

# GPS 携帯端末を用いた近隣バス停位置と バス停時刻表検索システムの試作

S003036

下田 宏志

計算機科学講座

田中研究室

2004 年 2 月 13 日

# 目次

第 1 章 序論.....	3
第 2 章 システムの概要.....	4
2.1 システム概要.....	4
2.2 システム機能.....	5
2.3 システム環境.....	6
第 3 章 システムを実現する要素技術について .....	8
3.1 DBMS .....	8
3.1.1 DBMS .....	8
3.1.2 利用者定義型（抽象データ型）と空間検索プラグイン.....	9
3.1.3 格納データ .....	10
3.2 Web サーバ .....	10
3.2.1 Web サーバ.....	10
3.2.2 JDBC.....	11
3.3 プログラミング言語 .....	11
3.3.1 埋め込み SQL.....	11
3.3.2 Java サブレット .....	12
3.4 携帯端末.....	12
3.4.1 各キャリア .....	12
3.4.2 携帯端末による Web コンテンツ .....	13
3.5 GPS .....	14
3.5.1 GPS .....	14
3.5.2 GPS の座標系 .....	14
3.5.3 携帯端末の GPS 機能 .....	14

<b>第4章 検索システムの機能構築</b> .....	<b>16</b>
4.1 今までのシステムとの相違点 .....	<b>16</b>
4.2 本システムの機能.....	<b>16</b>
4.2.1 GPS を利用した検索機能.....	16
4.2.2 地図からの検索機能 .....	18
4.3 本システムの仕組みと実装.....	<b>19</b>
4.3.1 基本システム .....	19
4.3.2 GPS による検索システム.....	22
4.3.3 au の GPS 携帯端末による検索システム .....	23
4.3.4 地図の表示システム .....	25
4.3.5 地図画像による検索システム .....	29
4.4 DB 格納データ .....	<b>32</b>
4.4.1 バス停位置テーブル .....	32
4.4.2 バス路線テーブル.....	34
4.4.3 時刻表テーブル.....	35
<b>第5章 検証実験及び考察</b> .....	<b>39</b>
5.1 検証実験.....	<b>39</b>
5.1.1 検証方法 .....	39
5.1.2 実験結果 .....	40
5.2 考察.....	<b>41</b>
<b>第6章 終論</b> .....	<b>42</b>
<b>謝辞</b> .....	<b>43</b>
<b>参考文献・資料・サイトなど</b> .....	<b>44</b>

## 第1章 序論

一般的に公共交通機関を利用する場合、分かりにくい部分がある。その中でもバスを利用する場合は、電車を利用する場合よりも分かりづらい事が多く、特に初めて利用する場合には利用しづらい面がある。初めて来たという場所であれば、現在位置の近くのバス停の場所さえも、どこにあるのか分からない場合も多い。そこで、初めての場所であっても近くのバス停の場所が簡単に分かり、そのバス停の時刻表まで知ることができるシステムがあれば、もっと気軽に安心して利用でき、便利である。

現在では、携帯端末(携帯電話)の普及率が高くなり、ほとんどの人が携帯端末を利用している。また、携帯端末の機能自体も高性能化し、現在使用されているほとんどの携帯端末でインターネットに接続し、様々な情報を得ることが可能である。最近では、GPS 機能を搭載し、現在位置の情報を取得できる携帯端末(以下 GPS 携帯端末)も販売されており、次第に普及しつつある。

また、この検索システムを構築する上で、バス停の情報、時刻表といったデータは松江市のデータだけでもかなり膨大な量であり、人間の手で高速かつ正確な検索、データの更新といったことは実用的ではないと言うよりは、むしろ不可能である。そこで、この問題を解決する方法として DBMS を使用した。この DBMS を使用することで、その時々が必要に応じて、適切なデータを高速かつ確実に利用することができる。また、データの更新の容易さやシステム稼働中のデータの更新が可能である。その上、セキュリティの高さの面でも優れている。そして、この DBMS は http 機能とも接続ができるという面も持つ。従って、Web サーバを構築する上で非常に便利である。

そこで、本研究では簡単に近くのバス停を検索する方法として、GPS 携帯端末を使用して、現在位置の情報を取得し、携帯端末のインターネット接続機能を用いて Web サーバにアクセスすることで、Web サーバで現在位置の近くにあるバス停を DBMS を使用して検索し、周辺の地図とバス停の位置、そのバス停の時刻表を検索し、携帯端末の画面に表示するシステムの試作を行った。

## 第2章 システムの概要

### 2.1 システム概要

GPS 携帯端末を用いて、携帯端末のインターネット接続機能で Web サーバにアクセスし、GPS で位置情報の取得を行うことで得られた位置情報をサーバへ送信することで、バス停およびその時刻表を簡単に検索することが出来るシステムである。このシステムのイメージは以下の図のようなものである。

