Design of a Universal Middleware Bridge for Device Interoperability in Heterogeneous Home Network Middleware

Kyeong-Deok Moon, Young-Hee Lee, and Chang-Eun Lee, Young-Sung Son, Non-Member, IEEE

Abstract —*This paper proposes the design of the software Universal Middleware Bridge (UMB) that can be used to solve seamless interoperability problems caused by the heterogeneity of several kinds of home network middleware. We verified that the proposed UMB dynamically maps physical devise in all different middleware domains into virtually abstracted devices in the UMB domain and enables all home devices overlaid on heterogeneous networks to be seen as virtually the same physical devices in the same middleware domain, as well as to detect and control each other. As a result, it is concluded that the proposed architecture provides commercial feasibility and cost benefic for the system implementations*[1]

Index Terms — interoperability, UMB, virtually abstracted devices, UMB domain.

I. INTRODUCTION

THE integration of computing-power and communication into the next generation of everyday device is one of the major factors that is driving the emergence of home network, which interconnects various appliances, enabling remote access to and control of those appliances, and any available services such as home entertainment, home office, and home automation. Nowadays, the increasing diversity of home devices implies that various kinds of home network middleware will be working in home. Because these devices typically have severely restricted computation resources, middleware technologies most appropriate for these devices are continuously turned up. It is highly improbable that there will be a single dominant home network middleware platform that would be good enough for various appliances and purposes in the near future. It is the key element for achieving digital home to provide interoperability among the heterogeneous home network middlewares which have quite different command and data delivery mechanisms. Because currently most available home network middleware, such as HAVI [1], Jini [2], LonWorks [3], and UPnP [4], have no compatibility with each other, home devices based on these heterogeneous middleware cannot communicate with one another, even though they are physically connected; therefore this paper proposes the design of the software Universal Middleware Bridge (UMB) that can be used to solve interoperability problems caused by the heterogeneity of several kinds of home network middleware (Fig. 1.).

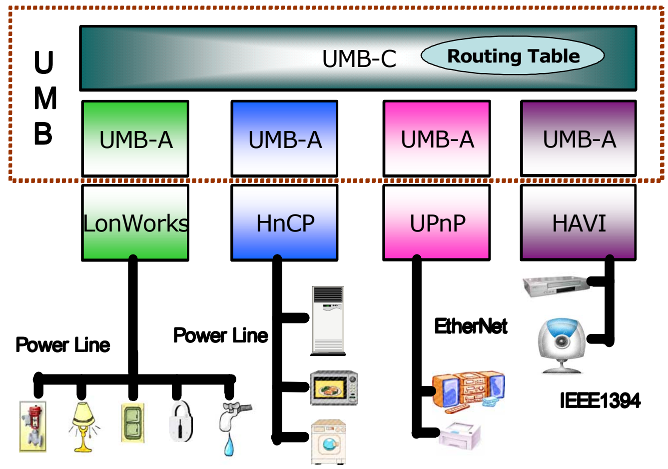


Fig. 1. The Architecture of Universal Middleware Bridge

II. RELATED WORKS

Developing home devices' interoperability with heterogeneous middleware creates a number of problems. Among these, the problem of protocol conversion is very important and is focused in this paper. There are several approaches to ensure the interoperability among the different home network middlewares. They are one-to-one and one-to-any protocol conversion approaches. Most of the existing bridge solutions used to settle this problem employ one-to-one protocol conversion. HAVi-to-UPnP bridge at Thomson provides the interoperability between UPnP and HAVi. It solves some problems with diversification of middleware. But, these are not enough to develop a single bridge that connects two specific middleware one to one, when new middleware will be developed one after another [5]. Therefore, though this approach offers interoperability that would be good enough for both types of middleware, the conversion complexity of this approach is too high, $(n*(n-1)) / 2$ where n is the total number of middleware. Recently T, Nakajima's paper of Waseda University proposed a virtual overlay network that has a one-to-any protocol conversion approach [6]. This Framework for Connecting Home Computing Middleware at Waseda University enables any appliances under any middleware's control to communicate any other appliances. It uses and deploys service of home without special conscious of diversification of middleware. This approach is relatively

