A Framework for Advanced Home Service Design and Management

Félix Jesús Villanueva, David Villa, María José Santofimia, Francisco Moya, Juan Carlos López
School of Computer Science, University of Castilla-La Mancha, Spain

***Abstract*—In this paper a distributed object-oriented framework (*DOBS, Distributed Object Based Services*) for home service design is presented. This framework eases the development of advanced services able to run in a variety of devices ranging from tiny wireless sensors to powerful multimedia servers. Special emphasis will be made on DOBS main features, its core components, the offered common services and the toolchain that has been developed to allow users and manufacturers to easily build advanced DOBS compliant services.**

I. INTRODUCTION

The lack of a common framework for home services development has been largely studied in last years. The main problem that has been identified is the heterogeneity and diversity of devices and services that exist in such type of environment.

Our proposal is built on the well-known Distributed Object Oriented (DOO) paradigm as a starting point for modeling, in a more efficient way than existent solutions, any type of service one can found (or envision) in home networks. Special care has been taken in the way required resources and bandwidth network can be optimized. In DOBS, all services are distributed objects that can run in low-footprint devices as well as in personal computers (having the manufacturers also the possibility of efficient ad-hoc hardware implementations). Additionally, a complete toolchain that allows manufacturers and even users to create advanced services, abstracting from middleware particularities, has been created.

II. STATE OF THE ART

In order to deal with heterogeneity, most of the research efforts in the area of home networking have been focused in the use of Java or Web Services -like platforms to create and manage the supported services. Examples of this type of middleware are OSGi[1], UPnP[2], etc.

Most of these solutions do not deal efficiently with resource constrained devices (e.g. wireless sensor network, WSN devices), or bandwidth limited networks (e.g. control networks). Moreover, they do not provide with powerful tools and methods that allow developers and manufacturers to substantially simplify the development of new advanced services (e.g. support for several languages, common services, a complete and integrated toolchain, etc.).

Other efforts have proposed object-oriented distributed platforms like the most efficient way to develop this type of environments [3]. However they are exclusively centered in interoperability between service oriented middlewares (HAVi, Jini or UPnP).

Our proposal is an alternative to such platforms overcoming their drawbacks and without overlooking their integration. An initial work that supports this proposal can be seen in [4].

III. DOBS CORE COMPONENTS

DOBS core components are:

- DOBS interfaces: They allow to model basic monitoring (temperature, humidity, presence, etc.) and control (light, doors, etc.) services.
- Audio and Video services: The AVStreams interfaces from the Object Management Group (OMG) have been adopted.
- Common services: Services such as service discovery, security mechanisms, bootstrap, management facilities, etc. are available to be used by the remaining home network services.
- Integration subsystems: These are specific subsystems that allow seamless integration of services from other platforms. As an example, subsystems for UPnP, X10 or Bluetooth services integration have been developed.
- Information model: A complete taxonomy of user services (with their attributes) has been developed so as the manufacturers can use a common nomenclature. This taxonomy is built up from UPnP templates and Bluetooth profiles and it has been completed with nomenclature and services from most relevant standards (Mobile Location Protocol, AVStreams, etc.). Together with the service type, the developer can consult the different service attributes. The idea behind this taxonomy is to create a basic set of home services that work as POSIX interfaces do for operating systems.

IV. DOBS COMMON SERVICES

We have implemented a set of common services that reduces configuration procedures and provides facilities to service developers. The main services available are:

- *ASDF*: The Abstract Service Discovery Framework allows implementing, in a DOBS environment, any existing model of service discovery protocol. It has been specially designed for interoperability with other well-known service discovery protocols (UPnP SSDP, Bluetooth SDP, SLP, etc.).
- *Bootstrap*: Together with the ASDF service, the bootstrap service enables a Place & Play philosophy for devices.
- *Security*: Our security infrastructure combines low level mechanisms like secure socket layer (SSL) communications with other more sophisticated procedures like digital certificates. This enables DOBS to be used in other more complex environments that precise of a high

This work has been funded by the Spanish Ministry of Science and Innovation under project SODA (TIN2005-08719) and by the Spanish Ministry of Industry under project CENIT Hesperia.

degree of security such as public spaces (airports, railway stations…) or critical facilities (industrial plants, power stations…), being adequate in general, for homeland security applications.

- *QoS*: By means of profiles, the network resources are assigned in function of the state of the environment (day/night, normal, intrusion alarm, etc.).
- *Service management*: A mechanism for service deployment, configuration, actualization… enabling both centralized and distributed management, has been developed.

V. DOBS DEVELOPMENT PROCESS

A complete description of basic and advanced home services has been done using an ontology. It includes the different relationships between services as well as other important semantic information (e.g. correspondences with other domains as UPnP or HAVi for interoperability purposes).

