

Java 対応携帯電話における マルチユーザ共有仮想空間の設計

細川武司[†] 佐藤文明[‡] 水野忠則[‡]

[†] 静岡大学大学院情報学研究科 [‡] 静岡大学情報学部

本稿では Java 対応携帯電話において、多人数でアバターを用いたコミュニケーションを提供する共有仮想空間の設計について述べる。Java 対応携帯電話では HTTP 通信のみの利用と、アプリケーションのダウンロード元サーバとのみの通信といった制限及び、メモリや計算資源の制限がある。これに対して、我々は仮想空間を背景とアバターに分離し、背景のレンダリングをサーバにまかせる疑似三次元空間を採用した。この設計に基づき試作システムを作成し評価した結果三次元空間としての感覚はある程度得られており、表示速度も操作に追従できる程度のものになった。

Design of multi-user sharing virtual space in the Java-compliant cellular phone

Takeshi Hosokawa[†] Fumiaki Sato[‡] Tadanori Mizuno[‡]

[†]Graduate School of Information, Shizuoka University

[‡]Faculty of Information, Shizuoka University

This paper describes a design of the virtual space shared between a lot of people which offer the communication using avatar in the Java-compliant cellular phone. In the Java-compliant cellular phone, there are restriction of use of only HTTP communication and communication of only the downloading agency server of application and restriction of a memory or calculation resources. In order to solve this problem, we divide virtual space into a background and a avatar, and adopt the pseudo-3-dimensional space which leaves the rendering of a background to a server. As a result of creating and evaluating a trial system based on this design, the feeling as 3-dimensional space is obtained to some extent, and became the thing of the grade to which display speed can also follow in footsteps of operation.

1 はじめに

現在、携帯電話はいつでもどこでも連絡が気軽にとれるという便利さから普及している。人々は携帯電話により場所の制約から解放されたが、音声通話やメールでは基本的に1対1のコミュニケーションに限られている。このため、多人数によるコミュニケーションや、さらにはコミュニティを形成することは、従来の携帯電話では不可能であった。しかし現在、携帯電話によるコミュニケーションは急速に普及しており、今後携帯電話同士によるコミュニティ形成の要求が高まることは十分に考えられる。

特に最近では Java 対応の携帯電話が登場し、プログラムが携帯電話で動くことで話題を集めている。これを受けて、携帯電話上で様々なアプリケーションの開発が活発に行われているが、ここで我々は Java 対応携帯電話をモバイルウェアのプラットフォームとして最適であると考えた。それは、これまでモバイルウェア研究のプラットフォームといえばノートパソコン・PDA などであったが、これらに比べて携帯電話は普及率や常時携帯性といった面で優れているためである。そこで本研究では、多人数コミュニケーションを支援す

るコミュニティウェアを Java 対応携帯電話へ実装することを提案する。

本稿では Java 対応携帯電話において、コミュニケーション支援を目的とした多人数参加型の共有仮想空間を提案し、その設計について述べる。Java 対応携帯電話の仕様である様々な制限と携帯電話自体の限られた計算資源の中で、いかにして共有仮想空間を実現するかといった具体的な提案と実装について述べる。

以下本稿は、2章で関連研究、3章で提案システム概要、4章で各コンポーネントの実装、5章で評価と今後の課題、6章でまとめについて述べる。

2 関連研究

これまでにさまざまな方法でコミュニケーション支援を行う研究が行われているが、多人数参加型のコミュニケーションには仮想空間という概念が主に利用される。仮想空間における情報のやりとりや共有はユーザに場の雰囲気をもたらすことができ、このため実世界のコミュニティと同様のコミュニケーション支援効果が期待できる。

2.1 共有仮想空間

モバイル向けのコミュニケーション支援として、移動体であることを活かした位置情報を利用する方法が

[†] Takeshi Hosokawa (cs6080@cs.inf.shizuoka.ac.jp)

