

Design and Implementation of Semantic Translation Engine In Ubiquitous Computing Environment

Hyun Jeong Lee, Jung-Sik Sung, Youngsik Chung, Euihyun Paik

Social Media Service Research Team, Green Computing Research Department

IT Convergence Technology Research Laboratory, ETRI

138 Gajeongno, Yuseong-gu, Daejeon, 305-700 KOREA

{ hjee294, jssung, yschung, ehpaik }@etri.re.kr

Abstract — In this paper, we design and implement semantic translation engine to provide continuous among several devices in ubiquitous computing environment. We use many kinds of devices and services according to the location, so it is difficult to provide seamless services among the devices. The proposed engine performs transform of protocols, frame rates, bit rates, and so on. Using our scheme, users can be provided a lot of services continuously among devices in ubiquitous computing environment.

Keywords — Semantic translator, Ubiquitous computing, One Source Multi Use.

1. Introduction

Ubiquitous computing aims to “enhance computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user” [1]. The idea of ubiquitous computing is that people’s life can be more convenient through using many invisible computers in ubiquitous computing environment. For example, people are able to be provided continuous service among many devices using the engine. The engine performs translation of frame rates, protocols, bit rates, and so on to accommodate various services in heterogeneous service domains through a standardized service execution environment without any modifications to original services.

We develop semantic translation engine and test it through real time camera AP (Access Point) monitoring service. The purpose of semantic translation engine is to provide continuous service to the users and reuse many services to several devices and domains. First, the engine connects to the camera and receives video data from it. And the camera clients connect to the engine and request real time camera data. Then the engine decides whether semantic translation is needed and it supports the requests. If translation is needed and it’s possible, the engine performs semantic translation, such as frame rates, bit rates, protocols, etc. to support the device profiles. Finally, it sends the translated camera data to client devices according to each device’s characteristics.

In this paper, we describe issues in semantic translation engine for the purpose of providing continuous service among devices irrespective of users’ location and devices. Also we propose system architecture and service flows for

these purposes, where the architecture consists of camera AP, semantic translation engine, and camera client devices. Using our mechanism, users can easily utilize continuous services among devices, and service developers can reuse one source code for several devices. Namely, one source multi use can be possible in ubiquitous computing environment.

The structure of this paper is organized as follows. In Section 2, we introduce three related projects for seamless service. We describe a scenario and issues for semantic translation for the purpose of maximizing users’ convenience of service use in Section 3. Also, we propose system architecture and a sequence flow for this purpose in Section 4, and conclude this paper in Section 5.

2. Related works

We introduce three related projects, such as Daidalos (Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimized personal Services), SPICE (Service Platform for Innovative Communication Environment), ARTEMIS (Advanced Research & Technology for EMbedded Intelligence and Systems), which provide users continuous services.

Daidalos project is an EU Framework Program 6 Integrated Project, and its vision is to seamlessly integrate heterogeneous network technologies that allow network operators and service providers to offer advanced and profitable services, giving users access to a wide range of personalized voice, data, and multimedia services [3]. Daidalos will lead the fundamental rethinking of network architectures that is necessary to create a new generation of user-centered manageable communication infrastructure for the future.

SPICE project addresses the unsolved problem of designing, developing and putting into operation efficient and innovative mobile Service creation/execution platforms for networks beyond 3G (the third generation of developments in wireless technology) [4]. Its platform will support multiple heterogeneous execution platforms allowing for new and innovative services to be spread across different operator domains and over different countries for realizing a variety of business models. For users, operators and service providers, the SPICE project will turn today’s confusing heterogeneity into an easily manageable and rich service environment by

exploiting the diversity of device capabilities and fostering service adoption. The SPICE approach will broaden business opportunities in the communications and associated business sectors. To achieve this vision, the SPICE consortium integrates the competence and knowledge of leading European telecom operators/service providers and main IT and telecommunications suppliers.

The vision of ARTEMIS is that embedded systems will realize the dream of ambient intelligence, where intelligent support for people will be embedded in everyday objects. Large-scale application will improve our quality of life. The result will be to help make life healthier and more secure. And to provide more comfort for Europe's ageing population. ARTEMISIA nurtures the ambition to strengthen Europe's position in embedded intelligence and systems and to attain world-class leadership [5].

As we describe above, providing continuous service to users aims for maximizing their convenience and the utilization of the circumference resources. We suggest semantic translation engine and its architecture for the purpose of realizing these advantages of ubiquitous computing environment.

3. Scenario of semantic translation based on device profile

In this paper, we describe a scenario for semantic translation engine based on device profile for the purpose of maximizing the capabilities of them and user's much convenience in the ubiquitous computing environment.

