SIP/SIMPLE-based Conference Room Management Method for the Voice Communication Medium "voiscape"

Yasusi Kanada
Central Research Laboratory, Hitachi, Ltd.
Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185-8601, Japan
kanada@crl.hitachi.co.jp

Abstract

A method for conference-room management for an auditory-virtual-space-based voice-communication medium called voiscape and a voice-communication system prototype called VPII, which used this method, were developed. With this method, conference rooms (called sound rooms) are managed through SIP and SIMPLE (a presence-related event-notification mechanism). A user can not only obtain a room list and enter (select) or exit from a room, but can also create, modify, or delete rooms by SIMPLE messaging. Rooms, users, and objects are managed by their "soft state"; i.e., they are deleted when a time out occurs. Users are informed of room membership, presence of a user, e.g., location and direction in the room, and presence of an object in the room by SIMPLE messaging, i.e., by SUBSCRIBE, NOIFY, and PUBLISH requests. To reduce the messaging overhead, the partial notification mechanism of SIMPLE is used in VPII.

Keywords: SIP (Session Initiation Protocol), SIMPLE, Virtual presence, Conference room management, Voiscape.

Contact Person: Yasusi Kanada (kanada@crl.hitachi.co.jp)

Contents

Introduction

- **Conference rooms** must be managed in teleconferencing systems.
- **Voiscape** — a voice-centered communication medium using an auditory virtual environment
- Each user voice is spatially located in a **sound room**.
- This paper focuses on the **sound-room management** in a voiscape prototype called VPII.

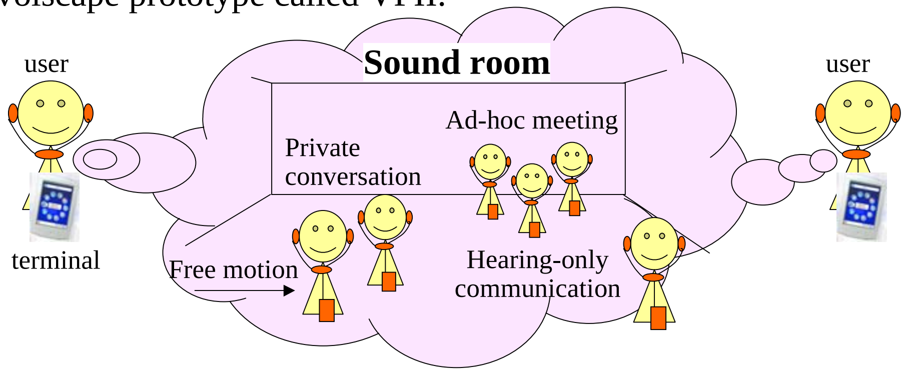


Figure 1. Sound-room concept

2

1. Introduction

Teleconferencing systems, including audio-only and audio-and-video conferencing systems, enable conversation between three or more persons. In many teleconferencing systems, there are multiple virtual conference rooms; i.e., two or more independent conferences can be held simultaneously. Each conference room has different users and different resources that must be managed by the teleconferencing system.

We are developing a voice-centered communication medium called *voiscape* [Kan 04]. Voiscape creates an auditory virtual environment with spatial audio technologies. The sound environment thus created is close to a face-to-face conversation environment. People enter a 3-D space, which is shared among the people in that space, to communicate with each other. This space is called a *sound room*. Each person in this space is represented by a spatially located sound, and people can move freely (see **Figure 1**). If a person moves close to another person, a localized conversation can be naturally initiated. A voiscape system is an audio conferencing system and the sound rooms that belong to the system are conference rooms. The sound rooms must be managed by the voiscape system.

A voiscape system prototype called VPII (Voiscape Prototype II) has been developed [Kan 05]. Because the properties and states of rooms, users, and objects that must be managed in VPII can be regarded as "presence", VPII manages them by using the presence-related event-notification mechanism of SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) [Roa 02] [Ros 04]. This event-notification mechanism enables a presence watcher to subscribe to a presence server, which notifies presence changes [Day 00], and enables a presentity to "publish" its presence change to a presence server [Nie 04].

This paper explains the conference-room management functions of voiscape and describes their implementation. Section 2 outlines communication using voiscape. Section 3 explains the room-management functions of voiscape, i.e., room list, property, user, and policy managements. Section 4 describes the framework and the method of room management using SIMPLE and their implementation. Section 5 concludes the paper.

Outline of Communication Using Voiscape

- ♦ The user selects a room from a menu and enter it.
- ♦ The user move in the room by pointing devices such as cursor keys — select persons or sound sources.
  - Forward/backward cursor keys: move forward/backward.
  - Left/right cursor keys: turn left/right 18 degrees.
- ♦ The user talks with other users or listen to a sound source.