[‡] Fumiaki Sato (sato@cs.inf.shizuoka.ac.jp)

[‡] Tadanori Mizuno (mizuno@cs.inf.shizuoka.ac.jp)

ある。これは実世界の位置情報を元に仮想世界の位置情報に投影することで仮想世界をわかりやすい形で実現できる利点がある。研究例として、[1]ではGPSを用いた周囲の人との位置依存の出会い支援、位置依存の同期型コミュニケーション支援を提供しており、実世界と仮想世界とを関連付けることで、仮想世界上でウェアネスをうまく表現している。ただ、これら位置情報依存の仮想空間の構築は、GPS利用のため屋外に限られることや、別の機器を携帯するというユーザへの負担を考えると、屋内でも利用することの多い携帯電話には不向きであると考えたため、提案するシステムでは位置情報を利用しない。

この手法の他に、3D仮想空間を用いることでコミュニケーション支援を行うものが多く研究されている。[2]のLadakhは既存のWWWサーバ・クライアントシステムを利用し、さらにサーバをマルチキャスト通信網で接続することにより多人数が同時に仮想空間に参加可能にしている。今回本システムのプラットフォームとしてNTT DoCoMoのiアプリを利用するが、この制限としてHTTP通信のみのサポートといったものがあるため、Ladakhと同様にWWWサーバ・クライアントシステムを採用する必要がある。

また3Dマルチユーザシステムにおいて会話を発展させるためのエージェントの研究として、[4]と[3][6]が挙げられる。まず[4]では、参加者同士の会話が中断して沈黙してしまった場合に、仮想空間エージェントがそれまでの会話をモニタリングしておき新たな話題を提供するといったものである。一方[3][6]のPAW²はユーザと一緒に行動する犬型のパーソナルエージェントを用いたものである。このパーソナルエージェントとは仮想世界と実世界を結ぶリンクとして機能するもので、ユーザの仮想世界への再訪を促すものである。しかしパーソナルエージェントの行動パターンには限界があるため、やはり時間の経過によりユーザが飽きてしまう可能性がある。

以上の3D仮想空間を用いたシステムはVRMLで記述されたものであるが、[5]ではビデオ画像を用いて擬似3次元空間システムを提案している。これは実世界をリアルタイムに仮想世界に反映することで、実世界と仮想世界とのインタラクションを実現できる利点があるが、実写を用いるためにカメラの設置や音声認識のための装置の設置が必要となってしまう、広域的なシステムの構築には不向きである。

2.2 インスタントメッセージサービス

また、共有仮想空間とはまったく違う方法で多人数でコミュニケーションを行うものとしてインスタントメッセージサービスがある。

パソコンで動作するコミュニケーションツールは数多くあるが、中でも有名なものにICQがある。このツールは相手ユーザのオンライン・オフラインを常に監視することができる。またメッセージを瞬時に相手

に届けることができ、相手のメッセージが届くと即座に音で知らせてくれるため、利用者がお互いにパソコンに常に向かっている時にはとても便利なツールである。このようなインスタントメッセージサービスを携帯電話で実現させた研究もあるが、携帯電話ではパソコンのように常にこのサービスを受けられるわけではなく、断続的なものになってしまうため、リアルタイムな応答が得られない可能性が高いため、提案するシステムではこのようなインスタントメッセージサービスではなく共有仮想空間によるコミュニケーション支援の方式をとる。

3 提案システム概要

提案システムはJava対応携帯電話において多人数参加可能な共有仮想空間を実現する。携帯電話で3D仮想空間を共有するシステムについての研究はまだない。それは、携帯電話という制限の厳しいプラットフォームのためであり、その問題を本研究では擬似的な3次元空間を用いることでシステムを実現する。ユーザは自分の分身であるアバタを操作することによって仮想空間を移動し、コミュニケーション手段としては文字によるチャットを利用する。また本システムが動作する携帯電話で、本稿執筆時に発売されている機種の平均的な仕様を以下の表に示す。