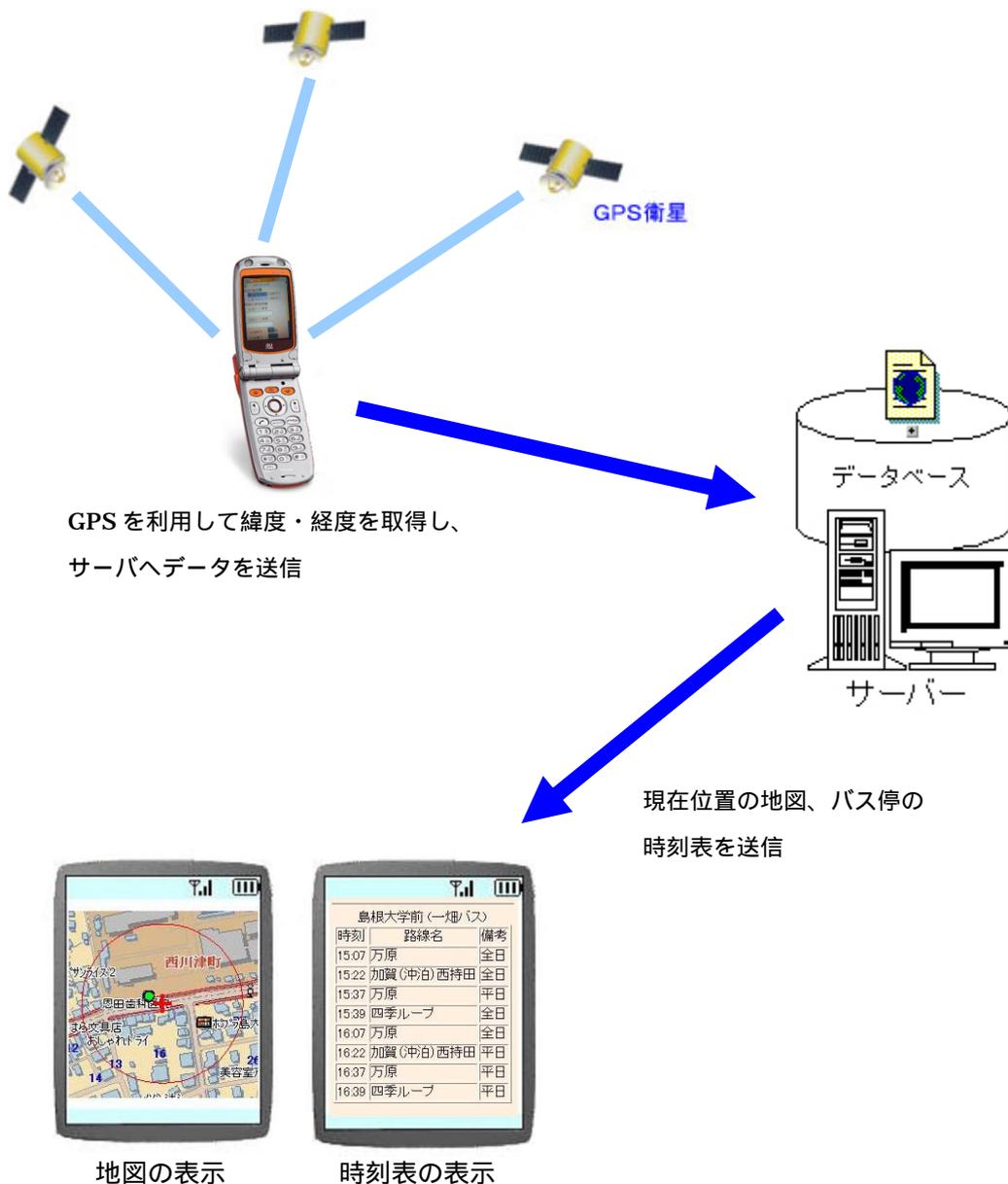


図 1. GPS 携帯端末を用いた検索システム

また、PC からアクセスした場合は、松江市の地図上のバス停をクリックすることで、そのバス停の時刻表を検索することも出来るシステムである。地図から検索するときのイメージは以下のようなものである。

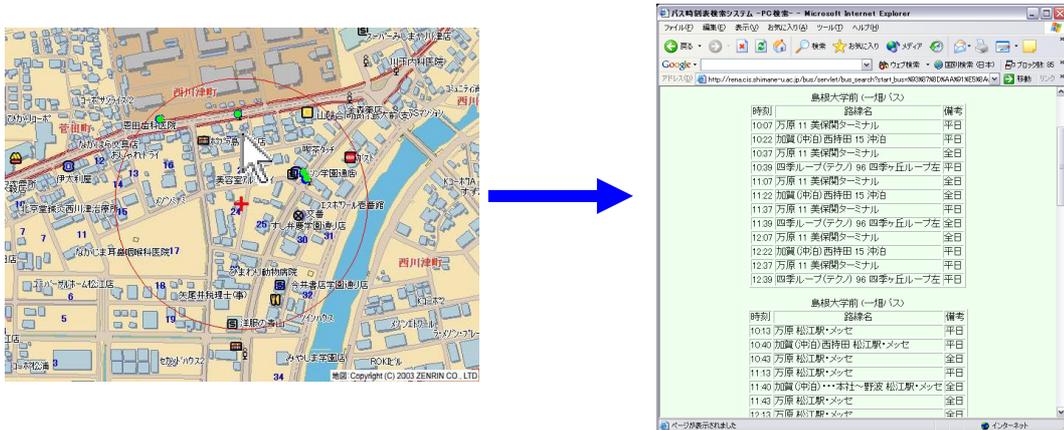


図 2. パソコンからの検索システム

## 2.2 システム機能

### ・GPS 携帯端末の場合

GPS 機能を利用して、図 1 の様に緯度・経度を取得し、そのデータを携帯端末のインターネット接続機能を利用して、Web サーバに送信し、サーバでその現在位置の近くにあるバス停を検索し、現在位置の地図と近くにあるバス停の時刻表を表示する。

