

仮想の“音の部屋”によるコミュニケーション・メディア *Voiscape* におけるポリシーベース・セッション制御

金田 泰

日立製作所 システム開発研究所

〒215-0013 神奈川県川崎市麻生区王禅寺 1099 番地

E-mail: kanada@sdl.hitachi.co.jp

あらまし 電話にかわるべき音声コミュニケーション・メディアの確立をめざした研究の一部として、多者間通信が可能な常時接続型コミュニケーション・メディア *voiscape* のアーキテクチャとプロトタイプを開発した。Voiscape においては 3 次元オーディオ技術によってつくられた仮想的な“音の部屋”を使用する。ユーザがマウスをつかって音の部屋内を移動すると、プレゼンスサーバを介して部屋内の位置などのプレゼンス情報が部屋内の他のユーザにつたえられる。相手にちかづいたり相手からとおざかったりすると、あらかじめ端末において定義されたポリシーにしたがって、SIP を使用して自動的に通信セッションの開始・終了などの動作がとられる。このポリシーベース・セッション制御によってプライバシー保護や通信量削減が可能になる。接続開始を要求する際に相手からも同時に接続開始を要求されることが頻繁におこるので、2重に接続したり“話し中”になったりせずに接続が確立する方法をくふうしている。

キーワード 電話, IP 電話, 音声通信, リアルタイム通信, 3次元オーディオ, 3Dオーディオ, 多者間通話, Voiscape.

Policy-Based Session Control in a Virtual “Sound Room” Based Communication-Medium Called *Voiscape*

Yasusi Kanada

Hitachi, Ltd., Systems Development Laboratory

Aso-ku Ozenji 1099, Kawasaki, 215-0013, Japan

E-mail: kanada@sdl.hitachi.co.jp

Abstract As part of research toward establishing voice communication media that shall replace telephone, we developed an architecture and a prototype of a continuously-connected multi-user communication medium called *voiscape*. A virtual “sound room” that is created by spatial audio technology is used in *voiscape*. When the user moves within the sound room by using a mouse, the presence information including the position in the room is distributed to other users of the room. If the user becomes closer to or more distant from another user, the communication session begins or ends automatically by using SIP according to predefined policy stored in the terminals. This policy-based session control enables privacy protection and reduction of communication. When a local site requires a session start, the remote site often requires a session start concurrently, so a method of establishing a connection without connecting doubly nor becoming busy was devised.

key words Telephone, IP telephone, Voice communication, Real-time communication, Spatial audio, Multi-user communication, Voiscape.

1. はじめに

電話は A. G. Bell によって発明されてから約 130 年たつが、いまでも人間どうしのコミュニケーションのためのメディアとしてもっとも重要である。しかし、電話のユーザ・インタフェースは発明当時からほとんど改善されることがなかった [Kan 93]。すなわち、電話によって会話するには、まず相手に回線を接続し、1 個のマイクと 1 個のスピーカをつかって 1 対 1 で会話し、会話がおわると回線を切断する。このインタフェースでは 2 個ある人間の耳とそれを前提にした聴覚能力がいかにされないだけでなく、多者間の会話が困難で、回線切断中はまったく情報がつたわらないという、メディアをおさない直接の会話とはまったくことなる不自然な状況がうみだされている。この不自然なインタフェースがこれまで変更されなかったのは、それがひとびとにうけいれられてきたからという理由もあるだろうが、電話網がかたいネットワークだったため、とくにそれが回線交換

網だったために変更できなかったのだとかがえられる。

しかし、いまや電話網は IP ネットワークによって置換されようとしている。固定電話はすでに IP 電話によって侵蝕されているが、携帯電話もいずれ IP 化することはまちがいないであろう。電話網がやわらかい IP ネットワークによって置換されれば電話のインタフェースを制約していた原因も消滅し、直接のコミュニケーションに匹敵する、人間の聴覚能力をいかした、多者間の自由な会話ができる、常時接続を前提とした新メディアによって電話はとってかわられるであろう。1 対 1 に制約されないということは、このメディアが基本的に会議 (conferencing) メディアであることを意味している。

このような状況のもとで、著者は電話にかわる音声コミュニケーション・メディアの一案として、3D (3 次元) オーディオによって実現される“音の部屋” (仮想音空間, virtual auditory space) を使用した音声コミュニケーション・メディアの開発をすすめている。作曲家

Murray Schafer が音によってつくられる風景を soundscape とよんだ [Mur 77] のにならって、このメディアを voiscap (声の風景、声景) とよびたい。この報告においては voiscap の通信アーキテクチャを記述し、プロトタイプの実装と実験結果についてもべる。

