Integrating Service Discovery Protocols with Presence-based Communications for Ad hoc Collaborative Scenarios

Ramiro Liscano
School for Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, K1N 6N5
rliscanoieee.org

Amir Ghavam
School for Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, K1N 6N5
ghavam@ieee.org

Michel Barbeau
School of Computer Science, Carleton University, Ottawa ON, Canada, K1S 5B6
barbeau@scs.carleton.ca

Abstract

This paper presents an approach for the integration of service discovery protocols with presence-based communication services. The goal is to be able to facilitate communications among 3rd party services in an ad hoc manner in particular for collaboration scenarios that occur in meeting rooms, face to face, or across enterprises. The presence protocol SIP/SIMPLE is extended to support descriptions of services for existing common service discovery formats like the IETF Service Location Protocol (SLP).

1 Introduction

Modern communication features have been transformed radically in the past 15 years with the advent of affordable wireless communications, the adoption of the Internet as a communication network, and the rapid adoption of presence-based Instant Messaging (IM) communication services.

With the advent of wireless networking technology like the IEEE 802.11 WLAN, Bluetooth WPAN, and IEEE 802.15 WPAN specifications, it has become easier to establish ad hoc connections to existing networks and therefore network resources. These technologies support the capability to have effective ad hoc networking among users but concentrate primarily on the lower networking layers and fail to offer a simple more dynamic access to network services and resources. The exception here is the Bluetooth protocol that from its inception has supported service discovery at the application layer and conformance to a set of application profiles among vendors.

Current service discovery protocols are not by themselves sufficient enough to facilitate the spontaneous sharing of services. There are several key factors that affect this. The first is that the service discovery procedure can be difficult, error prone, and lengthy. It can be difficult because at times one may not be aware of how the service query needs to be constructed. It can be error prone because one may not get back the "proper" service. By "proper" we mean a service that the user desired but did not actually get. It is generally a lengthy process in that several messages need to be exchanged between the client and the service directory before the client finally gets the interface to the service. For example, for a Bluetooth service request it can take up to several seconds to get a list of services from a device.

In this paper, we present a framework to enable and manage the automatic and pervasive access to a set of services among collaborating users. To do this, our design integrates the following technologies: a presence service as the basis infrastructure to connect users and services that share a common context with service discovery application profiles like IETF SLP

2 Presence Services

A presence service allows users of the service to "Subscribe" to another person's availability. Users that view another person's availability are called "Watchers". The user that projects their availability is called a "Presentity". This is in conformance with the definitions used the IETF RFC 2778 on the Common Profile for Instant Messaging [4]. A Presence service should offer two types of information:

- Projection Of Availability. This is the traditional user's availability information. It is an indication of the person's desire or willingness for immediate communication. The availability information is projected to other users. If people are willing to communicate they will appear available, otherwise they appear unavailable. The foremost contender as an "open" protocol standard for this is the SIP/SIMPLE protocol. Ironically it is not the protocol that is widely used due to the popularity of proprietary IM services like Microsoft Messenger, Yahoo Messenger, and AOL Messenger, but its "open" interfaces makes it the one contender for interoperability in this domain.
- Communication Contact Information. This second type of information reflects how the person is available for immediate communication. The contact information

Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04)
0-7695-2096-0/04 $20.00 © 2004 IEEE
IEEE COMPUTER SOCIETY

describes different media that the person is currently available on – the devices, software applications, etc. Examples of such would be – a telephone, an Instant Messaging via a certain provider, chat, video conferencing application etc.

There are 2 competing standards for presence information. This is the IETF Presence Information Data Format (PIDF) [5] and the IETF XMPP Instant Messaging [6] protocol supported heavily by the Jabber Consortium.

The PIDF standard defines a <tuple> element that carries a PRESENCE TUPLE, consisting of a mandatory <status> element, followed by any number of OPTIONAL extension elements (possibly from other namespaces), followed by an OPTIONAL <contact> element, followed by any number of OPTIONAL <note> elements, followed by an OPTIONAL <timestamp> element. An example of a PIDF <tuple> element is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
  <impp:presence
xmlns:impp="urn:ietf:params:xml:ns:pidf"
entity="pres:someone@example.com">
  <impp:tuple id="sg89ae">
     <impp:status>
       <impp:basic>open</impp:basic>
    </impp:status>
     <impp:contact priority="0.8">
        tel:+09012345678
     </impp:contact>
  </impp:tuple>
```

We see in this PIDF <tuple> element an availability status of "open" and contact information pertaining to a telephone number.

Presence technologies like SIP/SIMPLE [1][2] and Jabber[1] have concentrated on the modeling and projection of the availability of a user over the conventional communication protocols like voice, email, or IM and offer little insight into the projection of service information and interface data.

It is the contact information that is lacking sufficient information to be used effectively with other services. The PIDF draft simply states the <contact> element contains a URL of the contact address. It is left to the application to determine what protocol should be used to contact the user. This can be very ambiguous if a standard is not developed for the service specification.

The XMPP protocol is even worse since it simply states that for non-IM types of communications the VCARD specification will be used, but again there is no mention of a particular protocol specification for the contact data.

[1] http://www.jabber.org

3 Service Profiles

There are many forms of service profiles but the most generic are those used for service discovery. Of particular interest are the IETF Service Location Protocol (SLP). We chose SLP mostly because it is a simple protocol to implement.

The IETF SLP protocol [7] is one of the original service discovery protocols developed for the Internet. For this particular paper we shall describe in more detail its service profile. The SLP service scheme defines a service URL in the following manner:

