

## 家電機器間の相互接続性を考慮した連携サービスを実現する サービス指向フレームワークの提案

井垣 宏, 中村 匡秀, 松本 健一

奈良先端科学技術大学院大学 情報科学研究科

〒 630-0192 奈良県 生駒市 高山町 8916 番地の 5

あらまし ホームネットワークにおいて、異なるベンダの家電機器を協調連携させる高度な連携サービスを実現するためには、機器間の相互接続性を保証しつつ、機器かサービスの更新が柔軟に行えることが重要になる。本稿では、ホームネットワークの機器構成にサービス指向アーキテクチャを採用した新たなフレームワークを提案する。このフレームワークでは、全ての機器は機器の物理デバイス層とサービス層で構成される二層構造として設計される。物理デバイス層では、各機器の物理的な機能が提供され、サービス層ではその機能への汎用的なインタフェースが公開される。連携サービスは各家電のサービスを組み合わせることを実現する。このようにして提案フレームワークにもとづいて機器間の連携を行うことで、ベンダや機器種類に依存せず、相互接続可能なホームネットワークシステムを実現することが可能となる。

キーワード サービス指向アーキテクチャ, ホームネットワークシステム, 連携サービス, 相互接続性

## Service-Oriented Framework Achieving Interoperability between Networked Appliances in HNS Integrated Services

Hiroshi Igaki, Masahide Nakamura, and Ken-ichi Matsumoto

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama, Ikoma-Shi, Nara 630-0192

**Abstract** With integrated services which make two or more appliances orchestrate, an appliance framework in consideration of an interoperability problem between multi-vendor appliances should be needed. To achieve the interoperability, this paper presents a service-oriented framework for multi-vendor home network systems(HNS). We propose a two-layered design of an appliance, consisting of device and service layers. The device layer corresponds to the physical device of the appliance controlled by vendor-specific interfaces. The service layer exhibits features of the device as self-contained services accessible via device-independent interfaces. In our proposed framework, every integrated services are realized by these appliances appliances which adopt the design. Such self-contained appliances enable high interoperability between them.

**key words** service-oriented architecture, interoperability, evolution, home network system, integrated services

### 1. Introduction

Several companies have already released commercial networked appliances and *home network system*(HNS) services (e.g., [11, 6]), such as remote control of the appliance outside home and group control of lights. Although networking home appliances could add value to our daily life, the current HNS services are not so sophisticated that are not far from the conventional remote control. Also, a HNS service is usually limited within a single or small number of single-vendor

appliances.

The next step for the industries is to integrate and orchestrate appliances from multiple vendors flexibly, in order to provide more sophisticated value-added services for home users. We call such the next-generation HNS service as *HNS integrated services* in this paper. For instance, integrating a TV, a DVD player, speakers, lights and a blind would implement a HNS integrated service, say, *DVD theater service*. When a user turns on the DVD player, the lights become dark, the blind is closed, and the 5.1ch speakers are selected while the

volume is automatically adjusted.

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, since the conformance testing can be performed within a (single) vendor of the appliance. 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 [7]. 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 for HNS appliances. The proposed framework extensively utilizes the concept of *service oriented architectures* (SOA) [13].

In order to apply SOA to the networked appliance, we first propose a two-layered design of an appliance, where each appliance consists of two layers: device layer 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 is a core of the proposed framework.

The service layer exposes the features of the underlying device as *services*, based on the SOA principle. Specifically, the service layer aggregates the features of the appliance as self-contained services that are independent of other appliances. Also, for each service, 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 [4, 15]). Thus, each appliance can conform to a platform(i.e., implementation)-independent protocol, which achieves conformance.

Next, we construct HNS integrated services by combining the (existing) services of appliances, specifically executing exported methods in a certain order. The execution of the integrated service is performed with an external service integration mechanism. Thus, scenarios of the integrated services are separated from the appliances.

## 2. Preliminaries

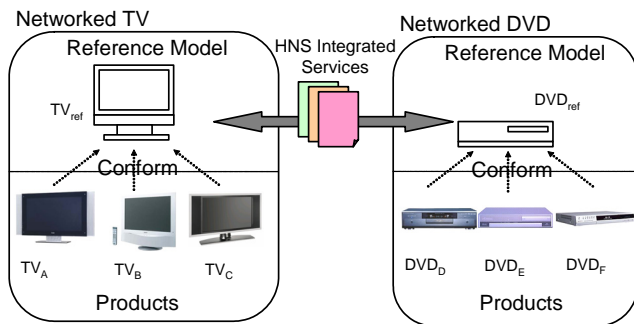
### 2.1 Networked Appliances and Integrated Services

A home network system (HNS) consists of one or more *networked appliances*, which are connected to a local area network. In general, each networked appliance has *device control interfaces* (i.e., APIs) by which users or external software agents can control the appliance via network. Thus, the appliance is supposed to own a processor, a storage (to store device applications or middleware), and a network interface to handle the API calls.