2. Voiscap のアーキテクチャ

この章では voiscap の機能、操作手順の概要とポリシーによる制御についてべる。Voiscap においては複数のサーバと複数のプロトコルを使用するが、この章ではそれらをくべつせず、一括して“サーバ”とよぶ。

2.1 機能の概要

クライアントを起動すると自動的にサーバにログインし、サーバからおくられた入室可能な部屋のリストが表示される。ユーザがそのうちのひとつを選択すると、その部屋に入室している他のユーザとのあいだで通信が開始され、部屋の様子が 3D オーディオによる声景 (voiscap) とグラフィクスとによって表現される。ユーザは部屋のなかを自由に移動し回転することができる。3D オーディオを使用している、移動して話者にちかづけば声はおおききこえるし、回転すればきこえる方向がかわる。しかし、音声だけでは話者の位置が正確にはわからないので、それを補助するためにグラフィクスを使用する。部屋は会議室やメイリングリストに相当するものであるが、3D オーディオを使用しているため、ひとつの部屋のはなれた場所で複数の会話あるいは会議を同時に進行させることもできる。

Voiscap においては、常時接続環境をいかして、部屋にいるすべてのユーザとのあいだでつねに通信するのが基本である。しかし、端末の能力やプライバシー保護からの要請によって、通信を制限する必要があるとかがえられる。したがって、距離がはなれたユーザとの接続を切断したり、距離がちかまったユーザと接続したりする機能をとり入れる。

2.2 操作手順

クライアントにはあらかじめユーザ名が入力されている。クライアント起動時にこのユーザ名 (SIP アドレス) をつかってサーバに対して自動的にログイン操作がおこなわれる。試作したプロトタイプのクライアントは Windows PC 上で動作する。そのクライアントのウィンドウを例として図 1 にしめす。

ログインするとサーバから部屋リストが送付されるので、クライアント・ウィンドウの左側にそのリストが表示される。ユーザは部屋リストのなかから部屋を選択して入室する。現在は一度に入室することができる部屋は 1 個だけである。したがって他の部屋を選択して入室するとそれまで入室していた部屋からは自動的に退室する。また、すべての部屋から退室して、全通信を終了させることもできる。

ユーザが部屋を選択して入室すると、部屋内の (近接した) 他のユーザとのあいだで自動的に音声通信が開始され、仮想空間に位置づけられた他のユーザが 3D オーディオとグラフィクスとによって表示される。ヘッドセットを通じて 3D 音声ユーザの両耳につたえられる。プロトタイプにおいてはそれと同時にユーザの前方の様子を 3D グラフィクスによって表示している。ユーザ自身は表示せず、直方体と円錐とをくみあわせて他のユーザのプレゼンスを表現している。これだけでは他のユーザのむきがわからないが、直方体の上部にユーザ名を表示して、それがだれであるかがわかると同時に他のユーザのむきがわかるようにしている。

ポインティング・デバイスを使用することによって、部屋のなかで自由に移動したり、方向をかえたりすることができる。この仮想空間における位置や方向は実空間におけるそれらとは無関係である。現在はポインティング・デバイスとしてマウスを使用している。マウス

の左ボタンによって前後にドラッグすれば前後に移動できる。また左右にドラッグすれば左右にむきをかえることができる。ひとが実空間内を移動するときは、通常、前方への歩行とむきの変更とによって移動するので、それに対応する操作ができるようにしている。

2.3 ポリシーにもとづく通信・表示の制御

プライバシー保護や通信量削減などの目的のため、ポリシーにもとづいて通信や表示を制御するのが有効であろう。Voiscap においては部屋にいるすべてのユーザとのあいだでつねに通信するのが原則である。しかし、端末の能力がひくい多数のユーザとのあいだで通信をおこなえず、通信相手や通信量を制限する必要がある。また、ネットワークの回線容量などによって通信量の制限が必要になる。さらに、プライバシーをまもるには部屋内で一定値以上の距離にいるユーザとの接続を切断したり、通信内容を制限したりするほうがよいであろう。たとえば、相手との距離によって音声の明瞭度や帯域幅を制御することによって、よりきこまかくプライバシー保護や通信量削減を実現できる。これらをクライアントに内蔵したポリシーによって制御するのがよいとかがえられる。ポリシーとは、特定の条件がなりたつときにどのような通信や表示の動作をするかを記述した規則またはそのあつまりのことである。

ポリシーとしてはさまざまな制御をおこなうものがあるが、現在は部屋内でユーザどうしの距離が一定値以下になったときに接続し一定値をこえたときに切断するというポリシーだけを実装している。このポリシーによって、相手の存在に気がつかないほど遠方のユーザによって話をぬすみきかれることがさげられる。そのため、このポリシーはプライバシー保護のために有効だとかがえられる。また、接続しても話をきけないほど遠方のユーザと通信するのはむだであろうが、このポリシーを使用することによってそれもふせげられる。そのため、通信量削減にも有効である。

