

仮想の“音の部屋”による
コミュニケーション・メディア voiscap の
JMF と Java 3D を使用した実装

金田 泰
(日立製作所 中央研究所)

発表のねらい

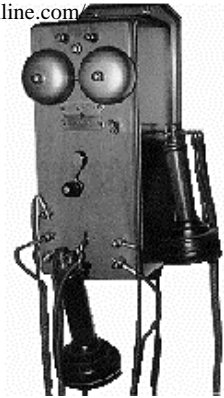
- Java を使用したマルチメディア・アプリケーション実装の紹介
 - ◆ Java によるリアルタイム通信と 3D オーディオ / グラフィックスのくみあわせの実現法
 - ◆ その問題点と部分的な解決策

電話のインタフェースの変化

■ 電話の基本インタフェースは発明以来まったく変化していない。

- ◆ 電話は A. G. Bell によって 1876 年 (約 130 年前!) に発明されて以来, 基本ユーザインタフェースはかわっていない。

1878 年ころの電話器
(<http://www.atcaonline.com/phone/coffin.html>)



- 相手に接続し, 1 個のマイクと 1 個のスピーカを使用して 1 対 1 で会話し, 接続をきる。
- このインタフェースにはいろいろ問題がある。

■ IP ネットワークによって基本インタフェースはかえられる。

- ◆ 現在の IP 電話は不必要に制約されたインタフェースを踏襲しているが, この状況は急速に変化するとかんがえられる。

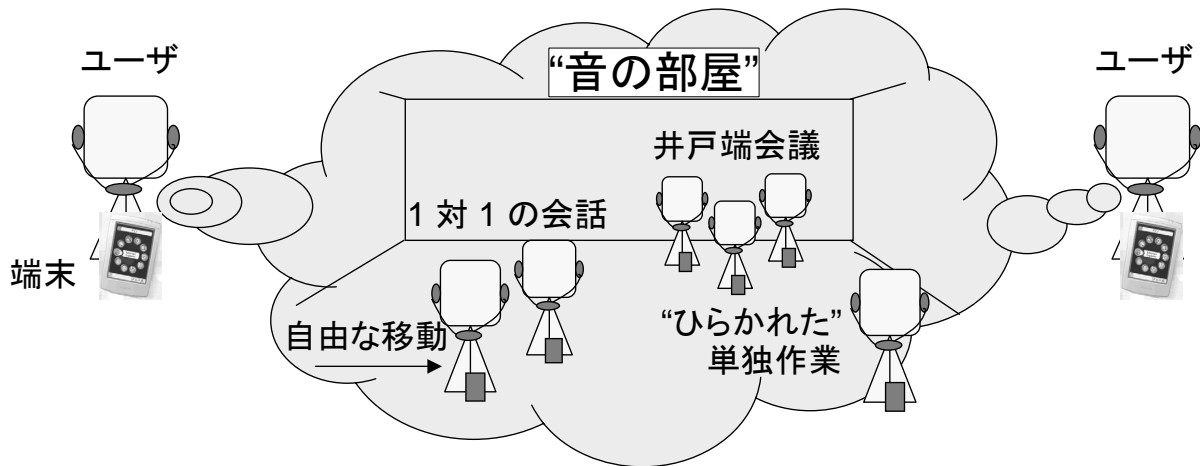
Voiscape: 携帯電話にかわるべきメディア

■ 電話にかわるべきメディアを voiscap とよぶ。

- ◆ 常時接続により回線の接続・切断は不要になり, つねに環境音などの情報 (“プレゼンス情報”) がおくれる。
- ◆ 多者間の自由な会話が可能になる – “会議” メディアになる。
- ◆ 人間の聴覚能力をもっといかすことが可能になる。
 - 両耳の使用など。
- ◆ Voiscape の語源
 - 作曲家 Murray Schafer による用語 soundscape にならう。
 - Soundscape = 音によってつくられる風景
 - Voiscape = 声の風景, 声景

想定する voiscap のすがた

■ 音によって 3D 仮想空間をつくり, そのなかで会話する.



■ リアルタイム通信と 3D オーディオのくみあわせ

- ◆ 3D オーディオを使用する理由: 3人以上で話をするときは, 方向感・距離感が重要.
 - それによって話者をくべつしやすい.
 - 2人以上が同時に話をして, カクテル・パーティ効果によって分離できる - 人間の聴覚能力をいかす.

Voiscap における会話のながれ

■ ログインして部屋リストを表示する.

■ 部屋を選択する.

- ◆ 右図では 4 つの部屋がある.
 - オフィス (Office)
 - プロジェクト X (Project-X)
 - 会議室 (MeetingRoom)
 - 家 (MyHome)

■ 入室すると部屋内が表示される.

- ◆ 部屋の様子が 3D オーディオとグラフィクスとで表示される.
- ◆ 部屋にいるメンバーのリストが表示される.