### ・GPS 機能のない携帯端末の場合

バス停名によるバス停の検索及び、そのバス停の時刻表の表示をする。また、目的となるバス停名を入力することで、目的のバス停まで行く路線を検索し、その時刻表だけに絞り込んで表示する。

### ・パソコンからの場合

図 2 の様に画面上に表示されている松江市の地図をクリックすることで、その場所の詳細地図が表示され、その地図のサークル内(半径 100m または半径 200m)にバス停がある場合に、点が表示されているので、その点をクリックすることで、そのバス停の時刻表を表示する。

また、緯度・経度を直接入力することでも、バス停を検索することができ、バス停名でも時刻表を検索することができる。

まとめると、本システムではバス停の検索に関して、各端末で以下のような検索に対応している。ただし、パソコンについては環境がそれぞれ異なるので、携帯端末については全ての機種で動作確認を行うことが出来ないで、正しく動作しない場合もあると思われる。逆に、今はパソコンのコンテンツも各携帯端末別のコンテンツも制限をしていないため、違う携帯端末のコンテンツなども見ることはできるので(文字化けや対応していないタグ、対応していない機能などがあるため正しくは表示されないと思われる。)、以下の表で×となっている所でも検索することが出来るものもある可能性はある。

		文字検索	GPS検索	地図検索
携 帯 電 話	au	GPSあり	1	×
		GPSなし	×	×
	DoCoMo	GPSあり 2	×	×
		GPSなし	×	×
	Vodafone	3	×	×
パソコン		4		

- 1 auの型番が5000系と3000系の両方に対応しているが、3000系に関しては検証は行えなかった。動作検証はA5501Tで行った。
- 2 2004年2月13日現在、DoCoMoからはF661iとF505iGPSの2機種が発売されているが、時間の関係上コンテンツを作成できなかった。
- 3 Vodafoneはi-mode用のコンテンツで表示が出来る可能性があるが、動作確認をすることが出来なかった。
- 4 GPSレシーバを用いてGPS携帯のような検索することには現在対応していない。経度と緯度を直接入力することで検索できる。

表 1. 各端末からの検索の対応状況

## 2.3 システム環境

まず、試作した検索システムを作成した環境は以下の通りである。

### 作成環境

OS : Microsoft Windows XP / 2000 Professional / 2000 Advanced Server

Java : Java2SDK と Borland JBuilder 8 Personal

C 言語 : Microsoft Visual C++

その他 : HiRDB 付属のプレコンパイラ pdcpp.exe

次に、本システムの動作環境は以下の通りである。

#### サーバ環境

CPU : Intel Pentium 866MHz

RAM : 256MB

HDD : 8.5GB

OS : Microsoft Windows2000 Advanced Server

DBMS : HITACHI HiRDB Version 6 および、プラグイン

HiRDB Spatial Search Plug-in Version 3

Web サーバ : Apache Tomcat 1.4

その他 : C プログラムの使用に Visual C++ の dll ファイルが必要

#### クライアント環境

- ・ PC からアクセスの場合

OS : Microsoft Windows2000 以降

ブラウザ : Microsoft Internet Explorer 5.0 以上

(Netscape Navigator には非対応)

その他 : JavaScript が有効

これらが正常に動作する環境で動作。

- ・ 携帯端末からアクセスの場合

au の場合

GPS を使用する場合、GPS 機能搭載の携帯端末。

GPS を使用しない場合では、画面が 120×120 ピクセル以上でカラー表示可能な携帯端末。(QVGA 表示が可能な携帯端末を推奨)

DoCoMo の場合

画面が 120×120 ピクセル以上でカラー表示可能な携帯端末。ただし現在は文字による検索のみ(QVGA 表示が可能な携帯端末を推奨)

vodafone の場合

現在是对应していない。(DoCoMo の検索システムで一部表示ができる可能性もあるが検証していない)

## 第3章 システムを実現する要素技術について

### 3.1 DBMS

#### 3.1.1 DBMS (DataBase Management System)

DBMS (データベース管理システム)とは、共有データとしてのデータベースを管理し、データに対するアクセス要求に応えるソフトウェアの事である。データの形式や利用手順を標準化し、特定のアプリケーションソフトから独立させることができる。DBMS を利用する利点として、一般に次のようなことが挙げられる。

