A Location-Adaptive Virtual Networked Appliance

Akihiko Kusumoto¹, Jin Nakazawa¹, Yoshito Tobe¹, and Hideyuki Tokuda¹ ²

{akihiko, jin, tobe, hxt}@ht.sfc.keio.ac.jp

¹Graduate School of Media and Governance, Keio University,
²Faculty of Environmental Information, Keio University
5322, Endo, Fujisawa, Kanagawa 252, Japan
Phone: +81-466-47-5000 (ext. 54280), Fax: +81-466-47-0624

Abstract

This paper presents a new way of constructing a Virtual Networked Appliance (VNA) using location information. In the VNA architecture which we proposed in our previous work, adaptation to the location of a user has not been clearly specified. We call such a location adaptive VNA “L-VNA.” In L-VNA, location management is encapsulated in the runtime of VNA, thereby eliminating a user’s explicit decision for location adaptation. We have implemented L-VNA over JDK1.2 and demonstrated its validity.

1 Introduction

The last several years have seen an evolution of techniques for networked appliances. Audio/visual appliances are equipped with digital interface such as IEEE 1394 [4]. In addition, even other appliances for our daily life such as an air-conditioner can be connected to a network. The progress in component-based software facilitates the use of such appliances over a network. Software for embedded devices such as light-weight Java enables network-affinity of the devices. Furthermore, plug-and-play technologies such as HAVi[5] and Jini[7] provide middleware-level connectivity as well. However, these techniques do not relieve the appliance users from complications in connecting the appliances.

To overcome the problem, we proposed Virtual Networked Appliances (VNAs) [5]. A VNA provides a user with one unified view of the constructed appliance. The VNA is composed of many logical functions distributed over the network, even over the Internet, beyond physical boundaries of the appliances. The VNA accompanies a set of function tables using an XML-based markup language, thereby enhancing flexibility and portability.

Although a VNA system is flexibly built, it did not explicitly define how a VNA is adapted to the location of a user. When the user of a VNA moves, the VNA should be dynamically re-built depending on the availability of constituent components. Therefore, we introduce a location management part in our VNA runtime. We call such an location-Adaptable VNA “L-VNA.” This paper describes the details of design and implementation of L-VNA.

The remainder of this paper is organized as follows: Section 2 gives discussions on location information in a home network. Section 3 explains the location adaptive VNA. Details of L-VNA are described in section 4, and its application in section 5. Sections 6 and 7 give related work and conclusion, respectively.

2 Location Information in Networked Appliance

Recent consumer appliances including audio/visual devices have rich computing resources and network connectivity [1]. These advantages occurred in consumer appliances made it possible that those appliances, i.e. information appliances, cooperate to provide users with more flexible utilization. In our middleware, called VNA, users can create a new appliance by assembling logical functions distributed in the network.
2.1 Location information in a home network

At home, we must consider that users who are not familiar with operation of appliances frequently. Therefore, we cannot expect the users to set up or control a network and appliances completely by themselves. We need to conceal complicated operations and provide the users with human-friendly operations. Such kind of operation can be partially automated by leveraging environmental information, as well as voice and image recognition. In this paper, we discuss user support and automatic operation using location information in networked appliances environment.

Most information appliances such as television and refrigerator strongly depend on physical location of appliances themselves and their users, because these appliances provide physical input and output such as sounds, temperature, and physical contacts. Also in this environment, it is likely that users and appliances move frequently. Users must lookup or set up appliances whenever the users’ and appliances’ location change. Furthermore location information is very useful for intuitively understanding connection and cooperation of appliances.

**Example A:** Bob is watching a TV program in the living room. When Bob has moved to his own room, VNA architecture should forward the A/V stream to the TV in his room. In addition, preferably, VNA architecture should turn on the room lights automatically.

2.2 Location Representation

In order for appliances to be location-aware, both user’s and application’s location need to be managed.