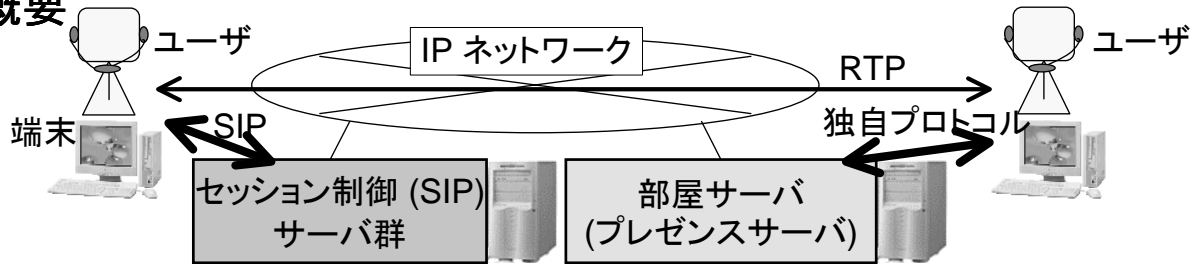


■ 部屋内で移動・回転して会話相手に接近する.

- ◆ 移動・回転にはポインティングデバイス (現在はマウス) を使用.

Voiscape のプロトタイプ

■ 概要



- ◆ 音声はユニキャスト P2P で RTP により通信.

■ RTP = Real-time Transport Protocol

■ 部屋サーバ (プレゼンスサーバ)

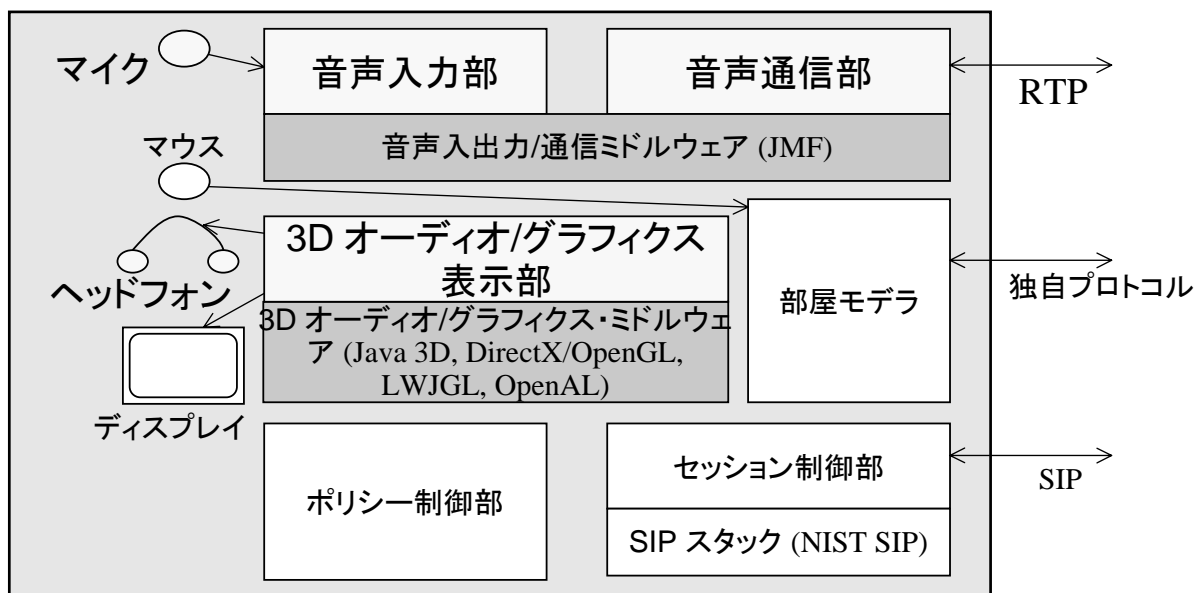
- ◆ 部屋属性と利用者 (ネットワーク接続, 在・不在, 位置など) を管理.
- ◆ ユーザエージェントとのあいだで独自プロトコルを使用.

■ ユーザインタフェース

- ◆ ヘッドセットによるバイノーラル音を使用.
- ◆ 3D グラフィクスによって前方を表示.
- ◆ Windows PC 上で動作.

ユーザエージェントの実装

■ 端末 (PC)



実装の方針

■ 新規開発部分すべてを Java で記述する.

- ◆ SIP スタックおよび SIP プロキシとしては NIST SIP を使用.
 - NIST SIP は米国 National Institute of Standards and Technology において開発された SIP スタックおよび SIP アプリケーション.
- ◆ 部屋サーバは Java を使用して独自開発.
- ◆ 音声の入力・通信には JMF を使用.
 - JMF (Java Media Framework) は J2SE の拡張 API.
 - J2SE = Java 2 Standard Edition
- ◆ 3D オーディオ/グラフィクスには Java3D の使用をこころみた.
 - Java3D も J2SE の拡張 API.

