User and Device Adaptable Service Managing Mechanism In Ubiquitous Computing Environment

Jung-Sik Sung , Jong-uk Lee, and Jaedoo Huh
Green Computing Department, IT Convergence Technology Laboratory, ETRI, Daejeon, Korea
jssung@etri.re.kr

Abstract*— Service mobility has become a new issue in the area of service convergence with the advent of versatile mobile devices. Hence, we propose a user and device adaptable service managing mechanism supporting service mobility. This mechanism is performed by adaptive data synchronization. The adaptive data synchronization service performs synchronization with part of data, not all of data, just used in frequent by user. Also, it manages data list in separate. So, it increases the performance of synchronization. The ubiquitous service framework presented in this paper suggests best available service for the user and the device when the user moves to other domains with other device. We implemented a prototype service framework to verify continuity and synchronization of service. Also, we showed a scenario for mobility of video conference service using the prototype service framework.

Keywords-data synchronization;service mobility; user adaptable service management; device adaptable

I. INTRODUCTION

In the past decade, many prototypes have been made for both ubiquitous computing and convergence of services, but it is not easy to service automatically according to users' and environment's context. Users want to be provided continuous and user adaptable service with multiple devices by moving the places. So, the design of a convergence service should address mobility, heterogeneity, and user-centric issues [1]-[3]. Service mobility is considered as maintaining a connection even when terminals or networks are changed due to user movement or personal preference [4]. There are several researches to support service mobility. In [5] it was suggested the method supporting service mobility by moving service components between devices in a serving network. But in this approach, information consistency is harder to achieve for a personal service because the data is scattered across several computers and some of them are disconnected to the network. In [6] it was proposed service mobility method based on Bluetooth but the user-centric mobility was not provided. Mobile agent based frameworks [7], [8] were proposed to provide personal mobility in accessing Internet services. In [7] it supports three Internet services, namely, Web, e-mail, and FTP using four assistants: user, HTTP, mail, and FTP assistants. Assistants operate at a proxy server close to the user. In [7] it supports a personalization scheme only. In order to provide true personal mobility that requires the integration of contact and personalization.

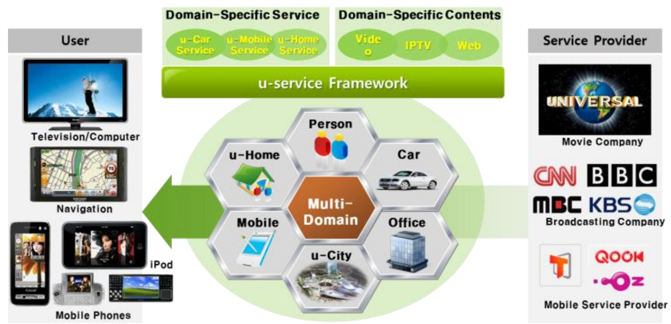

Figure 1. A conceptual model of u-service framework in ubiquitous computing environment

Focusing on service mobility under user movement and heterogeneous devices, a major problem with this service convergence is to build a platform that is applicable to services supported by heterogeneous service platforms and devices with their own platforms. A platform for convergence services plays the role of an infrastructure to execute content and application programs smoothly without obstacles, and provide interoperability between devices and services. We propose an user and device adaptable service managing mechanism in ubiquitous service(u-service) platform to provide the facilities stated above.

Fig. 1 shows a conceptual model of u-service framework in ubiquitous computing environment, where users can be provided continuous services although they move one place to another with different devices. There are several domains in the ubiquitous computing environments, where domains can be defined two terms, one is the device of users such as PC, TV, mobiles and smart phones, and the other is the location of the users, such as home, car, office, hot-spot, and so on. In existing environment, users could only utilize domain-specific services with specific devices. If users move to other domains with other devices, it is difficult to use the service properly, because the device or the network may not support the functions for the service. The u-service framework server presented in this paper suggests the best available service for the user and device, and then constructs and executes a service execution engine for the selected service on the connected device. As the user changes the device due to movement or personal preference, the service is provided continuously with transformed content suitable for the new device.

