

GPS 携帯電話を用いたマルチベンダ対応チャットとスケジューラの実装

Chat and Scheduler for Multi-vender using Mobile Phone with GPS

石山 慎^{*1} 佐々木 準人^{*1} 渡辺 雅人^{*1} 新美 礼彦^{*1} 高木 剛^{*1}
 Makoto Ishiyama Hayato Sasaki Masato Watanabe Ayahiko Niimi Tsuyoshi Takagi

小西 修^{*1} 宮本 衛市^{*1} 高橋 修^{*1} 携帯電話アプリケーション作成プロジェクト^{*1}
 Konishi Osamu Ei-ichi Miyamoto Takahashi Osamu The making Project of mobile Application

^{*1} 公立はこだて未来大学 システム情報科学部
 School of Systems Information Science, Future University-Hakodate

Recently, cellular phone has a lot of convenient functions and provides various performances and high quality services. However, the functions and interface of cellular phones depend on its vendors, nevertheless the same carrier. For example, Mail, Web Browser, Transmission Rate, etc. And also, there are few common application which is available for plural carriers. Therefore, we propose the common application for au and DoCoMo. Our project proposes chat and scheduler for multi-vender using mobile phone with GPS. And we implemented and discussed the application.

1. はじめに

近年、携帯電話は音声通話の他に電子メール、インターネットへのアクセス、カメラ機能、アプリケーション機能、GPS機能、そして最近では、非接触 IC カード技術を搭載するなど、単なる音声通話のためのツールから、生活するためには欠かす事のできない重要なツールへと変貌を遂げている。

このような進歩を遂げる一方、携帯電話に搭載されている機能は携帯メーカーによって異なっており、また機種によっても異なっている。このような機能の差異によって、ユーザが機能ごとに操作を覚えなくてはならないなどのユーザビリティに反する問題点が生じている。また、グループワークを行う場合、異なる機種間で動作する同一のアプリケーションが希少なため、グループメンバー全員と円滑なコミュニケーションを行いにくい。これらの問題点を解決するため、本稿では機種に依存せず、キャリアを越えて動作するグループワーク向けのアプリケーションを提案するとともに、その実装評価結果を報告する。

2. 提案するアプリケーション

本プロジェクトでは、既存のサービスおよびアプリケーションの問題点・改良点などを踏まえ、マルチベンダ対応の二つのアプリケーション「PersoMa」と「真ノ助」を開発した。

2.1 基本的な考え方

今回提案する二つのアプリケーションは、グループワークやグループコミュニケーションにとって必要な、マルチベンダ対応アプリケーションである。グループコミュニケーションに対しては、これまで携帯電話でコミュニケーションを行う上で注目されていなかった位置情報に着目し、位置情報を共有することで相手の状態を把握できる地図機能とチャットを備えたアプリケーション「PersoMa」を開発した。グループワークに対しては、グループワークを行う上でスケジュールの共有が必須であるため、グループメンバーの所有しているキャリアに関わらず、スケジュールの共有が可能なアプリケーション「真ノ助」を開発した。「PersoMa」は互いの位置情報を共有し、それに基づいたチャット機能を備えた

コミュニケーションツールであり、一方「真ノ助」はグループとグループメンバーのスケジュールをグループ内で共有する共有スケジューラである。

2.2 PersoMa

本アプリケーションは、近年携帯電話に搭載されつつあるGPS機能に着目して、モバイル端末である携帯電話を用いて個人の場所を取得し、グループで地図を見ながらチャットを行い、個人情報を共有しあうことができる新しいコミュニケーションツールである。