データを物理的・論理的にアプリケーションプログラムから独立させることが可能で、プログラムの負担を軽減できる。

データファイル間でのデータの重複を回避することができ、データの整合性を容易に確保することができる。

データをデータベースで一元管理することで、データの標準化や共有が可能である。

セキュリティの確保が容易である。

複数のユーザからのデータ操作を、データに矛盾を生じさせることなく行える。

データベースに障害が発生しても、DBMS に備わっている機能によって、これを復元できる。

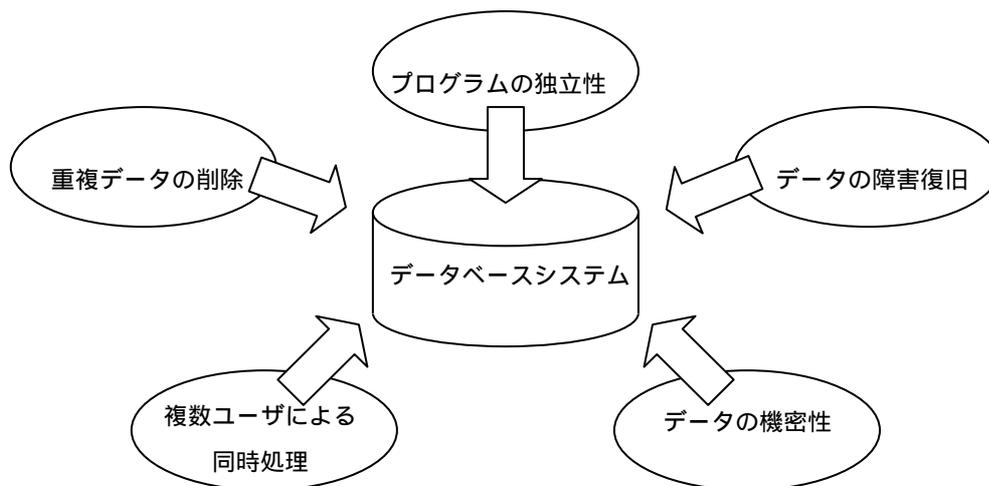


図 3. DBMS の利点

DBMS 中でも RDBMS (Relational DataBase Management Syaytem) と言われる管理方式がある。RDBMS とは、1 件のデータを複数の項目(フィールド)の集合として表現し、データの集合をテーブルと呼ばれる表で表す方式で、ID 番号や名前などのキーとなるデータを利用して、データの結合や抽出を容易に行なうことができる管理システムである。

本システムは、純国産 RDBMS である HITACHI HiRDB Version 6 を用いて時刻表やバス停情報といったデータベースの管理を行っている[7][28]。

### 3.1.2 利用者定義型 (抽象データ型) と空間検索プラグイン

本システムで使用している HiRDB には、抽象データ(GEOMETRY)型と呼ばれる利用者によって定義を行うことができる機能がある。また、この抽象データ型を利用して、HiRDB のプラグインという形で、空間検索を行うことができる空間検索プラグイン(HiRDB Spatial Search Plug-in)が提供されている。バス停の位置情報をこの GEOMETRY 型で定義して格納することで、比較的簡単な SQL 文の記述で、検索を行うことができる。

そこで、本システムにおいて、現在位置の近くのバス停を検索する機能を実現するためにこのプラグインを使用している。このプラグインは、データベースにこの抽象データ型で、定義したデータを格納しておくことで、様々な空間検索(座標を用いて、図形的(円形、多角形など)な検索を行う機能)を提供するものである。具体的には図 4.のような検索ができる[2]。

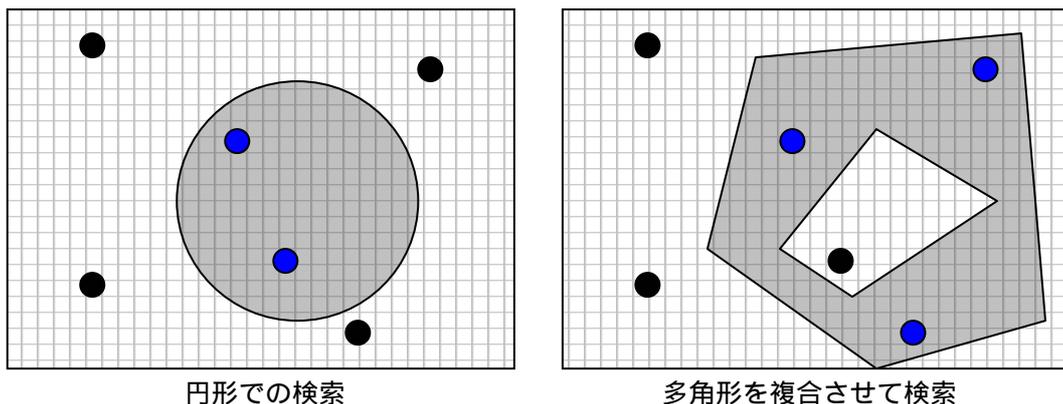


図 4. 空間検索プラグインの提供する検索(一部)のイメージ

### 3.1.3 格納データ

本システムの格納データは、バス停名、バス停の位置座標、バス路線名、バス路線情報、バス停の時刻表などである。このバスに関するデータは、松江市の方から提供頂いたものを使用させて頂いた。本システムの試作を開始する際に松江市の方に時刻表とバス停位置のデータの提供をお願いしたところ、たびねっと松江の時刻表・バス停データを学術研究のために提供をして頂いたものである。

その提供して頂いたデータを、本システムの利用に適した形へ変換(4.4 DB 格納データで説明する)をしたものを格納している。

## 3.2 Web サーバ

### 3.2.1 Web サーバ

Web サーバとは WWW システムにおいて、情報送信を行なうコンピュータ、または、WWW による情報送信機能を持ったソフトウェアのことである。

Web サーバは、HTML 文書や画像などの情報を蓄積しておき、Web ブラウザなどのクライアントソフトウェアの要求に応じて、インターネットなどのネットワークを通じて、これらの情報を送信する役割を果たしている。

本システムで、検索を行う場合、インターネットを利用してクライアント側からのデータを受け取り、サーバ側で検索を行い、その検索結果をクライアント側のブラウザで表示する形態であるので、当然ながら、クライアントと DBMS との仲介を行うための Web サーバが必要である。また、この Web サーバには、静的な HTML 文書だけでなく、DBMS へ接続して SQL 文を発行し、検索を行い、その結果を動的にブラウザに出力する機能が必須である。