表 1: 携帯電話の仕様

解像度	120 × 120 ドット
アプリケーションの容量	10K バイト
スクラッチパッドの容量	10K バイト
一度に通信できる容量	10K バイト
通信速度	9600bps
セキュリティ	Java アプリケーションは携帯電話のメモリ領域にアクセスできない
サーバアクセス	Java アプリケーションはダウンロードされたサーバとのみ通信できる

表1のスクラッチパッドは、各Javaアプリケーションが利用する専用の記憶領域である。

3.1 システム構成

提案システムは携帯電話である複数のクライアントと、空間情報やチャット情報を管理・仲介する1つのサーバで構成される(図1)。サーバが1つしか存在せず、サーバを仲介する必要があるのは、iアプリの仕

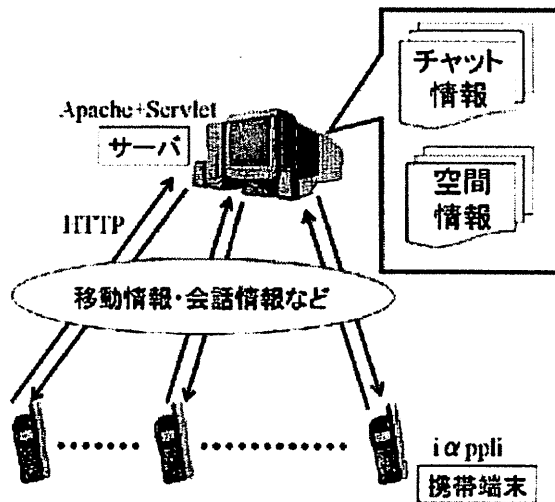


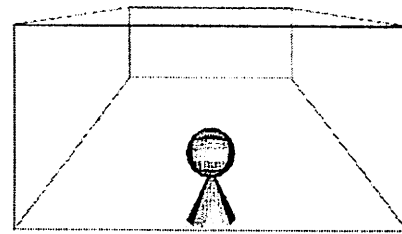
図 1: システム構成図

様のためダウンロード元サーバとのみ通信が許可されることと、クライアントとサーバの通信は HTTP 通信に制限されるためである。HTTP 通信はステートレスであり、連続的な双方向接続を行うことができない。そのため、各クライアントは定期的にこのサーバと通信する必要がある。サーバでは Web サーバプログラムとして Apache、サーブレットエンジンとして Tomcat を利用する。仮想空間の各ユーザごとの情報はサーバが管理する。

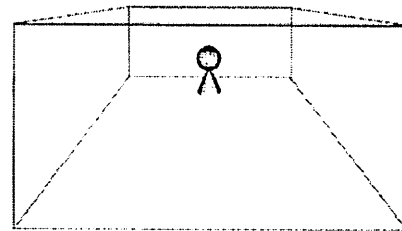
3.2 疑似三次元空間

通常の三次元空間の表示方法として、ユーザ自身のアバタの視点または自分のアバタの後ろの視点から、移動にあわせてリアルタイムにレンダリングを行う方法がしばしば用いられる。これに対し提案する疑似三次元空間は、ある固定した視点を設定し、その視界内でユーザはアバタを動かす(図 2)。そして移動量が閾値を越えたとき、視点が移動する方法である。この手法はレンダリングコストが削減できる利点があるが、その一方で三次元空間の表現力は減少してしまう。例えば背景が固定となるため、背景のオブジェクトとアバタとの重なりが表現できないといったことがある。しかし、現在の携帯電話の通信速度や処理能力を考えた場合、前者の表現法は現実的ではないため、後者の疑似三次元空間を採用する。

この疑似三次元空間の実現方法を次に示す。まず、サーバでは仮想空間のデータとして VRML ファイルを保持し、この空間の水平面での位置と方向をクライアントで管理する。水平面における格子点を設定し、各格子点での 4 方向をみたときの 3D 空間のレンダリング画像をサーバで生成する。その画像をクライアントで背景画像として用い、視点はその格子の線上のみを移動可能とする。アバタが移動して閾値を越えたとき