PersoMaで提案する主な機能は以下の通りである。

自分を中心とした周辺情報の通知

自分の位置を地図の中心とし、自分の周辺に他のメンバーがいるか、観光地があるかなどの周辺情報を通知する。

他者を中心とした周辺情報の通知

他のメンバーを地図の中心とし、他のメンバーがどの位置に居るか、他のメンバーの周辺に誰がいるか、周辺に何があるかなどの周辺情報を通知する。

チャットの送受信

周辺情報に基づき、他のメンバーとのチャットでメッセージを送受信する。1対1だけではなく、グループ間でのチャットもサポートする。

設定変更

地図の表示や、更新間隔、表示するメンバーの変更などを行う。

2.3 真ノ助

本アプリケーションは、殆どの携帯電話に搭載されているスケジューラ機能を基本とし、異なるキャリア間でもスケジュールを共有でき、また同一の操作で使用できる携帯オンラインスケジューラである。

真ノ助で提案する主な機能は以下の通りである。

複数人でのスケジュール共有

連絡先: 新美 礼彦, 公立はこだて未来大学, niimi@fun.ac.jp

スケジュールをグループ間で共有するために、個人のスケジュールはもちろん、グループのスケジュールを管理し、携帯電話上に表示する。

グループ管理

スケジュールを共有しあうグループの新規登録、作成を行う。また、グループへのメンバの参加、脱退、検索などを行う。

3. 実装方式

今回提案するアプリケーションの実装に際し、生じた主な問題点とその解決策を以下に述べる。

3.1 データの管理方式

異なるキャリア間でのデータ共有を前提とした、マルチベンダ対応アプリケーションを実現するために、アプリケーションに必要なデータをクライアントの機種やキャリアに依存しないサーバに蓄え、必要時にクライアントとサーバの間でデータの送受信を行わせるデータの管理方式を採用した。通信されるデータには、ユーザの個人情報、位置情報、スケジュール情報、そして、グループのスケジュール情報、グループ管理情報、地図の画像や地点情報、チャット情報などがある。クライアントがサーバに置かれているエージェントにアクセスし、エージェントから SQL 文をデータベースに発行して、アプリケーションに必要なデータを取得し、クライアントに送受信を行っている。au と DoCoMo では、モジュールごとの通信方式が異なるため、エージェントはキャリアごとに実装を行っている。クライアント、エージェントおよびデータベースの関係を図1に示す。

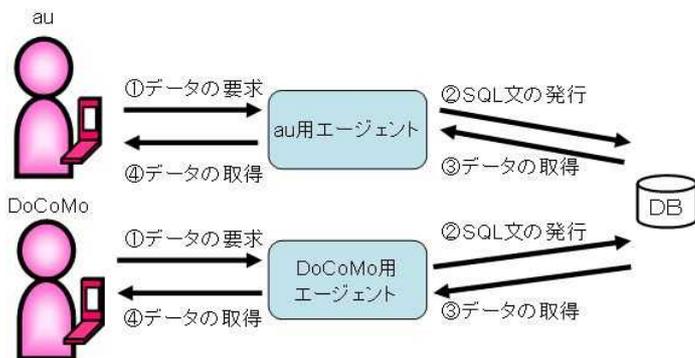


図1. データ管理方式

サーバに置かれているデータは、全て MySQL によって構築されたリレーショナルデータベースで管理されている。PersoMa が構築するデータベースは project11、真ノ助が構築するデータベースは prsdb および grsdb である。各種テーブルはモジュールごとに分けて用意し、モジュールごとに SQL 文を発行し、データの通信を行う。project11 では、PersoMa で必要とされるチャットや地図と位置情報、そしてグループ管理などに必要なテーブルが機能ごとに用意されている。真ノ助では、prsdb にユーザのスケジュールや個人情報を登録するテーブルが用意されており、grsdb にグループのスケジュールやグループ管理情報などを登録するテーブルが用意されている。

このようなデータの管理方式に基づき、キャリア間で異なる通信方式や内部処理の差異をシステムが隠蔽することにより、ユーザビリティを高め、アプリケーション起動時にサーバとの通信を行うことで機種やキャリア間に依存することなくデータを共有することを可能にした。

