M-43 Service state description for seamless service environment

Norihiro KAWASAKI† Hiroyuki KASAI† Yoko KIKUTA† Kenichi YAMAZAKI†

1. Introduction

Since many access methods and terminals will be used in the near future, the demand for a seamless service environment can only increase. The seamless service environment is a networking/computing environment that provides the capability of transferring services between all types of terminals. For example, a movie service first accessed on a PC can be transferred to a high-speed mobile phone without interruption in this environment.

We have already proposed the basic technology and described a prototype system for service transfer [1]. However the system has a strong limitation, that is, it merely can transfer a specific application. In this paper, we propose a service state description technology to transfer more general services.

2. Service transfer

Figure 1 shows an example of service transfer. In Figure 1, the user has a TV meeting in his company. When he has to go out, he transfers the service to continue the meeting via cellular phone. In addition, in another environment (e.g. in a bus), the content is adapted to suite the environment.

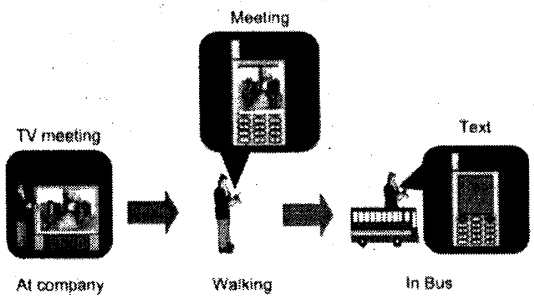

Figure 1 Example of service transfer.

Our current target services for service transfer are non-realtime communication services such as web browsing services and streaming services. Realtime communication services such as voice communication are not our current target. (We think these services can be supported by extending our system.) Since stand alone applications can be considered as a subset of non-realtime communication services, our system can transfer them in the same manner.

We have already proposed the basic technology and introduced a prototype system for service transfer. The system transfers a streaming service between different terminals using a set of information about the suspended service. We call it the "service state". (The service state of this system was content URL and elapsed time of the streaming media.) The service transfer process consists of suspending a service, transfer of the service state, and resumption of the service. Figure 2 illustrates this process.

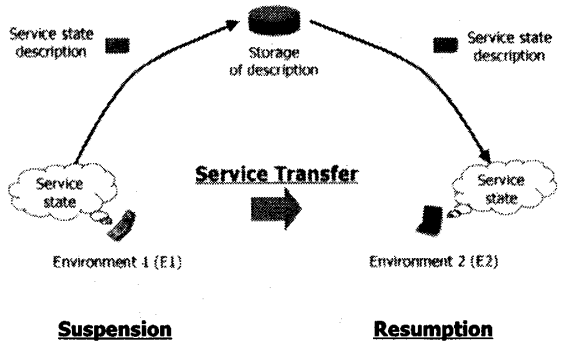

Figure 2 Transfer of service state.

Since the service state of this system is represented in an ad hoc manner, the applicable area is restricted.

3. Service state and the issues involved

To extend the target service, we have to redesign the service state description. The main issue of the service state is that the level of description abstraction creates a trade-off between reproducibility level and the variety of environments to which the service can be transferred.

A concrete description of service state enables us to resume exactly the same service as the original one in the same environment. However, if the new environment is slightly different from the original one, resumption might be harder. On the other hand, an abstract description provides larger generality in terms of supporting new environments, though it would be difficult to reproduce the original service precisely.

4. Proposed system

In order to solve these problems, we propose a method that describes service state in a low level of abstraction and realizes various flexibilities by intelligent processing.

Figure 3 shows the structure and basic steps of the proposed system.

Figure 3 shows an example in which the user has played 24min 30sec of the content http://hoge.com/neko.avi via the application "Real Player" in environment 1 (E1). In the following sections, we explain the important steps in detail using Figure 3.

† Network Laboratories, NTT DoCoMo, Inc.