(a) 移動前



(b) 移動後

図 2: 疑似三次元空間におけるアバタ移動

に、背景画像をその格子点かつその方向の画像に更新する。各格子点において方向は自由に変えることで、その方向に対する背景画像をみることができる。各格子点における空間を以下シーンと呼ぶ。

3.3 クライアントの設計

クライアントの内部モジュール構成を図 3 に示す。ユーザからの操作によりアバタの移動やチャットの入力を行う。この移動情報とチャット情報を定期的にサーバへ HTTP で送信する。また、その応答としてサーバより他のユーザのアバタ移動情報とチャット情報、そして参加・脱退情報があれば受け取る。この情報を元に参加者管理モジュールがアバタの移動やアバタオブジェクトの生成などを行う。

仮想空間はシーンで分けられることを 3.2 節で説明したが、仮想空間の共有はシーンごとに行い、同一シーン内のクライアント同士のみと通信する方式をとり、システム全体のスケーラビリティを向上させる。表示領域の都合上、アバタ同士の衝突判定は行わない。なおアバタの画像を奥行きを表現するためにサイズを変化させる予定であったが、現時点における i アプリの容量制限では実装は難しいと判断し、今回はこの実装を見送ることにした。

クライアントで必要な画像はサーバより適宜ダウンロードするが、アバタ画像は利用する頻度が高いため、スクラッチパッドへ保存しておく。これにより、同じアバタ画像のユーザが参加した場合には新たにダウンロードする必要がなくなる。画像管理モジュールはス

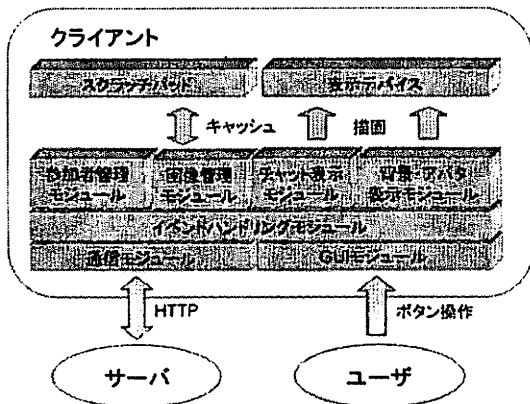


図 3: クライアントの内部モジュール構成

クラッチパッドにどの画像を保持しているかを管理し、ダウンロードする必要があるかどうかを判断する。

クライアントがサーバに送るアバタの移動データとしては、どこの座標まで動いたかという絶対値と、その座標までどのように動いたかという相対値の両方を利用する。それは、相対値の方がデータ量が少ないためと、途中参加者のためには絶対値がないと参加直後の他のアバタの座標を特定できないためである。つまり、定期的な通信では相対値による移動データを利用し、ユーザの参加脱退などの際には絶対値を利用する。また、前述した通り HTTP 通信によりこの動作データは定期的にサーバへ送らなければならないが、今回の実装では 5 秒ごとに一回送ることにした。すなわちクライアント側でアバタを動かした動作データはこの間バッファリングされまとめて送ることになる。

チャットの表示に関しては、仮想空間とアバタの表示によってほぼ画面の全てを使ってしまうため、チャット表示領域として別途スペースを用意することは難しい。このため画面の切り替えもしくはオーバーレイ表示が考えられるが、今回はアバターの移動をしながらチャットを円滑に行うことが可能なオーバーレイ表示を用いる(図 4)。なお、チャットを行うメンバーについても、同一シーン内のユーザ同士で行うものとする。これにより、シーンごとに仲間を集め話題によってシーンを選ぶことが可能である。