また、この方式により、グループメンバなどの登録情報などを全てサーバに置くために、携帯電話の保存領域を使用する必要がなく、したがって携帯電話の保存領域の限界に左右されることが無くなった。

3.2 GPS 座標の取得方法

GPS から位置座標を取得しようとする場合、au の端末を用いれば携帯端末そのものから位置座標を獲得できるが、DoCoMo の端末の場合、プライバシーの観点から NTT DoCoMo 公式のコンテンツプロバイダ以外には GPS の仕様を公開していないため、端末上では GPS 情報を扱うことが出来ない。そのため、DoCoMo の端末では図2に示すように、携帯端末からアプリケーションサーバにアクセスし、アプリケーションサーバを介して DLP (DoCoMo Location Platform) サーバにアクセスし、GPS 情報を取得する。

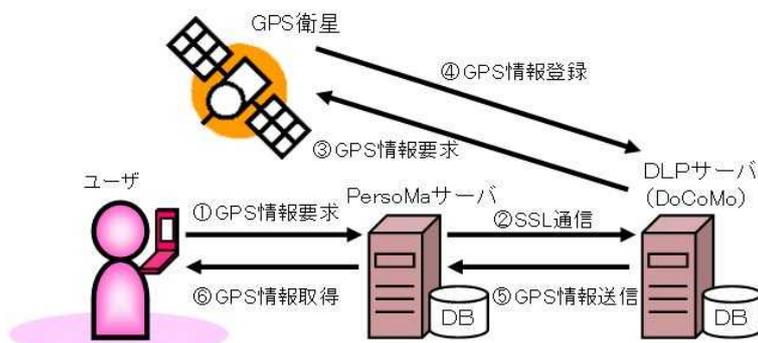


図2. DoCoMo 端末での GPS 情報取得

3.3 地図座標の特定方式

GPS を用いて位置情報を取得した後の、取得した位置情報から地図上の座標の割り出しは、携帯電話の処理能力や記憶領域を勘案し、またサーバの有する処理能力の高さから、携帯電話上ではなくサーバ上で処理を行っている。この際、au 端末では直接緯度経度情報を受け取っているが、DoCoMo 端末では緯度経度情報を Degree 度単位で受け取っている。

地図の X, Y 座標を割り出す際のサーバ処理の流れを図3に示す。ここで、図3における地図の始点とは、用いる地図の左下端の点であり、終点とは用いる地図の右上端の点である。この時の終点(a,b)は実数であり、求める点(x,y)は変数である。