Location information can be represented in several ways. An example is coordinates. The representation of coordinates is useful to measure liner distance between two points in the same room. Another example is a Fully Qualified Domain Name (FQDN) format. FQDN is suitable for description of spaces. Description of location information strongly depends on application which uses it. It is important to choose appropriate representation to be suitable for applications.

**Example B:** Bob’s room is described as “5215.delta.sfc.5322.endo.fujisawa.kanagawa.japan”. If there are two TVs in the room, the nearest one is selected by coordinates calculation.

2.3 Location Management

To look up appliances around users and vice-versa, user’s and appliance’s location should be administrated. There are two ways of administrating location. One is area server configuration. In the area server configuration, location of physical objects such as users and devices are administrated in a certain place. The area server configuration, thus, can recognize who resides or what exists at a certain place. However it cannot recognize the location of a user or an object if they are scattered all over the world. The other way is home server configuration. In home server configuration, users and devices notify their location to a home server whenever they move. The home server configuration detects where a user lives or a device exists. However the home server configuration cannot recognize who lives or what exists in a certain area. By administrating both user’s and device’s location, we can reduce management cost for utilizing information appliances.

**Example C:** Bob’s location is administrated by a home server, and area servers recognizes TV’s location in each area. Consequently his father in the remote office can show a message to the TV nearest to him.

3 L-VNA

We have developed a flexible and dynamically adaptive middleware, called Virtual Network Appliances (VNA) architecture, aiming at integration of information appliances. This section describes its concept regarding upon achieving location awareness.

3.1 Virtual Networked Appliances

Aiming at integration of heterogeneous appliances on heterogeneous networks, we focus on composing logical appliances, VNAs, by assembling functional components regardless of physical boundaries of the appliances. A VNA is defined with a markup language (VNA Markup Language: VML). To make VNAs portable among various physical appliances, we introduced a new mechanism called Virtual Plug & Play. The
key advantage in our architecture is a property of a server-less distributed configuration.

Our new mechanism enables users not only to define VNAs themselves, but also to acquire pre-defined VNAs from vendors and simply utilize them with neither installation nor management cost. We call the former as bottom-up integration, and the latter as top-down integration. The top-down integration is our challenge which existing middlewares do not aim to. In top-down integration, users virtually plug a pre-defined VNA into a home network, by loading the VNA into our middleware. Virtual Plug&Play subsequently creates an instance of the VNA by mapping its function templates with Serdgets, which are functional components running on physically existent information appliances. Virtual Plug&Play includes the following three mapping policies: location mapping, QoS mapping, and time mapping. Mappings between function template and Serdgets are dynamically updated, and Virtual Plug&Play provides users with a dynamically adaptive environment. Since our architecture does not force users to install any servers, users can achieve flexible utilization of information appliances with little management cost.

3.2 The Location mapping module for Virtual Networked Appliance

We have developed a location mapping mechanism called L-VNA. A location-aware mapping mechanism between function templates and Serdgets in VNA, which is one of the aforementioned mapping policies, is called L-VNA. We call this location mapping “Location-based Virtual Plug&Play”. Location based Plug&Play is performed in the following cases.

VNA Instance Creation

When a new instance of a VNA is created, L-VNA maps its function templates with Serdgets by using their location information. In the absence of user’s preference on location, L-VNA maps the function templates with the nearest information appliances.

Location Change

When an information appliance has moved, L-VNA senses the location change and remaps function templates. A VNA is, thus, constructed by Serdgets considering the user’s preference or location at all times. As mentioned above, L-VNA is dynamically adaptive to location changes, thus users does not need to reconfigure location information of Serdgets.

3.3 Design Issues on location mapping module

Lightweight The key advantage in VNA architecture is server-less property. In order to realize this server-less property, it is necessary that L-VNA is installed in VNA runtime on each appliances. Therefore, L-VNA should retain small footprint and interchangeability with other modules in VNA runtime.

