Ubiquitous Cloud: Managing Service Resources for Adaptive Ubiquitous Computing

Koichi Egami[†1] Shinsuke Matsumoto[†1] Masahide Nakamura[†1]

To facilitate the management of ubiquitous service resources, this paper presents a novel platform called *ubiquitous cloud*, borrowing the concept of the cloud computing. The ubiquitous cloud supports various stakeholders to use appropriate ubiquitous objects in infrastructure, platform and application levels. We present an architecture consisting of four key components: the service resource registry, the adaptive resource finder, the context manager and the service concierge.

1. SOA for Ubiquitous Services

Ubiquitous services and applications are required to be more *adaptive* to users and surrounding environments[1)]. In reality, however, it was quite difficult to implement practical adaptive ubiquitous services, since there was no standard platform to guarantee the interoperability among ubiquitous resources (devices, sensors, interfaces, etc).

The concept of the *Service-Oriented Architecture* (SOA) is well applied to ubiquitous/pervasive environment to achieve the programmatic interoperability among heterogeneous and distributed devices. Wrapping proprietary operations and protocols by Web services provides loose-coupling and platform-independent access methods for external software that uses the devices[2)3)].

As more and more ubiquitous devices become available as SOA services, the next challenge is how to manage such service resources efficiently and effectively. Our key idea is to borrow the idea of the *cloud computing*, regarding everything as a service (XaaS).

For the various ubiquitous resources deployed in the Web, different requirements will be expected from different *stakeholders*. The hierarchy of requirements fits well the structure of IaaS, PaaS and SaaS in the cloud computing. We propose an architecture of *ubiquitous cloud*, to manage service resources for adaptive ubiquitous services.

†1 神戸大学大学院
Graduate School of System Informatics, Kobe University

2. Stakeholders

The ubiquitous cloud is designed to deal with service resource managements at different levels of abstractions for the following stakeholders:

Ubiquitous Resource Providers (URP): People who want to provide own ubiquitous resources in the service-oriented ubiquitous network.

Custom Applications Developers (CAD): People who want to develop their own applications by using available service resources as they are.

Adaptive Services Developers (ASD): People who want to develop large-scale adaptive services that dynamically use service resources based on given requirements and contexts.

Adaptive Service Providers (ASP): People who want to provide adaptive applications as commercial or non-commercial services for end users.

End Users (EU): People who want to use the published adaptive services as tailor-made services.

3. Architecture of Ubiquitous Cloud

The ubiquitous cloud adopts a three-layered architecture, where each layer respectively corresponds to the IaaS, PaaS and SaaS of the conventional cloud, as shown in Figure 1.

Infrastructure Layer: Provides fundamental features to publish and discover ubiquitous objects as service resources. The infrastructure layer is supposed to be used by URP and CAD.

Platform Layer: Provides development platforms to facilitate development of adaptive ubiquitous services. The platform layer is supposed to

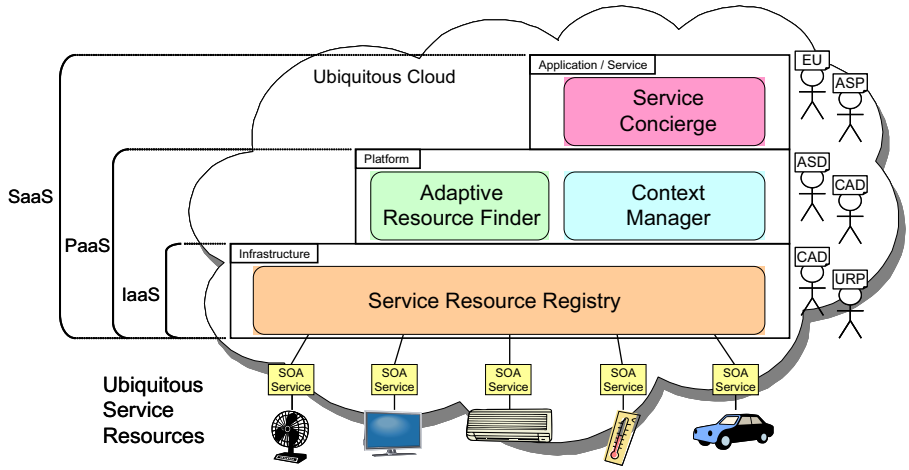

図 1 Ubiquitous Cloud for Adaptive Ubiquitous Services

be used by CAD as well as ASD.

Application/Service Layer: Provides showcases where adaptive services are deployed and provided. The Application/Service layer is supposed to be used by ASP and EU.

4. Four Key Components

In the three-layered architecture, we are currently developing the following four key components.

4.1 Service Resource Registry

Situated in the infrastructure layer, the service resource registry stores meta-data explaining service resources that are registered by the URPs. It also supports the CADs (as well the ASDs) to identify service resources and operations by various attributes: device class, operation type, physical location, purpose, users, etc.

4.2 Adaptive Resource Finder

Situated in the platform layer, the adaptive resource finder recommends most appropriate service operations based on given requirements as well as the current contexts. It understands the requirements, and transforms the requirements into concrete queries for the service resource registry. The results of the queries are filtered by the contexts gathered by the context manager (explained below). Thus, the adaptive resource finder works as a powerful tool for the ASDs and the CADs to implement the adaptive services.

4.3 Context Manager

Situated in the platform layer, it gathers various contextual information from real world, using the underlying sensor services. The gathered information is interpreted as explicit contexts characterizing user's status, requirements, tastes, environmental situation. The contexts are extensively used by the adaptive service applications as well as the adaptive resource finder. Thus, the context manager supports the ASDs and the CADs to obtain contextual information easily.

4.4 Service Concierge

Situated in the application/service layer, the service concierge provides a showcase for the developed applications and services, which are deployed by the ASP. It also recommends and executes published adaptive applications as tailor-made services of the EUs. Thus, the EU can use the services the services without concerning whatever service resources are used. Based on user's request, the behaviors of the service are dynamically determined.

参 考 文 献

1) K. Carey, D. Lewis, S. Higel, and V. Wade, "Adaptive Composite Service Plans for Ubiquitous Computing," *Workshop on Managing Ubiquitous Communications and Services (MUCS)*, 2004.

2) M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. Matsumoto, "Constructing Home Network Systems and Integrated Services Using Legacy Home Appliances and Web Services," *International Journal of Web Services Research*, vol.5, no.1, pp. 82–98, 2008.

3) J. King, R. Bose, H. I .Yang, S. Pickles, and A. Helal, "Atlas: A Service-oriented Sensor Platform Hardware and Middleware to Enable Programmable Pervasive Spaces," *Proc. on Local Computer Networks (LCN)*, pp. 630–638, 2006.