From this ontology, and using the DOBS OWL compiler, a candidate interface (expressed in an interface definition language, IDL) for the selected service can be obtained. This specification is the contract between the service and the client that wants to use it. From this specification and using the IDL compiler, developers get the stubs and skeletons for both, client and server, in the desired programming language. We have also developed tools that allow the generation of stubs and skeletons for services and clients that are intended to run on small microcontrollers (e.g. WSN devices). It is also possible to automatically obtain a hardware implementation (VHDL). More information on these two possibilities can be found in [5] and [6].

Regarding the use of the DOBS common services, templates and examples of use are available (see [7]), so the developer can center on the service functionality taking advantage of the use of these common services to improve the final service implementation in some general (but important) aspects (security, management, etc.).

We have selected the Internet Communications Engine (ICE)[8] from ZeroC for the DOBS implementation. ICE is a CORBA-like middleware that uses a specific protocol (ICEP) and a specific interface description language (Slice).

The DOBS tools (DOBS OWL compiler, DOBS VHDL compiler and DOBS WSN compiler), together with the SLICE compiler (from ICE), compose a toolchain (Fig. 1) able to generate in an automatic way service implementations (servers) for a variety of scenarios (medium size computers as set-top-boxes or residential gateways, small processors as WSN nodes, or even ad-hoc hardware versions). These tools allow the developers to get rid of annoying service communication details (a very error-prone development task) while better focusing on the service functionality itself.

On the other hand, from the client developer's point of view, the access to a specific service is transparent and can be dealt with in the same way no matter whether it is implemented in a residential gateway, in a WSN device or in hardware.

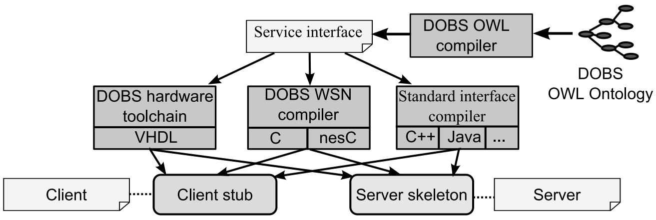

Figure 1- DOBS toolchain diagram

VI. PROTOTYPE

In order to show the features of the proposed middleware, a set of user services, integrating different types of devices, has been developed. Some working scenarios can be found in [7]:

- A generic server for RTSP cameras (VCC4 and AXIS).
- Presence detection platform with mica2 WSN devices.
- Integration of X10 devices adding SDP capabilities.
- An *Inspector* to monitor and control any device or service in a DOBS environment.
- UPnP SSDP protocol (used by AXIS cameras) integration.

VII. CONCLUSION

Meanwhile OSGi provides developers with a Java-based platform for service management without any consideration about user services, and UPnP offers templates for services without management service procedures, DOBS incorporates both aspects and also provides a common set of services to help in the development of advanced services. Besides, DOBS avoids the use of virtual machines, XML parsers and web servers, what has a clear impact on the final resource requirements. This is a key feature regarding the final implementation cost, specially considering the type of devices that have to support home services (white goods, cook machines, sensors, actuators, etc.). Finally DOBS provides with a toolchain that eases the implementation of both, the service itself (in a variety of alternatives) and the client that is supposed to use it.

REFERENCES

[1] OSGi Alliance, *OSGi Service Platform Core Specification*, Release 4, version 4.1, April 2007.

[2] UPnP Forum, *UPnP Device Architecture 1.0*. At www.upnp.org, 2003.

[3] Joo-Yong Oh, Jun-Ho Park, Gi-Hoon Jung, Soon-ju Kang. *CORBA based Core Middleware Architecture Supporting Seamless Interoperability between standard home network Middleware. IEEE Transactions on Consumer Electronics, vol. 49, no 3. August 2003.*

[4] F. Moya, J. C. López. *SENDA: An alternative to OSGi for Large Scale Domotics*, Networks, pp. 165-176, World Scientific Publishing, 2002.

[5] F. Moya, D. Villa, F. J. Villanueva, J. Barba, F. Rincón, J. C. López. *Embedding Standard Distributed Object-Oriented Middlewares in WSNs*. Special issue on Distributed Systems of Sensors and Applications, Wireless Communication and Mobile Computing, Wiley, 2007.

[6] J. Barba, F. Rincón, F. Moya, F. J. Villanueva, D. Villa, J. Dondo, J. C. López. *OOCE: Object-Oriented Communication Engine for SoC Design*. X EUROMICRO Conf. on Digital System Design (DSD). Germany. 2007.

[7] http://arco.esi.uclm.es/dobs

[8] M Henning and M. Spruiell. *Distributed Programming with ICE*, ICE version 3.3. At http://www.zeroc.com, May 2008.