A Service-Oriented Framework for Networked Appliances to Achieve Appliance Interoperability and Evolution in Home Network System

Hiroshi Igaki, Masahide Nakamura and Ken-ichi Matsumoto
Nara Institute of Science and Technology
8916-5,Takayama-cho, Ikoma Nara 630-0192, Japan
{hiro-iga, masa-n, matumoto}@is.naist.jp

Abstract

In the next-generation home network systems(HNS), interoperability among multi-vendor appliances is a challenging issue to implement value-added integrated services. This paper presents a service-oriented framework to enable both evolution of HNS and the appliance interoperability. The key idea is to construct the integrated services by combining the existing services deployed by the appliances. This allows to eliminate the reference model, which had been an obstacle of the evolution of the HNS.

Keywords: *service-oriented architecture, interoperability, evolution, home network system, integrated services*

1. Introduction

The recent ubiquitous technologies allows home electric appliances, such as TVs, lights, DVD players, refrigerators, to be connected with a local area network at home.The next step for such home network system is to *integrate* different appliances from multiple vendors flexibly, in order to provide more sophisticated value-added services for home users. We call such the next-generation services as *HNS integrated services* in this paper.

In order for the multi-vendor appliances to communicate with each other, a common network protocol is necessary for the HNS. For this, several protocols (e.g., DLNA [2], ECHONET [3]) are being standardized. Once a common protocol is given, the vendor of each appliance has to achieve two issues: *protocol conformance* and *appliance interoperability*. The protocol conformance is that each appliance must conform to the given HNS protocol. On the other hand, the appliance interoperability is that multiple appliances conforming to the same protocol must operate together to achieve the HNS integrated services.

In general, checking the protocol conformance of an appliance is not very hard.However, validating the appliance interoperability is much more difficult, since it requires the actual integration of appliances under the collaboration of multiple vendors. The conventional approach to achieve the appliance interoperability is that the alliance of the vendors determines a *reference model* with a rigorous specification, for each kind of appliances [4]. The rigorous specification minimizes vendor-specific implementation. Thus, each vendor can perform the interoperability testing in isolation, by checking if the developed appliance can work with the reference model. However, this approach lacks flexibility in changing the reference model, which significantly limits the evolution of home network systems. Specifically, the reference model may prevent appliance vendors from implementing vendor-specific features exceeding the model. Also, the variety of integrated services is also limited within the reference model.

To enable the evolution of HNS with sustaining the appliance interoperability, this paper proposes a new framework extensively utilizing the concept of *service oriented architectures* (SOA) [8]. We first propose a two-layered design of an appliance, where each appliance consists of two layers: device layer and and service layer. The device layer corresponds to the physical device of the networked appliance. On the other hand, the service layer is a software application on top of the former two layers, which exposes the features of the underlying device as *services* based on the SOA principle. For each serice, the service layer exposes a platform-independent and strictly-typed interface, by wrapping proprietary device APIs with a certain SOA framework (e.g., Web Services). Thus, each appliance can conform to a platform-independent protocol, which achieves conformance. We describe the details of our framework for solving the existing interoperability problems in subsequent chapters.

2. Conventional Approach for Appliance Interoperability

Even if multiple appliances achieve the protocol conformance, it does not mean that the appliances can successfully work together to provide the HNS integrated services. Let us consider the following HNS integrated service:

Auto-DVD Service : When a user switches on the DVD player, the TV is turned on in DVD mode, the 5.1ch speakers are selected, and the volume of the speaker is automatically adjusted.

To implement the Auto-DVD service with multi-vendor appliances, the TV, the DVD and the Speaker must conform to a common protocol. In addition, they must achieve the interoperability. This issue is known as *appliance interoperability*. The conventional approach to circumvent this problem is to introduce a *reference model* for each class of appliances [2] [3] with rigorous requirements and specifications. For instance, let us consider a case with the networked TV and the networked DVD player with Figure 1.