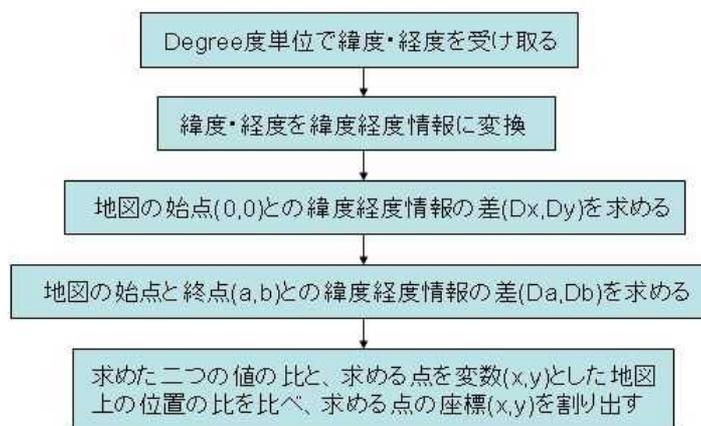


図3. 地図座標取得処理のフロー

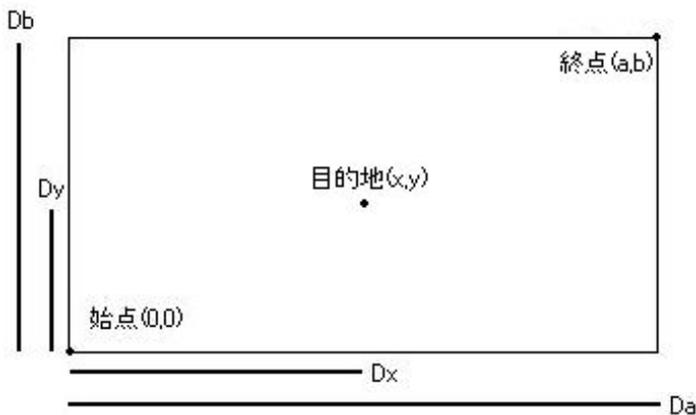


図4. 地図上の位置関係例

上記の図4を用いて位置情報処理の例を示す。この時、 (x,y) , (a,b) の単位はピクセル、 (D_x, D_y) , (D_a, D_b) の単位は緯度経度情報である。まず、始点から目的地までの緯度経度情報の差を (D_x, D_y) 、始点から終点の緯度経度情報の差を (D_a, D_b) とする。始点・目的地・終点の座標は図4に示す通りであり、座標はピクセルで表されている。これより緯度経度情報の差と座標との関係を表すと、以下の式(1)、式(2)のようになる。

$$D_x : D_a = x : a \dots (1)$$

$$D_y : D_b = y : b \dots (2)$$

これらの式より、目的地の地図上のピクセルでの座標である x, y の値を求めると、式(3)、(4)のようになる。

$$x = aD_x / D_a \dots (3)$$

$$y = bD_y / D_b \dots (4)$$

上式から、指定された地点の地図上のピクセルでの座標を得ることができる。

3.4 地図画像の取り扱い

3.3の方法を用いて、サーバはGPS情報から取得した位置情報を地図画像における座標に変換した後、その座標を中心として地図画像から携帯電話の表示領域の大きさで画像を切り取った。切り取った画像をクライアントである携帯端末に送信する方式はDoCoMo 端末と au 端末で異なる。DoCoMo 端末では、切り取った JPEG 画像をそのままクライアントに送信すればよいが、au 端末ではこれができないため、サーバ内のエージェントで画像データをBASE64にエンコードを行い、データベースにlongblobとして登録した後にクライアントである au 端末に送信した。

クライアントに送信する際は、クライアントが受信できるデータサイズに制限があるために、エンコードされたデータをいくつかに分け、通信を繰り返し行うことでデータの受け渡しを実現した。全てのデータを受信し終えた後にクライアントがデコードを行い、クライアントで取り扱うことのできる画像のデータ形式に変換した上で画像表示を行った。

地図の拡大縮小機能や、地図移動機能などのユーザからの要求に対しては、その要求事項をサーバに送ってサーバに画像処理を依頼し、サーバより目的の画像を取得している。

3.5 スケジュールの共有方式

従来のスケジュールリングアプリケーションの場合、ユーザのスケジュール情報は端末のみに記録され、複数の人に関わるようなスケジュールでも個人のスケジュールとしてしか扱うことができなかった。そこで真ノ介ではサーバにスケジュールをアップロードし、閲覧権限をもつグループ参加者間でスケジュールを共有するという方法で解決している。スケジュールのアップロードはグループに参加している人全員が行うことができ、新規にスケジュールを作成、またはスケジュールを修正した場合にグループ参加者に通知されるようになっている。真ノ助の通信は以下の図5の通りである。

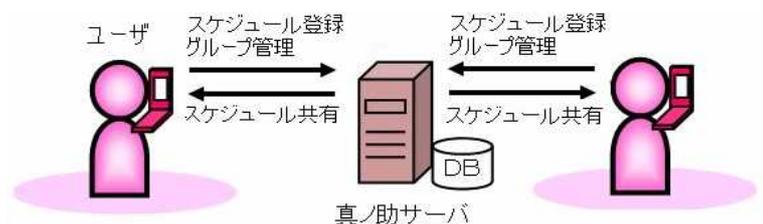


図5. スケジュールの共有方式

4. 実装と評価

アプリケーションの実装は、開発環境の導入から始まり、エミュレータ上で動作するアプリケーションの開発、エミュレータ上で開発したアプリケーションを携帯電話上に移植という手順で行った。