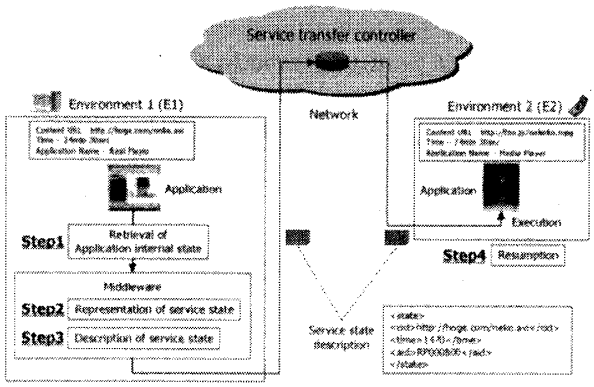

Figure 3 Proposed system (terminal based).

4.1 Suspension

◆ **Retrieval of application internal state (Step1)**

The system retrieves the internal states of the application i.e. the elements of service state. There are three retrieval methods. If we define APIs for information retrieval, newly developed applications can be written to provide such APIs. The second method is to reuse existing APIs. A typical example is ActiveX in Microsoft Windows. Using the API, we can retrieve the elapsed time from Windows Media Player. We use this method in our prototype system. The last method is to provide some kind of proxy to determine the service state. This method requires no modification of the applications, however a new proxy should be developed for each new service.

◆ **Representation of service state (Step 2)**

Retrieved information is transformed based on the representation policy in order to create a description of service state. There are two policies: representation by value and representation by reference. The former includes the content itself in the service state. The latter indicates the location of the content in the service state.

Here, in the streaming service, the content URL is used as the reference of content.

◆ **Description of service state (Step 3)**

In this step, the system creates a concrete description of service state aiming at high reproducibility. Three approaches are possible in describing service state: (1) use an original markup language, (2) use standard markup language such as CC/PP, SMIL, and so on, and (3) use our own markup language based on DTD or XML Schema. Approach (1) is not general-purpose, and approach (2) is not enough to describe service states. We choose approach (3) because it can describe the service state and its description is readable by humans, and it is available in various environments. An example is given below,

```
<state>
<cid>http://hoge.com/neko.avi</cid> <!-- content ID -->
<time>1470</time>                   <!-- elapsed time -->
<aid>RP000800</aid>                 <!-- application ID -->
</state>.
```

To achieve high reproducibility, the system has sufficient capability to represent all information. Table 1 shows all elements of service state information.

Table 1 Elements of service state information.

Information category	Typical Example
Service Information	Content (Contents URL, elapsed time), History (Event, URL)
Layout Information	Desktop state (window location, size)
Environment Information	Device, Application, Preferences, Communication, AAA, Time

4.2 Resumption (Step 4)

After receiving and analyzing the service state description, the system selects a suitable application from among the application candidates in order to support environment 2 (E2). If there is the same application as the application used in E1 (e.g. "Real Player"), it is used. Otherwise, an alternative application is selected automatically. In this case, the system selects "Media Player" as a substitute, because E2 has "Media Player" but not "Real Player". By using the information (content URL and elapsed time) collected from the application in E1, the system can resume streaming service in E2. The user plays content "http://foo.jp/nekoko.mpg" from the 24min 30sec point by executing "Media Player" in E2.

Thus, since applications are selected in the new environment and service state is interpreted adaptively, our system can resume various service states in various environments and hence achieves generality.

5. Conclusions and Future works

In this paper, we have proposed a service state description for the seamless service environment. The full and concrete description of service state and the intelligent interpretation of this information in the resumption step achieve high reproducibility and generality at the same time.

We intend to carry out a series of seamless experiments to demonstrate the effectiveness of the proposed system.

References

[1] Yoko KIKUTA, Kenji SAKAMOTO, Kenichi YAMAZAKI, "Resource-aware seamless service architecture," NS, IEICE, 2001.