Suppose that each of vendors A, B and C wants to develop a networked TV (TV_A, TV_B or TV_C) to achieve Auto-DVD service. While, each of D, E and F tries to develop a networked DVD player (DVD_D, DVD_E or DVD_F). Then, the alliance of the vendors determines a reference model for each class of TV (TV_{ref}) and DVD player (DVD_{ref}). Each vendor develops the product so that it conforms to the reference model. Thus, TV_A, TV_B and TV_C are supposed to conform to TV_{ref}. Therefore, if DVD_D (or DVD_E, DVD_F) is developed so that it can work with TV_{ref}, then DVD_D achieves interoperability with all of TV_A, TV_B and TV_C. Note that this interoperability validation can be performed by D without having TV_A, TV_B and TV_C.

The method of interoperation (e.g., how to turn on the appliance and how to change the inputs, etc.) is strictly predefined among the reference models. Hence, the HNS integrated services are implemented in details according to the reference models. The conventional approach with the reference model currently gives a realistic solution. However, in the near future it will place serious obstacles in *evolution* of HNS, which are summarized as follows:

E1: Evolution of reference model

Since the reference model is tightly coupled with all the appliances, it is quite difficult to change or update the reference model.

E2: Evolution of networked appliances

By the reference model, each appliance has to be strongly aware of how the appliance should be interoperated with other appliances. This fact would significantly limit the vendor-specific features, which prevents the evolution of the appliances.

E3: Evolution of integrated services

Developing integrated services requires vendor's expert knowledge, and customizing the integrated services is limited within the framework of the reference model. Thus, the reference model would limit the possibility of future *programmable services*, where home users integrate arbitrarily appliances.

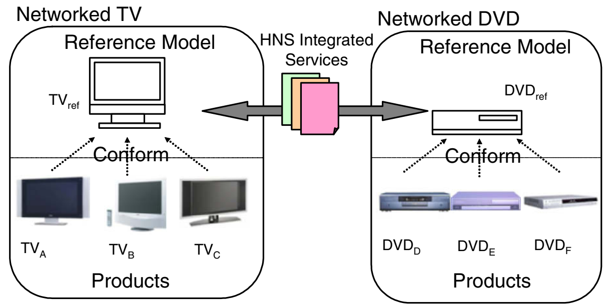

Figure 1. Conventional approach to assure the appliance interoperability

3. Service-Oriented Framework for Home Networked Appliances

3.1 Key Idea

Our key idea is to apply service oriented architecture (SOA [8]) to HNS in order to *eliminate* the reference model, without loosing the appliance interoperability.

SOA is a system architecture to integrate different systems distributed over a network. Each system exports own features to the network as a unit of *service*(a set of tasks, which is coarser than an object). Each service is *self-contained*, that is, a service operates independent of the context or states of other services. The internal logic and implementation are encapsulated in the system. The system exposes only interfaces of the service in form of strictly-typed *exported methods*. To achieve the SOA-based framework in HNS, we assume that each networked home electric appliance satisfies the following conditions.

Condition C1: Each appliance has *device control interfaces* that can be accessed by software (e.g., APIs).

Condition C2: Each appliance has a storage to store *application software* (server and device control application), a processor to execute the application, and a network interface.

Device control interfaces of each appliance change with kinds of the appliances. In our proposal, this differenct is generalized by the application installed in appliance strage area(C2). Self-contained services of the appliances are realized by implementing such applications as the service of SOA. The HNS integrated services are created as a service set of appliances. Such integrated services are not influenced by implementing for every appliance, or the dependency between appliances. As a result, the interoperability in the HNS integrated services improves.

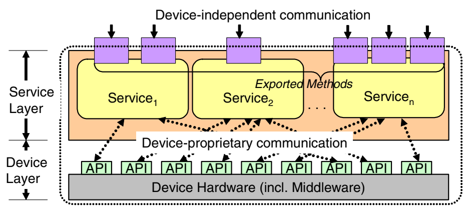

Figure 2. Two-layered design of an appliance

Our originality here is to apply SOA extensively to the home network system, where heterogeneous and multi-vendor appliances (satisfy Condition C1,C2) must collaborate to provide the HNS integrated services.