ポリシーは通信相手ごとに個別に調停される。相手がこのポリシーをもつときは、よりよいプライバシー保護を実現するポリシーが双方に適用される。たとえばユーザ間距離による接続・切断に関するポリシーを双方がもっているばあいは、みじかいほうの距離にあわせる。この調停はクライアント間で P2P で実行され、サーバは関与しない。

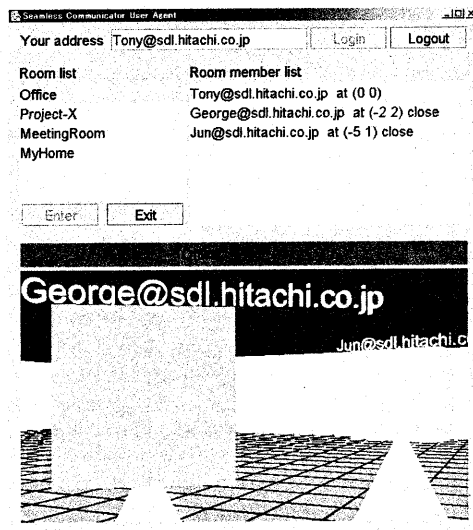


図 1 クライアント・ウィンドウ

3. プロトタイプの実装

この章においては、voiscape プロトタイプの構成と各部の実装について、セッション制御に重点をおいて説明する。

3.1 全体構成

図3に試作システムの全体構成をしめす。試作システムはおおきくわけてサーバ群と複数のクライアント(端末)によって構成されている。現在、クライアントはMicrosoftのWindows XP または Windows 98を動作させたPC上で動作させている。

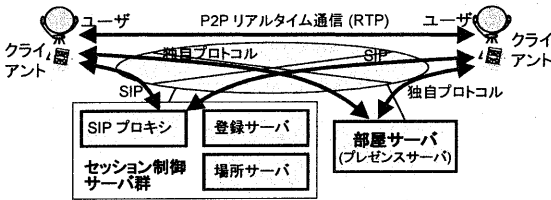


図3 試作したプロトタイプの全体構成

サーバ群のなかには、セッション制御のためのSIPメッセージを中継するSIPプロキシ(Proxy)、ネットワークにクライアントが接続するとその情報をSIPプロキシ経由でうけて場所サーバに登録する登録サーバ(Registrar)、クライアントのIPアドレスなどの情報を管理する場所サーバ(Location Server)、クライアントがネットワークに接続されているかどうかなどのプレゼンス情報を管理し、voiscapeにおける仮想会話空間すなわち部屋とその利用者を管理するプレゼンスサーバ(または部屋サーバ)がある。SIPプロキシ、登録サーバと場所サーバの動作は通常のSIPとほぼおなじであり、サーバ群のなかでvoiscapeに特徴的なものはプレゼンスサーバである。

試作システムにおける動作シーケンスの概要を図4にしめす。Voiscapeにおいては、基本的にはネットワーク・アクセス可能な場所でユーザがクライアントの電源を投入すると(図4の(1)以下のシーケンスを参照)、クライアントやユーザの情報がSIPプロキシ、登録サーバを通じて場所サーバに登録される。また、ユーザがクライアントに対して特定の部屋ROOM1への入室を指示すると(図4の(2)以下のシーケンスを参照)、クライアントからプレゼンスサーバ

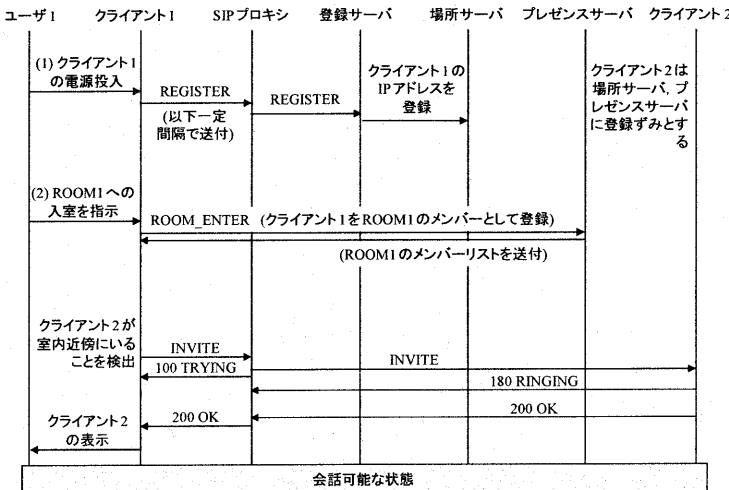


図4 試作したプロトタイプにおける接続までの動作シーケンス概要

にユーザをROOM1のメンバーとして登録するとともに、ROOM1での座標がおくられる。すると、プレゼンスサーバはROOM1のメンバーリストすなわちそのクライアントに対しておなじ部屋にいる他のユーザの部屋内での位置を通知し、その結果として部屋内で近傍にいるクライアント間でP2Pで通信が開始される。試作システムにおいてはここで使用されるメディアとして音声と動画の両方が実験可能になっているが、動画はオプションである。この通信は基本的にはユーザが入室してから退室するまで継続される。詳細な動作に関しては次節以降で説明する。