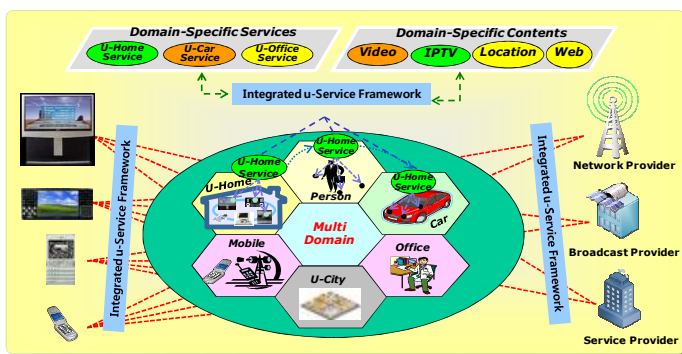


Figure 1. A conceptual model of the ubiquitous computing environment

Figure 1 shows the conceptual model of the 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 TV (Television), computer, PMP (Portable Multimedia Player), PDA (personal digital assistant), and mobile 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. For example, a user

watches a TV program in a home, but it isn't easy to do when he (or she) is in the street with a PMP. But, using our semantic translation engine, a user can watch a TV program continuously in any place with any device, since the engine transforms the data and context properly to the device.

In this paper, we consider real time camera AP monitoring service from one device to another. In the home or office, we can watch the camera data with PC or laptop based on Windows XP. When we move to hot-spot zone, we can watch the data with the PDA, PMP, mobile phone which are based on Windows CE/Mobile. Using this method, users can take care of the children, the old and the weak, and can monitor their office and jewelry shop more convenient in the ubiquitous computing environment. Also, service developers can reuse their service for many devices.

4. Design and Implementation of Semantic Translation Engine

4.1 TECHNICAL ISSUES

We use camera AP to provide real-time camera streaming service to users. The scenario is as follows: A man watches his jewelry shop with his PC at home. Then he goes to his shop with watching it using the displayer in the car or mobile phone. In the shop, he can watch his home using the laptop for his parents and children. This scenario requires many techniques, such as several devices, adequate camera clients (displayers) for the devices, networked camera.



Figure 2. User interface of semantic translation engine

Figure 2 shows user interface of semantic translation engine, which is developed using Java program language in Eclipse development kit, so it can be on the Windows and Linux platform. The engine relays or translates semantics of real time video data from camera to several devices of users. Camera AP List in left upper shows the list of camera AP which is found by the "SEARCH CAMERA" button below. "PREVIEW" button performs preview function of camera video data using video display program, such as ffplay in Linux. "TRANSMIT" button performs transmit video data to clients which connects to the semantic translation engine. Video Configure in the center performs translation video data from camera AP to the adequate format to the client. The elements for translation are video rate, bit rate, resolution,

encoding, frame rate, and platform. “Service Device Info” in the right of the figure displays information of devices which has camera client. “Result Console” displays the result after each button is clicked and log data of console.

4.2 ARCHITECTURE FOR SEMANTIC TRANSLATION

We describe an architecture for semantic translation in this section. It consists of semantic translation engine, camera AP server, and camera viewers. Semantic translation engine can perform translation of protocols from TCP/UDP to RTSP ((Real Time Streaming Protocol)), codec from mpeg 4 to jpeg, and can set video parameters such as rate control(VBR/CBR), bit rate according to the VBR/CBR, resolution with CIF and D1, encoding option with I reference and general mode, frame rate from 2 to 30, and control platform such as Windows and Linux. When camera clients request camera video data, semantic translation engine compares the required entity for devices with supporting entity of the engine, which are in device profiles. The profiles are predefined and based on W3C CC/PP (World Wide Web Consortium Composite Capabilities/Preference Profiles). If the engine provides the supported entity, then it performs transform the data and sends to the camera clients which are based on windows, android, and mobile devices. The camera clients can connect and display 4 camera data channel.