[1] Kyeong-Deok Moon is with the Electronics and Telecommunications Research Institute, Korea (e-mail: kdmoon@ etri.re.kr)
Young-Hee Lee is a professor at Information and Communications University, Korea (e-mail: yhlee@icu.ac.kr)
Chang-Eun Lee is with the Electronics and Telecommunications Research Institute, Korea (e-mail: celee@ etri.re.kr)
Young-Sung Son is with the Electronics and Telecommunications Research Institute, Korea (e-mail: ysson@ etri.re.kr)

Manuscript received January 15, 2005

simple but cannot automatically support dynamic service activation by combining any functions of any appliances and the device's discovery scheme in without user intervention.

III. DETAILS OF PROPOSED FRAMEWORK

To overcome the defects of the previous works, the proposed UMB architecture, which consists of UMB Core (Fig. 2.) and UMB Adaptor (Fig. 3.) aims at two goals.

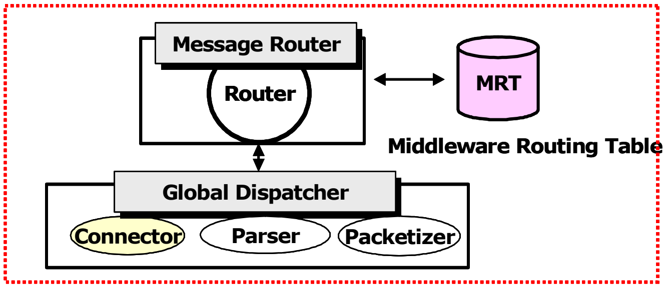


Fig. 2. The Structure of UMB Core (UMB-C)

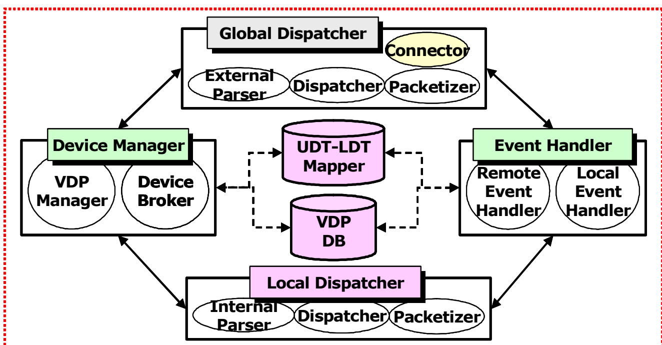


Fig.3. The Structure of UMB Adaptor (UMB-A)

First, since home appliances in different middleware domain must be seen as virtually the same physical devices in the same middleware domain, the UMB Adaptor allows physical devices with its own middleware to virtually expose the abstracted devices that wrap the original appliances to all the middleware's domain. In the case where the device is plugged in on specific middleware, the UMB Adaptor overlaid on that middleware converts the physical device into the virtually the abstracted one described by Universal Device Template(UDT), which consists of a Global Device ID, Global Function ID, Global Action ID, Global Event ID, and Global Parameters (Table. 1.). If the UMB Adaptor, which contains any other middleware, receives the exposed virtual device with UDT through the UMB Core, it creates the Virtual Device Proxy (VDP) database (Table.2.) and announces this locally to connected devices.

Second, in order to insure that devices in various middleware network control and monitor one another, the UMB Adaptors translate the local middleware's message into global metadata's message. For this process, the Device Manager (DM) of the UMB Adaptor tries to match each local device's function, which is expressed by the Local Device Template(LDT), consisting of the Local Device ID, Local Function ID, Local Action ID, Local Event ID, and Local Parameters with appropriate UDT by the UDT-to-LDT Mapper (Fig. 4.).

```
<?xml version="1.0"?>
<UDT>
   <deviceType>Device Type</deviceType>
   <function>
      <functionId>Function ID</functionId>
      <functionType>Function Type</functionType>
      <actionList>
         <action>
            <actionId>Action Name</actionId>
            <parameterList>
               <parameter>
                  <parameterId>Parameter ID</parameterId>
                  <dataType>Data Type</dataType>
                  <direction>Direction</direction>
               </parameter>
            </parameterList>
         </action>
         more actions go here if exist…
      </actionList>
      <eventList>
         <event>
            <eventId>Event ID</eventId>
            <dataType>DataType</dataType>
         </event>
      </eventList>
      more actions go here if exist…
   <function>
   more functions go here if exist…
</UDT>
```