II. U-SERVICE FRAMEWORK ARCHITECTURE

We propose a u-service framework as an open service framework to support convergence services. It is a technology to generalize a ubiquitous computing environment by providing an environment that eases execution and combination of domain-subordinated services and/or contents by organically integrating independent service domains. Therefore, the u-service framework is an optimized integrated service framework that provides continuous services that are not constrained by physical user environments. Fig. 2 shows an architecture of u-service framework proposed in this paper.

The u-service framework supports the registration of service and execution engines using service profiles, the registration of user and device using user and device profile when users login in u-server framework server. Users and service providers can make profiles and rules using a profile authoring tool which generates and edits profiles of device, execution engine and service, vocabularies, and rules. The functions of u-SF Middle in Fig. 2 are provided by the u-service framework manager. The u-service framework manager supports the registration and management of service and execution engines, the management of service profile, device, subscriber, and service category. It controls user accounts and u-service session. The prototype provides a Web- and proprietary GUI-based method for access of the framework so that devices with conventional Web browsers connect to the framework through a Web site, while devices without Web browsers use a proprietary GUI to access the framework. When services, devices or users are registered or modified to the u-service framework, the u-service framework manager updates service lists and service category lists that are suitable for the user. But service lists that are suitable for both the user and the targeted terminal device are computed in real-time when the targeted terminal device is connected to the u-service framework.

The proposed framework can recommend currently available services based on user preference and device characteristics. Fig. 3 describes a learning algorithm according to user preference gathered by examining the service usage history and a recommendation algorithm for services that can be run on the used device. When the user simply selects one of the recommended services, and the according execution engine is then downloaded automatically/dynamically forming an optimized service execution environment. The dynamic configuration of a service execution environment includes the following procedures: user device profiling, a search for user and device specific services, execution engine search procedure to find an engine that suites both the service selected by the user and the user device profile, transfer of selected execution engine to the user's device, and automatic installation/execution of downloaded engine. The downloaded engine is managed by the execution engine loader and updated automatically and periodically.

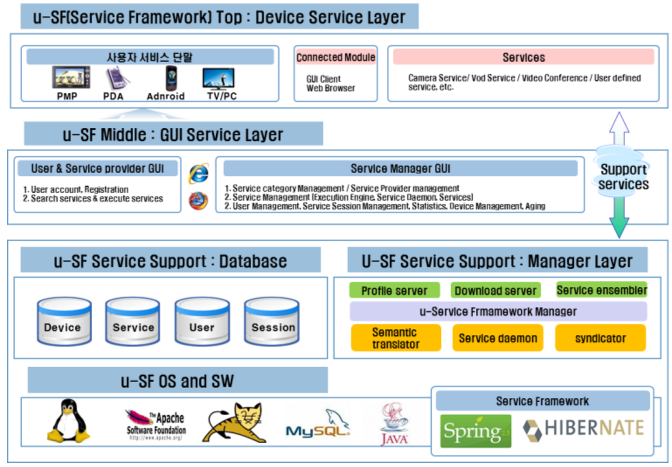

Figure 2. The Proposed u-Service Framework Architecture

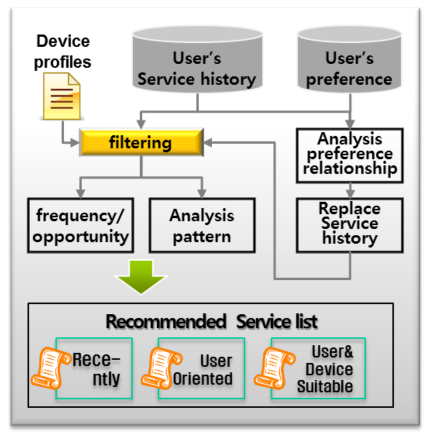

Figure 3. User & device available service recommendation

The framework performs automatic detection of a suitable service execution engine and also supports a search function for services that are suitable for the user and the targeted terminal device. Moreover, it supports real-time content adaptation for targeted terminals, semantic translation including a communication protocol translation, and seamless service continuity so that a user can continue using a service across different terminals. Fig 4 shows processing in series provided by u-service framework in order to support seamless service synchronization at last when a user connects to the u-service framework server.

