Personalization for SIP Multimedia Communications with Presence

Dongmei Jiang[1], Tet Hin Yeap[1], Luigi Logrippo[1,2], Ramiro Liscano[1]

[1] University of Ottawa, 800 King Edward Ave., Box 97, Ottawa, Ontario, K1N 6N5, Canada
[2] Université du Québec en Outaouais, Département d'informatique et ingénierie
[1] {djiang, tet, luigi, rliscano}@site.uottawa.ca

Abstract- **This paper presents a 3-layer service architecture for the Session Initiation Protocol (SIP) that incorporates other protocols to provide multimedia and presence communications. Beyond SIP basic services, personalized services are also addressed. To enrich presence related services, presence information is extended from the well known "online" and "offline" indicators to a much broader meaning that includes "location", "lineStatus", "role" and "availability". Based on this, the Call Processing Language (CPL) is extended to describe presence related personalized services in both call-processing systems and presence systems using information such as a person's presence status, time, address, language, or any of their combinations.**

Keywords: **SIP; presence; multimedia communications; SDP; presence extensions; CPL extensions**

I. INTRODUCTION

Presence in communications conveys the willingness and the ability of a user to communicate with others on a network. With awareness of the presence information of other users, unwanted and interrupting calls can be avoided. Multimedia communications in Session Initiation Protocol (SIP) with presence can provide uninterrupted communications via the formats of instant messaging, audio call, video call, multiparty conferencing, etc. After "I Seek You" (ICQ) was introduced in 1996, numerous variations of instant messaging with presence, and more recently, presence based multimedia communications have come to the market very quickly. Currently, presence information involves "online" and "offline" indicators only and no personalized services are offered to users. After a brief introduction to the SIP communication system, this paper will address the enrichment of presence information and the description and control of personalized services for multimedia communications, especially presence related.

II. SIP COMMUNICATIONS

In the three-layer architecture seen in Fig.1, a call-processing system (italic) and a presence system are shown to illustrate how a SIP communication system works. SIP [1], the signaling standard from Internet Engineering Task Force (IETF), is responsible for establishing, modifying and terminating multimedia sessions and calls. It has two types of components: user agents and network SIP servers. By sending SIP messages [1][2], a user agent allows a caller (or a watcher in a presence system) to initiate a request and allows a callee (or a presentity in a presence system) to answer the request. A SIP message can contain media session information in the Session Description Protocol (SDP) [3], which helps to determine in what type of media (e.g. audio, video, etc.) the communication will be realized later. A SIP server directs SIP messages to where they should go for transport. A registrar accepts SIP REGISTER requests and keeps the user information to provide location service, which is the key to achieve mobility. Defined on top of the transport layer, SIP messages can convey arbitrary payloads: session description, instant messages, presence documents, etc. This makes SIP natural and easy to combine with voice services and data services. Once a SIP session is established, the real time media inputs are sampled, converted to digital format and encapsulated in the Real Time Protocol (RTP), whose session quality is monitored via the Real Time Control Protocol (RTCP). The media inputs are then delivered most often via the User Datagram Protocol (UDP), which is more efficient than the Transmission Control Protocol (TCP).

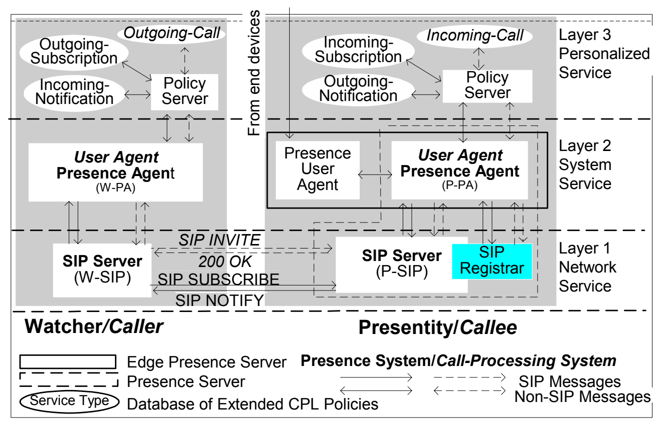

Fig. 1. Presence System and Call-Processing System Architecture

III. THREE-LAYER SERVICES

We propose a three-layer architecture for SIP communication services, as shown in Fig. 1. SIP servers provide network services in Layer 1. In Layer 2, user agents (or presence agents in the presence system) provide system basic services to all users fairly. The system services include sending requests for a caller (or a watcher in the presence system), replying to the call for a callee or authenticating and authorizing the watcher's request in the presence system, and notifying the watcher once its request is approved. In Layer 3, personalized services (policies) (ovals in Fig.1) are described in extended Call Processing Language (CPL) [4]. These