Table. 1. XML Description of UDT

Property	Field Semantic	Example
Index	DB index	0
Global/Local	Global or Local Device	Local
GID	Global ID	LonWorks, LID
LID	Local ID	0x112233445566
GDType	Global Device Type	LAMP
LDType	Local Device Type	0x05 0x01 (LonMark Object)
Local address	Local Device Address	SubnetID/NodeID
Function List	Available Function List	F1
		F2,…
Action List	Available Action List	…

Table. 2. VDP Database.

If a translator object (UDT) is found, the Global Dispatcher (GD) of the UMB Adaptor creates and send a Universal Metadata Message (Table. 3.), which has UDT as parameters to the UMB Core. Universal Metadata Message is used as follows : UMB Adaptor finds the socket connection of UMB Core by SSDP discovery message. QueryDeviceList message requests all devices list on specific middleware. NotifyOnlineStatus message notifies the UMB system event to all UMB Adaptors whenever specific UMB Adaptor or

Device is Plugged in/out. NofifyDeviceState message notifies the device event on specific middleware to all UMB Adaptors in case of changing the device's status. QueryDeviceState and QueryAction message monitors and controls the device. The major role of the UMB Core is routing the universal metadata message to the destination or any other UMB Adaptors by the Middleware Routing Table (MRT). The Event Handler (EH) of the UMB Adaptor sends and receives the events generated by changing the device's status. EH invokes the Remote Event Handler and Local Event Handler. Local Event Handler delivers local device event to the remote device which subscribe that event, when the status of subscribed event is changed. Remote Event Handler delivers the received event to the local device which subscribes that event.

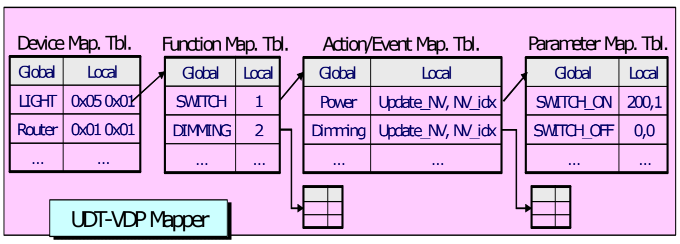


Fig.4. UDT-LDT Mapper

Item	Message Type	Argument (Target/ReQ/Resp)
Middleware	SSDP Discovery	IP Port
	QueryDeviceList	
Event	NotifyOnlineStatus	Middleware ID Device ID Online status Device ID
	NotifyDeviceState	Device ID Function ID Action Name State variable value Out parameters
Control / Monitor	QueryDeviceState	Device ID Function ID(O)
	QueryAction	Module ID Device ID Function ID

Table. 3. Universal Metadata Message

Now, we describe the flow when a device is entered and controlled over home network. Fig. 5. shows the sequence of adding a new device in UMB. When a new device is plug-in, the device sends the announcement message to its sub-middleware. When the sub-middleware receives the announcement message, it notifies that event to the UMB Adaptor. It also gets the information on that device and configures that device. The UMB Adaptor overlaid on that middleware converts the physical device into the virtually the abstracted one (UMB-VD) described by Universal Device Template(UDT). After that, UMB Adaptor translate the local middleware's message into Universal Metadata's Message and sends a UMB message to the UMB core. The UMB core (UMB-C) routes the UMB message to the destination or any other UMB Adaptors by the Middleware Routing Table(MRT).