4.1 実装

アプリケーションの実装を行う際、まず必要なのは携帯電話アプリケーション向けの開発環境の整備である。使用した開発環境は DoCoMo 側ではプログラム環境として Eclipse3.0 + Doja4.0、テスト用端末として FOMA P700i・movaf505iGPS、au 側ではプログラム環境として BREW SDK V2.1.1 Ja、テスト用端末として A5502K・A5506T をそれぞれ用いた。また、サーバ側の環境としては Red Hat Linux 9.0 を OS として用い、その上で Apache 2.0.55・PHP 5.0.5・MySQL 4.0.15・OpenSSL 0.9.8a といったソフトウェアを動作させる環境を整えた。

実際にアプリケーションの実装を行う過程で、発生した問題点と解決策を以下に記す。

(1) 文字コードの違いによる文字化け

サーバと携帯電話では、使用している文字コードが異なるため、どちらか一方のフォーマットにあわせて文字コードを設定すると文字化けが発生し、文字データが正常に登録・表示できなかった。そこで、サーバ上に設置している PHP 全てに、携帯電話から受け取った文字をサーバ上の文字コードに変換する処理と、サーバ上の文字を携帯電話の文字コードに変換する処理の二つの処理を加え、サーバと携帯電話双方向で文字の変換を可能にすることで、問題を解決した。

(2) 処理できる関数や変数の型の制限

一部携帯電話においては、小数点を含む数を扱えない、C言語などのプログラミング言語の一部関数が使用できないなどの理由から、処理できる関数や変数が限られていた。au 端末である BREW では、小数同士を計算する独自の関数が用意されており、それを使用することで計算を行った。また、小数から整数へのキャストを明示的に行ったとしても、コンパイルを行うことができないので、小数を一度文字列に変換し、atoi 関数を使用して再びその文字列を整数に変換を行うという処理を行った。DoCoMo は小数の処理をサーバ上で行い、その結果を整数に変換した値のみを携帯に受け渡す手法を用い、またプログラミング言語上の制約もサーバに処理を委ね、携帯端末側で処理しなくてもすむようにすることで問題を解決した。

4.2 評価

開発に伴うアプリケーションに対する定量的なユーザ評価は行っていないが、携帯電話を主に用いる大学生ユーザや、携帯電話アプリケーションを作成している企業からの試用後の評価を受けた。評価の中には、アプリケーションを共有することで様々な機種から参加できるものになっているのは良い、キャリアの差を埋めるというアプローチは良い、といった肯定的な意見が多々見られた。しかし、中にはパケット量を少なくするアルゴリズムの探求・実装まで考慮すべき、既存のアプリケーションを組み合わせただけにみえる、既に同じようなアプリケーションが存在しているのではないか、という意見もあった。

PersoMa と類似するアプリケーションとしては、Ez ナビウォーク [KDDI 04]、Team Factory [KDDI 05]、イマドコサーチ [NTTDoCoMo 06] が上げられる。これらのアプリケーションと PersoMa を比較したのが表1である。同表の機能、は項2.2であげた PersoMa の機能である。

表1. PersoMa と他アプリケーションの機能比較一覧表

	機能	機能	機能	対応端末
PersoMa				au, DoCoMo
Ezナビウォーク		×	×	au
Team Factory			×	au, DoCoMo
イマドコサーチ	×		×	DoCoMo

表1の通り、既存のアプリケーションでは PersoMa で提案する機能を全て兼ね備えたものは存在しないと言える。さらに、対応端末では Team Factory が PersoMa と同様に au, DoCoMo となっているが、前提として Team Factory では au ユーザが最低でも一人おり、au ユーザの承認を受けなければ DoCoMo ユーザはサービスを利用できない。一方、PersoMa では au, DoCoMo どちらのユーザでも承認を受けることなく利用が可能である。よって、PersoMa は導入を含めてキャリア間を越えて動作するアプリケーションであると言える。