0-7803-8971-9/05/$20.00 ©2005 IEEE

policies are associated with and owned by a particular user and triggered only when the request is for the user. As a simple example of personalized services, Tom can reject calls from anonymous callers. Personalized services are programmed by end users, managed by a policy server and executed by user agents or presence agents. Only system basic services will be provided if personal policies are not available. In presence systems, a notifying process can be induced by any of the following events: a presentity registers its presence information; the presence information is updated or a watcher's subscription request is approved [5].

IV. PRESENCE INFORMATION EXTENSIONS

As mentioned before, currently presence information contains "on-line" and "off-line" indicators only, which is very limited concerning a person's communication status. An extension was proposed in [6] to include "location", "lineStatus", "role" and "availability".

location: indicates a presentity's current location. The parameter value can be "office", "meeting room" or "car" etc. These locations must be equipped with sensors in a network. The sensors can identify and monitor the user who has registered in the network.

lineStatus: indicates if a presentity is occupying a telephone line or not. It has two possible values, "on" and "off". Value "on" indicates that the user is on a phone and value "off" indicates that the user is not on a phone.

role: indicates the position status of a presentity, such as "professor", "consultant", etc. A user can have multiple role values when the user takes more than one position simultaneously. For example, Sharon works as a consultant for a company and as a professor for a university. She has two roles, "consultant" and "professor".

availability: indicates whether a presentity wishes to communicate with others. The value "yes" indicates that the user is ready to communicate with others and "no" indicates that the user does not wish to communicate with others currently. On status "yes", other users have a better chance to get a successful communication with the user than on status "no". Availability is not the presentity's communicating ability; it is his willingness. On status "no", the user can still communicate with others if necessary. For example, he can still answer emergency phone calls.

When any of the presence parameters is changed, the user's presence status is changed and his presence agent is informed of the change. The presence agent then notifies all watchers of this change according to the user's notification policies.

The extensions of presence information can be defined in the two formats: Document Type Declaration (DTD) or Extensible Markup Language (XML) schema [6]. The XML schema is preferred in this paper due to its capability of being extended. Presence documents are written in Presence Information Data Format (PIDF) [7] using XML syntax.

```
NOTIFY sip 厖

<presence ?
 <tuple id="?   >

  <status>
   <basic>open</basic>
     <!- able accept instant message? -->
   <epidf:location> office </epidf:location>
   <epidf:lineStatus> on </epidf:lineStatus>
     <!- talking on a phone? -->
   <epidf:role> doctor </epidf:role>
   <epidf:availability> no </epidf:availability>
     <!- like a talk now? -->
   </status>

  </tuple>
</presence>
```

Fig. 2. Presence in PIDF carried in a SIP NOTIFY

As an example of presence extension, Fig. 2 shows Bob's presence information sent to Alice indicating that Bob can accept instant messages; he is working as a doctor in his office and he is talking on his phone; also he prefers not to communicate with others at this time (he is not available). Knowing this, Alice decides not to bother Bob with a call.

V. CPL EXTENSIONS FOR PRESENCE

CPL [4] was accepted as a standard by the IETF in 2004. It is designed for end users to describe and control their specific telephony services. CPL itself is a programming language in XML syntax. Its script represents a tree of decisions in terms of tags of nodes and links. Each node or link corresponds to a tag in CPL. A node specifies an action to take or a decision to make. A link specifies the result of an action and displays which decision was taken. CPL is independent of signaling protocols. It can work on top of either SIP from the IETF or H.323 from the International Telecommunications Union-Telecommunications Standard Sector (ITU-T). This paper only addresses the co-operation of CPL with SIP. With the two directions of "incoming" and "outgoing" considered, current CPL can process SIP INVITE messages and deal with call-processing services.

X.T. Wu was the first to propose an extension of CPL for presence and published his work in an IETF Internet Draft [8]. He added to CPL the capability to describe presence system services, but his focus was on basic system services rather than user personalized services. We extend his work by defining four top-level actions, five operations and a presence-switch [6] to process SIP SUBSCRIBE and SIP NOTIFY messages with the two directions of "incoming" and "outgoing" considered.

