producer when it provides application data or services.

Consumer role: A node acts as a Consumer when it asks for a service or consumes application data without any request or in response to its request.

Relay role: A node acts as a Relay when its current communication task is to forward application data, service requests and their replies.

In peer-to-peer applications, a node may play any of these three defined roles and the peer-to-peer application dynamically determines which one is to be used. Based the roles described above, we define two PUCG communication modes (Figure 4).

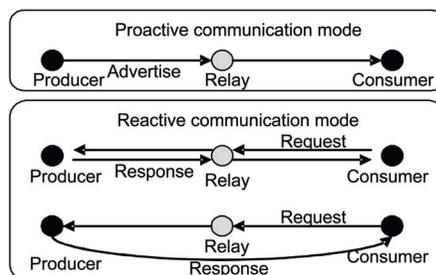


Figure 4 PUCG Communication Model

Proactive communication mode: This mode represents the unsolicited transmission of information that does not require any specific response. It is generally used by a PUCG node to notify other PUCG nodes of its own existence or resources it holds.

Reactive communication mode: This mode represents the transmission of information that requires a response. It is generally used by a PUCG node which requests certain services or resources provided by other PUCG nodes.

As shown in Figure 5, the PUCG architecture also supports basic communication types such as unicast (includes multi-hop unicast), broadcast, and multicast.

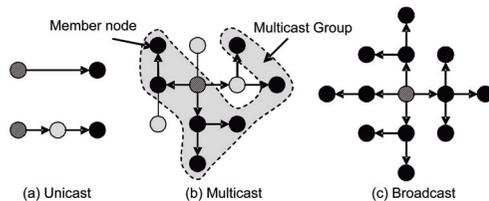


Figure 5 PUCG Communication Types

In PUCG networks, each PUCG node should have a unique name. This name will be extensively used for various purposes such as node search, routing information to a node and cache table management. PUCG node names should satisfy the following requirements.

Uniqueness: PUCG node names should be unique within the scope of a community.

Manageability, Simplicity: PUCG node names should be able to be autonomously generated in particular for pure ad hoc PUCG networks. Its naming structure should use simple for manageability and generality.

Scalability: The PUCG naming system should support extremely a large number of nodes.

Anonymity, Privacy and security: Nobody should be able to infer any type of private information from PUCG node names. For Security, PUCG Node names should not be arrogated.

Independency: PUCG Node names should be independent of location, user, transport protocol, application, and so on.

Considering the above requirements, we have defined PUCG node names that are based on UUID

[12] where a node name is assigned by the node itself. UUID is, however, not human-readable, and an alias name system for human-readability may be required.

Message routing is one of the key mechanisms for realizing efficient and reliable peer-to-peer/overlay communication. In the traditional Internet, routing is performed by a router according to a routing table it holds. However, since a PUCG node freely enters and leaves a PUCG network, the topology of a PUCG network changes very frequently. Routing based on a stable routing table is hence inadequate and inefficient for a PUCG network. Since PUCG nodes communicate with each other across heterogeneous network environments, a transport layer independent routing mechanism is required. In the PUCG architecture, name-based routing mechanism is adopted. This is a mechanism that finds a source route toward a destination PUCG node in a heuristic manner. Good examples of this type of routing mechanism are ad hoc network routing mechanisms such as Dynamic Source Routing (DSR [13]). One advantage of this method is its simplicity because it does not need complicated routing protocols and a particular server. Obviously, it consumes a lot of network resources and incurs long delay. Hence, performance and efficiency could not be sufficiently ensured in large PUCG network environments.

3.2 PUCG Protocols

The PUCG protocols are designed to realize the PUCG architecture. The PUCG protocols described herein are designed, considering the following requirements.

Extensibility: PUCG protocols should be layered, generic, and have extensibility so that they can support various peer-to-peer applications.

Utilization of existing technology: PUCG protocols should leverage existing technologies such as XML, and support existing network infrastructures such as the Internet to simplify implementation and deployment.

Independence of transport protocols: PUCG protocols should be independent of transport protocols for realizing peer-to-peer applications over heterogeneous network environments (e.g. the Internet,

home networks and ad hoc networks).