3.2 Two-layered Design of Appliance

To achieve the key idea, we propose a *two-layered design* for networked appliances. Specifically, we divide each appliance into two layers: a *device layer* and a *service layer*, as shown in Figure 2.

The device layer refers to the hardware portion of an appliance ,while the service layer is implemented as a software application, which is a core of the proposed framework. The service layer aggregates the features of the appliance as a set of *services*, and exports the services to the network with *exported methods*.

As depicted in Figure 2, each service invokes a set of APIs of the underlying device layers, in order to implement a *self-contained* feature of the appliance. The communication between the service and the device layers may require vendor-specific procedures and/or proprietary protocols, which is depicted by dotted arrows in Figure 2. Then, the service layer exhibits the interfaces of the service as a set of exported methods (depicted by boxes on the services). The exported methods are opened to network with a strictly-typed *interface definition*. These methods can be accessed (executed) from external software in a *device-independent manner*, which does not depend on the underlying device implementation of the appliance. For this, we use a generic SOA framework such as Web services (with WSDL and SOAP/XML).What must be achieved in the service layer are that;

Requirement S1: Each service must be self-contained, which does not depend on the context or states of other appliances.

Requirement S2: The service must be executed with device-independent communication using a standard SOA framework such as Web services.

Requirement S3: Once the service is deployed, its interface definition must not be changed.

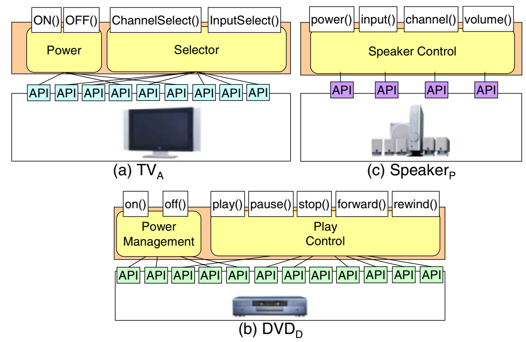

Figure 3. Appliances with the SOA framework

These requirements are quite reasonable in the context of SOA. Figure 3 shows an example design of the networked TV, DVD player, Speaker and Light. In the figure, we assume that $\texttt{TV}_A$, $\texttt{DVD}_D$ and $\texttt{Speaker}_P$ are developed by different vendors A, D, P, respectively, with the proposed two-layered design. The role of each method is as read in the figure. We assume that for each exported method, a usage manual and an interface definition specifying the type of parameters and return value are given by the vendor.

3.3 Service Integration to Achieve HNS Integrated Services

We implement an integrated service by executing exported methods in accordance with a certain order. For example, let us implement Auto-DVD service with $\texttt{TV}_A$, $\texttt{DVD}_D$, and $\texttt{Speaker}_P$ shown in Figure 3. Then, the following two sequences of method invocations can achieve an implementation of these integrated services. In the following, an invocation of an exported method `m()` of an appliance `App` is denoted by `App.m()`. Also, the sequences are called *service scenarios*, denoted by SS_2.

SS_2: **Auto-DVD Service** (with $\texttt{TV}_A$, $\texttt{DVD}_D$, $\texttt{Speaker}_P$)

2.1 $\texttt{DVD}_D$`.on();` `/*`$\texttt{DVD}_D$ `is turned on*/`
2.2 $\texttt{TV}_A$`.ON();` `/*`$\texttt{TV}_A$ `is turned on */`
2.3 $\texttt{TV}_A$`.InputSelect(DVD);` `/*Input is set to DVD*/`
2.4 $\texttt{Speaker}_P$`.power(ON);` `/*`$\texttt{Speaker}_P$ `is turned on*/`
2.5 $\texttt{Speaker}_P$`.input(DVD);` `/*Input is set to DVD*/`
2.6 $\texttt{Speaker}_P$`.channel(5.1);` `/*5.1ch is selected */`
2.7 $\texttt{Speaker}_P$`.volume(80);` `/*Sound is set to 80db */`
2.8 $\texttt{DVD}_D$`.play();` `/*`$\texttt{DVD}_D$ `plays contents*/`

To execute the exported methods over different appliances, we need a certain *service integration mechanism*. For this, two different methods in SOA can be used.