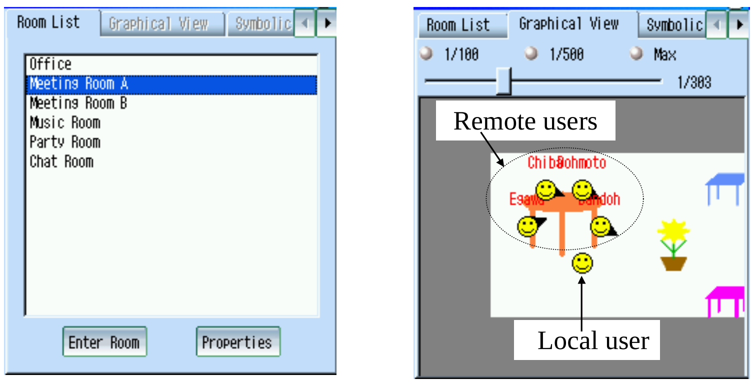


Figure 2. User interface of Voiscape Prototype II

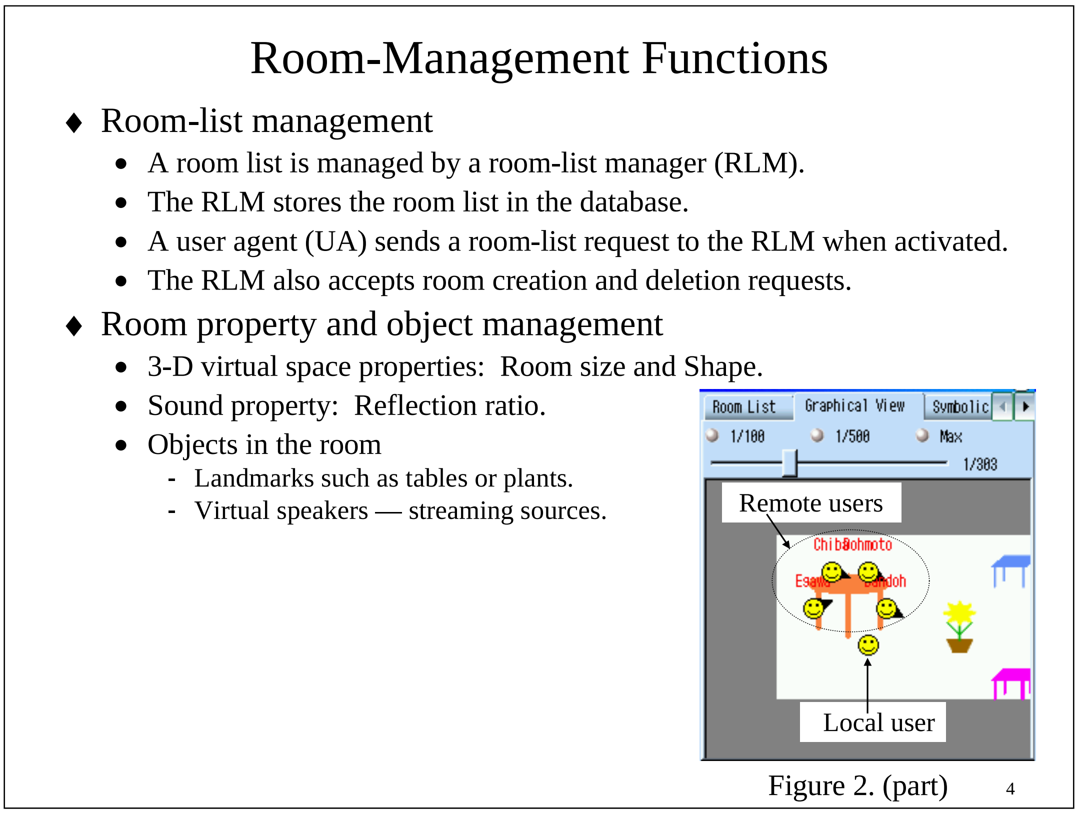
3

2. Outline of Communication Using Voiscape

The user interface of a UA is illustrated in **Figure 2**. The user first selects a room to enter from the room list, shown on the left. The UA then displays a map of the sound room, shown on the right. The walls are displayed in gray. The scale of the map can be changed using radio buttons (or a slider). A unique icon can be used for each user. The orientations of the other users are displayed by arrows.

A user moves one-foot forward by pushing the forward (arrow) key and one-foot backward by pushing the backward key. The user turns left 18 degrees by pushing the left (arrow) key and right 18 degrees by pushing the right key. The user's icon is always displayed immediately below the screen center, and its orientation remains fixed. This means that the displayed wall moves downward as the user moves upward and that the room display and other users rotate right as the user turns left. By moving, the user can select a person or persons in the room and talk with them, or can select a sound source in the room and hear them.

In the prototype implementation, a Sharp Zaurus (SL-B500, SL-C860, or SL-5600) and a Linux-based PDA were used as the terminals. Qt middleware developed by TrollTech was used in the Zaurus to provide a light-weight window system and some additional functions such as XML parsing. Since Qt works on Microsoft Windows and Apple Mac OS X, software run on Qt can be easily ported between these environments.



Room-Management Functions

- Room-list management
  - A room list is managed by a room-list manager (RLM).
  - The RLM stores the room list in the database.
  - A user agent (UA) sends a room-list request to the RLM when activated.
  - The RLM also accepts room creation and deletion requests.
- Room property and object management
  - 3-D virtual space properties: Room size and Shape.
  - Sound property: Reflection ratio.
  - Objects in the room
    - Landmarks such as tables or plants.
    - Virtual speakers — streaming sources.

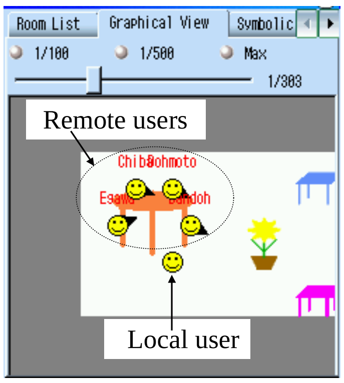


Figure 2. (part)

4

3. Room Management Functions

This section explains the room-management functions of voiscape; i.e., room list, property, user, and policy management functions. These functions include those not yet implemented in VPII.