As shown in Figure 6, the PUCG protocols are currently defined over TCP/IP and Non-IP Protocols such as Bluetooth [14], IEEE 1394 [15] and OBEX/IrDA [16]. PUCG protocols are defined as overlay networking protocols in the application layer. PUCG protocols hence allow communication among home devices, HGWs, mobile phones and cloud servers across heterogeneous network.

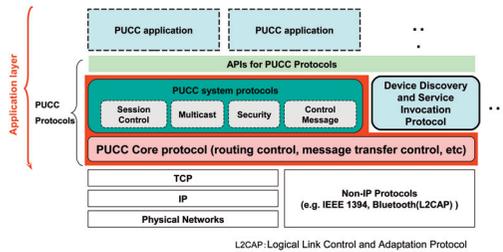


Figure 6 Architecture for PUCG Protocols

PUCG protocols are application-independent and general-purpose protocols for various PUCG applications, which include home devices control from a mobile phone, multicast video streaming, sensor network applications and so on. The APIs for PUCG protocols are defined for developing various PUCG applications. The PUCG protocols stack comprises the two layers of the PUCG core protocol and the other PUCG protocols over the PUCG core protocol as shown in Figure 6. The PUCG protocols are briefly described below.

(1) PUCG Core Protocol

This Protocol processes PUCG messages according to PUCG communication model. Three message types are defined in Section 3.1. Request and response messages are defined for the reactive communication mode while an advertise message is defined for the proactive communication mode. A message is sent to the destination node either directly or using multi-hop unicast. A node sends a broadcast message to all adjacent nodes. The message routing mechanisms of PUCG core protocol are independent of underlying transport protocols. Other PUCG protocols are defined over this protocol.

(2) PUCG System Protocols

PUCG system protocols define common protocols for various PUCG applications, which include

PUCG basic communication protocol, PUCG multicast communication protocol and PUCG control message protocol.

PUCG basic communication protocol establishes and releases PUCG sessions. In the PUCG architecture, all communications are based on a PUCG session between pair adjacent PUCG nodes. This protocol also has the function of exchanging node resource information such as names of its adjacent PUCG nodes, and security functions such as encryption and authentication.

PUCG multicast communication protocol constructs a multicast distribution tree for multicast member nodes and forwards multicast messages along it. A node finds a member node of a multicast group, and sends a Join message to it, and a multicast routing table is generated at each node along the path toward it at the same time. When a node wants to send a multicast message, it sends the message towards the adjacent member nodes based on the multicast routing table. The multicast messages are forwarded along the multicast distribution tree using the bi-directional shared tree mechanism. When the node leaves the multicast group, it sends a Leave message to the adjacent member nodes. This protocol is mainly used by multicast streaming applications.

PUCG control message protocol provides ancillary functions such as notification of message forwarding error, keep-alive for PUCG sessions, and first PUCG node discovery in PUCG network environments. For example, an ErrorReport message is used to notify the source node of the forwarding error of a message. A Diagnose message is used to measure RTT (Round Trip Time) between PUCG nodes. A Lookfor message is used to find the first peer node to which a node should connect in PUCG network environments.

(3) PUCG application protocols

In addition to PUCG system protocols, PUCG application protocols can be defined over PUCG core protocol for various PUCG applications. For smart home applications, PUCG device discovery and service invocation protocol is the most important PUCG application protocol. This protocol is described below in more details.

To control and manage home devices that are connected to a home network, it is first necessary to discover each home device, its capabilities, and the services offered by it. The overview of this protocol is shown in Figure 7. We define PUCG general-purpose device metadata template. The device metadata for each device is defined, using this device metadata template. Such devices include home appliances, healthcare devices, surveillance camera, sensor devices and so on.