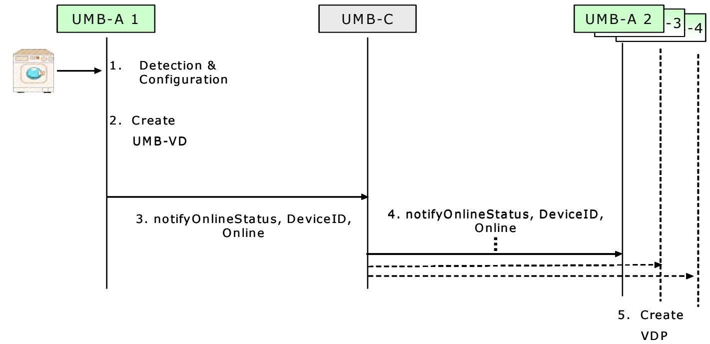
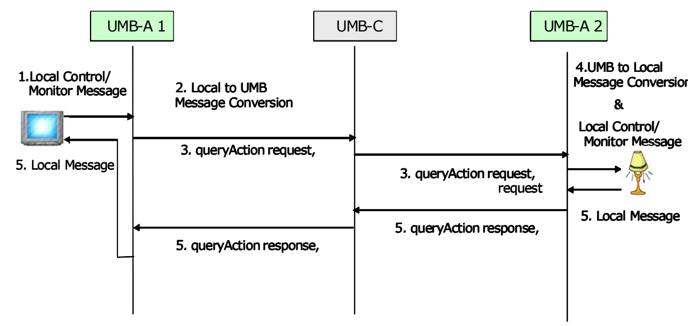


Fig.5. Flow when a new device is plugged in

If the processing mentioned above is done properly, the Physical Light device on LonWorks Middleware would be registered as a Virtual UPnP Light device on UPnP Middleware and the UPnP control point should have exactly the same physical UPnP Light device experience (Fig. 6.).



6. Flow when a device is controlled and monitored.

IV. EXPERIMENTS RESULT

In order to verify the feasibility of the proposed middleware, we've implemented an event-based surveillance service using a HAVi-camera and HAVi-display on IEEE1394, a LonWorks-motion sensor on PLC, and a Jini-display on TCP/IP. Event-based surveillance service monitors the intrusion with LonWorks-motion sensor. If someone broke into the house, LonWorks-motion sensor notifies the event. Then, HAVi-camera would send the captured-image to display appliance nearest to user immediately upon receiving an event from the LonWorks-motion sensor. UMB and several sub-middlewares are executed over home server, which is the central of home network, and manages and controls the appliances on the home network without user intervention. We have implemented our home server with JDK1.3.1 and Linux. A user point PDA to select the room where he is. Then intelligent service application can dynamically combine the HAVi-camera and the display-appliance nearest user that receives and displays the captured-image from HAVi-camera. If a user move to different room, he switches the location by pointing PDA. As soon as intelligent service application receives the change of user's location, existing interactions between the appliances will be hand over to the new display-appliances resulting in uninterrupted use of service according to change of user's location. Fig. 7. shows the test-bed. This experiment shows that a intelligent service provides the context-aware and adaptive service with considering the user's location and circumstances change through dynamically combining by Universal Middleware Bridge (UMB). Furthermore, UMB provides users with single image view of digital home, and an appliance can communicate with other appliances under different middleware.