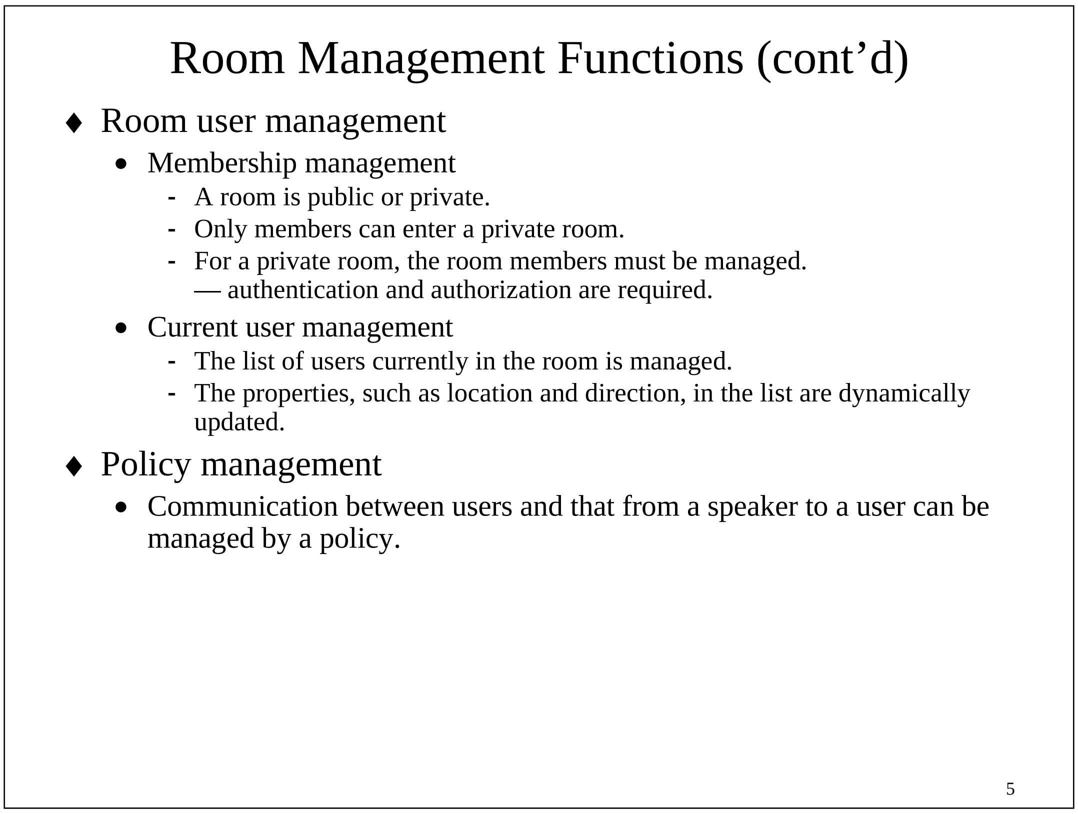
3.1 Room-list management

When a user agent (UA, i.e., a terminal program) is activated, it requests the room list from a room-list manager (RLM), which then returns the requested list to the agent. The RLM stores the list in the room-list database. The RLM also accepts requests for creating or deleting a room and records them in the database. In our design, each room belongs to only one room list. A room creation thus means addition of a room to the list, and a room deletion means removal of a room from the list.

3.2 Room-property and object management

Room properties are managed by the room manager. A sound room has many properties. It is not just an abstract conference room, but it is a 3-D virtual space and has sound-related properties. The virtual space properties include room size (i.e., width, depth, and height, if the room is rectangular). The room size is also used as a sound-related property. Another sound-related property is the reflection ratio of the walls. This is used with the room size for calculating early reflections (reverberations) [Kan 05].

A sound room can also contain silent objects such as tables or plants (see Figure 2). They can be used as visual landmarks from which users can distinguish their direction in the room and places to meet with other users. The objects in the room have their locations and icons as their properties. A sound room can contain a speaker object that plays a sound file has its address (URL) as a property.



Room Management Functions (cont'd)

- ♦ Room user management
  - Membership management
    - A room is public or private.
    - Only members can enter a private room.
    - For a private room, the room members must be managed. — authentication and authorization are required.
  - Current user management
    - The list of users currently in the room is managed.
    - The properties, such as location and direction, in the list are dynamically updated.
- ♦ Policy management
  - Communication between users and that from a speaker to a user can be managed by a policy.

5

3.3 Room-user management

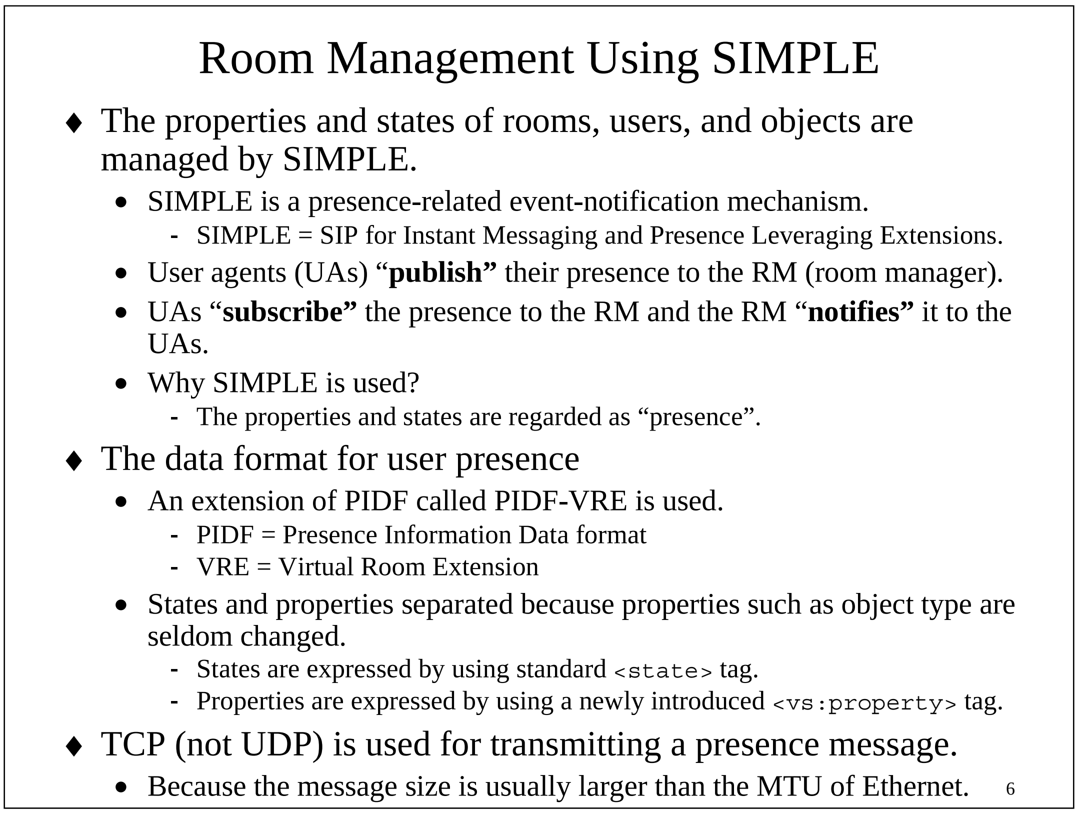
Room users are also managed by the room manager. Two types of user management are required.