また、真ノ助に関してはほぼ同様のアプリケーションが存在するが、最終的には一つのアプリケーションへの統合を想定しているため、単体としての目新しさよりも操作性と完成度を重要視したアプリケーションとなっている。

完成したアプリケーションを動作させた結果、操作上や表示画面などにおいて au, DoCoMo 間での大きな差異は出ず、ほぼ同一の性能を発揮した。一方、利用環境でのレスポンスには差異が発生した。au 側の環境ではレスポンスがやや遅く、DoCoMo 側の端末のうち、FOMA 端末は問題なく動作し、レス

ポンスも速く問題はないが、FOMA 端末以外の端末ではレスポンスがやや遅く、au 側の端末とほぼ同程度の速度であった。この時、au 側端末の通信速度は 144kbps、DoCoMo 側の通信速度は FOMA 端末では 384kbps、movia 端末では 28.8kbps であった。

一方、取得した GPS 情報を元にした地図上の位置への変換では、GPS との通信が最も良好に行える状態で、縦方向横方向に au 端末では約 3mm ずつ、DoCoMo 端末では約 1cm ずつ画面上で位置のずれが発生した。用いた地図は縮尺 1/25000 であったため、実際の距離での誤差に換算すると、au 端末では $3\text{mm} \times 25000 = 75000\text{mm} = 75\text{m}$ 、DoCoMo 端末では $1\text{cm} \times 25000 = 25000\text{mm} = 250\text{m}$ のずれとなる。また、au 端末と DoCoMo 端末での GPS 誤差は、au 端末では最大で 30m 前後であり、DoCoMo 端末では最大で 50m であった。

5. 終わりに

本プロジェクトでは、サーバを用いて携帯電話の機能格差を無くし、ユーザ側には格差を感じさせることのないアプリケーションを実現した。しかし、サーバを用いると、レスポンスが携帯端末のみでアプリケーションを動作させる場合よりも低下するため、より速いレスポンスを返すための処理の改善などが必要であることを実感した。今後は、パケット量の減少や、ユーザインターフェースの改良・GPS 精度の向上など、個々のアプリケーションとしての機能を洗練し、これら二つのアプリケーションを一つのアプリケーションへ統合することを予定している。

6. 謝辞

本プロジェクトでは、KDDI、NTT DoCoMo、日立公共システムエンジニアリング株式会社、日本情報通信コンサルティング株式会社の四社からの懇切な技術協力を受けたことに対し、深甚なる謝意を表します。

参考文献

- [鈴木 01] 鈴木、土屋 i モード Java プログラミング、株式会社アスキー、2001年。
- [茂木 04] 茂木、関根、近藤、白論地 BREW プログラミング、実践バイブル、株式会社インプレス、2004年。
- [KDDI 04] KDDI(株)、Ez ナビウォーク、
http://www.au.kddi.com/ezweb/service/ez_naviwalk/, 2004年
- [KDDI 05] KDDI(株)、TeamFactory、
http://www.au.kddi.com/ezweb/service/team_factory/, 2005年
- [NTTDoCoMo 06] NTTDoCoMo(株)、イマドコサーチ、
<http://www.nttdocomo.co.jp/service/imadoco/index.html>, 2006年
- [NTTDoCoMo 06] NTTDoCoMo(株)、GPS 機能への対応について、
<http://www.nttdocomo.co.jp/service/imode/make/content/html/outline/gps.html>, 2006年
- [memn0ck 05] memn0ck KDDI V.S. ドコモでナビ対決！, ITmedia +D mobile, <http://plusd.itmedia.co.jp/mobile/articles/0510/18/news004.html>, 2005年