Controlling only a single networked appliance does not offer much added value compared to the traditional appliances [9]. The main advantage of the HNS lays in *integrating* the control of multiple appliances together. This yields value-added and more powerful services, which we call *HNS integrated services* in this paper.

The communication among networked appliances is performed by an underlying protocol. Assuming that multiple vendors participates in developing networked appliances, a number of protocols for the appliances have been proposed and standardized. Typical protocols include X-10 [16], HAVi [5], Jini [8], UPnP [14], ECHONET [3] and DLNA [2], although their purposes and operating layers may vary. In order for two or more appliances to collaborate with each other, basically the appliances have to conform to the same protocol, which we call *protocol conformance*.

Checking the protocol conformance is not very hard, since it can be performed by individual vendor of the appliance, independently. Tools for the conformance testing could be given. For instance, DLNA provides CTT [7](Conformance Test Tool) for digital audio/video appliances.



**Figure 1. Conventional approach to assure the appliance interoperability**

## 2.2 Example of HNS Integrated Services

For a more comprehensive discussion, here we introduce a simple example of HNS integrated services. In the example, we suppose a HNS consisting of three appliances (a DVD player, a TV, a speaker). We prepare the following two *service scenarios* of the HNS integrated services.

**Auto-TV Service :** When the user turns on the TV, the speaker's channel is set to 2ch, and the volume of the speaker is automatically adjusted for the TV mode.

**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.

## 2.3 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. Major reason is that the protocol conformance is validated by each vendor in isolation, which does not cover the combined behavior of multiple appliances. This issue is known as *appliance interoperability*.

A straight-forward way for validating the interoperability is to run and test the HNS integrated services thoroughly on the actual appliance implementations. However, it is difficult for the vendors to predict exactly which appliances are combined in a HNS by the individual home users. Worse in the multi-vendor environment, the number of possible combinations grows in exponential as the number of appliances. Thus, achieving *complete* interoperability is a quite difficult problem.

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 (see Section 2.2). 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 pre-defined among the reference models. Hence, the HNS integrated services are implemented in details according to the reference models.

## 2.4 Problem on Conventional Approach

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. Any changes in the reference model force the vendors to update all the appliance products, which would corrupt the interoperability among the new and the existing appliances.

### E2: Evolution of networked appliances

The reference model rigorously prescribes over the multiple vendors how the appliances should behave. Each appliance has to be strongly aware of how the appliance should be interoperated with other appliances. Also, all appliances conforming to the same model are regarded to be *equivalent*. These facts would significantly limit the vendor-specific features, which prevents the evolution of the appliances.

### E3: Evolution of integrated services

The HNS integrated services are implemented in a *ready-made* form by appliance vendors or service providers, according to the reference model. Hence, 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 as they like and create their own integrated services.

### 3. Service-Oriented Framework of Home Networked Appliances

To achieve the appliance interoperability in a more flexible way, we introduce the concept of the *service oriented architecture*[13] in the HNS.

#### 3.1 Key Idea

Our key idea is to apply service oriented architecture (SOA [13]) to HNS in order to *eliminate* the reference model, without losing 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.

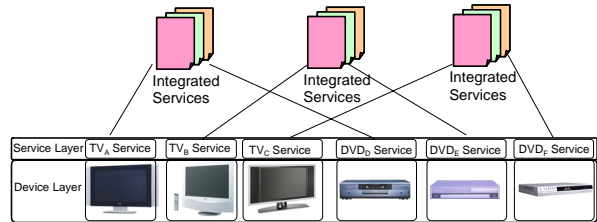
**Condition C1:** Each appliance has *device control interfaces* that can be accessed by software (e.g., APIs).

**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 difference is generalized by the application installed in appliance storage 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.

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.

Figure 2 represents an overview of the proposed framework, involving the same set of appliances shown in Figure 1. As will be explained in the next subsection, each appliance has two layers: service layer and device layer. HNS integrated services are created for each combination of appliances, by combining services provided by individual appliances.



**Figure 2. Overview of the proposed framework**

#### 3.2 Two-layered Design of Appliance

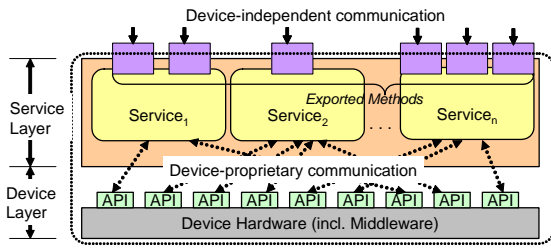
To achieve the basic principle, we propose a *two-layered design* for networked appliances. Specifically, we divide each appliance into two layers: a *device layer* and a *service layer*, as shown in Figure 3.

The device layer refers to the hardware portion of an appliance. According to Condition C1, each appliance has a set of APIs, by which a software application can control the appliance. The device layer can also involve a middleware to drive the APIs.

On the other hand, the service layer is implemented as a software application installed in the storage of the appliance (see Condition C2), 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 3, 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 3. 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).