プログラムの大半の部分はJavaによって記述している。これは、各サーバの開発・改造を容易にし、複数のプログラミング言語のくみあわせから生じる煩雑さをさけるためである。

開発にあたっては、Javaで記述されたSiptrex (<http://www-siptrex.net/>)のSIPスタックを使用し、Siptrexのサーバ群を参考にした。SiptrexはUCL(University College London)において開発され、SIPスタックだけでなく単純なSIPプロキシ、登録サーバ、場所サーバ、クライアントがJavaによって記述され、公開されている。SIPスタックはJAIN SIP [SUN 03]の仕様に準拠している。これらにくみあわせることによって、SiptrexだけでIP電話の実験ができる。Javaによって記述されたSIPスタックとしては米国の国立機関であるNIST(National Institute of Standards and Technology)において開発された著作権フリーのものがあり[NIST]、Siptrexはその旧版をもとにしている。NIST SIPでなくSiptrexを使用したのは、そのほうが各種のサーバやクライアントなど、手本とするべきプログラムがおおきく、しかもそれらの構造が比較的単純だからである。

3.2 サーバによるセッションとプレゼンスの管理

Voiscapeにおける各サーバの機能についてのべる。

3.2.1 SIPサーバ

SIPメッセージの処理とそれに必要な状態管理のため、SIPプロキシ、登録サーバ、場所サーバを使用している。これらは旧版のSIP(RFC 2543)に準拠しているが、特に新規な機能はもっていないので、こまかい説明は省略する。

3.2.2 プレゼンスサーバ

プレゼンスサーバは利用可能な部屋のリストと各部屋の属性を管理し、部屋内のユーザの位置情報を収集して管理し、他のユーザに通知する。プレゼンスサーバにおいてはユーザのクライアントがネットワークに接続しているかどうかを把握する必要があると同時に、接続しているときにはそのユーザがどの部屋のどの位置にいるかをあわせて把握する必要がある。そのため基本的な方法としてつぎの2とおりがかんがえられる。

- イベントにもとづく方法: ユーザはつねに移動しているわけではないので、移動を検出したときにイベントを発生させ、それをネットワーク内で伝搬させる。ある部屋に入室したユーザはその部屋への他のユーザの入退室イベントの通知を予約すればよい。この方法はイベント発生がまればあいは通信頻度がひくくてよいが、イベントが頻繁に発生するときはオーバーヘッドがおおきい。

- ポーリングにもとづく方法: クライアントが

他のユーザの位置を把握したいときにプレゼンスサーバに対して要求メッセージをおくる。この方法によれば通信オーバーヘッドが一定値以下におさえられる。しかし、ユーザの移動がないときにもむだな要求メッセージをおくることになり、またクライアントが他のユーザの移動にただちに追従できないという問題点がある。ただし、この問題点はポーリング周期の調整によって軽減させられる。

SIP においてはその拡張仕様としてイベントの通知とその予約 (subscription) のためのメッセージがきめられている [Roa 02]。イベント通知には NOTIFY メッセージ、イベント通知予約には SUBSCRIBE メッセージが使用される。プレゼンスサーバがあつかう情報は、これらのメッセージを使用すれば実装できる。しかし、使用した SIP スタックにイベント通知拡張が実装されていなかったため、このプロトタイプにおいてはクライアントからの通知兼ポーリングにもとづく独自のプロトコルによって部屋の管理をおこなっている。ポーリングの周期はクライアントが決定するが、現在はそのクライアントを使用するユーザに最接近した他のユーザとの距離によって周期を変化させている。他のユーザが接近しているときは1秒にしている。ユーザが急速に接近してくるばあいなどはかならずしも十分にみじかいたとはいえないが、プレゼンスサーバの負荷を考慮してこの周期にしている。

プレゼンスサーバとクライアントとの通信がすべてクライアントからのメッセージによって起動されるようにしたので、この独自プロトコルにおけるメッセージはつぎの8種類とした。

