Just Another Service Composition Approach for Smart Home

Tsung-Chi Wu, Cheng-Liang Lin, Ping-Chung Po, Chih-Pin Freg, Che-Wei Hu ,Ting-Wei Hou
Department of Engineering Science, National Chen-Kung University
Tainan City 70101, Taiwan, R.O.C
{n96014138, n96014170, hou}@mail.ncku.edu.tw; {chenlianglin,superjohnathon}@gmail.com, cpfong@fotech.edu.tw

***Abstract*—we propose a service composition approach to provide users a simple graphic interface and to help users configure service compositions easily. We developed a first version OSGi based prototype, which could translate the graph based representations into a physical service composition and control flow. A second prototype is now under development to have an Android based GUI and more flexible composition rules.**

Keywords—Service Composition, OSGi

I. INTRODUCTION

Basic smart home devices are dedicated. Alone with the thriving of diverse consumer electronic devices to provide a variety of services, users would begin to think of combing environmental information and consuming electronic devices, together with their functions, respectively, in order to create more diverse services. Then "service composition" occurs. However, there are issues in performing service composition. The first reason is that service composition usually would require a user to have higher technical background. A user may need to learn to use the dedicated service, and the newly composed service. The second reason is that each developer would have one's own proprietary devices and services, which may be difficult to compose. This research will focus on this two disadvantages to develop a simple and easy-to-use service composition model.

II. PROPOSED SERVICE COMPOSITION

Because of the appearance of Service-Oriented Architecture (SOA) [1][2], it is easier to integrate heterogeneous equipments, software components and data sharing than before. One of the typical integration methods is the introduction of SOA middleware, such as OSGi (Open Service Gateway initiatives). SOA middleware not only can hide the details of communication but it also can easily customize data descriptions and data sharing. By applying SOA middleware, we can achieve data reusability and high intercommunication between heterogeneous resources.[3]

We adopted OSGi in our study, and propose an approach called Lightweight Service Composition (LSC) to service composition. The LSC has two principles: (1) The user has centralized control; (2) the system is light-weight. The first principle is from a user's view. We are to help a user participate in service composition procedures much more intuitively. The second principle emphasizes on a simplified system and technique. It is less complex in building and developing services for developers, users and service providers. We have experiences in OSGi field. We have experiences on OSGi and smart home applications [4-8] and Android based systems [9,10]. We have been working on the proposed service composition model and its prototype since 2010.

Beside the technical background, we are to provide a relative simple interface for users. Because the users are much familiar with graphic user interface (GUI), we develop an interface for the prototype to manifest services and users can compose personal services by operating the service control panel, as shown in

Figure 1. A user can choose the services from the service list and use the buttons for sequence, fork and join to compose the services to create a new scenario or service. From the system side, the service control panel requires a language which can describe service composition. We introduced Process Definition Language (PDL) to present the binding between services and control flows. In other words, we use this language to model the user-defined service composition information and transform the created service into a form that system can recognize and execute accordingly. The architecture of the first phase design is shown in Figure 2. The light-weight service composition engine is composed of service registration, service control handler, and service executor. This engine would interpret the service composition described in PDL and start service composition control flows.

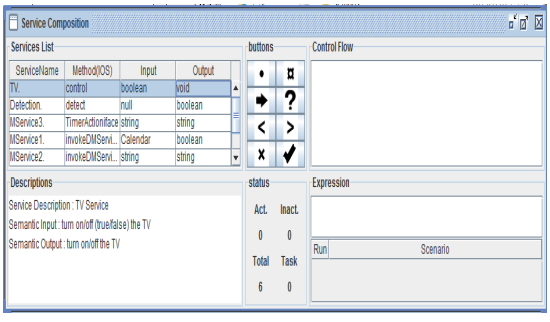

Figure 1 Service Control Panel

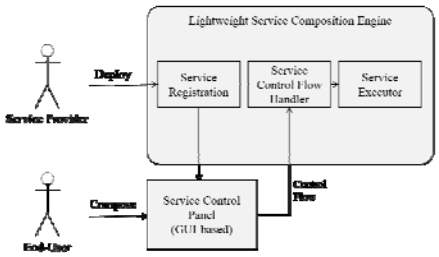

Figure 2 Lightweight Service Composition

III. CURRENT STATUS

Currently, based on the first version prototype, we consider enhancing it in three ways: (1) in sensor services which can provide big amount of data to variety services. (2)to have a eas to use GUI in Android. (3)Add some heuristic or decision support in finding patterns for service composition. For sensor data service, we propose a service, called Sensor Information Service (SIS), to collect, request and response sensor information. This service will be mounted in OSGi. The Sensor Information Service module is shown in Figure 3.