Our framework does not force the appliance vendors to adopt any reference model. Moreover, for a given



**Figure 3. Two-layered design of an appliance**

appliance, how to implement the service layer is completely up to the vendor. Also, it is possible for the vendor to modify or update the device and the service layers, as long as the interface definition of the service is not changed. 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.

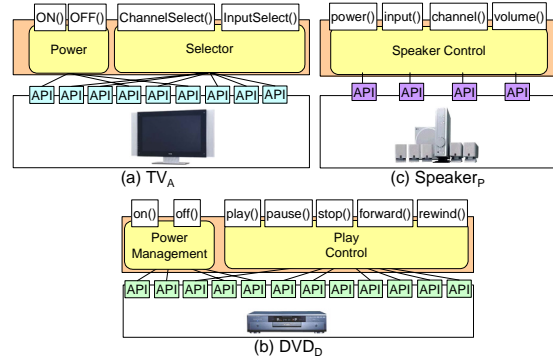
**Requirement S3:** Once the service is deployed, its interface definition must not be changed.

These requirements are quite reasonable in the context of SOA.

Figure 4 shows an example design of the networked TV, DVD player, Speaker and Light. In the figure, we assume that  $TV_A$ ,  $DVD_D$  and  $Speaker_P$  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  $TV_A$ ,  $DVD_D$ , and  $Speaker_P$  shown in Figure 4. 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  $SS_2$ .



**Figure 4. Appliances with the SOA framework**

$SS_2$ : **Auto-DVD Service** (with  $TV_A$ ,  $DVD_D$ ,  $Speaker_P$ )

- 2.1  $DVD_D.on()$ ; /\* $DVD_D$  is turned on\*/
- 2.2  $TV_A.ON()$ ; /\* $TV_A$  is turned on \*/
- 2.3  $TV_A.InputSelect(DVD)$ ; /\*Input is set to DVD\*/
- 2.4  $Speaker_P.power(ON)$ ; /\* $Speaker_P$  is turned on\*/
- 2.5  $Speaker_P.input(DVD)$ ; /\*Input is set to DVD\*/
- 2.6  $Speaker_P.channel(5.1)$ ; /\*5.1ch is selected \*/
- 2.7  $Speaker_P.volume(80)$ ; /\*Sound is set to 80db \*/
- 2.8  $DVD_D.play()$ ; /\* $DVD_D$  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.

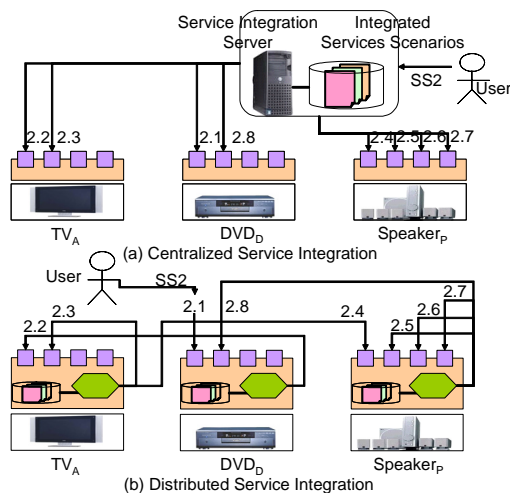
The first method is to use a standard service integration framework in SOA, such as BPEL4WS [1](as shown in Figure 5(a)). The framework specifies an execution order and logic among distributed services, and integrate and execute them as a *business workflow*[10].

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 5(b)). In our previous work [12], we put an application module (depicted by an octagon in Figure 5) 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

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



**Figure 5. Service integration mechanisms**

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.

## 5. Conclusion

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] C. Ethan, "Web Services Essentials – First Edition", *O'Reilly & Associates Inc.*, United States of America, 2002.
- [5] HAVi - <http://www.havi.org/>
- [6] Hitachi Home & Life Solutions inc., "horaso network" - <http://ns.horaso.com/>
- [7] Intel Corporation, "Framework for Designing Interoperable Digital Home Devices" - <http://www.intel.com/technology/comms/cn11031.htm>
- [8] Jini - <http://www.jini.org/>
- [9] M. Kolberg, E. H. Magill, and M. Wilson, "Compatibility issues between services supporting networked appliances", *IEEE Communications Magazine*, vol. 41, no. 11, Nov 2003 pp. 136-147
- [10] 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.
- [11] Matsushita Electric Industrial Co., Ltd., Kurashi net - <http://national.jp/appliance/product/kurashi-net/>
- [12] 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.
- [13] M.P.Papazoglou, D. Georgakopoulos, "Service-Oriented Computing", *In Communications of the ACM*, Vol. 46, No.10, pp.25-28, Oct. 2003.
- [14] UPnP Forum - <http://www.upnp.org/>
- [15] W3C Web Service Activity - <http://www.w3.org/2002/ws/>
- [16] X-10 - <http://www.x10pro.com/>