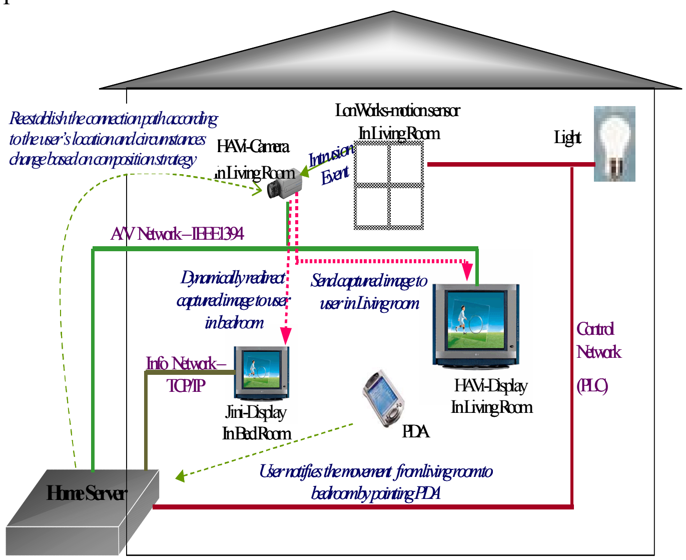


Fig. 7. Testbed with Heterogeneous Home Network Mddileware

V. QUALITATIVE EVALUATION

We evaluated our research from qualitative perspectives to clarify the similarities and differences between our study and related researches following particular criteria toward a framework for providing seamless interoperability under heterogeneity. Our qualitative evaluation compares features of middleware supporting interoperability among heterogeneous middleware. Table 3 shows the results of our qualitative evaluation among middleware supporting interoperability.

UMB and Waseda's approach provide the interoperability via a one-to-any protocol conversion based on meta-protocol, therefore the cost of implementation is lower than that of one-to-one bridge, such as Thomson's approach. Our approach enables interoperability without any modification of appliances under heterogeneous middleware, and globally managing the entire home network. But, Waseda's approach can globally manage the entire home, but applications or appliances should be modified in order to enable interoperability. One-to-one bridge, such as Thomson's approach need be modified the appliances for providing interoperability, and it locally manage the entire home network.

Since UMB is implemented based on hybrid discovery system, UMB can provide flexible query. Waseda's approach is implemented based on server-based, it can also provide flexible query. However, since Thomson's approach is implemented based on server-less, it only supports the simple query. UMB abstracts the functionality of appliances with well-defined script based on XML, thus it can dynamically bind the functions of appliances without any modification of the application during execution. Therefore abstraction of UMB efficiently enables the value-added applications. However, since Waseda's and Thomson's approach does not provide well-defined abstraction and only support static binding on initializing execution, thus they is not suitable for value-added applications.

Moreover, UMB automatically converts a protocol which is specific to a certain middleware into other middleware-specific protocols by mapping the middleware-specific protocol into meta-protocol defined by UMB via a UDT-LDT mapping relation. Therefore a newly coming appliance will be easily integrated into UMB by simply describing the UDT-LDT mapping relation for that appliance. Moreover, UMB can efficiently handle optional function through dynamically extracting and defining user-defined set for a newly added appliance. Waseda's approach also converts a middleware-specific protocol into other middleware-specific automatically via a javassist tool. However, Waseda's approach cannot handle optional functions dynamically, and if any functions of an appliance are changed, it needs to re-generate interfaces through javassist tool. Since Thomson's approach manually converts the protocol between middleware, it can easily support optional functions. Now, we describe the features of UMB for supporting dynamic substitution of appliances. Coarse-grain composition is command or process execution appropriate. On the contrary, fine-grained composition is performed within an application process without restarting the process. Thus, fine-grain granularity is better than coarse-grain granularity for self-adjustability. Therefore, UMB supports fine-grain granularity.

	Ours	Waseda	Thomson etc.
Interoperability under heterogeneity	One-to-any conversion based on defined meta middleware	One-to-any conversion based on defined meta middleware	One-to-one conversion based on defined bridge
Cost for interoperability	n-1, 1	n-1, 1	n*(n-1)/2, n-1
Modification of legacy appliance	X	O	O

Resource management	Global	Global	Local
Resource discovery	Hybrid	Server-based	Server-less
Appliance abstraction	O	X	X
Binding	Dynamic binding during executing	Static binding on initialization	Unsupported
Query Level	Flexible	Flexible	Simple
Support of manufacturer-specific function	Support	Limited	Support
Mapping Mechanism	Automatic mapping based on UDT	Automatic mapping based on javassit tool	Manual

Table. 4. Evaluation of interoperability among heterogeneous middleware from qualitative perspectives.

VI. CONCLUSION

In this proposed approach, a new middleware can be used in our UMB framework effortlessly without any modification of existing middleware layer, but by simply adding the UMB adaptor module for the UMB mechanism. We verified the commercial feasibility of the proposed UMB architecture through experimental results. As a result, we have concluded that the proposed UMB architecture provides interoperability necessary to easily monitor and control appliances connected home network through UMB.