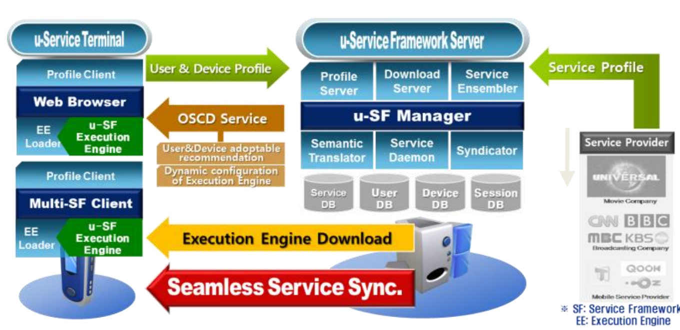

Figure 4. Seamless service synchronization in u-service framework

III. USER & DEVICE ADAPTABLE SERVICE MANAGEMENT

We suggest the use and device adaptable service managing mechanism via data synchronization. The syndicator is located on middleware and it provides seamless service in ubiquitous computer environment by supporting data synchronization between terminals and users. Existing data synchronization service has some inefficient problems such as data duplications on terminals and low performance of synchronization. We propose an adaptive data synchronization service mechanism in order to solve these problems. The adaptive data synchronization service does processing of synchronization with part of data used frequently by user not all of data and it manages data list in separate. So, it increases the performance of synchronization. Fig. 5 shows functional components of the syndicator, adaptive data synchronization block and a syndicator manager.

The Syndicator manager maintains information of users and devices, which are independent with a synchronization target. It also communicates with each service synchronization terminal.

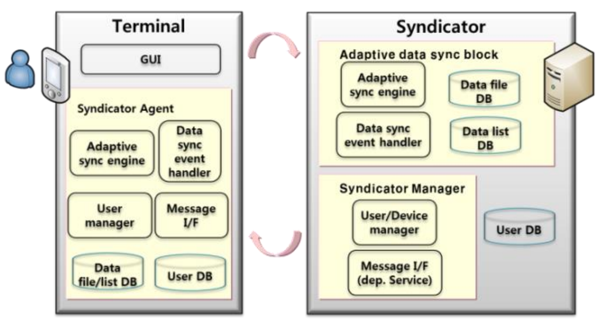

Figure 5. An architecture of the syndicator

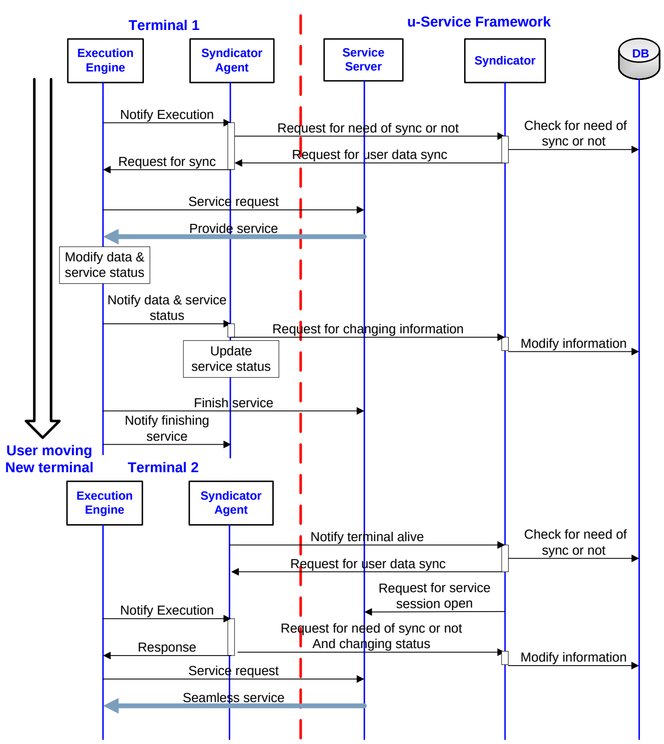

Figure 6. The procedure of seamless service synchronization

The adaptive data synchronization block supporting a data synchronization function composes of a data sync event handler and an adaptive sync engine. The data event handler provides data synchronization based on data update event, while the adaptive sync engine generates adaptive synchronization files based on terminal's characteristics. The adaptive sync engine supports data synchronization with data file DB, data catalogue DB, and user information DB. The followings are functions in order to provide data synchronization.