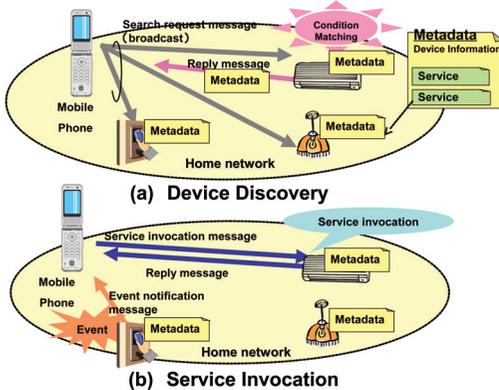


Figure 7 PUCG Device Discovery and Service Invocation Protocol

This protocol uses PUCG device metadata defined for various devices to discover and control devices and the services they offer, as shown in Figure 7(a). Each device on a home network has metadata that describes its capabilities and the services it offers. A user searches for home devices by broadcasting a PUCG search request message over the PUCG overlay network, which contains device functions, device types or key words as search conditions. Each device that receives the PUCG search request message compares its own device metadata with the search conditions and returns a reply message that contains the device metadata if the description included in the device metadata matches the search conditions, as shown in Figure 7(b). The user selects the service to be invoked from the services described in the received device metadata and then sends the PUCG service invocation message, to control the device. Since a mobile phone automatically creates a graphical user interface for controlling home devices, based on the description included in the device metadata, a user easily controls a home device, by using the GUI generated on

the display of the mobile phone.

In addition, the PUCG device discovery and service invocation protocol defines the subscription and notification functions. For example, a cloud server subscribes to a home device such as healthcare device, using the PUCG subscription message. If an event (e.g. measuring healthcare data from healthcare device or sensor data from a sensor device) occurs, the device notifies the cloud server of the event, using the PUCG notification message. Event conditions (e.g. notification of an event once every hour) can be also described in the metadata of a device. The PUCG subscription and notifications functions are applied to communications between home devices connected to a home network and a cloud server, via a HGW. Those functions may be applied to various PUCG applications, including healthcare applications and Home Energy Management System (HEMS).

Figure 8 shows a basic sequence when a PUCG node participates in a PUCG network and exchange resource information with a first-peer node in the PUCG network. At first, Node A sends a Lookfor message using network specific broadcast or multicast mechanisms (e.g. IP multicast) and receives the corresponding LookforResponse messages from certain nodes. Then, Node A sends a Hello message to one of discovered PUCG nodes (in this example, Node B) to participate in a PUCG network. When a HelloResponse Message from Node B is received, a PUCG session is established between Node A and Node B. Resource information are exchanged using Resource Information Advertisement messages in the next step.

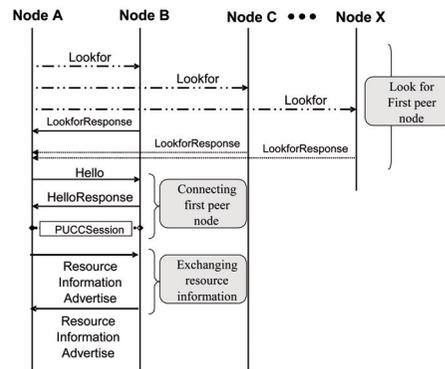


Figure 8 Example of PUCG Protocol Sequence

3.3 PUCG Device Metadata

Structure of the PUCG device metadata is shown in Figure 9. The PUCG device metadata describes information on the device including device name, type, device attributes and the services the device offers, using XML format.

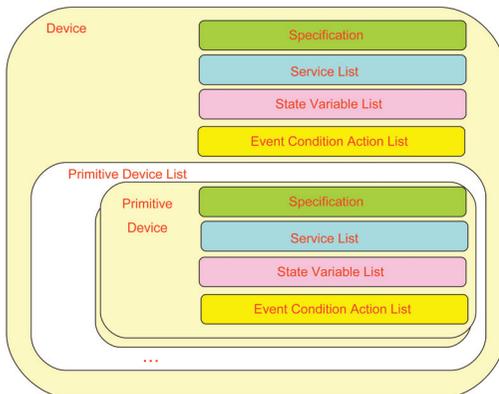


Figure 9 Structure of PUCG Device Metadata

Static information on a device is described in the “specification” field, which includes device name, device manufacturer, model number of the device, serial number of the device and Universal Product Code. The various functions provided by the device are described as a list of services in the “Service List” field, which include function name, input parameters and output parameters. When defining a service, it is possible to specify the service as a series of functions. For example, a “view DVD” service could be specified as the following series of functions. Each service has a unique Uniform Resource Identifier (URI), and the URI is used as an identifier for service discovery and service invocation.