```
service_url = "service:" service-type ":"
service-access-info
```

The service-type allows for the specification of service abstract types and particular recognized URL scheme names like http, ftp, telnet, etc. Some examples of this are "service:lpr:", "service:http:", "service:ftp:", "service:chat:". Service types can utilize a generic service name as well as the specific service name to help the user understand the URL. For example, the generic service type "printer" can be used to represented a series of protocols that relate to communicating with a printer. The service in this case can be specified using "service:printer:lpr".

The service-access-info describes how that service is to be accessed. In our particular case, this has information about how an actual user is to be contacted. Service access information consists of the following format:

```
service-access-info  = "/" address-family
'/" address-spec [ "/" [url-path] [";"
attribute-list]]
```

The address-family indicates the network layer. For example "//" indicates and IP address. The address specification is the address required to communicate with the presentity. A url-path is the most protocol-specific part of the URL, it specifies details of how to access the service and similar to the attributes it is an optional item. Attributes are specified in the following manner: attribute-id "=" value.

In SLP these services are defined using a service template and are registered with the Internet Assigned Numbers Authority (IANA) to represent the service media type.

4 Integration of S.D. Profiles into Presence

The link between these service specifications and the presence services is the modification of the contact information in the availability message. This is actually a relatively straightforward exercise since the SIMPLE protocol can be extended to accommodate for these changes.

IEEE COMPUTER SOCIETY

The simplest way to do this is to leverage the ability to define custom namespaces and XML schemas for a contact that supports and SLP format for services. The following is as example of how this new contact information would appear in as part of an impp message.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence
xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:imppslp="http://www.w3.org/2001/XMLSch
ema-instance"
xsi:noNamespaceSchemaLocation="impp-
slp.xsd">
entity="pres:someone@example.com">

  <impp:tuple id="sg89ae">
     <impp:status>
       <impp:basic>open</impp:basic>
     </impp:status>
     <imppslp:contact priority="0.8">
       <simppslp:slpURL version=1.0 dn=2100>
       service:http//webset2100@ABC.com
       </simppslp:slpURL>
     </imppslp:contact>
  </impp:tuple>
```

In this example the extensions that were defined to support the SLP service specification for contact information are defined in the `slpimpp` name space. These new entities are defined using an XML style sheet for the `slpimpp` namespace. The SLP representation of an SLP service is straight forward in that it is composed of an SLP URL name and any number of attributes. Below is the style sheet used for defining the new SLP style of contact entity.

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="contact"
   type="slp-contact" minOccurs="0"/>
   <xs:complexType name="slp-contact">
   <xs:sequence>
   <xs:element ref="slpURL"
     maxOccurs="unbounded" />
   </xs:sequence>
   <xs:attribute name="priority"
     type="qvalue"/>
   </xs:complexType>