A. Four New Top-level Actions

1) incoming-subscription - the action that is performed on the presentity-side when a SIP SUBSCRIBE message arrives and the message's destination is the script owner, i.e. the presentity.

2) outgoing-subscription - the action that is performed on the watcher-side when a SIP SUBSCRIBE message is ready to be sent and the message's originator is the script owner, i.e. the watcher.

3) incoming-notification - the action that is performed on the watcher-side when a SIP NOTIFY message arrives and the message's destination is the script owner, i.e. the watcher.

4) outgoing-notification - the action that is performed on the presentity-side when a SIP NOTIFY message is ready to be sent and the message's originator is the script owner, i.e. the presentity.

B. Five New Operations

1) subscribe - this action causes the SIP server to send a SIP SUBSCRIBE message to the specified presentity.

2) notify - this action causes a SIP server to send a SIP NOTIFY message to the specified watcher. The NOTIFY message contains a presence document in PIDF.

3) approve - this action tells a PA of a presentity that the watcher request is approved with a time limit. The PA then can start to prepare the notification message.

4) accept – by this action, the PA of a watcher accepts the received notification. The presence information is displayed or refreshed.

5) call – this action combines action "accept" for presence with a new "call" action for the call-processing service. It causes the SIP server to send a SIP INVITE message to a specified callee. The script owner is the caller who launches a new call. The action leads to the co-operation of the presence system and the call-processing system i.e. the presence agent in the presence system co-operates with the SIP user agent in the call-processing system.

C. A New Presence Switch

1) presence-switch – this switch enables an end user to make decisions based on the presence status of a presentity. The presentity may be the user himself or somebody else. This is a very important feature. If the presentity is the user himself then the presence information can be acquired from the user's presence user agent. Otherwise if the presentity is somebody else then the user must have captured the presentity's presence information from a previous SIP NOTIFY message or from the incoming SIP NOTIFY message depending on which event triggers the CPL script.

Node "presence-switch" has two mandatory parameters, "presentity" and "timeout". Parameter "presentity" identifies a presentity, and parameter "timeout" gives the CPL executor (i.e. a user agent in a call-processing system or a presence agent in a presence system) a time limit to retrieve presence information. "presence-switch" is followed by the link node "presence", which specifies a presence status to verify whether the presentity matches the status or not.

The definition of CPL extensions for presence can be defined in the format of either DTD or XML schemas. These definitions will be used to validate CPL scripts. With these extensions of CPL, personalized services are much enriched in both call-processing and presence systems. They can be handled based on a person's status, time, address, etc.

VI. APPLICATIONS

These new services in extended CPL are event services. They can be caused by the events of "incoming-call" or "outgoing-call" in call-processing systems, or by the events of "incoming-subscription", "outgoing-subscription", "incoming-notification" or "outgoing-notification" in presence systems.

Based on the types of actions in extended CPL, these new services can be classified into three main types: screening service, forwarding service and auto-call service. The first two types can be applied to both call-processing systems and presence systems. The auto-call service is based on a presence event and it is initiated by a presence system. These types of services are shown with the following four examples described in extended CPL.

```
<cpl xmlns = ?
<cplPresence:incoming-subscription>

  <address-switch field ="origin">
  <address is = "sip:SharonBoss@example.com">
    <time-switch tzid="America/New_York"
                    tzurl="http://zones.example.com/tz/America/New_York" >
    <time dtstart="20000703T090000" duration="PT8H" freq="weekly"
            byday="MO,TU,WE,TH,FR" >

      <cplPresence:approve/>
    </time>
    <otherwise>
      <reject/>
    </otherwise>
    </time-switch >
  </address>
  </address-switch>

</cplPresence:incoming-subscription>
</cpl>
```

Fig. 3. Conditionally Authorization for Presence

The first example is Sharon's policy of conditional authorization to her watchers as shown in Fig. 3. Sharon screens her incoming-subscription requests. She accepts her boss's requests only during working hours, i.e. from 9:00am to 5:00pm, Monday to Friday. This is a "pure" screening example for presence subscription where the trigger is an "incoming" SIP SUBSCRIBE request.

