A Service Discovery Framework for Ubiquitous Computing*

Xiaoguang Gu, Hongzhou Shi, Jian Ye, Zhenmin Zhu

Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
xggu@ict.ac.cn

Abstract

This paper presents a novel service discovery framework for ubiquitous computing called Ubiquitous Computing Service Discovery Framework (UCSDF). UCSDF offers a more flexible and scalable architecture which can combine the local services with remote services. That characteristic makes UCSDF different from many other service discovery frameworks and more adaptive to ubiquitous computing environments.

1. Introduction

Ubiquitous computing[1] makes the actual computing and communication essentially transparent to the users[2]. Supporting devices and services to properly discover, configure, and communicate with each other is essential to achieving such sophistication[3]. Service discovery is the technology for that aim.

Most of existing service discovery frameworks emphasize on either local ambient environment or wide area scope. But combining local services with remote services can provide users with more powerful functions. Jini[4] and SLP[5] have partial similar character, but both of them have some limitations. In Jini, user needs prior knowledge about address of remote administrative domain when requesting a service which is not in local domain. In SLP, because the method for DA[5] discovery is based on DHCP and multicast, SLP can not discovery remote services outside of local scope. This paper offers a more flexible and scalable architecture than Jini and SLP to extend the scope of service discovery from local to remote. And a novel service query strategy is presented to meet the new requirements.

2. Service Discovery Infrastructure

The service discovery infrastructure of UCSDF is shown in Figure 1. Global Manage Server (GMS) and Local Manage Server (LMS) are novel characteristic different from other service discovery frameworks. GMS is set on WAN and maintains a mapping table from a service to a set of LMSs. LMS is set in a LAN and maintains a mapping table from a service to a set of SDAs (Service Dictionary Agent) or service providers. SDA is set in a sub network of a LAN and maintains a mapping table from a service to a set of service providers.
sub-network or LAN or WAN. So the infrastructure of UCSDF is more flexible and scalable than any other existing service frameworks.

3. A Novel Service Query Strategy

The query strategy is two-phased. In the first phase, SDA, LMS and GMS perform keyword based query. In the second phase, filtered service providers do complex accurate match. As a result, on the one hand, the search space becomes smaller in the first phase; on the other hand, the complex query becomes distributed and parallel in the second phase. This strategy is efficient when the query scope is extended from the local domain to the large-scale remote domains.

Table 1 Format of Request

<table>
<thead>
<tr>
<th>Key Word</th>
<th>Attributes</th>
<th>Requester ID</th>
<th>Response Address</th>
<th>Expired Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Algorithm for Service Requester

Begin
1. Create a Request;
2. Ret = Find Key Word in Cache;
3. If Ret == Success Then
   Return Service Address;
   Else
   Multicast the Request to Service Provider and Unicast the Request to Local SDA;
4. Wait for Response Until Overtime;
5. Process Response to Return an Optimization or NULL;
End

Table 3 Algorithm for SDA

Begin
1. Receive a Request;
2. Ret = Find Key Word in Mapping Table;
3. If Ret == Success Then
   If the Request is from LMS Then
     Modify the Response Address by Own Address and Expired Time;
     Send the Request to Service Providers Belong to the Key Word;
     Wait for Response Until Overtime;
     Process Response and Send to LMS;
   Else
     Send the Request to Multi Service Providers Belong to the Key Word;
   Else
     If the Request is from Service Requester Then
     Modify the Response Address and Expired Time;
     Send the Request to LMS;
     Wait for Response Until Overtime;
     Process Response and Send to Service Requester;
End

Table 4 Algorithm for Service Provider

Begin
1. Receive a Request;
2. Ret = Match the Service Attributes of Request with Own

Table 5 Algorithm for LMS

Begin
1. Receive a Request;
2. Ret = Find Key Word in Mapping Table;
3. If Ret == Success Then
   Modify the Response Address by Own Address and Expired Time;
   Send the Request to SDAs and Service Providers Belong to the Key Word;
   Wait for Response Until Overtime;
   Process Response and Send to Requester;
   Else
   If the Request is from SDA or Service Provider Then
   Modify the Response Address and Expired Time;
   Send the Request to LMSs Belong to the Key Word;
   Wait for Response Until Overtime;
   Process Response and Send to Requester(LMS);
End

Table 6 Algorithm for GMS

Begin
1. Receive a Request;
2. Ret = Find Key Word in Mapping Table;
3. If Ret == Success Then
   Modify the Response Address by Own Address and Expired Time;
   Send the Request to LMSs Belong to the Key Word;
   Wait for Response Until Overtime;
   Process Response and Send to Requester;
End

References