●クライアントからの ROOM_ADD とそれに対するサーバ応答: ROOM_ADD メッセージはクライアントからサーバへの部屋への入室通知とあたらしい部屋の生成のために使用する。このメッセージは、部屋名と入室するユーザ名とを指定し、指定ユーザ名をもつユーザによって送信される。指定部屋名をもつ部屋がすでにあれば、ユーザがその部屋に入室することを意味する。指定部屋名をもつ部屋がまだないときは、あたな部屋を生成して、その部屋にそのユーザが入室することを意味する。部屋名が指定され、ユーザ名が指定されないときは、部屋の生成だけをおこなう。異常がおこらなければサーバはこのメッセージに対して成功応答をかえす。

●クライアントからの ROOM_REMOVE とそれに対するサーバ応答: ROOM_REMOVE メッセージはクライアントからサーバへの退室通知のために使用する。このメッセージは、部屋名とその部屋から退室するべきユーザ名とを指定し、指定ユーザ名をもつユーザによって送信される。指定部屋名をもつ部屋はユーザの退室後も存在しつづける。異常がおこらなければサーバはこのメッセージに対して成功応答をかえす。

●クライアントからの PRESENCE_REFRESH とそれに対するサーバ応答: PRESENCE_REFRESH メッセージはクライアントからサーバにそのクライアントを使用するユーザのプレゼンス (ネットワーク・アクセス可能かどうかの区別と部屋内の位置など) の変更を通知し、他のユーザのプレゼンスと部屋のリストの送信をもとめるのに使用する。サーバはこれに対して全ユーザのプレゼンスと全部屋のリストをふくむ応答をかえす。

●クライアントからの PRESENCE_REFRESH とそれに対するサーバ応答: PRESENCE_REFRESH メッセージはクライアントからサーバにそのクライアントを使用するユーザのプレゼンス (ネットワーク・アクセス可能かどうかの区別と部屋内の位置など) の変更を通知し、他のユーザのプレゼンスと部屋のリストの送信をもとめるのに使用する。サーバはこれに対して全ユーザのプレゼンスと全部屋のリストをふくむ応答をかえす。

¹ DirectX (Microsoft の登録商標) と OpenGL (Silicon Graphics 社の登録商標) はグラフィクス API, OpenAL はオーディオ API, LWJGL (Light-Weight Java Game Library) は SourceForge.net において開発された、Java から OpenGL, OpenAL を使用するための API である。

●クライアントからの ROOM_DESTROY とそれに対するサーバ応答: ROOM_DESTROY メッセージはクライアントからサーバに部屋の抹消をもとめるのに使用する。異常がおこらなければサーバはこのメッセージに対して成功応答をかえす。

3.3 クライアント

クライアントの構造、動作の概要、ポリシーによる制御の順に説明する。

3.3.1 クライアントの構造

クライアントの構造は図5のとおりである。すなわち、クライアントは入力デバイスとしてマイクロホンと位置指定デバイス(マウス)をもち、出力デバイスとしてイヤホンまたはヘッドホンとディスプレイをもつ。マイクロホンからの出力は音声入力部によってデジタル信号にエンコードされる。エンコードされた信号は通信相手が存在するときには音声通信部におくる。現在は 8000 Hz でサンプリングし、ITU-T の標準である G.711 u-law 64 kbps の信号として RTP (Real-Time Transport Protocol) [Sch 96] によって他のクライアントに P2P で送信している。他のユーザの音声信号は RTP によって受信し、3 次元オーディオ / グラフィクス表示部におくる。音声の入出力と RTP 送受信には Sun Microsystems が Java SE (Standard Edition) の拡張 API として提供している JMF (Java Media Framework) [Gor 98] [Fai 00] を使用している。

プレゼンスサーバとのあいだのメッセージはこのプロトタイプに独自のプロトコルによって部屋モデラが送受信する。部屋モデラは位置指定デバイスからの出力を受けとり、ユーザの部屋内における位置をもとめて、独自プロトコルによってプレゼンスサーバに送信する。プレゼンスサーバからは他のユーザの部屋内における位置を受信する。

3 次元オーディオ / グラフィクス表示部においては、受信した RTP 信号を部屋モデラの情報にしたがって 3 次元音場に位置づけ、その信号に対応するユーザのプレゼンスを 3 次元グラフィクスによって表示する。この表示には Sun Microsystems によって Java SE (Standard Edition) の拡張 API として提供されている Java 3D を使用している。Java 3D には 3D オーディオ表示の機能も提供されているが、リアルタイム通信とくみあわせて使用できないことがわかったため、独自開発した Java 3D のオーディオ API (図 5 の JA3D) を JMF にはめこんで使用した。サウンドカードとしては現在、HRTF (Head Related Transfer Function) [Beg 00] 機能をもつものを使用している。HRTF 機能を使用すれば左右・前後の方向感をえることができるが距離感は一再現されにくいので、残響を使用し距離感をえている。