     <xs:complexType name="slpURL">
       <xs:simpleContent>
        <xs:pattern value=
           "service:[A-Z]{*}"/>
        <xs:anyAttribute />
       </xs:simpleContent>
     </xs:complexType>
</xs:schema>
```

This rather simple addition to the contact information increases the potential of SIP SIMPLE to be able to project to users meaningful information about enterprise service s that are available for the particular context that the user is in with out the tedious search routine that is generally required when the service discovery protocol is used directly.

5 Ad hoc Collaborative Framework

This integration of SLP service profiles with presence services is a crucial component of a larger scale framework that defines a software entity known as a presence association. Some preliminary work and details of this framework have been previously presented [12] and we present an overview here focusing more on details related to the integration of service profiles with presence.

A presence association is a software entity that manages presence and resource access for a group of persons that share a common context. It can be viewed as a group manager that maintains a common set of presence and service access policies for its members. This presence association is the software entity that integrates the service discovery component with a presence service. This relationship is depicted in figure 1.

One of the first types of presence associations we envision is a location-based presence association. For example room associations. These associations would be persistent objects with the resources of that room already registered with a particular presence association. This is not an unreasonable assumption, since many resources in meeting rooms are generally stationary. On the other hand, mobile resources most often belong to users and need to be dynamically registered with the presence association.

Location-based presence associations will most likely be the primary association that users will indirectly interact with. They behave as a first entry point for users to a corporate network. All users in a particular location will be subjected to a set of policies for network and resource access. The best way to show this is with an example.

Let's take a meeting room example. A meeting room presence association needs to be created by a user (step 1, figure 1). The process of creating an association results in a particular URL being created for that association after which it is possible for all public services and users to register with the association (step 2, figure 1). The procedure can be partially automated by using location information from a location server and RFID tags to

IEEE COMPUTER SOCIETY

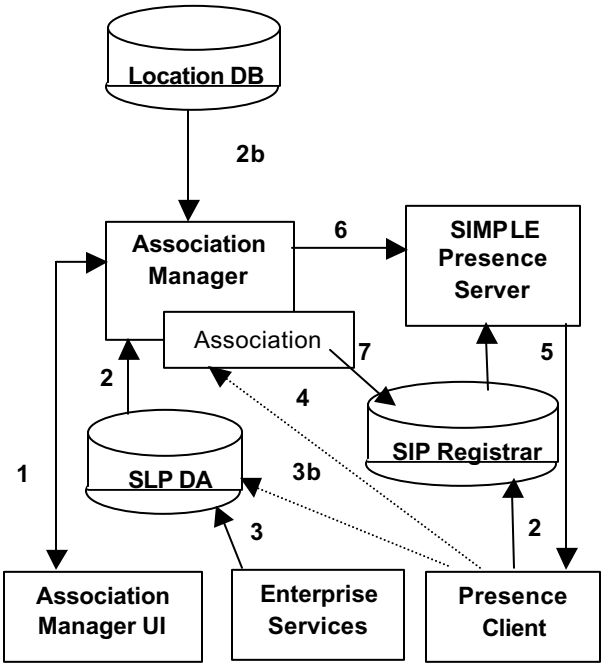

Figure 1. The enhanced presence server

correlate user and service IDs with the location of the meeting room (step 2b, figure 1).

It is also necessary to make the distinction between personal communication services that users have on their desktop to those public services that are available for a anyone to connect to. These public services are registered with an SLP DA (step 3, figure 1). while those private communication services are part of the SIP registration process (step 4, figure 1). This is also a distinguishing point between the services that are available using the conventional `impp:contact` information to those that are available using the `imppslp:contact` entity. Users may choose to also introduce into the association a public service that should be registered using the SLP DA (step 3b, figure 1).

These services along with any other communication services that users bring into the association are projected among members of the association using a presence service (step 5, figure 1). The association maintains a tight relationship with the presence server (step 6, figure 1). The association interacts with the presence service to create a presence group with the same name as the association and creates presence agents for each of the members of the association as well as the enterprise and public services. Since the public services do not have any users to register them with the presence service the association must do this for them (step 7, figure 1). The association is the sole representative for the service with respect to the presence server.

The association is a special group-based component of the presence service. It is an extension of other presence group-based such as private groups and role-based groups that were developed in an effort to investigate the integration of presence with call control [3]. Each presence agent contains a set of notification and subscription policies that are used to determine under which context the users can see the availability of each other. These policies are asserted into the presence server by the association. It also manages the automatic subscription between all the presence agents representing the users and services that are members of the association. This is required in order for each person to see each other's availability as well as the availability of a service. Details on these polices are part of a patent submission on "role-based presence" [3].