Modularity of Search Mechanism The description and search algorithm of location information strongly depend on applications which use them. Therefore, search mechanism of L-VNA should be modular, and L-VNA rearranges its algorithms.

Autonomy In order to dynamically adapt VNA architecture to location changes, L-VNA needs to have autonomy. L-VNA should notify VNA runtime of location information proactively whenever location changes.

4 Design and Implementation

In this section, we describe the design and the implementation of the prototype L-VNA. Location-based Virtual Plug&Play maps function templates to Serdgets, which run on information appliances, using location information.

4.1 Location-based Virtual Plug&Play

We have implemented a prototype of the location mapping mechanism by LocationMetamorphicComponent class, AreaManager class, Area class and AttributeSet class. An instance of LocationMetamorphicComponent class has three states as shown in Figure 1: mapping, mapped, and sleep phases. The LocationMetamorphicComponent instance starts from Location mapping phase. In this phase, the instance searches for a Serdet which has appropriate attributes by using the L-VNA module. In case such a Serdet is found, the instance metamorphoses into it. In other cases, the instance sleeps and returns to Location mapping phase.
When a device, on which a Serdget is running, moves to another location or disconnects, the Serdget receives a LocationChangeEvent from L-VNA module. The Serdget, which has received a LocationChangeEvent, returns from Mapped phase to Location mapping phase. Location mappings and event notification are realized by using L-VNA module.

![State diagram of LocationMetamorphicComponent](image)

**Figure 1. State diagram of LocationMetamorphicComponent**

### 4.2 System Configuration

#### 4.2.1 Hardware Configuration

In order to realize Location-based Virtual Plug&Play, we assume that tags and badges are installed in each information appliances, and location detection sensors exist in each area. In the current prototype, E-code Spider [11] and its tags are used for location detection. As a prototype of information appliances, we use industrial network computers named DUONUS[12]. DUONUS has several interfaces (RS-232C, DI/DO, ether) and a flash memory socket. In the current prototype, FreeBSD3.4[9] and Java Virtual Machine (JVM) runs on DUONUS. Table 1 shows the specification of DUONUS.

<table>
<thead>
<tr>
<th>Table 1. the specification of DUONUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL</td>
</tr>
<tr>
<td>CPU</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>Compact Flash</td>
</tr>
</tbody>
</table>

### 4.2.2 Software Configuration

L-VNA consists of four parts (VNA runtime, L-VNA module, location detection module, and VNA components). Each information appliance needs a VNA runtime and a L-VNA module. Moreover, location detection modules work on computers which have location detection sensors. VNA components work on VNA runtime and provide various services by composing other VNA components. Refer to the following section for further details.

### 4.3 L-VNA module

L-VNA provides four functions: lookup, registrar, location update and event notification. The L-VNA module is implemented as a Serdget and runs on VNA runtime. A L-VNA module communicates with VNA and other Serdgets by using Ports which realize message communication. Figure 2 summarizes interactions between L-VNA, VNA runtime and other Serdgets.

![The relationship between L-VNA module and VNA runtime](image)

**Figure 2. The relationship between L-VNA module and VNA runtime**

#### 4.3.1 Registrar Function

A Serdget is registered as an Attribute Set in the L-VNA module at the moment the Serdget is loaded into a VNA runtime. The Attribute Set contains VNA Identifier, device name, tag ID, and other information. The Attribute Sets are stored in an Area class. The Area represents a space such as a building, a floor and a room. Area Manager administrates Areas in a tree configuration. Figure 3 shows relation between Area and Attribute Sets.
4.3.2 lookup function

A LocationMetamorphicComponent instance sends its area information, search policy and other attributes to L-VNA. Afterwards, L-VNA looks up an appropriate Serdget by using the tree architecture. L-VNA provides the following three search policies.

Search Same Area L-VNA searches a Serdget which completely matches the area attribute of LocationMetamorphicComponent. By using this policy, we can bind a display with speakers in the same room and compose a virtual television for example.

Search Adjacent Area L-VNA detects sibling nodes of the area attribute, which are maintained by LocationMetamorphicComponent.