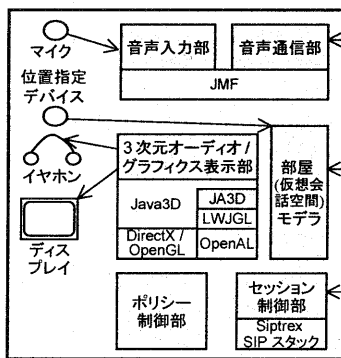


図5 クライアントの構造¹

セッション制御部においては SIP (Session Initiation Protocol) [Ros 02] を使用し、SIP プロキシを経由して他のクライアントとのあいだの RTP 通信の開始・終了等を制御する。SIP スタックとしては Siptrex SIP スタックを使用している。ポリシー制御部はおもにセッション制御部の機能を制御するためのポリシーを保持し、それにもとづいて通信などを制御する。セッション制御とポリシー制御の詳細に

については3.3.3節においてのべる。

3.3.2 クライアントの動作概要

クライアントはユーザによって起動されたあと、ユーザインタフェースとネットワークからのイベントにもとづいて動作する。ユーザ名および各サーバのIPアドレスはクライアント起動のためのバッチファイルに記述されていて、起動時に自動的にログインする。SIPのREGISTERメッセージをSIPプロキシに送信することによってログインする。このとき、SIPプロキシはこのメッセージを登録サーバにつたえ、登録サーバはユーザのネットワーク上の位置(IPアドレス)を場所サーバに登録する。本来はログイン時にユーザ認証をおこなうべきだが、現在は認証はおこなっていない。

ユーザインタフェースから発生するおもなイベントはつぎの3つである。

- 入室: メニューにおいて指定されたなまえをもつ部屋に入室する。プレゼンスサーバにENTER_ROOMメッセージを送信する。
- 退室: 現在入室している部屋から退室する。プレゼンスサーバにEXIT_ROOMメッセージを送信する。
- 移動と回転: マウスのドラッグ操作により前後の移動と回転をおこなう。マウスの左ボタン開放時に現在の位置をPRESENCE_REFRESHメッセージによってプレゼンスサーバに送信する。

クライアントはユーザの移動にもとづいてイベントを自律的に生成する。すなわち、プレゼンスサーバに対してその端末を使用するユーザの部屋内の位置を送信し、他のユーザの位置を受信する。

他のクライアントに起因するおもなイベントはつぎの2つである。ただし、現在のプロトタイプにおいてはこれらのイベントはポーリングによって他のユーザの位置を取得したときに発生するので、その契機はクライアント自体がつくっていることになる。

- 他のユーザの接近: 他のユーザとの通信はポリシーによって制御される。現在あたえられているポリシーはつぎのように動作する。他のユーザがクライアントのポリシーによってきめられた距離内にちかづくとき、SIPのメッセージングによってそのユーザとの通信が開始される。
- 他のユーザの離遠: 前記のポリシーを使用するばあいは、他のユーザがクライアントのポリシーによってきめられた距離外にとおざかると、SIPのBYEメッセージをそのユーザにおくり、そのユーザとの通信が切断される。

3.3.3 セッション制御部

クライアントのセッション制御部はSIPユーザエージェントをふくんでいる。SIPスタックとしてはSiptrexのものを使用している。前節でのべたように、クライアントはそれを使用するユーザと他のユーザとの部屋内の距離などに応じてクライアント間の通信を開始または終了する。クライアントが自律的に送信するメッセージは、他のユーザと接近したときにおくるINVITEメッセージと、他のユーザから離遠したときにおくるBYEメッセージの2種類である。一方、他のクライアントからSIPメッセージを受信することによって生じるイベントにはつぎの4つがある。

- INVITEメッセージ受信: 相手が自分の接近を検知するとINVITEメッセージを送信してくる。
- INVITEに対する応答メッセージの受信: INVITEメッセージに対しては通常200 OKメッセージによってこたえる。IP電話であればクライアントはただちに180 RINGINGメッセージをかえし、ユーザが受話器をとったときに200 OKメッセージをかえすが、

voiscapeにおいては自動的に接続するため、すぐに200 OKメッセージがかえされる。

- ACKメッセージ受信: 200 OKメッセージの送信だけでは通信がきちんと確立されないばあいもあるので、SIPではACKメッセージ送信によって通信が最終的に確立する。
- BYEメッセージ受信: BYEメッセージを受信すると、通信の終了にともなう処理をおこなう。この時点で通信はすでに切断されているはずである。