The first method is to use a standard service integration framework in SOA, such as BPEL4WS [1](as shown in Figure 4(a)). The framework specifies an execution order and logic among distributed services, and integrate and execute them as a *business workflow*[6].

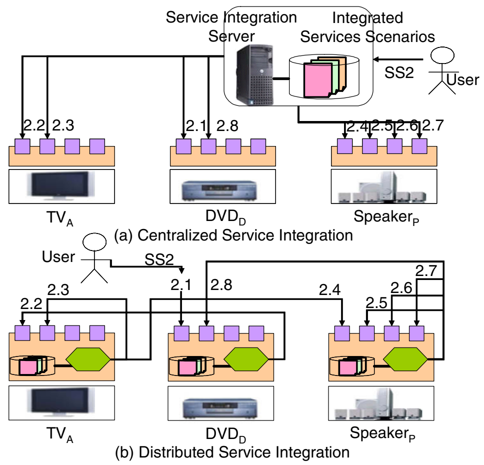

Figure 4. Service integration mechanisms

The second method is to delegate the service integration to the appliance themselves, so that the integrated services are executed in an autonomous and distributed way (as shown in Figure 4(b)). In our previous work [7], we put an application module (depicted by an octagon in Figure 4) into the service layer of each appliance. The module dynamically selects and triggers an appropriate exported method, according to a given service scenario.

No matter which method we use for the service integration, a HNS integrated service can be implemented as a sequence of invocations of exported methods.

4. Discussion and Conlusion

If our proposed framework is achieved, then it would be unnecessary to take problem E1 (see Section 2) into account. In the proposed framework, each appliance provides its features in a form of *self-contained services* with a standardized interface. Here, the appliance can export *any* services without concerning how the services are used by other appliances.

The vendor of each appliance can freely update and modify the implementation of the appliance, as long as the vendor does not change the interface definition. This would help to cope with problem E2.

A service designer just chooses a set of exported methods and specifies an execution order and logic. For this, the designer just has to know the interface definition of the exported method, but does not need to concern implementation or dependency among the appliances. With an appropriate assistance to understand the interface definition, even a home user would be able to design own services. This would help to resolve problem E3.

In this paper we have presented a service-oriented framework for networked appliances in HNS. The proposed method extensively used to enable flexible evolution of HNS with sustaining interoperability. We are currently developing tools to support *non-expert users* to construct integrated services easily. To design integrated services, a service designer has to first choose appropriate services for each appliance, and then understand the service interface definitions. This would impose a tough task for non-expert users.

References

[1] Business Process Execution Language for Web Services, Version 1.1: http://www-106.ibm.com/developerworks/library/ws-bpel/ .

[2] Digital Living Network Alliance - `http://www.dlna.org`

[3] ECHONET Consortium - `http://www.echonet.gr.jp/`

[4] Intel Corporation, "Framework for Designing Interoperable Digital Home Devices" - `http://www.intel.com/technology/comms/cn11031.htm`

[5] T. Kimura, H. Tamada, H. Igaki, M. Nakamura and K. Matsumoto, "A visual framework for monitoring and controlling distributed service components", *Proc. 1st Korea-Japan Joint Workshop on Ubiquitous Computing and Networking Systems (ubiCNS 2005)*, Jun. 2005. (to Appear)

[6] S. W. Loke, "Service-oriented device ecology workflows", *Proc. of 1st Int'l Conf. on Service-Oriented Computing (ICSOC2003)*, LNCS2910, pp.559-574, Dec. 2003.

[7] M. Nakamura, H. Igaki, H. Tamada and K. Matsumoto, "Implementing integrated services of networked home appliances using service oriented architecture", *Proc. of 2nd International Conference on Service Oriented Computing (ICSOC2004)*, pp.269-278, Nov. 2004.

[8] M.P.Papazoglou, D. Georgakopoulos, "Service-Oriented Computing", *In Communications of the ACM* , Vol. 46,No.10, pp.25-28, Oct. 2003.