(1) Membership management: Every user can enter a public room, but users can enter a private room only if they are members of that room. The room manager manages the membership and authorize users when they request room entry. If membership management is required, the users must be authenticated when (or just before) requesting a room list or when requesting room entry. The member properties include the URLs of user's visual and auditory icons if the user specifies icons that are not default. The visual icon is used in the map (see Figure 2), and the auditory icon may be used when the user moves close to another user [Kan 05]. The member properties may include the default virtual locations and directions of a member. The default location of a member represents the location where he/she appears when he/she enters a room, and it is displayed on the map even when he/she is not in the room.

(2) Current-user management: A member can enter a room or exit from it. The room manager has a list of users currently in the room. This list contains not only the identifiers of the users but also contains their current virtual locations and directions. These properties are updated when the users move or turn around in the room.

3.4 Policy management

Communication between room users and that from a speaker to a room user can be controlled by policies [Kan 04], which are also managed by the room manager.



Room Management Using SIMPLE

- ♦ The properties and states of rooms, users, and objects are managed by SIMPLE.
  - SIMPLE is a presence-related event-notification mechanism.
    - SIMPLE = SIP for Instant Messaging and Presence Leveraging Extensions.
  - User agents (UAs) "**publish**" their presence to the RM (room manager).
  - UAs "**subscribe**" the presence to the RM and the RM "**notifies**" it to the UAs.
  - Why SIMPLE is used?
    - The properties and states are regarded as "presence".
- ♦ The data format for user presence
  - An extension of PIDF called PIDF-VRE is used.
    - PIDF = Presence Information Data format
    - VRE = Virtual Room Extension
  - States and properties separated because properties such as object type are seldom changed.
    - States are expressed by using standard `<state>` tag.
    - Properties are expressed by using a newly introduced `<vs:property>` tag.
- ♦ TCP (not UDP) is used for transmitting a presence message.
  - Because the message size is usually larger than the MTU of Ethernet.

6

4. Room Management Using SIMPLE

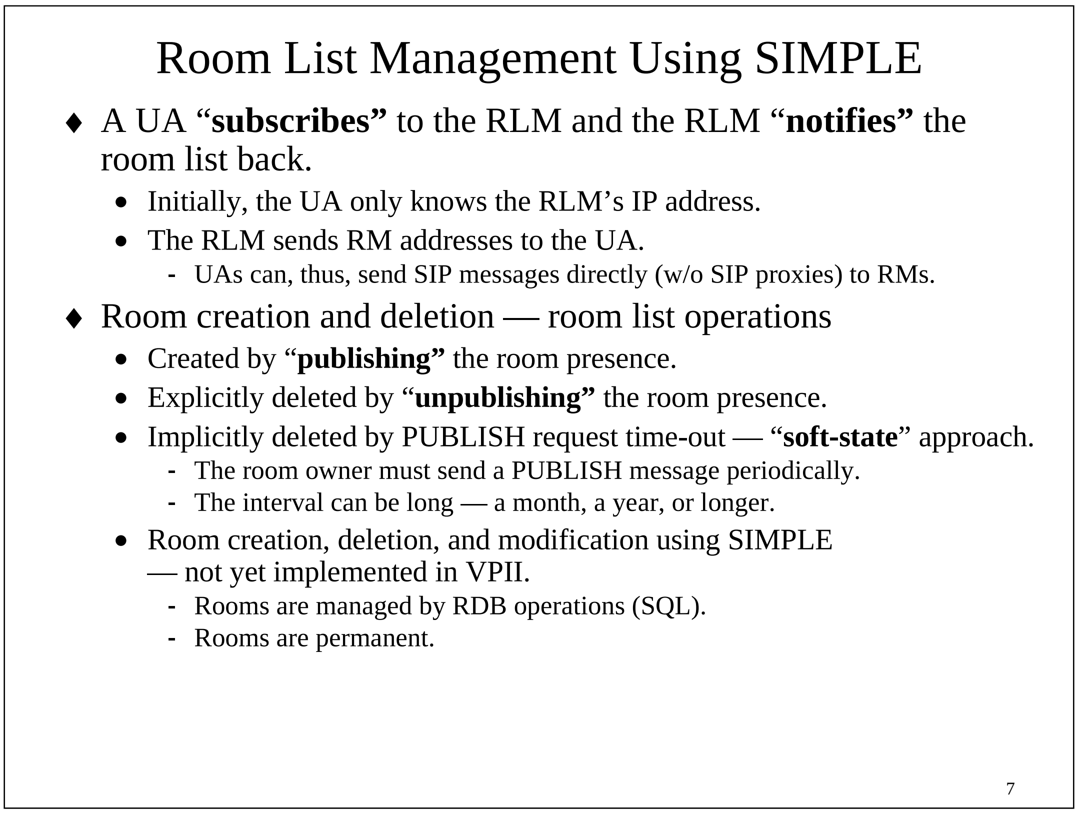
This section describes the framework and the method of room management using SIMPLE and the implementation in VPII.

4.1 Outline

The properties and states of rooms, users, and objects that must be managed in VPII can be regarded as "presence". VPII thus manages them by using the presence-related event notification mechanism of SIMPLE. SIMPLE is an extension of SIP (Session Initiation Protocol) [Ros 02]. Two request messages, i.e., SUBSCRIBE and NOTIFY, are added in this extension, and PUBLISH message is added by another document [Nie 04]. In VPII, the user agent (UA) sends a PUBLISH message to the room manager (RM) or room list manager (RLM) when a presence changes. The RM or RLM stores the presence in a database and sends the updated presence by using a NOTIFY message. The UA requests the room-presence information, which includes the presence of users and objects in the room, to the RM by sending a SUBSCRIBE message.

User presence is expressed using an extended PIDF (Presence Information Data Format [Sug 04]) document. This extended format is called PIDF-VRE (Virtual Room Extension). Presence is usually regarded as a status and is thus indicated by `status` tags (`<status>` and `</status>`). While a status can be changed easily, presence, in a general sense, contains properties that are not easily changed. Although it is not necessary to propagate unchanged properties every time a presence message produced by a status change is sent, they must be propagated the first time the user or object appears. Such properties should be distinguished from the status. However, PIDF has no tag for properties. A new tag, `vs:property`, was thus introduced to PIDF-VRE. This tag includes new tags such as `vs:type` and `vs:room-size`. The former indicates the type of entity, and the latter indicates the coordinates of the object.