- Synchronization recent file and data list according to the events of a terminal
- Generation of an adaptive synchronization file list according to the characteristics of a terminal
- Remote synchronization with a terminal using service

First and the last functions are executed by the data sync event handler while second one is executed by the adaptive sync engine. Fig. 6 shows the procedure for supporting service synchronization seamlessly using the syndicator of a u-service framework server and the syndicator agent of a terminal.

IV. A SCENARIO FOR SUPPORTING SERVICE SYNCHRONIZATION

We composed a scenario as shown Fig. 7 for supporting service synchronization such as managing service status continuously and providing the data synchronization. The syndicator of the u-service framework is implemented using Java and GNU C++ developing languages. Three kinds of terminals like Android, Windows 7, and openSUSE11 were used to test service synchronization seamlessly by switching user's terminals. The scenario is as follows: User-1 is talking with video conference server using a smartphone on his way office. When he arrived at his office, he wanted to keep video conference using his laptop. So, he moves his call from the smartphone to the laptop, the session between the two devices (the smartphone and the video conference server) is closed.

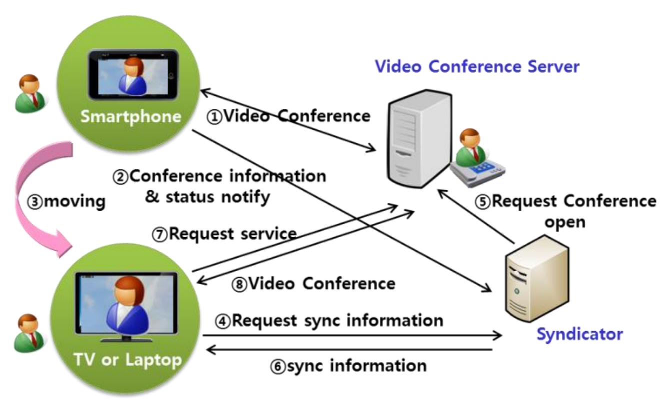

Figure 7. A video conference service scenario for supporting service synchronization

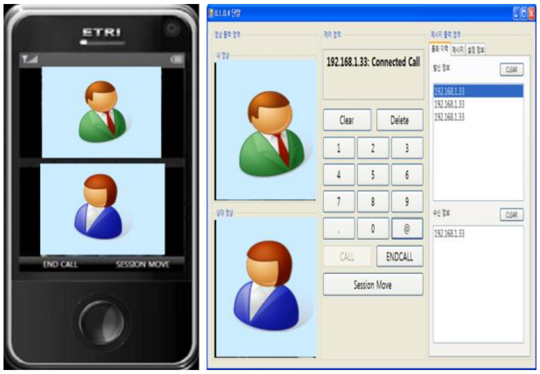

Figure 8. Video conference program capture screens of the smartphone & the laptop

A new link is established between User-1's new device, the laptop and the video conference server. After completing the new connection process between two devices, User-1 can seamlessly talk with the video conference with his new device, the laptop. This service synchronization was provided by the syndicator. The syndicator of the u-service framework supports data synchronization between the smartphone and the laptop and manages video conference information and status continuously. It also asks the video conference server to open the conference with the new device and notifies the synchronization information to the new device. Fig. 8 shows the actual demonstration environment for the scenario in Fig. 8. The video conference execution engine program can detect the device's camera and display the owner in the upper side of the displayer and the other in the lower side as shown in Fig. 8. If the device has no displayer, then only voice can be transferred between users. The left side of Fig. 8 illustrates the program for mobile devices such as smartphone, PMP, PDA, and so on, while the right one is the execution engine program on windows 7. Using the "Session Move" button in Fig. 8 users can change their devices during a call. If they press the "Session Move" button, the agent of syndicator in device notifies the conference information to the syndicator. So, the user moves one domain to another, he can be on the continuous phone with the other using his device that he has at that time. Using the proposed video conference service, users can use video conference services anywhere, any devices and any network.