3.4 サーバの設計

サーバの内部モジュール構成を図 5 に示す。クライアントの設計でも述べたとおり、シーンごとの仮想空間の共有を行うため、まずクライアントから HTTP で受信したデータはシーン管理モジュールに渡される。ここで、同一シーン内のユーザ同士のアバタの位置と動きデータは参加者管理モジュールが管理し、チャットのやり取りに関してはチャット管理モジュールが管理する。また、シーンの変更が生じた場合は、シーン管理モジュールから背景画像生成モジュールへ要求を出

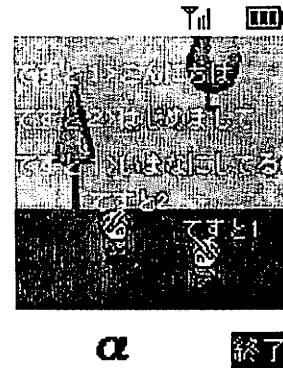


図 4: オーバレイ表示によるチャットの例

し、クライアントへ新たなシーンの背景画像を HTTP の応答として返す。

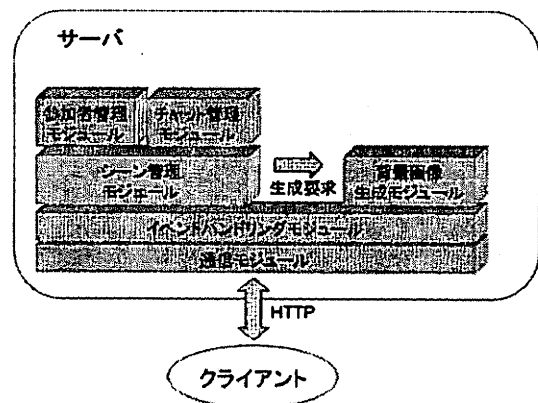


図 5: サーバの内部モジュール構成

サーバではクライアントの HTTP による要求をすべてサーバレットで処理する。すなわち、ログイン処理などはもちろん画像データなどのダウンロードに関しても、まずサーバレットと通信した後でファイルヘリダイレクトする。これによりサーバ側でクライアントの画像データの要求と同時に、その画像のファイル名からクライアントの仮想空間内におけるシーンの特定を行い、通信コストを抑えている。

サーバレットはクライアントごとの情報を管理する簡単なデータベースを内部に持ち、クライアントのログインによりデータベースへの追加、ログアウトによるデータベースからの削除が行われる。そのためサーバ側でクライアントの情報をファイルに書き出すようなことはなく、すべてメモリ上で処理を行う。すなわち、クライアントのログインの際にサーバはデータベースのキーを生成し、応答としてキーである ID を返す。今回の実装では 8 個のランダムな数字の羅列をキーとした。

また、クライアントへ渡す仮想空間の背景画像については、将来的に VRML によって記述された仮想空

間をサーバ側のレンダリングにより動的に生成する予定だが、今回の実装ではある地点ごとの視点からレンダリングしておいた画像をあらかじめ用意し、サーバに保存しておくことにした。

4 各コンポーネントの実装

この章ではクライアントの i アプリの実装、サーバのサーブレットの実装について述べる。まずサーブレットで管理するクライアント情報のデータベースについて述べる。

4.1 クライアント情報データベース

クライアントは最初にサーバへログインすることで、サーバから ID を受け取る。サーバはこの ID をキーとして、ハッシュテーブルを利用したデータベースに登録する。このキーに対するデータとしては、ユーザの名前・アバタの種類・アバタ位置の座標など複数に渡るため、String クラスの配列を用いる。格納データの一つに、このキーのユーザ以外のアバタの動きデータを時系列で格納するキュー（以下動作キューと呼ぶ）を用意している。動作キューには例えば、次の図 6 に示す形でデータを格納する。