TCP, instead of UDP, is used for transmitting a presence message, because the message size is usually larger than the MTU of Ethernet. SIMPLE and PIDF are computationally quite heavy, so if the status is updated frequently, presence propagation requires many computational resources. This problem and solutions are explained in Section 4.4.



Room List Management Using SIMPLE

- A UA "**subscribes"** to the RLM and the RLM "**notifies"** the room list back.
  - Initially, the UA only knows the RLM's IP address.
  - The RLM sends RM addresses to the UA.
    - UAs can, thus, send SIP messages directly (w/o SIP proxies) to RMs.
- Room creation and deletion — room list operations
  - Created by "**publishing"** the room presence.
  - Explicitly deleted by "**unpublishing"** the room presence.
  - Implicitly deleted by PUBLISH request time-out — "**soft-state**" approach.
    - The room owner must send a PUBLISH message periodically.
    - The interval can be long — a month, a year, or longer.
  - Room creation, deletion, and modification using SIMPLE — not yet implemented in VPII.
    - Rooms are managed by RDB operations (SQL).
    - Rooms are permanent.

7

4.2 Room-list management

In addition to room users and objects, a room list is also regarded as "presence" in VPII, and it is requested and sent by using SIMPLE. That is to say, the UA sends a SUBSCRIBE request to the RLM and the RLM replies with a NOTIFY request with a room-list PIDF document that contains a list of room presence. If the UA requests one-time notification, the effect is similar to a request-response protocol such as HTTP. However, if the UA requests subsequent notifications, the RLM reports room additions, changes, and deletions to the UA.

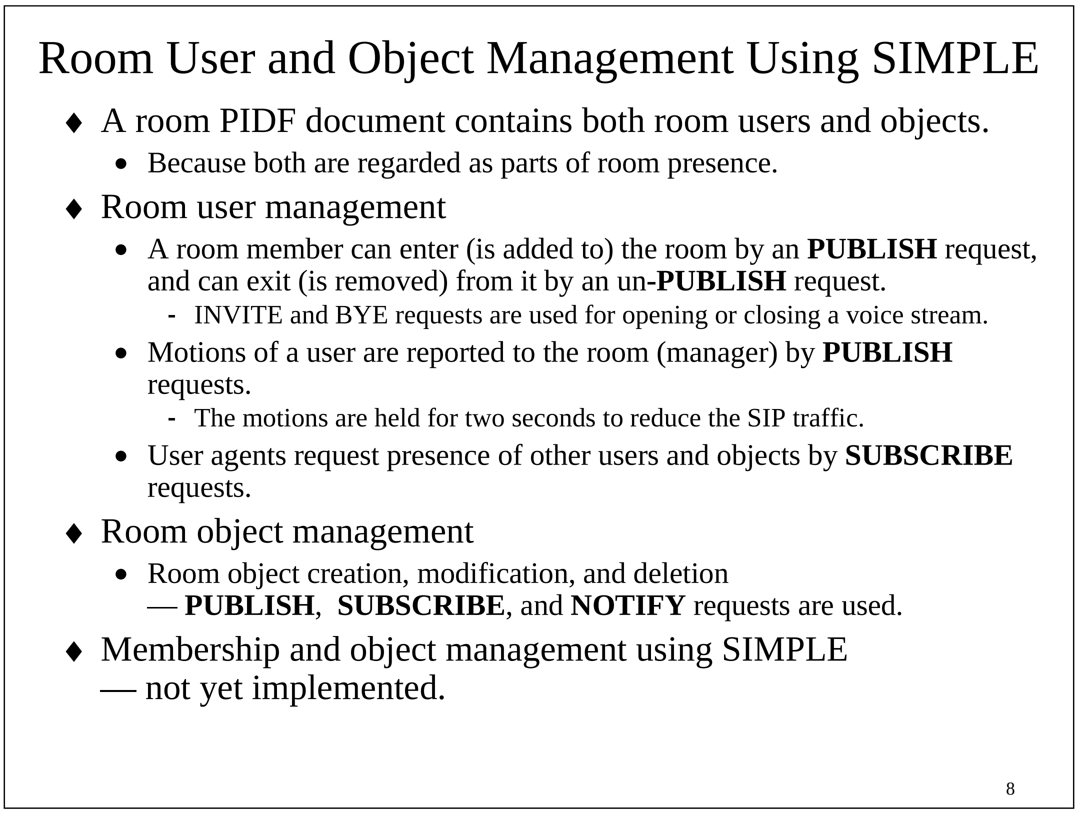
The following part of the PIDF document describes the presence of a sound room.

```
<tuple id="Office@serverdomain.hitachi.co.jp">
  <nickname>Office</nickname>
  <contact>sip:Office@1.2.3.4:5060</contact>
  <status><basic>open</basic></status>
  <vs:property>
    <vs:type>room</vs:type>
    <vs:room-size>50,30,5</vs:room-size>
  </vs:property>
</tuple>
```

This tuple indicates that the entity is a room. Its identifier is `Office@serverdomain.hitachi.co.jp`, which is an SIP URI, and its short name is `Office`. The room size is $50 \times 30 \times 5$ m. The contact address contains the IP address and port of the RM. This enables direct access to the RM and avoids high load of SIP proxies and the delay they cause. If the IP address or port of the RM is changed, UAs are informed about this change by sending a room-list notification message.

A room creation or deletion can be ordered by a UA by using a PUBLISH request. If a PUBLISH request for creating a room is valid, i.e., the user is allowed to create a room with the specified properties, a new room is created and its URI becomes ready to receive messages such as INVITE, BYE, or SUBSCRIBE. The creating user is the owner of the room unless it transfers the ownership to another user or the administrator. Because this event notification mechanism is based on a soft-state approach, i.e., the effects of SUBSCRIBE, NOTIFY, and PUBLISH messages disappear if these messages expire, the created room is kept alive only while the RM continues receiving PUBLISH requests from the owner. However, the interval of the PUBLISH requests can be a month, a year, or longer. This approach is applied to users and objects too.

In the current VPII, the RLM and RM are integrated into a single machine. The room-creation and deletion requests have not yet been implemented. The rooms are thus permanent and, at present, can be added or removed only by the system administrator.