The second example shown in Fig. 4 represents an outgoing-call screening service based on the callee's (Sharon's boss') presence status. Sharon thinks it is not a good time to call her boss when he is talking on his phone with his availability "no". Sharon makes her outgoing-call screening policy to prevent such calls. In her policy as shown in Fig. 4, she blocks her calls to her boss when the boss is talking on his phone and his availability status is "no". The calls will be processed based on the callee's status, which is triggered by Sharon's "outgoing" SIP INVITE messages.

```
<cpl xmlns ......>
<outgoing>
  <address-switch field ="original-destination">
  <address is = "sip:SharonBoss@example.com">

    <cplPresence:presence-switch presentity= "sip:SharonBoss@example.com">
    <cplPresence:presence lineStatus = "on" availability = "no">
      <cplPresence:success>
        <reject/>
      </cplPresence:success>
    </cplPresence:presence>
    </cplPresence:presence-switch >

  </address >
  </address-switch>
</outgoing>
</cpl>
```

Fig. 4. Screening Outgoing-calls

```
<cpl xmlns ......>
<incoming>

  <cplPresence:presence-switch presentity =  "sip:SharonBoss@example.com">
    <cplPresence:presence lineStatus = "on" availability = "no">
      <cplPresence:success>
        <location url="sip:SharonBossVoiceMail@example.com">
          <proxy/>
        <location>
      </cplPresence:success>
    </cplPresence:presence>
  </cplPresence:presence-switch >

</incoming>
</cpl>
```

Fig. 5. Presence-based Incoming-call Forwarding

The third example shown in Fig. 5 is a call-forwarding service for Sharon's Boss. He forwards his incoming-calls to his voice mail when he is talking on his phone with his availability status "no". He prefers to deal with these voice mails at a later time. His policy is shown in Fig. 5. These calls are processed based on callee's (the boss's) status, which is triggered by his "incoming" SIP INVITE messages.

The last example is an auto-call policy of Peter. As shown in Fig. 6, Peter asks to initiate an auto call to Sharon as soon as he is notified that Sharon arrives at her office. This auto-call is based on the callee's (Sharon's) presence status, which is carried in the "incoming" SIP NOTIFY from Sharon to Peter.

```
<cpl xmlns = ?   厖
<cplPresence:incoming-notification>

  <address-switch field="origin">
  <address is="sip:Sharon@example.com">

    <cplPresence:presence-switch presentity ="sip:Sharon@example.com">
    <cplPresence:presence location = "office">
      <cplPresence:success>
        <location url="sip:Sharon@example.com">
          <cplPresence:call/>
        </location>
      </cplPresence:success>
    </cplPresence:presence>
    </cplPresence:presence-switch>

  </address>
  </address-switch>

</cplPresence:incoming-notification>
<cpl>
```

Fig. 6. Presence-based Auto Calls

VII. CONCLUSION

This paper has described an architecture for personalized services for SIP multimedia communications with presence that uses extended presence information and extended CPL. Based on this work, end users can have call-processing services and presence services, which are processed according to a person's presence status, time, address, or any combinations.

ACKNOWLEDGMENT

This work was motivated by a grant from Mitel Networks. It was also funded partially by Communications and Information Technology Ontario and the Natural Sciences and Engineering Research Council of Canada. We thank Tom Gray and Romelia Plesa for having participated in parts of this work and Jacques Sincennes for technical advice.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, etc. "SIP: Session Initiation protocol", IETF RFC 3261, June 2002.

[2] A.B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification", IETF RFC3265, June 2002.

[3] M. Handley and V. Jacobson, "SDP: session description protocol", IETF RFC 2327, April 1998.

[4] J. Lennox, X. Wu, H. Schulzrinne, "Call Processing Language (CPL): A language for User Control of Internet Telephony Services", IETF RFC3880, October 2004.

[5] J.Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)", IETF RFC 3856, August 2004.

[6] D. Jiang, Internet Telephony Services for Presence with SIP and Extended CPL, Master's Thesis, University of Ottawa, 2004.

[7] H. Sugano, S. Fujimoto etc. "Presence Information Data Format (PIDF)", IETF RFC 3863, August 2004.

[8] X.T. Wu, et al., CPL Extensions for Presence, IETF Internet Draft, June 1, 2000, Accessed May 2004, http://www1.cs.columbia.edu/~xiaotaow/rer/Research/Paper/draft-wu -cpl-presence-00.txt.