また、クライアントからのデータの受け渡しや DBMS への検索の命令、検索結果の動的な作成には、Java サブレットを使用することにした。それは、サブレットは Java 言語で記述されているため、特定の OS やハードウェアに依存することがなく、サブレット API を実装したあらゆる Web サーバで稼働させることができる。また、CGI などの他のサーバサイドプログラムと異なり、一度呼び出されるとそのままメモリに常駐するため、高速な処理が可能である。また、データを永続的に扱うことができるため、複数のユーザ間で情報を共有することもできるためである。

そこで、Web サーバには、フリーで使用でき、Java サブレットが使用できる Jakarta Tomcat 1.4 を使用することにした[28]。

### 3.2.2 JDBC (Java DataBase Connectivity)

JDBCとはJavaプログラムからリレーショナルデータベースにアクセスするためのAPIのことである。SQL言語による命令を発行してデータベースの操作を行なうことができ、データベースの種類によらない汎用性の高いプログラムを開発することが可能である。

JDBCドライバは、各DBMS用のドライバが必要である。HiRDBには標準でこのHiRDB用のJDBCドライバが付属しているため、このドライバを使用して、JavaサーブレットからDBMSへSQL文を発行して、検索と検索結果の受け取りを行っている。

HiRDBのJDBCドライバはODBC (Open DataBase Connectivity)をマッピングすることで実現されているため、処理が重くDBMS内での検索は高速であるにもかかわらず、データの受け渡しで時間がかかってしまうという問題がある[28]。

## 3.3 プログラミング言語

### 3.3.1 埋め込みSQL

埋め込みSQLとは、C言語プログラム等の中に直接SQL文を埋め込んでしまう方式である。当然そのままCコンパイラでコンパイルは出来ないが、前処理によってCコンパイラが処理可能な形式に変換する。具体的にはEXEC SQLというキーワードをSQL文の前に記述する。SQL文の中にCプログラムの変数等を埋め込むことができるため動的なSQL文の発行が可能である。本システムでは、HiRDBにはpdcpp.exeというプリプロセッサが付属しているため、それを用いて、埋め込みSQLを通常のCプログラムにプレコンパイルすることで、使用することが出来る。

本システムでは、この埋め込みSQLを使用して、時刻表の検索を行っている。Javaサーブレット内でこの埋め込みSQLを用いたCプログラムを呼び出して、検索を行い、その検索結果をJavaサーブレットで受け取ることで、クライアント側に検索結果を表示しているという間接的な使用方法をとっている。また、JavaサーブレットでJDBCドライバを使用した場合と比較して、埋め込みSQLを使用したCプログラムの方が、高速に検索と表示ができる[3][28]。

### 3.3.2 Java サブレット

Java サブレットとは、Web サーバ上で実行されるモジュール(部品)化された Java プログラムのことである。サブレットを追加することにより、Web サーバの機能を拡張することができる。前述したようにサブレットは Java 言語で記述されているため、特定の OS やハードウェアに依存することがなく、サブレット API を実装したあらゆる Web サーバで稼働させることができ、CGI などの他のサーバサイドプログラムと異なり、一度呼び出されるとそのままメモリに常駐するため、高速な処理が可能である。また、データを永続的に扱うことができるため、複数のユーザ間で情報を共有することもできるというような特徴をもっている。

本システムでは、この Java サブレットを用いることで、クライアントとサーバとのやり取りを行っている。全ての処理をこの Java サブレットでやるのが可能であるが、HiRDB の JDBC ドライバを使用した場合に受け渡すデータの量が増大した場合(本システムでは時刻表の検索)に時間がかかってしまうため、実使用には耐えうる結果となってしまったために、埋め込み SQL で一部機能を行うことで、この問題を解決している[28]。

## 3.4 携帯端末

### 3.4.1 各キャリア

2004 年 1 月末現在、携帯端末の契約台数は 80,128,800 台と言われている(社団法人 電気通信事業者協会(TCA)調べ)。個人向けの携帯端末の事業者は、シェアの大きさで、NTT DoCoMo、au(KDDI、TuKa)、vodafone の 3 つに大きく分けられる。各事業者の契約台数はそれぞれ、45,365,900 台、15,977,300 台、14,774,000 台である。また、現在ではインターネットに接続する機能を搭載した携帯端末の方が圧倒的に多い。

その中で現在、GPS 機能を搭載した携帯端末を販売しているのは 2 つの事業者である。まず、2 番目に契約台数の多い au の比較的新しい携帯端末のほとんどの機種に、GPS 機能を搭載している。そこで、これらの携帯端末をメインにしている。また、契約台数がトップの DoCoMo にも、携帯端末に GPS 機能を搭載したものを販売しているが、出荷台数が少ないことと、時間の都合でシステムを作成することができなかった。

### 3.4.2 携帯端末による Web コンテンツ

携帯端末の事業者によって、表示できる Web コンテンツに多少違いがある。ほとんどの場合、同じ事業者のものであれば対応しているが、同じ事業者の携帯端末であっても、開発したメーカーや発売時期などによってもコンテンツの対応状況が異なっている場合もある。

また、Web コンテンツの作成については以下のような記述言語を使用することになっている。

#### ・ DoCoMo(i-mode)

i モード対応 HTML(Compact HTML)と呼ばれる記述言語で作成された Web ページを閲覧することができる。これは、文法自体は基本的に HTML 2.0 / 3.2 / 4.0 と同じで、狭いディスプレイでは表示しきれないタグなど不要なものが省かれた、サブセットとなったものである。そのため HTML との親和性が高く、HTML で記述されたコンテンツをコンパクト HTML 向けに移植しやすい、というメリットがある。i モード対応 HTML では、電話番号へのリンクを埋め込む機能などが独自に追加され、文字コードは、シフト JIS のみに対応し、独自に拡張された機種依存文字(絵文字)がある。また、半角カナの使用も認められている。