前節でのべたように、2人のユーザ間の通信において両者がポリシーをもっているときには調停が実施される。現在はユーザ間距離に関するポリシーだけが実装されているので、そのときの具体的な調停法について説明する。ポリシーにおいて指定された接続すべき距離が両者のあいだでちがうばあいは、みじかいほうの距離によって実際の通信開始時刻がきめられる。すなわち、ポリシーによってきめられた通信開始距離が d_1 のユーザ u_1 と d_2 ($d_2 > d_1$)のユーザ u_2 とがしだいに接近するとき、通信シーケンスはつぎのようになる。まず両者の距離が d_2 以下になったとき、 u_2 が u_1 にINVITEメッセージをおくる。ところが u_1 はこのとき接続すべき距離にはまだ達していないため、これを拒否する。両者の距離が d_1 以下になると u_1 が u_2 にINVITEメッセージをおくる。このときは u_2 にとっても接続すべき距離に達しているため、通信が成立する。

通信を切断するばあいは、同様にみじかいほうの距離によって実際の通信終了時刻がきめられる。すなわち、すなわち、ポリシーによってきめられた通信終了距離が d_1' ($d_1' > d_1$)のユーザ u_1 と d_2' ($d_2' > d_2$, $d_2' > d_1'$)のユーザ u_2 とがしだいに離れるとき、通信シーケンスはつぎのようになる。両者の距離が d_1' になると u_1 は u_2 にBYEメッセージをおくるとともに、通信を切断する。両者の距離が d_2' になったときには、通信はすでに切断されている。なお、ここでそれぞれの通信開始距離は通信終了距離よりおおきい、つまり $d_1' > d_1$, $d_2' > d_2$ がなりたつとする。これは、通信終了距離を通信開始距離とひとしくするとユーザどうしがその距離にあるときに通信の切断と接続をくりかえすおそれがあるからである。

以下、INVITEメッセージを送信した相手が、それがとどくまえにINVITEメッセージを送信しているばあいについて考察する。これは、電話でいえば双方が同時に受話器をあげたケースに相当する。電話であればこのばあいは双方が話し中になるため通信は確立されないが、voiscapeにおいてはこのケースにおいても通信が確立されなければならない。

この問題を解決するために、現在ではどちらのINVITEメッセージに対しても200 OKメッセージを返信し、さらにACKメッセージをおくるという方法をとっている。この方法においては、SIPメッセージングに関しては2重の接続関係が生じるが、RTPの接続は1重(ただし双方向)にする。すなわち、このメッセージングの結果としてただひとつの通信路がひらかれる。この方法においては2個のクライアントに関してメッセージが対称に交換される。この2重接続の機構はすべてクライアント側の実装されていて、SIPサーバは通常のシーケンスを実行するだけである。すなわち、サーバは通常の接続が2個あるようにあつたらう。他の方法として、なんらかの方法で2個のINVITEメッセージに順序ないし優先をつけて対称性をやぶり、一方をとりけす方法もある[Len 03]。SIPにおいてはユニークなcall idが使用されるので、INVITEメッセージのcall idの大小によって選択すればよい。このほうが通常の方法だろう。

SIPメッセージングが2重になっているときは基本的にはBYEメッセージも2重化する必要がある。これは、このメッセージングに関与しているSIPプロキシがステートフルであるばあいに、そのプロキシがもつ状態をクリアする必要があるからである。具体的には

BYEメッセージの送信時または受信時にもう一方の接続に関してもBYEメッセージを送信すればよい。受信側からBYEメッセージをおくるほうが対称性はよい。

4. 関連研究¹

HP LabsのLowら[Low 98]は電話より柔軟なメディアの開発をめざして、3Dオーディオを使用した会議システムを開発している。セッション制御にはSIP、SDPを使用することも念頭においているが、このプロトタイプではMBONE[Kum 95]をベースとしている。

Lennoxら[Len 03]は会議サーバを使用せず、セッション制御も音声通信も端末間で直接ユニキャストする会議の方法を開発している。この方法は全情報をP2Pで交換する“完全メッシュ型会議”である。しかし、集中型セッション制御と同様に会議参加者のリストを参加者間で共有するために、その管理方法はかなり複雑になっている。Lennoxらも3.3.3節でのべたような2重に接続するという問題(the double-dialog glare problemとよんでいる)の解決をせまられているが、そのために非対称な方法を使用していて、そのためにアルゴリズムの複雑さを増大させている。

UCLにおいて開発されたRAT(the Robust-Audio Tool)[Har 96]はマルチメディア会議のためのオーディオ・ツールである。話者の音声を容易に分離できるようにするために比較的単純な3Dオーディオ技術を使用しているが、会話中に移動することは意図していない。研究の重点が通信にあるため、この論文には、マルチメディア会議の実装上の技術課題やヒントが多々、記述されている。