- (1) Turn on a TV
- (2) Set input of the TV to “video”
- (3) Turn on a DVD player.

The “State Variable List” field is used to define a list of state variables. Each state variable holds a value that a device measures. For example, a state variable named “temperature” is used as a state variable in a thermometer device metadata, to hold the value of temperature that the thermometer measures.

The “Event Condition Action List” field can be

used, to define a list of Event-Condition-Action (ECA) rules among devices. An ECA rule has three parts: an event, a condition, and an action. The semantics of an ECA rule are: when the event is detected, the condition is evaluated, and if the condition is satisfied, the action will be executed. For example, using this field, the following ECA rules can be defined.

- When a thermometer shows over 28 degrees Celsius, turn on an air-conditioner.
- When a thermometer shows below 24 degrees Celsius, turn off an air-conditioner.

The “Primitive Device List” field is used for defining individual components in a multi-function device. For example, Blu-ray recorder device may be defined as a multi-function device. In this case, metadata for Blu-ray player, Blu-ray recorder and HDD recorder are defined in each “Primitive Device” field included in the “Primitive Device List” field.

The PUCG device metadata is compatible with UPnP metadata to ensure interoperability with UPnP-compliant devices, and has been designed as general-purpose device metadata, by enabling it to be applied to metadata description for other devices including sensor devices and healthcare devices, and adding functions for describing ECA rules.

To interwork with home devices which are compliant with existing standards on home devices (e.g. UPnP/DLNA and ECHONET), PUCG device metadata are defined to be compatible with such standards. When controlling such devices, a home gateway converts the PUCG device discovery and service invocation protocol into existing standard protocols for controlling home devices.

Figure 10 shows framework for PUCG device metadata description using XML.

```

<Device type="URI" id=" global unique ID " name="user-friendly title">
  <Specification>
    <Manufacturer>manufacturer name</Manufacturer>
    <ManufacturerURL>URL to manufacturer site</ManufacturerURL>
    <ModelDescription>user-friendly title</ModelDescription>
    <ModelName>model name</ModelName>
    .....
  </Specification>
  <StateVariableList>
    <StateVariable name="user-friendly title" datatype="data type" sendEvents="yes or no">
      .....
    </StateVariable>
    .....
  </StateVariableList>
  <ServiceList>
    <Service name="user-friendly title" type="URI" id="global unique ID"/>
    .....
  </ServiceList>
</Device>

```

Figure 10 Framework of PUCG Device Metadata Description

4. Implementation

We have developed the PUCG middleware platform, based on the PUCG specifications described.

The requirements on PUCG middleware platform are as follows.

- (1) PUCG middleware platform should support a wide range of OS environments.
- (2) PUCG middleware platform should run over a wide range of devices
- (3) PUCG middleware platform should provide general-purpose APIs for software developers who will implement various PUCG applications.

Considering above requirements, we have developed the PUCG middleware platform over Java 2 SE and Java 2 ME environments. As shown in Figure 11, Main components of the PUCG middleware platform are PUCG protocols module, PUCG node manager and Transport Adapter. PUCG protocols module implement PUCG protocols including PUCG core protocol, PUCG systems protocols and PUCG application protocols. The role of PUCG node manager is the management of PUCG nodes (e.g. invocations and termination of PUCG nodes) on a device. The role of PUCG transport adapter is to map PUCG protocols onto various underplaying transport protocols (e.g. TCP/IP, Bluetooth, IEEE 1394 and OBEX/IrDA).

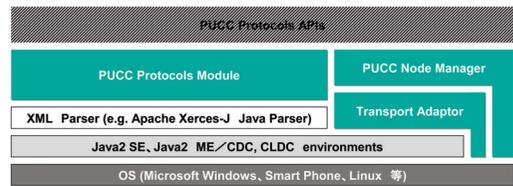


Figure 11 PUCG Middleware Platform

Since the PUCG middleware platform run over Java 2 SE environments, it works on various operating systems including Microsoft windows, Unix/Linux and Apple Mac OS. Since the PUCG middleware platform run over Java 2 ME environments, it works on various small and embedded devices including mobile phones. For example, we have implemented the PUCG middleware platform as a Java application called i-Appli, on NTT DoCoMo’s feature phones [17]. i-Appli can be developed over Java 2 ME CLDC environments, which are supported by NTT DoCoMo’s feature phones.