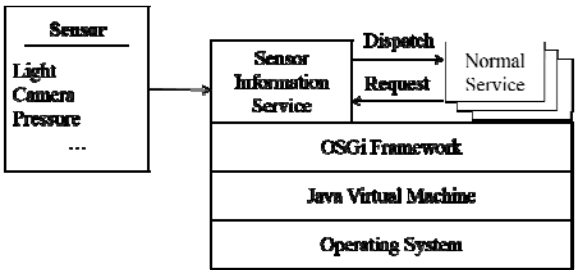

Figure 3 Sensor Information Service

We divide the system into client and server sides. A client, can an Android device to set conditions to operate devices. Then, the system would determine whether the condition is correct. For example, it is an unreasonable condition that start the air conditioner when the temperature is 15°C. A refined LSC that contains service binder, scenario executor with Android user interface is shown in Figure 4. A snapshot of the prototype of Android UI is shown in Figure 5.

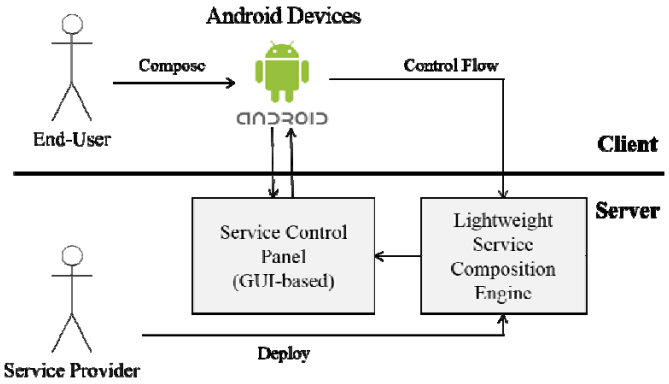

Figure 4 LSC with Android Client

In Figure 5, a user sets a service by choosing the devices/services for the menu. The server receives the control flow from the client and sends it to service binder. Service binder and scenario executer compute and translate control flow to the language representation that environment devices knowing. Finally, Service Executor activates devices according the conditions set by the user. Currently, we are trying to find a better way for users to have a home layout easily and to find a way to embed some intelligence to derive scenarios automatically for the user.

Figure 5 A Snapshot of UI

ACKNOLEGEMENTS

The research is supported in part by National Science Council under contracts NSC101-2221-E-006-071 and NSC102-2221-E-006-138

REFERENCES

[1] "Open Service Gateway initiative Alliance," http://www.osgi.org/
[2] R. S. Hall and H. Cervantes, "Challenges in building service-oriented applications for OSGi," IEEE Communications Magazine, vol. 42, no. 5, May 2004, pp. 144-149.
[3] N. Ibrahim and F. Le Mouël, "A Survey on service composition middleware in pervasive environments," International Journal of Computer Science Issues, vol. 1, no.1, Aug. 2009, pp. 1-12.
[4] Cheng-Liang Lin, Pang-Chieh Wang and Ting-Wei Hou, "A wrapper and broker model for collaboration between a set-top box and home service gateway," IEEE Transactions on Consumer Electronics, vol. 54, no. 3, Aug. 2008, pp. 1123-1129.
[5] Pang-Chieh Wang, Cheng-Liang Lin and Ting-Wei Hou, "Resource-aware service deployment for open service gateway in home network," IEEE Region 10 Conference, Oct. 2007, pp. 1-4,.
[6] Chin-Yang Lin, Cheng-Liang Lin, and Ting-Wei Hou, "A Graph-based approach for automatic service activation and deactivation on the OSGi platform," IEEE Transactions on Consumer Electronics, vol. 55, no. 3, August, 2009, pp. 1271-1279.
[7] Pang-Chieh Wang, Cheng-Liang Lin and Ting-Wei Hou, "A service-layer diagnostic approach for the OSGi framework," IEEE Transactions on Consumer Electronics, vol. 55, no. 4, Nov. 2009, pp. 1973-1981.
[8] Cheng-Liang Lin, Pang-Chieh Wang and Ting-Wei Hou, "Classification and evaluation of middleware collaboration architectures for converging MHP and OSGi in a smart home," Journal of Information Science and Engineering, vol. 25, no. 5, Sep. 2009, pp. 1337-1356.
[9] Tien-Yen Ma, Shu-Wei Hsu, Wei-Tsong Lee, and Ting-Wei Hou, "A novel disply content stabilization scheme for mobile access," Journal of Internet Technology. (Accepted July 2013.)
[10] Chi-Pin Freg, Ting-Yen Ma, Kuo-Yang Tseng, Wei-Tsong Lee and Ting-Wei Hou, " An Android launcher with hand tremor filtering capabilities for the Elderly", Journal of Internet Technology. (Accepted July 2013).