We see an example of the formation of these groups in figure 2 for a presence client displaying users that are part of a private group. The association component is an extension of the private group formalism except the name is changed.

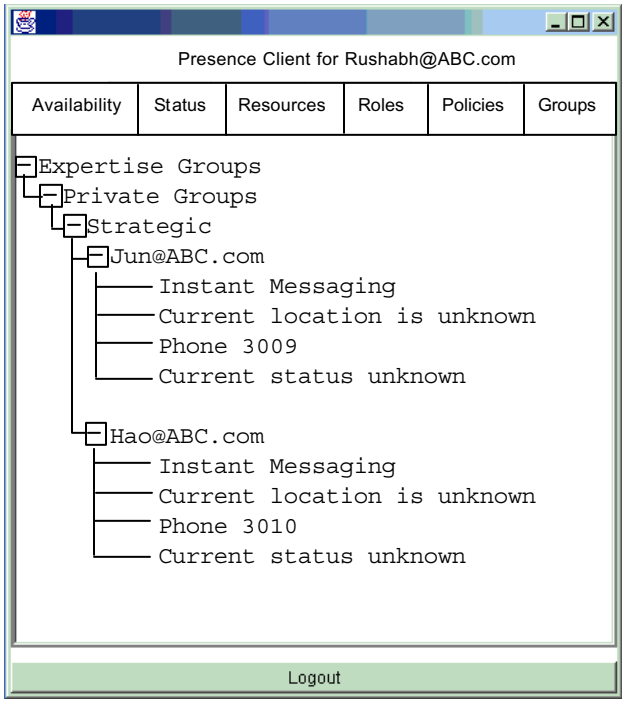

Figure 2. Presence interface showing members of a private group

In this configuration the presence service manages the distribution of service contact information among all the members of the association. The Association also has the

IEEE COMPUTER SOCIETY

ability to monitor the state of the enterprise services and behaves like a Presence User Agent (PUA) [2] to the presence server and sends change status events to the presence server so that users of a service are aware of the availability of the service. The presence server uses the SIP SIMPLE protocol for communication with the presence clients.

6 Related Work

In this section we will discuss other approaches that are closely related to the . We have not seen much work published in the area that combines service discovery and presence.

Some of the original developers of the SIP protocol have proposed the use of SLP for discovering the SIP registrars. This has been proposed to IANA as an SLP supported service [8]. This is definitely not the way that we propose that SLP be used with SIP.

In a recent paper by Wu et al. on the integration of Web Services with SIP Presence [9] a methodology and framework was presented that supported SIMPLE presence messages with a proprietary collaboration Web-Service framework called XGSP. The emphasis was placed more on the translation of SIP messages to XGSP so that the SIP protocol could be supported by XGSP. In our framework the XGSP Web service would be a service that would need to register with the enhanced Presence server in order that it be projected to the appropriate users of the presence service.

A paper similar to this work, by Berger et al. [10], uses SLP to locate services relevant to a user's context and controls those devices using SIP SIMPLE messages that are sent to the device. The types of messages that are sent are based on a pre-defined set of user preferences. It does not try to project these services to the users in the system.

The Service Peer Discovery Protocol developed by Cascella [11] follows along similar lines as that presented in this paper. In that work a new service discovery protocol was invented that leveraged extensions to the SIP messages to send the request to other peers in a network.. It does not try to propose the use of existing protocols since it felt that these protocols were not suited for service discovery in a peer-to-peer network.

7 Conclusion

This paper presented an extension to the SIP/ SIMPLE protocol in order to support SLP service profiles. With this extension the user receives much more information about the service that is required to make a connection.

We would like to extend this work to handle the projection of Web Service interfaces using presence. With the rapid adoption of UDDI and SOAP as mechanisms for service discovery for Web service it is imperative that the framework be able to manage these types of services.

We are also currently investigating the use of this approach for the management of secure services through the investigation of dynamic access control mechanisms and security profiles.

8 References

[1] J. Rosenberg, H. Schulzrinne, G. Camarilli, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 2543 Draft #9, February 2002.

[2] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)", IETF Internet Draft <draft-ietf-simple-presence-10.txt>, January 2003.

[3] R. Liscano, K. Baker, N. Balaba, and J. Zhao, "Role-based Presence", UK Patent Submission File #0218711.0, 2002.

[4] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging", IETF RFC 2778, February 2000.

[5] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson, "Presence Information Data Format (PIDF)", IETF Internet Draft <draft-ietf-impp-cpim-pidf-08.txt>, May 2003.

[6] P. Saint-Andre and J. Miller, "XMPP Instant Messaging", IETF Internet Draft < draft-ietf-xmpp-im-11>, May 04, 2003.

[7] E. Guttman, C. Perkins, J. Veizades, and M. Day. "Service Location Protocol". RFC 2608, July, 1999.

[8] J. Kempf and J. Rosenberg, "Using SLP for SIP Server Discovery", IANA SLP Service Specification.

[9] W. Wu, A. Uyar, H. Bulut, and G. Fox, "Integration of SIP VoIP and Messaging with the AccessGrid and H.323 Systems", Proc. of 1st Int. Conf. on Web Services Las Vegas June 2003.

[10] S. Berger, H. Schulzrinne, S. Sidiroglou, and X. Wu, "Ubiquitous Computing Using SIP", NOSSDAV'03, Monterey, California, USA, June 1–3, 2003.

[11] R. Cascella, *Reconfigurable Application Networks Through Peer Discovery and Handovers*, MSc Thesis, Royal Institute of Technology (KTH), Dept. of Microelectronics and Information Technology Stockholm, Sweden. June 2003.

[12] A. Ghavam, R. Liscano, M. Barbeau, T. Gray, and N.D. Georganas, "Enabling Secure Ad hoc Communications in the Enterprise", Proc. Ad hoc Networks and Wireless (AdHoc-NOW 2002), Toronto, Canada, September 20-21, 2002.