Search Nearest Area L-VNA tries to detect the Serdget which exists in the nearest room, floor and building. A L-VNA searches tree architecture, which is administrated by L-VNA.

4.3.3 location update function

The Location update function of L-VNA receives location update information from location detection devices that can detect location changes of physical objects. Location update information is sent in the following format:

device_name://location/ID

where device_name represents location detecting device, location is described with the FQDN format, and ID identifies an information appliance with the Serdget running on it. In L-VNA, an area is defined as units of some space such as a room, a floor, and a building. Any location detecting device will do as long as it has area boundary. For example E-code Spider is suitable for our middleware, since we can set area boundary as we like by adjusting radio strength. On the other hand, GPS is hardly adoptable, since it does not have functions which recognize area boundary.

For example, a device which has No.59 as tag ID of E-Code Spider and exists in our laboratory, is described as follows.

![Spider](https://example.com/spider.png)

-------------------

SPIDER_1.34://S215.2.delta.sfc.5322.endo.fujisawa.kanagawa.japan/59

4.3.4 Event Notification Function

Whenever L-VNA module receives location update information from location detection devices, L-VNA module updates its area tree. Serdgets, which are interested in its update, need to implement AreaTreeChangeListener interface. The L-VNA module notifies Serdgets including LocationMetamorphicComponent through the interface when location of Serdgets changes.

5 Applications

We present the following services as applications of L-VNA.

Location adaptive Virtual-FAX (LV-FAX)

A Virtual-FAX is composed of a printer, a scanner and a telephone. In addition, L-VNA adds location adaptive function to Virtual-FAX. We call this LV-FAX. Due to L-VNA, LV-FAX is composed of the Serdgets near each other. Thus if someone takes the printer out of the space, LV-FAX is reconstructed using another printer which has become the closest. By using LV-FAX, users can use a function of faxes without consciousness and administration of information appliance’s location.
Location adaptive Virtual air conditioner (LV-air conditioner)

Suppose that there is an air conditioner and many sensors. The air conditioner can control room temperature. In addition, thermometers and hygrometers obtain surrounding temperatures and humidity and detect its own location using E-code Spider. The air conditioner, thermometers and hygrometers compose a Virtual air conditioner. By switching sensors with users’ movement, it is possible to control room temperature automatically considering bodily sensations.

6 Related Works

Jini provides searches using several attributes and realizes Plug&Play with Java technology. Service Location Protocol (SLP)[10] also provides service discovery with location information. However, since their attributes of location have only texts as construction of attributes, it is impossible to realize flexible search by using location information.

Ultrasonic Location System [3] obtains three dimensional location information using ultrasonic sensors. Bat Teleporting is an application using Ultrasonic Location System. This system administrates users’ location and detects their movement. Bat Teleporting switches the image shown on the user’s display to the nearest display. Accordingly, this system realizes the dynamic adaptation to location information. Unfortunately, Ultrasonic Location System lacks modularity of search mechanisms and search services.

Service Discovery Service (SDS)[2] is developed as a part of Ninja Project[8]. SDS administrates the service location hierarchically. Moreover, SDS has various search policies (e.g. location-base, administration-base and network topology-base). It is, thus, possible for SDS to select the service by using many search algorithms. However SDS requires several servers and is too heavy. Consequently, if SDS is applied to home networks, it brings users a management cost. In addition, Virtual Plug&Play cannot be realized only with SDS, since SDS lacks mechanisms of composing functions.

7 Conclusion

In this paper we have described the design and implementation of a location-adaptive VNA which realizes “Location-based Virtual Plug&Play”. L-VNA holds lightweight, modularity of search and autonomy as its design issues. L-VNA is a mechanism which maps functional templates to Sergets using its location information. We have implemented the prototype of L-VNA and suggested its applications. Finally L-VNA lowers management cost of appliances, and realizes location-adaptive utilization of information appliances.

References