5. 試作の結果と検討

まだユーザによる試用が可能な状態ではないので、試作の過程やシステム・テストにおいてわかった現象を中心にのべる。

- 接続にかかる時間の評価:** 2台のクライアントが接続するべき距離に達してから実際に相手の音声がききとれるようになるまでの時間は約7秒である(他に接続がない状態で測定した)。この時間は実用上ながすぎる。SIPのメッセージは1秒程度で交換される(この時間はほぼポーリング周期でできる)が、受信側に最初の音声データがとくまで約6秒の時間がかかっている。その原因はつきとめられていない。なお、切断時は交渉をしないため即座にきれる。
- プレゼンスサーバの実装に関する結果と検討:** ユーザの移動がすくないときはユーザ位置の把握のためにイベントにもとづく方法をとるのがよいが、現在はポーリングにもとづく方法をとっている。そのため、クライアントが3台でもユーザどうしが接近しているとIntel Pentium III 800 MHzのサーバ負荷が50%をこえることがある。クライアント数を数10にふやせばこのPC 1台では処理しきれなくなるであろう。したがって、ユーザ位置把握をイベント・ベースにするとともに、イベント発生頻度をおさえる必要がある。
- 音質劣化と遅延:** プロトタイプにおいてもっとも重大かつ解決困難だった問題が音質劣化と最大で約6秒におよぶ遅延に関する問題だった。しかし、これは通信に直接関係しない問題なので詳細は省略する。
- リアルタイム処理上の問題点:** プロトタイプにおいては約70個のスレッドが生成される。このスレッド数はOSにおいてリアルタイム処理をどここおりなくすすめるためには過大だとかんがえられる。前項の音質劣化や遅延にはスレッドのスケジューリングが関与し

ている可能性がたかいとかんがえられる。

6. 結論

リアルタイム音声通信の技術と3Dオーディオ/グラフィクスの技術をくみあわせ、IPネットワークの常時接続性をいかにした新メディアvoiscapcを提案し、Javaでプロトタイプを開発した。Voiscapcにおいては3Dオーディオの使用によって人間の聴覚能力がいかされる。また、端末上に定義されたポリシーにしたがって自動的に通信セッションの開始・終了などの動作がとられるので、プライバシー保護や通信量削減が可能になる。さらに、接続開始時に2重に接続したり“話し中”になったりせずに接続する方法をくふうした。

前記の各技術をくみあわせて動作させることは想像していたよりはるかに困難であり、JavaのAPIがくみあわせられないという問題や、音質劣化やおおきな遅延が発生した。しかし、おおきな問題は一部をのぞいてすでに解決できた。今後のおもな課題は、これらの中でのこった問題を解決するとともに、サーバやネットワークの負荷を軽減して常時接続使用を可能にすること、小型化・ウェアラブル化をはかること、認知的な評価をおこなうことなどである。

参考文献

- [Beg 00] Begault, D. R., “3-D Sound for Virtual Reality and Multimedia”, NASA/TM-2000-XXXX, NASA Ames Research Center, April 2000, http://human-factors.arc.nasa.gov/i/hh/spatial-papers/pdfs_db/Begault_2000_3d_Sound_Multimedia.pdf
- [Gor 98] Gordon, R. and Talley, S., “Essential JMF – Java Media Framework”, Prentice Hall PTR, November 1998.
- [Har 96] Hardman, V., and Iken, M., “Enhanced Reality Audio in Interactive Networked Environments”, *Framework for Interactive Virtual Environments (FIVE) Conference*, December 1996.
- [Kan 93] 金田 泰, “仮想の‘音の部屋’によるコミュニケーション・メディア Voiscapc”, 電子情報通信学会 技術研究報告 (MVE研究会), 2003-10-7.
- [Kum 95] Kumar, V., “MBONE”, New Riders Publishing, 1995. 訳書: 橋本 博之 監訳, “インターネットマルチキャスト MBONE”, インプレス・ブレンティスホール, 1996.
- [Len 03] Lennox, J., and Schulzrinne, H., “A Protocol for Reliable Decentralized Conferencing”, *13th Int'l Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'03)*, pp. 72-81, June 2003.
- [Low 98] Low, C., and Babarit, L., “Distributed 3D Audio Rendering”, *7th International World Wide Web Conference (WWW7)*, 1998, <http://www7.scu.edu.au/programme/fullpapers/1912-com1912.htm>.
- [NIST] “Project: Internet Telephony / VOIP”, <http://www-x.antd-nist.gov/proj/iptel/>.
- [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.
- [Mur 77] Murray Schafer, R., “The Tuning of the World”, 訳書: 鳥越 けい子他訳, “世界の調律”, 平凡社, 1986.
- [Sch 96] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, IETF, January 1996.
- [SUN 03] O'Doherty, P., “SIP Specifications and the Java Platforms”, Sun Microsystems, 2003, <http://java.sun.com/products/jain/SIPAPIS.pdf>.

¹ここでは3Dオーディオを使用し通信にフォーカスした研究を中心に紹介する。仮想環境に関する研究については金田[Kan 93]が紹介している。