図 6: 動作キュー

このキューを String クラスで実装し、スペースで区切られたデータの羅列とする。時系列ごとに ID とその ID の動作の対を、この動作キューである String クラスへ各 ID のユーザからの配信が発生する度に追加していく。図 6 の例では、ID1 のユーザのあとに ID3 のユーザ、さらにそのあとに ID2 のユーザのデータが追加されていく様子を示している。動作キューのキーである ID のクライアントが、定期的に HTTP 通信する際に応答としてこの動作キューを受け取る。その後、動作キューをクリアし、再び他のユーザのデータを追加していくというメカニズムである。この方式によりサーバ側はクライアントへの配信データを管理することが容易になる。動作キュー以外のデータベースの要素の一覧を次の表 2 に示す。

4.2 参加脱退処理

サーバへのログインは、クライアントが名前とアバタの種類を決めることで行う。この情報をサーバが受信すると、サーバ側ではクライアントのデータベースへの登録処理が実行される。この際にデータベースのキーとなる ID が生成され、応答としてクライアントが ID を受け取る。

次に今回の実装ではログイン直後は仮想空間の特定のシーンから始まるが、この同一シーン内の他のユー

表 2: データベースの要素一覧

要素 1	ユーザの名前
要素 2	アバタの種類
要素 3	アバタの x 座標
要素 4	アバタの y 座標
要素 5	動作キュー
要素 6	ユーザの視点の向き
要素 7	ユーザの視点の x 座標
要素 8	ユーザの視点の y 座標

ザへログイン通知が行われる。すなわちシーンへの参加処理を行うわけだが、これには前述した動作キューを利用する。動作キューには ID とその ID の動作が対で追加されると 4.1 節で述べたが、この動作の部分にログインを示す文字列と、初期のアバタの座標を代わりに格納することで実現する。またシーンからの脱退処理に関しても同様の方法で行うことができ、動作キューへログアウトを示す文字列を格納する。

このログイン通知を受けた各クライアントは、通知元のクライアントの動作キューへ応答としてアバタ情報を追加する。その動作キューを得ることで仮想空間への参加処理が完了する。

また、クライアントでシーン切り替えが発生した場合には、元のシーンからの脱退、次のシーンへの参加ということになる。そのため元のシーン内の各ユーザにはログアウトの際に利用した文字列を動作キューへ加える。同様に、次のシーン内の各ユーザにはログインの際に利用した文字列、シーンへ参加するときのアバタの座標を動作キューへ加える処理が必要である。

4.3 チャットメッセージ

チャットは同一シーン内のユーザ同士で行う。メッセージの受け渡しは基本的には、同一シーンのメンバーへのログイン・ログアウト通知やシーンの参加・脱退通知の手法と同様である。つまりクライアントでメッセージを作成し、送信ボタンを押すことで、サーバへ定期チェックの際に送信するアバタの動作データに、メッセージを加えてサーバへ渡す。そして、サーバではこの ID のユーザが発言した旨を通知する文字列を同一シーン内に存在するクライアントの動作キューへ追加する。このようにすることで、クライアントでは受け取った動作キューの解析時に発言通知があると、うけとったメッセージ即座に表示し、ユーザがアバタ操作中であってもメッセージを見逃すことなくチャットすることを可能としている。

4.4 動作キューの解析

3.3節で述べたように、クライアントは5秒に一度アバタの動作をまとめてサーバへ送信する。この際に送信されるデータはアバタの相対値の羅列である。サーバでは同一シーン内のユーザをデータベースから検索し、各該当ユーザの動作キューへ動作データを加えるが、この相対値を元にシーン内の絶対座標を計算する。相対値と絶対値を併用することで、通信のコストの削減と不定期なユーザの参加を可能としている。さらに相対値は次に示すような動作キューの解析に利用している。

クライアントに渡される動作キューのIDの対となる部分は、アバタの動作だけではなく参加脱退処理やチャット情報の配信にも利用すると述べたが、これは動作データを相対値の羅列としていることで可能となっている。すなわち相対値としてアバタの動作は上下左右の4方向であるから、それを表す以外の値を動作キューに格納することで、容易に動作情報かチャット情報かが判別できる。