また、この i モード用のコンテンツを作成することで、他の事業者の携帯端末でも閲覧することが可能な場合が多い(共通した部分が多く、他の事業者がその事業者用のコンテンツに変換などで互換表示する場合もあるが、正しく表示されない場合もある)。また、mova 用と FOMA 用の i モード対応 HTML があり、バージョンによって、対応した機能が異なっている。最近では Flash に対応したコンテンツも作成できる[15]。

#### ・ au(ezweb)

au では、最初 HDML(Handheld Device Markup Language)と呼ばれる au 独自の携帯端末用に開発された記述言語を用いて、コンテンツを作成していたが、現在では、XHTML Basic と呼ばれる HTML の仕様(用語)と同じものを、XML の構文を使って改めて仕様化した言語で、携帯端末向けのもので記述している。XHTML を使用(携帯端末向けにしてある)していることで、今までのコンテンツよりも実現できる機能が豊富である。また、文字はシフト JIS のみに対応し、機種依存文字(絵文字)もある[16]。

- **vodafone(Vodafon Live!)**

ボーダフォンライブ！向け HTML と呼ばれる、ボーダフォンライブ！ウェブで利用可能なマークアップ言語で記述された Web ページを閲覧することができる。au と同じ XHTML Basic に沿った仕様となっている。ただし、スタイルシートには対応しておらず、対応文字はシフト JIS のみと言ったような制限もある。多様なコンテンツを作成することが可能である[17]。

どの事業者の Web コンテンツ作成用の記述言語でも、最近の携帯端末の処理能力などの向上やメモリ容量の増大によって様々なマルチメディアコンテンツを作成することができる。ただし、どのキャリアの携帯端末にもページ容量の制限も存在する。また、Web コンテンツを表示するとそのパケット量に応じて課金される仕組みとなっているので、ページは可能な限り小さくするべきである。

## 3.5 GPS

### 3.5.1 GPS (Global Positioning System)

人工衛星を利用して自分が地球上のどこにいるのかを正確に割り出すシステム。高度約 2 万 km の 6 つの円軌道に 4 つずつ配された米国防総省が管理する GPS 衛星からの電波を利用し、緯度、経度、高度などを数十メートルの精度で割り出すことができる[28]。

### 3.5.2 GPS の座標系

座標系には、国内では世界測地系(WGS54)と日本測地系(TOKYO)の 2 つの座標系があり、使用されている。世界測地系は、世界標準の測地系であり、日本測地系は国内で標準となっている(現在の国内の地図などは日本測地系で表記されている場合が多いが、この 2 つの測地系が混在していたり、世界測地系で表記されている場合もある)測地系である。この測地系を間違えると、約 400m ほどの誤差が生じてしまう事態になってしまう[20]。

### 3.5.3 携帯端末の GPS 機能

au の GPS 携帯端末

- **gpsOne**

QUALCOMM 社が開発した、cdmaOne/cdma2000 端末向けの位置情報取

得技術。GPS(全地球測位システム)を応用した技術で、衛星から取得したデータを基地局にそのまま送信し、実際の位置確定は基地局によって行なう。GPS衛星からの受信についても基地局から端末に衛星を探索するための情報が送信され、衛星からの受信開始に必要な時間を大幅に短縮している。GPSからの電波が届かない屋内では、基地局からの電波を利用して測位を行なう。位置情報の精度は通常のGPSと同等の5m~10mレベルとなる。

#### ・eznavigation

KDDIが同社の携帯端末サービス「au」の情報サービス「EZweb」で提供している、現在位置検索サービス。同社販売の携帯端末のうち、5000/3000の両シリーズで利用できる。eznavigationでの現在位置割り出しには、QUALCOMM社が提供している「gpsOne」を利用している。対応端末にはGPS受信機が内蔵されており、GPS衛星からの電波を受信するほか、基地局からの情報も利用して現在位置を割り出すシステムとなっている。このため、eznavigationの精度はGPS受信機と同等以上の高さを誇る一方、EZwebと連動して使用するサービスのため、cdmaOneのサービスエリア外では一切のサービスを受けることができない。

auのGPS携帯端末は当初、携帯端末で受信したデータをauのサーバに送信して、計算をサーバ側で行い、その結果を携帯端末に送信してからWebコンテンツのサーバへその情報を送信するという形態をとっていた。しかし、最近のGPS携帯端末の中には受信したデータを携帯端末側で計算し、その情報を直接Webサーバへ送信することができるものがあり、計測速度が向上している。

また、auのGPS携帯端末は、世界測地系(WGS84)と日本測地系(TOKYO)の2つの測地系の計測が可能である。本システムでは日本測地系(TOKYO)を使用しているため、現在位置を取得する場合は、日本測地系で測位を行っている[28]。

#### DoCoMoのGPS携帯端末

DoCoMoの場合、GPS携帯端末で受信したデータをDoCoMoのサーバへ送信し、サーバ側で経度と緯度を計算し、その計算結果を携帯端末へ送信してからWebコンテンツへその情報を送信している。また、世界測地系(WGS84)の座標系でしか測定することができないため、本システムで使用する場合は、本システムが日本測地系(TOKYO)を使用しているため、日本測地系(TOKYO)に変換する必要がある[18]。