APPENDIX

Appendices, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The authors wish to thank Professors Young-Hee Lee for his contributions to design of UMB.

REFERENCES

[1] HAVI Home Pages, *http://www.havi.org*
[2] *JINI Home Pages, http://www.sun.com/jini*
[3] *Echelon Co., "LonTalk Protocol Specification ver 3.0", 1994*
[4] *UPnP Home Page, http://www.upnp.org*
[5] *Bichot Guillaume, Ramaswamy Kumar, Burklin Helmut, and Stahl Thomas, " Methods for bridging a HAVi sub-network and a UPnP sub-network and device for implementing said methods," Thomson Multimedia, 2002*
[6] *Eiji Tokunaga, Hiro Ishikawa, Makoto Kurahashi, Yasunobu Morimoto, and Tatsuo Nakajima, " A Framework for Connecting Home Computing Middleware," ICDCSW'02, 200*
[7] *Tatsuo Nakajima, Daiki Ueno, "A Virtual Overlay Network for Integrating Home Appliances", Proceeding of the 2002 Sysmposium on Applications and the Internet (SAINT'02).*
[8] *Bill Rose, "Home Network: A Standards Perspective", IEEE Communication Magazine, December 2001*
[9] *Gerado O'Driscoll, "The Essential guide to Home Networking Technology", Prentice Hall, 2001*
[10] Kwangroh Park, Yeunjoo Oh, Kyungshik Lim, Kyoung-Rok Cho, "A Dynamic Packet Recovery Mechanism for Realtime Service in Mobile Computing Environments," .ETRI Journal, vol. 25, pp.356-368, Oct. 2003
[11] *Koon-Seok Lee et al., "A New control protocol for home appliances-LnCP", Indeustrial Electronics, 2001. proceedings ISIE 2001. IEEE International Symposium on, vol. 1, pp.286-291, 2001.*

BIOGRAPHY

KyeongDeok Moon received the B.S. and M.S. degrees in computer science from Hanyang University, Korea in 1990 and 1992 respectively. From 1992 to 1996, he was researcher at System Engineering Research Institute where he worked on high performance computing and clustering computing. Since 1997, he has been a senior researcher and a team leader of Information Appliance Control Software Research Team at Electronics and Telecommunications Research Institute, where he develops the home network middleware and Java embedded architecture. He has been PhD student at ICU since 1998. His research interests in home network middleware, Java, active network, and pervasive computing

YoungHee Lee received the B.S. and M.S. degrees in EE from Seoul National University, Korea in 1976 and 1980 respectively. He received Ph. D degrees in computer science from Universite de Technologie de Compiegne, France in 1984. From 1984 to 1997, he had been working at ETRI as a various research team leader including as a director of protocol engineering center. He was invited scientist at IBM T. J. Watson Research Center from 1986 to 1987. From 1998 to 2002, he was the dean of school of engineering in Information and Communications University, Korea and now he is a professor at ICU. From 1994, he was serving as a vice chairman of ITU-T SG7, and chairman of WP3. His research interest is in Internet protocols, active networking and network supports for pervasive computing.

ChangEun Lee received the B.S. and M.S. degrees in Electronics Engineer from Hanyang University, Korea in 1996 and 1998 respectively. From 1998 to 2000, he was researcher at LG Industry System, where he worked on intelligent building automation systems. Since 2001, he has been a researcher in Information Appliance Control Software Research Team at Electronincs and Telecommunications Research Institute, where he develops the home control network middleware and embedded system architectures. His research interests are in home network middleware, embedded system, and wearable computing.

YoungSung Son Son Young-Sung received a B.S. and M.S. degrees in computer science from Pusan National University in 1995 and 1997 respectively.From 1997 to 1999, he worked for developing file system and vod server of linux clustering software at Electronics and Telecommunications Research Institute. Since 1999, he joined embedded software center for developing the home network middleware and Java embedded architecture. He was PhD student at Pusan National University from 2002. Currently, he is interested in home network middleware, P2P computing, and pervasive computing.