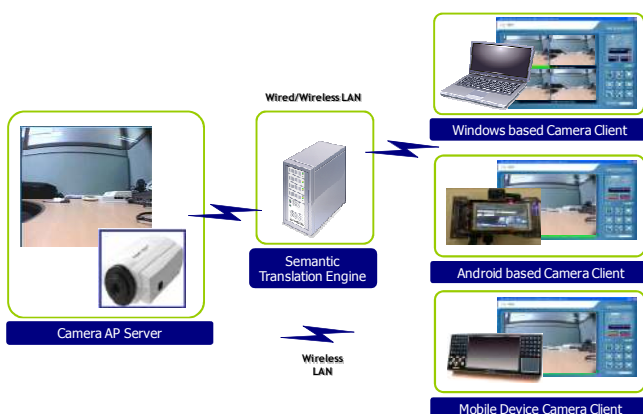


Figure 3. Configuration of camera AP monitoring service

An architecture of semantic translation engine is illustrated in Figure 4. It is developed using Java based on Eclipse, so it can be operated on the Windows and Linux platform. Semantic translation engine reads and interprets device profile written in XML, and provides adequate service to the devices for the service users. There are 4 blocks to perform the translation, such as interface, XML (Extensible Markup Language) parser, protocol translator, codec translator. First, interface block manages the sockets to communicate with user devices, camera AP, device profile, and operates first in the engine. After the engine connects with camera AP, it waits the connection request from clients. When a client connects to the engine and request the camera data, the engine reads and parse the predefined device profile in XML parse

block. In the parser block, we use DOM parser for java, because the device profiles are text document. If the engine supports the parsed function, it calls the translation blocks, such as protocol and codec. Protocol blocks transforms the camera data from TCP/UDP to RTSP, and codec blocks does the adequate codec to the client device. If the device doesn't need the transformation, then the original camera data is sent to the client.

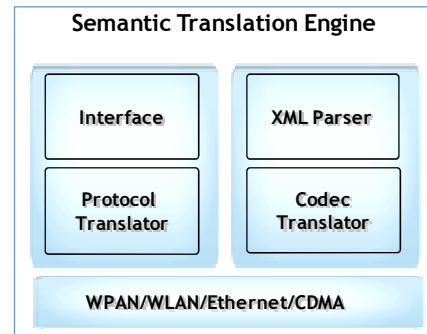


Figure 4. Architecture of Semantic Translation Engine

4.3 SERVICE FLOWS

We describes the flow chart of camera AP monitoring service as shown in Figure 5. First, semantic translation engine searches and requests connection with cameraAP. After camera AP responses the request, the engine requests video data to the camera AP. Then, the engine connects with several clients, and sends video data to them. The video data may be transformed before sending according to the device profile. If the client requests video data using RTSP, then the engine sends video data through RTSP streaming service. Or, if the clients needs to codec translation, then the semantic translation performs the translation according to the device profile.

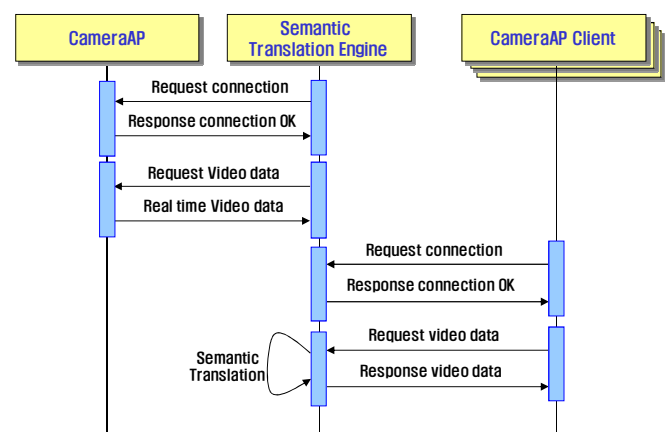


Figure 5. Flow of camera AP monitoring service

5. Conclusions and future works

The purpose of semantic translation is to provide seamless service to the users and to reuse many services to several devices for service developers.

In this paper, we describe the concept of the semantic translation engine for ubiquitous computing environment, which a user moves from one domain to another with being provided seamless service through his (or her) device. Also we implement semantic translation engine for this service. Using our scheme, users can use continuous services with any devices.

We will define general service APIs for service developers and add functions to make execution engine automatically for many devices.

6. Acknowledgement

This work was supported by the IT R&D program of MKE/KEIT, [2008-S036-01, Development of Integrated u-Service Framework for One-Service-Cross-Domain].

References

- [1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," Commun. ACM, Vol. 36, no. 7, pp. 75–84, July 1993.
- [2] Hyung J. Jang, Jung K. Uhm, Sang H. Lee, Joon S. Lim, Design and Implement of Ubiquitous Healthcare System Using Portable Hemadynamometer, The 3rd Asian Fuzzy Systems Symposium Tutorial, vol.2, no. 1, 2007, pp.652–656.
- [3] Daidalos project, <http://www.ist-daidalos.org/>
- [4] SPICE project, <http://www.ist-spice.org/>
- [5] ARTEMIS project, https://www.artemis-association.org/about_artemis