V. Conclusion

In ubiquitous computing environment, users move from one domain to another with many kinds of devices usable in each domain. In that case, users want to use services seamlessly irrespective of their location and devices they have. The technique necessary in this case is called service mobility. This paper proposed a ubiquitous service framework that supports convergence services including heterogeneous service platforms and devices with their own independent platforms. It plays the role of an infrastructure to execute content and application programs with a dynamic configuration using mechanisms such as user preference learning, service and execution engine profiling, and real-time device profiling. It also supports service mobility to provide continuity and service synchronization when terminals or networks are changed due to user's movement or a change of personal preference. Existing data synchronization service has some inefficient problems such as data duplications on terminals and low performance of synchronization. But our proposal adaptive data synchronization service mechanism solved these problems. It processed synchronization with part of data used frequently by user, not all of data and it manages data list in separate. So, it increases the performance of synchronization

We implemented a prototype service framework to verify continuity and synchronization of service. We describe a scenario for mobility of video conference service in ubiquitous computing environment where a user moves from one domain to another with being provided seamless service through his device. Also, we design and implement the model for video telephony mobility and its clients. Using our scheme, users can use video telephony anywhere, any devices and any network. We showed not only service mobility of one user's migration but also of multiple users' sharing one single session among them. In order to minimize the delay time for seamless service mobility [9], a further study such as a reliable prediction about the movable target terminal through the analysis of user's context information is needed.

Acknowledgment

This work was supported by the IT R&D Standardization program of MKE/KATS, [2011-PM10-02, Development of Smart Utility based Green WPAN Standardization]

Reference

[1] E. Lavinal, N. Simoni, M. Song, and B. Mathieu, "A Next-Generation Service Overlay Architecure," Annals of Telecommunications, vol. 64, no. 3-4, Apr. 2009, pp. 175-185.
[2] P. Maniatis, et al., "The Mobile People Architecture," Mobile Computing and Communications Review, vol. 1, no. 2, July 1999, pp. 36–42.
[3] H. J. Wang, et al., "ICEBERG: An Internet-core Network Architecture for Integrated Communication," IEEE Personal Communications, Aug. 2000, pp. 10–19.
[4] H. Si, Y. Wang, J. Yuan, and X. Shan, "A Framework and Prototype for Service Mobility," 2009 World Congress on Computer Science and Information Engineering, Mar. 2009, pp. 315-319.
[5] Z. Chen, C. Lin, and X. Wei, "Enabling On-Demand Internet Video Streaming Services to Multi-terminal Users in large scale," IEEE Transactions on Consumer Electronics, vol. 55, no. 4, Nov. 2009, pp. 1988-1996.
[6] M. Hasegawa, H. Morikawa, M. Inoue, U. Bandare, H. Murakami, and K. Mahmud, "Cross-device handover using the service mobility proxy," in Proceedings of 6th International Symposium on Wireless Personal Multimedia Comunications (WPMC2003), vol. 2, Yokosuka, Japan, October 2003, pp. 357–361.
[7] A. D. Stefano and C. Santoro, "NetChaser: Agent Support for Personal Mobility," IEEE Internet Computing, Mar. 2000, pp. 74–79.
[8] B. Thai, R. Wan, and A. Seneviratne, "Integrated Personal Mobility Architecture: a Complete Personal Mobility Solution," ACM Mobile Networks and Applications, Feb. 2003, pp. 27–36.
[9] Y. Cui, K. Nahrstedt, and D. Xu, "Seamless User-Level Handoff in Ubiquitous Multimedia Service Deliver," Multimedia Tools and Applications, vol. 22, no. 2, 2004, pp. 137–170.