Room User and Object Management Using SIMPLE

- ♦ A room PIDF document contains both room users and objects.
  - Because both are regarded as parts of room presence.
- ♦ Room user management
  - A room member can enter (is added to) the room by an **PUBLISH** request, and can exit (is removed) from it by an un-**PUBLISH** request.
    - INVITE and BYE requests are used for opening or closing a voice stream.
  - Motions of a user are reported to the room (manager) by **PUBLISH** requests.
    - The motions are held for two seconds to reduce the SIP traffic.
  - User agents request presence of other users and objects by **SUBSCRIBE** requests.
- ♦ Room object management
  - Room object creation, modification, and deletion — **PUBLISH**, **SUBSCRIBE**, and **NOTIFY** requests are used.
- ♦ Membership and object management using SIMPLE — not yet implemented.

8

4.3 Room-user and object management

Both current room users and objects in the room are regarded as parts of the room presence. A room PIDF document thus contains both. An object can be added to or removed from a room by a PUBLISH request. A room user can enter (be added to) a room by sending a PUBLISH request to the room, and the user can exit from the room by sending an un-PUBLISH request (a PUBLISH request with zero expiration time) to it in VPII. When entering or exiting a room, INVITE and BYE requests are used for opening or closing a voice stream between the UA and the media server. The UA sends a SUBSCRIBE request concerning room presence changes to the room URI, and they are reported to users by sending NOTIFY requests. The NOTIFY requests contain PIDF documents.

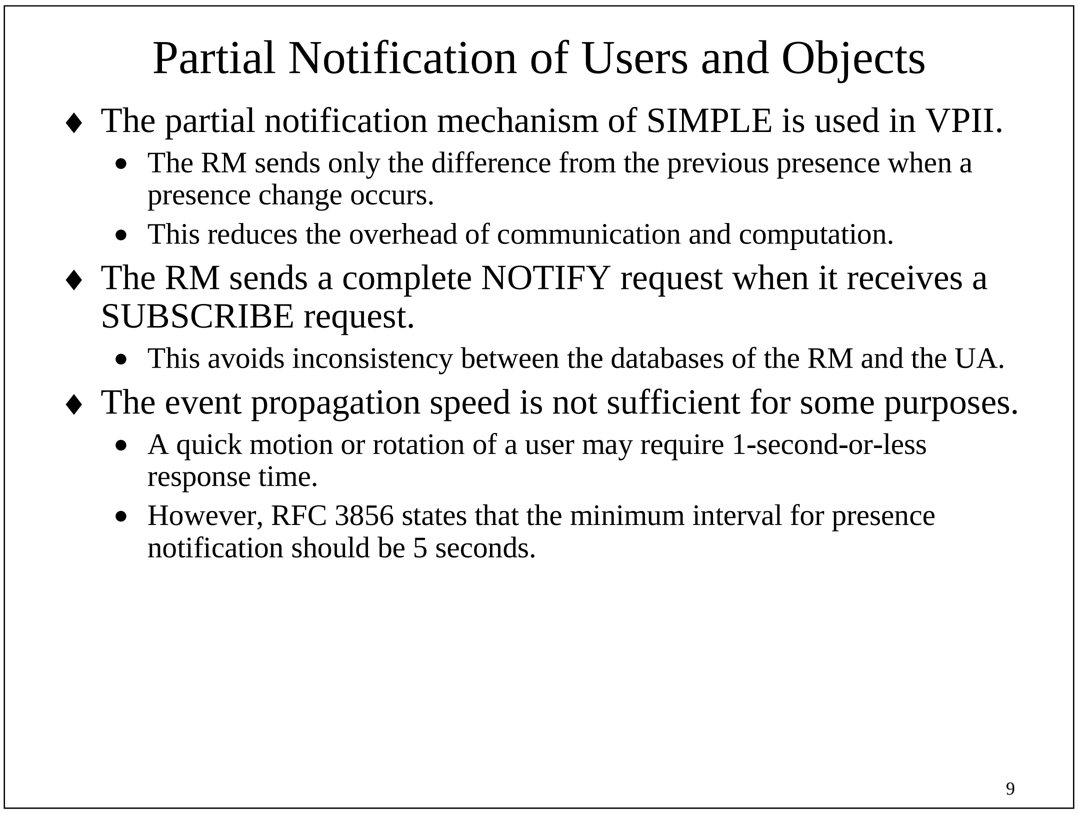
The following part of PIDF document (a tuple) describes the presence of a user.

```
<tuple id="George@userdomain.hitachi.co.jp">
  <nickname>George</nickname>
  <icon>http://hitachi.co.jp/icons/George.bmp</icon>
  <vs:auditory-icon>http://hitachi.co.jp/auditory-icons/George.wav
  </vs:auditory-icon>
  <status>
    <basic>open</basic>
    <vs:location>10,5,0</vs:location>
  </status>
  <vs:property><vs:type>human</vs:type></vs:property>
</tuple>
```

This tuple describes the user's properties and status. The identifier is `George@userdomain.hitachi.-co.jp`, and his short name is `George`. The 2-D and auditory icons of George are specified by their URLs. The property shows that he is a person. The status includes George's location and direction (orientation).

A NOTIFY request must contain a list of all the users and objects even if there is no presence change [Roa 02]. This means that if the number of users and/or objects is large, the message size becomes very large. In addition, if a user (or users) moves very frequently, notification requests are sent very frequently. The overheads of transmission, analysis, and creation of such a message is thus very large. This problem has been solved by the method described in the next subsection.

In the current VPII, objects cannot be added or removed by using SIMPLE messaging but only by the system administrator, and room membership management functions are not yet implemented.



Partial Notification of Users and Objects

- ♦ The partial notification mechanism of SIMPLE is used in VPII.
  - The RM sends only the difference from the previous presence when a presence change occurs.
  - This reduces the overhead of communication and computation.
- ♦ The RM sends a complete NOTIFY request when it receives a SUBSCRIBE request.
  - This avoids inconsistency between the databases of the RM and the UA.
- ♦ The event propagation speed is not sufficient for some purposes.
  - A quick motion or rotation of a user may require 1-second-or-less response time.
  - However, RFC 3856 states that the minimum interval for presence notification should be 5 seconds.