## 第4章 検索システムの機能構築

### 4.1 今までのシステムとの相違点

本システムと今までのシステムの違いは、まず、携帯端末のGPS機能を用いて、現在位置の近くのバス停を検索できる機能である。電車の駅などの場合は対応している場合があるが、バス停の場合は対応していない場合がほとんどである。

また、PC上からアクセスした場合に文字からの検索だけでなく、地図をクリックすることで、バス停を検索し、そのバス停の位置を地図上に丸い点として表示し、その点をクリックすることで、そのバス停の時刻表を検索ができるので、直感的に操作が行える機能である。

そして、検索にDBMSを使用していることある。DBMSを使用することで、検索が高速かつ確実に行え、セキュリティも高く、データの更新もいつでも行うことができ、また、使用者の権限を設定することで、Webサーバの運用中に管理者のみならず、各バス会社のデータならばそのバス会社の方で更新できるように設定することもできる。

従って、本システムは今までのシステムよりも、検索するユーザ側はより簡単な操作で使用でき、管理者側はセキュリティも高く、メンテナンスが簡単であるシステムである。

### 4.2 本システムの機能

#### 4.2.1 GPSを利用した検索機能

GPS携帯端末(現在の環境ではauの携帯端末)を使用して、緯度と経度を取得し、そのデータをWebサーバに送信することで、現在位置の近くのバス停を検索し、現在位置の地図とバス停の時刻表を携帯端末の画面に表示する。

まず、Webサーバにアクセスし、その中の”位置情報取得”というリンクを押す。(GPSを利用して現在位置の情報を取得し、その情報をWebサーバに送信することまで自動で携帯端末が行う。)そして、送信された経度と緯度の情報からWebサーバが、現在位置を中心に半径100m以内にあるバス停を検索する。もし、バス停があれば、検索されたバス停を携帯端末に表示し、バス停がなければ、半径200mで検索をし直し、それでもなければ半径300mで検索し直し、バス停があれば表示し、なければ”バス停はありませんでした。”と表示する。