We have also converted the PUCG middleware platform into Java format called “OSGi bundle” specified in OSGi Alliance. The PUCG middleware platform then runs over OSGi-compliant HGWs. We have converted the PUCG middleware platform into Android software, which runs over Dalvik Virtual Machine (VM), provided by Google. The PUCG middleware platform then runs over Android devices including smartphones and tablets.

The PUCG middleware now runs over a wide-range of devices including PCs, HGWs, mobile phones, tablet devices and cloud servers.

We have also developed Java APIs for PUCG application developers. Java APIs roughly consist of PUCG protocols APIs and PUCG management APIs. PUCG protocols APIs provide PUCG application developers with functions for using PUCG protocols including PUCG core protocol, PUCG system protocols and PUCG application protocols (e.g. PUCG device discovery and service invocation protocol). PUCG application developers also design their own PUCG application protocols, using APIs for PUCG core Protocol). PUCG management APIs provide PUCG application developers with functions management functions such as invocation and termination of PUCG nodes on a device, and error handling.

5. Smart Home Applications using the Pucc Middleware Platform

To verify the feasibility and usefulness of Pucc technologies, Pucc have developed prototype systems implementing Pucc middleware platforms for mobile phones, PCs, mobile phones, tablet devices, HGWs, cloud servers, and so on. Pucc have also developed Pucc applications such as multimedia content search from mobile phones, home appliances control from mobile phones, healthcare applications, AV devices (e.g. HD/DVD recorders) control from mobile phones and home security applications. Pucc conducted several demonstrations on Pucc technologies at world-famous exhibitions such as CES 2008, CES 2009, MWC 2009 and CeBIT 2009 [7]. Among them, we described several smart home applications using Pucc middleware platforms. One of typical smart home applications is described below

5.1 Home Appliances Control from Smart Devices

We have also developed the home appliances (e.g. air-conditioner) control system from smart devices (e.g. smart phone and tablet device)[18].

Much attention has been paid to ECHONET Lite specifications as standards for smart home applications in Japan. The main purpose of ECHONET lite is to apply ECHONET specifications to Home Energy Management System (HEMS). Some ECHONET lite-compliant commercial products are now available in Japan. In this system, we use ECHONET lite-compliant air-conditioners and lamps. OSGi-compliant commercial HGWs are also available. Using those devices, we have developed ECHONET lite-compliant home appliances from smart devices. Since we have already implemented the Pucc middleware platform on Android devices, we use Android smart phones and Android tablets in this system.

An overview of this system is shown in Figure 12. This system consists of smart devices, HGW and ECHONET lite-compliant air-conditioners and lamps. This system works as follows.

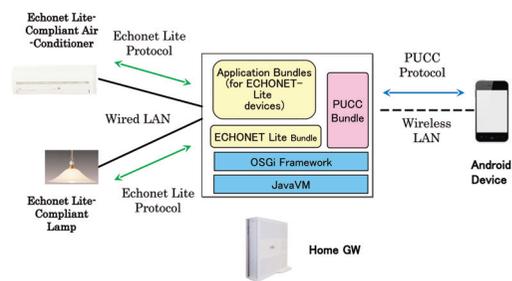


Figure 12 Overview of Home Appliances Control System

- (a) OSGi bundles including the Pucc bundle can be downloaded from a management center, if necessary.
- (b) A smart device (i.e. Android smart phone or Android tablet) looks for Pucc devices using A Look-for message, and finds a HGW. The Pucc session is then established between the smart device and the HGW, using Hello messages. The smart device sends a Discover request message to the HGW, to search ECHONET Lite-compliant devices.
- (c) When the HGW is received the Discover request message, the application bundle for ECHONET-Lite devices converts it to the corresponding ECHONET Lite protocol using the ECHONET Lite bundle, to search ECHONET Lite-compliant devices. When ECHONET Lite-compliant devices are discovered (in this case, air-conditioners and lamps), the HGW sends a Discover reply message to the smart device, which contains ECHONET Lite device metadata for air-conditioners and lamps.
- (d) When the smart device receives the Discover reply message, it generates the GUI for controlling ECHONET Lite-compliant devices, using ECHONET Lite device metadata contained in it. When a user want to control ECHONET Lite-compliant devices, the user can controls them using the GUI on the smart device. For example, when the user wants to turn on the air-conditioner, the smart device sends an Invoke request message, which contain the service request (i.e. turn on the air-conditioner) to the HGW. When the HGW receives it, the application bundle for ECHONET-Lite devices converts it to the corresponding ECHONET Lite protocol using the ECHONET Lite bundle, to turn on the air-conditioner. As a result, the air-conditioner is turned on.