9

4.4 Partial notification of users and objects

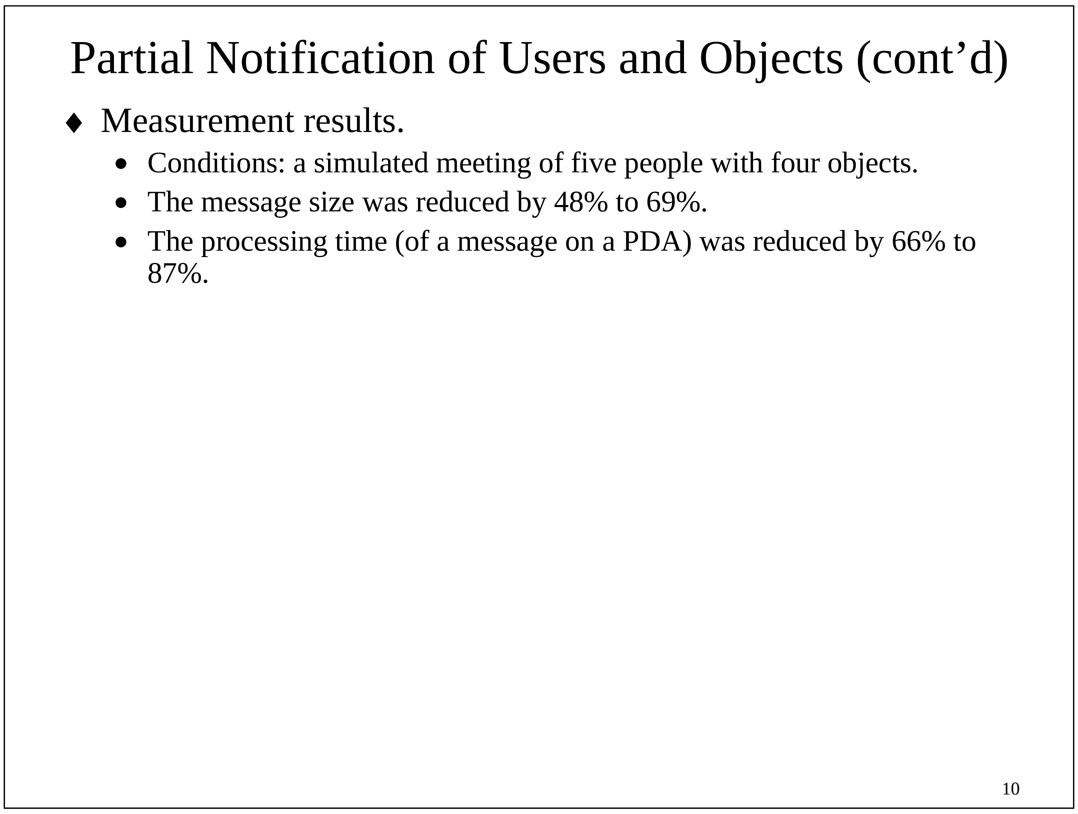
To reduce the overhead of communication and computation, a partial notification mechanism [Lon 04a] [Lon 04b] is used in VPII. When the presence is updated, only the difference from the previously reported presence is sent to the UAs by using a partial PIDF document; i.e., it does not contain unchanged parts. If there is only a small change, the document size is much smaller than the full notification message. For example, an update of user location and direction is notified by the following partial PIDF document.

```
<?xml version="1.0" encoding="UTF-8"?>
<pidf-diff xmlns="urn:ietf:params:xml:ns:pidf"
           xmlns:vs="urn:ietf:params:xml:ns:virtual-space"
           entity="pres:meetingA@serverdomain.hitachi.co.jp" version="2">
  <replace sel="presence/tuple[id=&quot;Listener51&quot;]/status/vs:location/
text()">5,4,0</replace>
</pidf-diff>
```

Both the full and half PIDFs contain a comma-separated list: “`10,5,0`” and “`5,4,0`”. They do not follow typical XML syntax; i.e., each list element is not enclosed by a tag. The reason such a syntax is used is that it reduces the size of partial PIDF documents.

Presence changes always generate partial notification messages. However, full notification messages are also sent to a UA when it sends a SUBSCRIBE request to the RM. Because SUBSCRIBE requests are sent periodically, full notification messages are also sent periodically. As a result, even if some inconsistency between the RM database and the UA database occurs because of bugs or a temporary communication trouble, the inconsistency is resolved when the UA receives a full notification message.

Although the size of partial notification message is small, RFC 3856 [Ros 04] states that the minimum interval for presence notification should be five seconds. Thus, if a presence change such as user motion is quick, the system cannot track the change properly. In the current VPII, the interval is set to two seconds. However, this response time is still not sufficient for some applications. A protocol different from SIP/SIMPLE, such as RTP (Real-time Transport Protocol), should be used for user location and direction notification in such applications. Shirmohammadi and Georganas [Shi 00] discussed update messages and proposed a new protocol called Synchronous Collaboration Transport Protocol.