5 評価と今後の課題

今回提案したシステムをiアプリにより実装し、実機における動作を確認した。その結果、初回の画像のダウンロードにはやはり時間がかかってしまうが、2度目以降の同じ画像の表示はスクラッチパッドからの読み込みによりスムーズに行えた。他のユーザの参加・脱退やアバタの動きの反映については、タイムラグが多少生じるものの仮想空間を矛盾なく表示し、仮想空間の共有が確認できた。

評価方法として実際にシステムを利用してもらうことで、疑似三次元空間が効果的に仮想空間を表現しているかということや、通常の三次元空間に比べて利用者にとってどのような違いがでるかなどをアンケート調査により行うことを考えている。

今後の課題として、ただ共有仮想空間という場を提供しただけでは、ユーザは時間がたつにつれ飽きてしまいコミュニティの発展は見込めないという問題がある。最低限ユーザをとどめるためには、コミュニティに文脈を与えることや、メンバーのプロフィールを知ること、そしてメンバー同士の信頼を築くことが必要で、これらを達成するためには、メンバー・データベースの構築と管理が欠かせない[7]、すなわちシステムからログアウトすると、データベースからユーザを削除してしまうのではなく、データベースサーバを別途用意しそこへプロフィール、友達リストなどを作成できるようにする必要がある。

今後システムへ追加する予定の機能として、今回クライアント側における仮想空間の背景画像はあらかじめレンダリング済みのものを利用したが、それをサーバでVRMLから動的に生成をする機能がある。そしてアバタの画像のサイズ変更による奥行きを表現機能

は、疑似三次元空間を表現する効果的な機能となるため、追加する必要がある。

また、現在のシステム構成では1つのサーバに負荷が集中し、ユーザ数の限界が予想される。このため、この限界を調べる必要があるとあり、RMIによってサーバの負荷分散を行う改良を考えなければならない。

今回はiアプリによる実装を行ったが、今後J PhoneのJava対応携帯電話や、ez plusといった携帯電話で動作するJavaが登場予定であるため、その他のプラットフォームへの移植が考えられる。

6 まとめ

本稿ではJava対応携帯電話における多人数参加型の共有仮想空間の設計と実装について述べた。未だ発展途上であるが、限られた計算資源や制限されたプロトコルの中でも、実際に共有仮想空間システムを実装することで、Java対応携帯電話上のコミュニティウェアの可能性を示した。そして疑似三次元空間を採用することで、これまで実現されなかった携帯電話上における三次元空間の共有システムを実現した。実際に実装してみることで、低速な回線・少量のリソースという制限の中で、共有仮想空間が実現可能であることがわかった。そして、今回実装したシステムに必要な改良や、研究すべき点を示し、また今後のシステム開発の方向性を示した。

参考文献

- [1] 太田 賢, 町田基宏, 大辻清太, 杉村利明: 実世界コミュニティプラットフォームの提案, DICOMO2000, pp.601-606
- [2] 中村暢達, 里田浩三, 平池龍一, 根本啓次: インターネット対応 3D マルチユーザシステム Ladakh, 信学論 (D-II), Vol.J82-D-II, No.5, pp.982-991, 1998
- [3] 松田晃一: パーソナルエージェント指向仮想社会PAW(第2版)の設計と構築, DICOMO'99, pp.637-642
- [4] 中澤 諭, 中西 英之, 石田 亨: 会話を発展させる仮想空間エージェント, DICOMO2000, pp.19-24
- [5] 小川剛史, 中野昭宏, 塚本昌彦, 西尾章治郎: ビデオ画像を用いた疑似3次元空間に基づくコミュニケーションシステム, DICOMO2000, pp.691-696
- [6] 松田晃一, 谷島 直: パーソナルエージェント指向仮想社会PAW'2におけるユーザアクティビティの分析と考察, DICOMO2000, pp.775-780
- [7] Amy Jo Kim: ネットコミュニティ戦略, 翔泳社