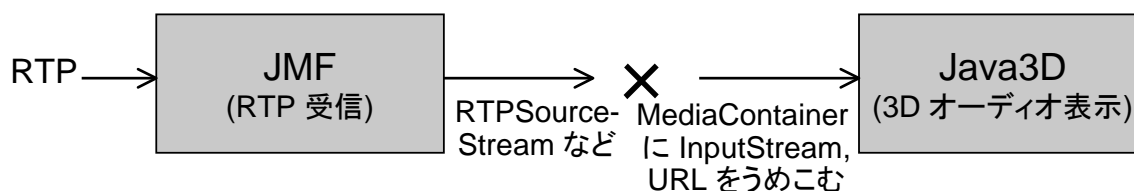
JMF + Java 3D による実装の問題点

■ 3D オーディオレンダラの問題点

- ◆ Java3D は 2 種類の 3D オーディオレンダラをもっているが、いずれも voiscap の実装には使用できない.
 - 2 種類のレンダラ: JavaSoundMixer と HeadSpaceMixer
- ◆ JavaSoundMixer: リアルタイムで動作するが 3D 化しない.
- ◆ HeadSpaceMixer: 3D 化するがリアルタイムで動作しない.
 - ファイルを最後までよんでから再生する.

■ JMF と Java 3D は直接つなげない.

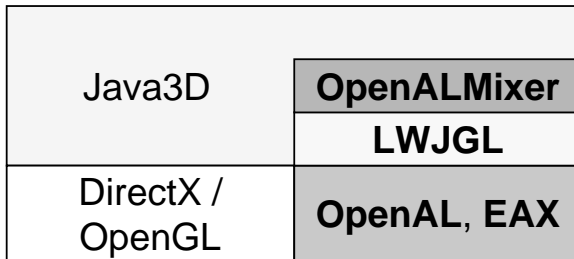
- ◆ JMF が出力可能なクラスと Java3D が入力可能なクラスにかさなりがない.



解決策

■ 3D オーディオレンダラの問題点

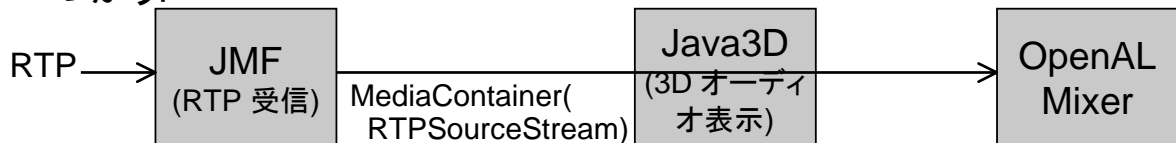
- ◆ 他の API を利用して独自のレンダラ OpenALMixer を開発した。



- **OpenAL**: 業界標準 API とするべく開発された PC オーディオ API.
- **EAX**: DirectSound, OpenAL の 3D オーディオむけ拡張 API (Environmental Audio Extensions).
- **LWJGL**: OpenGL, OpenAL を Java から利用するための API (Light-Weight Java Game Library).

■ JMF と Java 3D の接続法

- ◆ Java3D が本来はあつかえないクラス (RTPSourceStream) を “だまして” わたす。
- ◆ RTPSourceStream は Java3D をトンネルして OpenALMixer であつかう。



HRTF の実現法

■ HRTF (Head Related Transfer Function) とは?

- ◆ 人頭周辺における音の伝達関数.
- ◆ HRTF をシミュレートすれば, 前後・上下の方向感をあたえられる.
 - 左右の耳における音の時間差, 強度差だけでは前後・上下のくべつはつかない.

■ PC サウンドにおける HRTF

- ◆ メジャーなのは Sensaura 社の技術.
 - Sensaura 社は Creative Technology 社の子会社 (2003-12 以降)
 - 英国 Central Research Laboratories, Ltd. の技術を使用.
- ◆ この技術に対応した PC サウンド用の LSI を使用する.
 - C-Media CMI-8738, Analog Devices AD1985 (SoundMax™) など.

■ プロトタイプにおける HRTF の実現

- ◆ CMI-8738 搭載のサウンドカードを使用した。

実装上の問題点

■ 音質劣化と遅延

◆ 当初の状況

- 片方向で約 6 秒の遅延.
- バッファを強制的にクリアして遅延をなくすと、ほとんどききとれないほどの音質劣化.

◆ 現在の状況

- 音質はよくないが、通常はききとるうえで問題ない.
- 最高 6 秒の遅延があり、突発的な遅延はたびたびおこる.
- CPU 負荷は通常 15% 以下 (Pentium III, Athlon 等).

◆ 原因

- はっきりとはつかめていないが、寄与しているものは
 - Java / OS のスケジューリング
 - 3D グラフィクス
 - サウンドカード周辺の処理
 - 静寂検出をしていない.

まとめ

■ Java を使用してリアルタイム通信と 3D オーディオ / グラフィクスをくみあわせたアプリケーションを開発した.

- ◆ その実装法、問題点と部分的な解決策を紹介した.

■ この方法を改良すれば、さまざまなプロトタイピングに使用できるであろう.

- ◆ ただし、のこされた問題点を解決する必要がある.