6. Conclusion

We first describe the current trend in standardization on home networks and devices. We then describe issues with such standardization activities from the mobile phone and cloud server point of view. To resolve such issues, we have developed PUCC overlay networking protocols and devices metadata technologies as a solution for unified control and management of various home devices connected to home networks, from a mobile phone and a cloud server.

We have developed the PUCC middleware platform, based on the PUCC specifications. Since the PUCC middleware platform run over Java 2 SE environments, it works on various operating systems including Microsoft windows, Unix/Linux and Apple Mac OS. The PUCC middleware now runs over a wide-range of devices including PCs, HGWs, mobile phones, tablet devices and cloud servers. To verify the feasibility and usefulness of PUCC technologies, PUCC developed several PUCC applications including smart home applications.

Future work will include further developments of PUCC application such as HEMS, applications for smart grid, healthcare applications, distributed autonomous control of home devices and M2M applications towards realizing smart home and smart city, integration of PUCC technologies with social networking services (e.g. Facebook) and further efforts toward International Standardization.

ACKNOWLEDGEMENT

This work was supported by NTT DOCOMO, Ericsson, Kyoto University, Keio University and PUCC.

This work is partially supported by Special Research Program of Komazawa University, 2013 and 2014.

Reference

- [1] UPnP Forum, <http://www.upnp.org/>
- [2] DLNA, <http://www.dlna.org/>

- [3] ECHONET CONSORTIUM, <http://www.echonet.gr.jp/english/index.htm>
- [4] OSGi Alliance, <http://www.osgi.org/Main/HomePage>
- [5] Continua Health Alliance, <http://www.continuaalliance.org/index.html>
- [6] ISO TC 215 Health informatics, http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54960
- [7] N. Ishikawa, PUCC Activities on Overlay Networking Protocols and Metadata for Controlling and Managing Home Networks and Appliances, Proceedings of THE IEEE, Vol. 101, No.11, pp.2355-2366, 2013
- [8] N. Ishikawa, T. Kato, H. Sumino, S. Murakami and J. Hjelm, PUCC Architecture, Protocols and Applications, 4th IEEE Consumer Communications and Networking Conference (CCNC 2007), 2007
- [9] The Gnutella protocol specification v0.4, http://web.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf
- [10] J. Liang, R. Kumar and K. W. Ross, Understanding KaZaA, <http://infosec.pku.edu.cn/~p2p/slides/%5BUnknown%5D%20Understanding%20KaZaA.pdf>
- [11] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, Freenet: A distributed anonymous information storage and retrieval system, In Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, California, 2000.
- [12] P. Leach, M. Mealling and R. Salz, A Universally Unique Identifier (UUID) URN Namespace, RFC 4122, 2005
- [13] D. Johnson, Y. Hu and D. Malz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728, 2007
- [14] Bluetooth, <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>
- [15] IEEE 1394, http://standards.ieee.org/develop/wg/1394_WG.html
- [16] Infrared Data Association, <http://www.irda.org/>
- [17] H. Sumino, N. Ishikawa and T. Kato, Design and implementation of P2P protocol for Mobile Phones, 3rd IEEE Workshop on Mobile Peer-to-Peer Com-

puting (MP2P06), IEEE PerCom 2006, 2006

- [18] S. Saito, N. Ishikawa and Y. Tsuchiya: Development of ECHONET Lite-Compliant Home Appliances Control System Using PUCP Protocols from Smart Device, 3rd IEEE International Workshop on Consumer Devices and Systems (CDS 2015), 2015