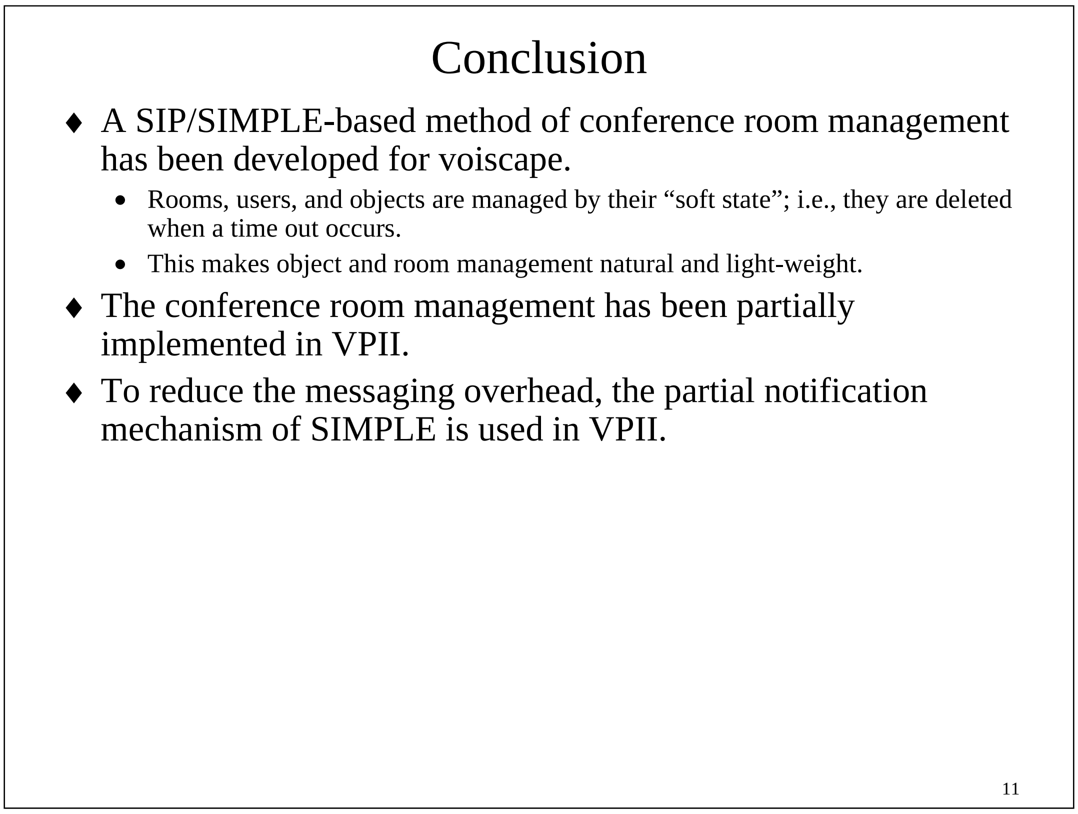


Partial Notification of Users and Objects (cont’d)

- ♦ Measurement results.
  - Conditions: a simulated meeting of five people with four objects.
  - The message size was reduced by 48% to 69%.
  - The processing time (of a message on a PDA) was reduced by 66% to 87%.

10

Message sizes and corresponding processing times were measured in the case of a simulated conference of five people with four objects. The average TCP message size of a full notification was 2200 bytes, and that of a partial notification was 680 bytes (with one replace tag) to 1150 bytes (with five replace tags). The message size was thus reduced by 48% to 69%. The average elapsed time from when a Zaurus (SL-C860) UA receives a NOTIFY message to when an 200 OK message is received for a full notification message was 130 ms, and that for a partial notification message was 17 to 31 ms. Processing time was thereby reduced by 66% to 87%. This reduction ratio is smaller than the size reduction ratio because the XML part of the message, which requires much more parsing time, is much smaller.



Conclusion

- A SIP/SIMPLE-based method of conference room management has been developed for voiscape.
  - Rooms, users, and objects are managed by their “soft state”; i.e., they are deleted when a time out occurs.
  - This makes object and room management natural and light-weight.
- The conference room management has been partially implemented in VPII.
- To reduce the messaging overhead, the partial notification mechanism of SIMPLE is used in VPII.

11

5. Conclusion

A SIP/SIMPLE-based method of conference-room management have been developed for a voice-centered communication medium called voiscape. Conference members are informed of room membership, presence of a user, e.g., location and direction in the sound room, and presence of an object in the room, by SIMPLE messaging, i.e., by SUBSCRIBE, NOIFY, and PUBLISH requests. A user can not only obtain a room list and enter (select) or exit from a room, but also create, modify, or delete rooms by SIMPLE messaging. Rooms, users, and objects are managed by their “soft state”; i.e., they are deleted when expired. This makes user, object, and room management natural and light-weight in terms of computational and programming resources. This conference-room management method has been partially implemented in VPII.

To reduce the messaging overhead, the partial notification mechanism of SIMPLE is used in VPII. This reduces SIP message sizes by 48% to 69% and reduces processing time by 66% to 87% in the case of a simulated meeting with five people and four objects.

Acknowledgment

The author wishes to thank Nobuyuki Oshima, Hiroshi Sukeda, Akira Kagami, and Takashi Hotta of Hitachi, Ltd. for supporting this research and development. The author also wishes to thank Vasu Murthy, Hidefumi Higuchi, and other engineers at Zygox, Inc., who refined the design of VPII, implemented it, and managed this project.

References

[Day 00] Day, M., Rosenberg, J., and Sugano, H., “A Model for Presence and Instant Messaging”, RFC 2778, IETF, February 2000.

[Kan 04] Kanada, Y., “Multi-Context Voice Communication Controlled by using an Auditory Virtual Space”, *2nd International Conference on Communication and Computer Networks* (*CCN 2004*), pp. 467-472, 2004.

[Kan 05] Kanada, Y., “Multi-Context Voice Communication In A SIP/SIMPLE-Based Shared Virtual Sound Room With Early Reflections”, *15th ACM International Workshop on Network and Operating System Support for Digital Audio and Video* (*NOSSDAV 2005*), pp. X-X, 2005.

[Lon 04a] Lonnfors, M., Costa-Requena, J., Leppanen, E., and Khartabil, H., “Partial Notification of Presence Information”, Work in progress, IETF.

[Lon 04b] Lonnfors, M., Leppanen, E., and Khartabil, H., “Presence Information Data Format (PIDF) Extension for Partial Presence”, Work in progress, IETF.

[Nie 04] Niemi, A., Ed., “Session Initiation Protocol (SIP) Extension for Event State Publication”, RFC 3903, IETF, October 2004.

[Roa 02] Roach, A. B., “Session Initiation Protocol (SIP)-Specific Event Notification”, RFC 2543, IETF, June 2002.

[Ros 02] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., “SIP: Session Initiation Protocol”, RFC 3261, IETF, June 2002.

[Ros 04] Rosenberg, J., “A Presence Event Package for the Session Initiation Protocol (SIP)”, RFC 3856, IETF, August 2004.

[Shi 00] Shirmohammadi, S. and Georganas, N.D., “Collaborating in 3D Virtual Environments: A Synchronous Architecture”, *IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (*WETICE 2000*), pp. 35-42, 2000.

[Sug 04] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and Peterson, J., “Presence Information Data Format (PIDF)”, RFC 3863, IETF, August 2004.