ユーザ側から見ると、次の図のようにほぼ自動的に近くのバス停を検索することになる。

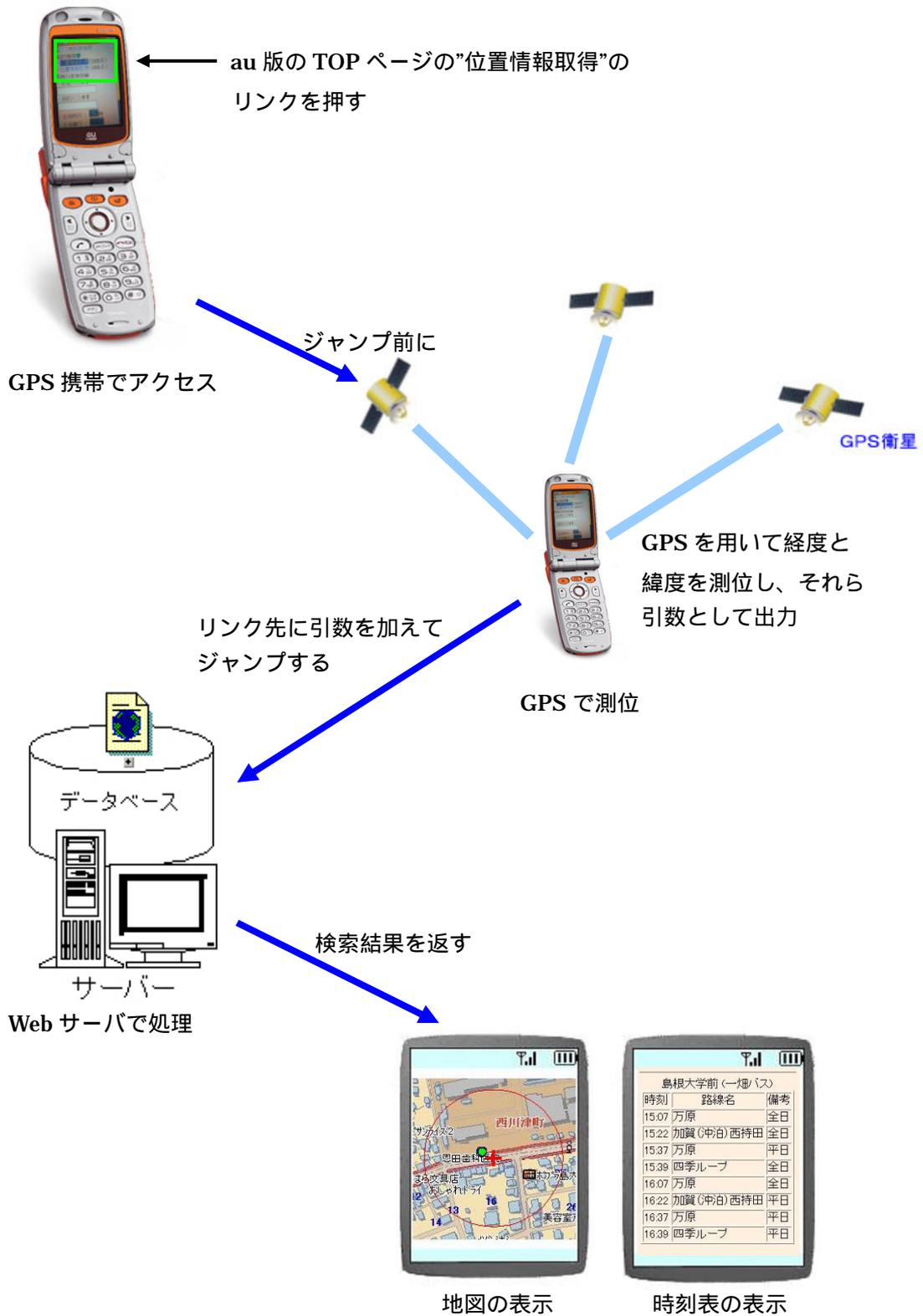


図 5. GPS 携帯端末の利用

#### 4.2.2 地図からの検索機能

今までの検索システムと同じように文字からの検索だけでなく、地図をクリックすることで、バス停の検索と検索された地図上のバス停をクリックすることで、そのバス停の時刻表を検索することができる。

まず、Web サーバにアクセスし、松江市全体の地図の検索したい場所をクリックする。クリックした場所の詳細地図が表示され、もしクリックした場所から半径 100m(または半径 200m)以内にバス停があれば、青または緑の点でバス停が表示される。点以外の地図上をクリックすると、クリックして場所を中心に地図が移動するので、調べたいバス停が半径 100m(200m)以内に入るように移動して、青または緑の点で表示されたら、その点をクリックすることで、そのバス停の時刻表を検索できるので、直感的な操作で検索できる。

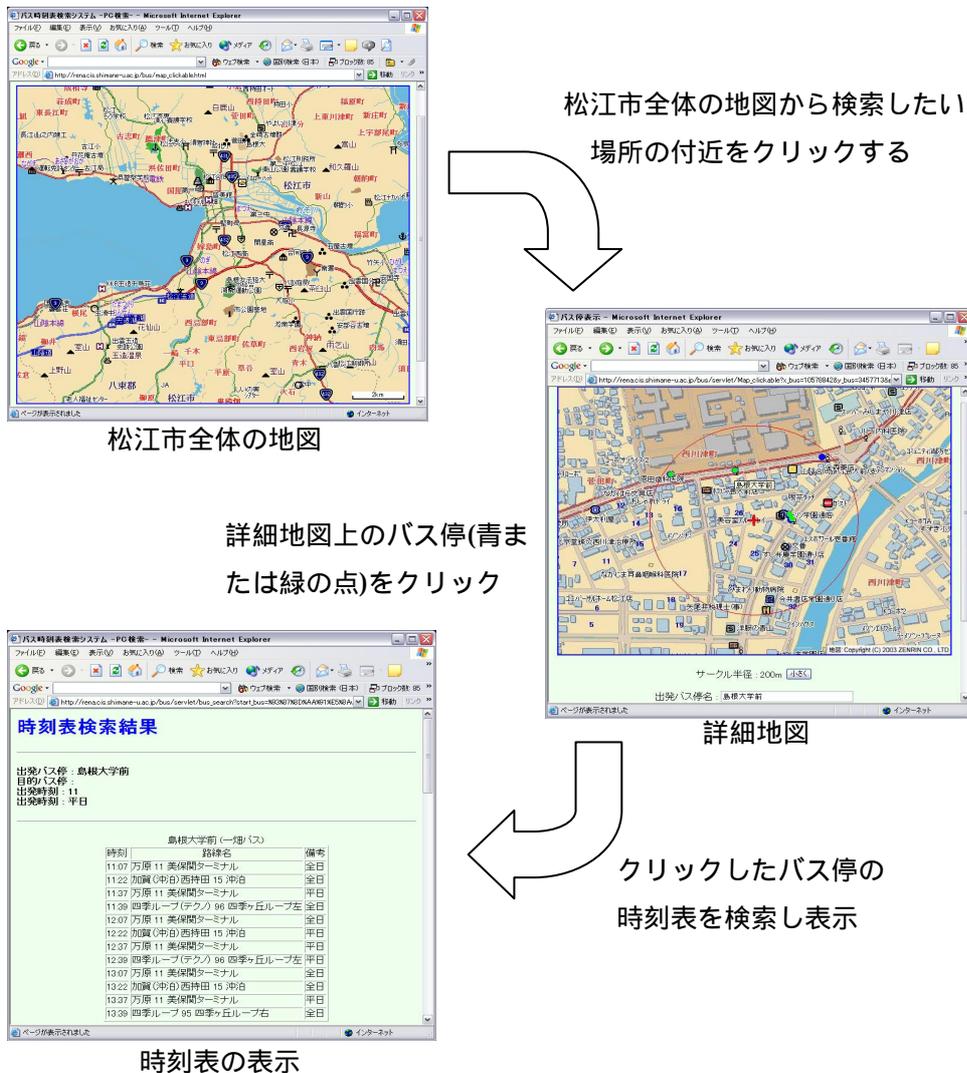


図 6. 地図からの検索

## 4.3 本システムの仕組みと実装

### 4.3.1 基本システム

#### ・システム概要

まず、各データ(バス停の名前、バス停の位置データ、バスの路線のデータ、時刻表のデータ)は、DBMS である HiRDB を使用して、格納している。そして、この HiRDB を用いて、検索も行う。

ユーザからの検索の際は、インターネットを利用するので、Web サーバアプリケーションが必要となるので、そのアプリケーションには、今回、Java サブレットを用いて、動的に検索結果の表示を行うので、Tomcat を使用し、動的に HiRDB へ SQL 文を発行し、その結果を各ユーザの端末へ送信して表示を行っている。

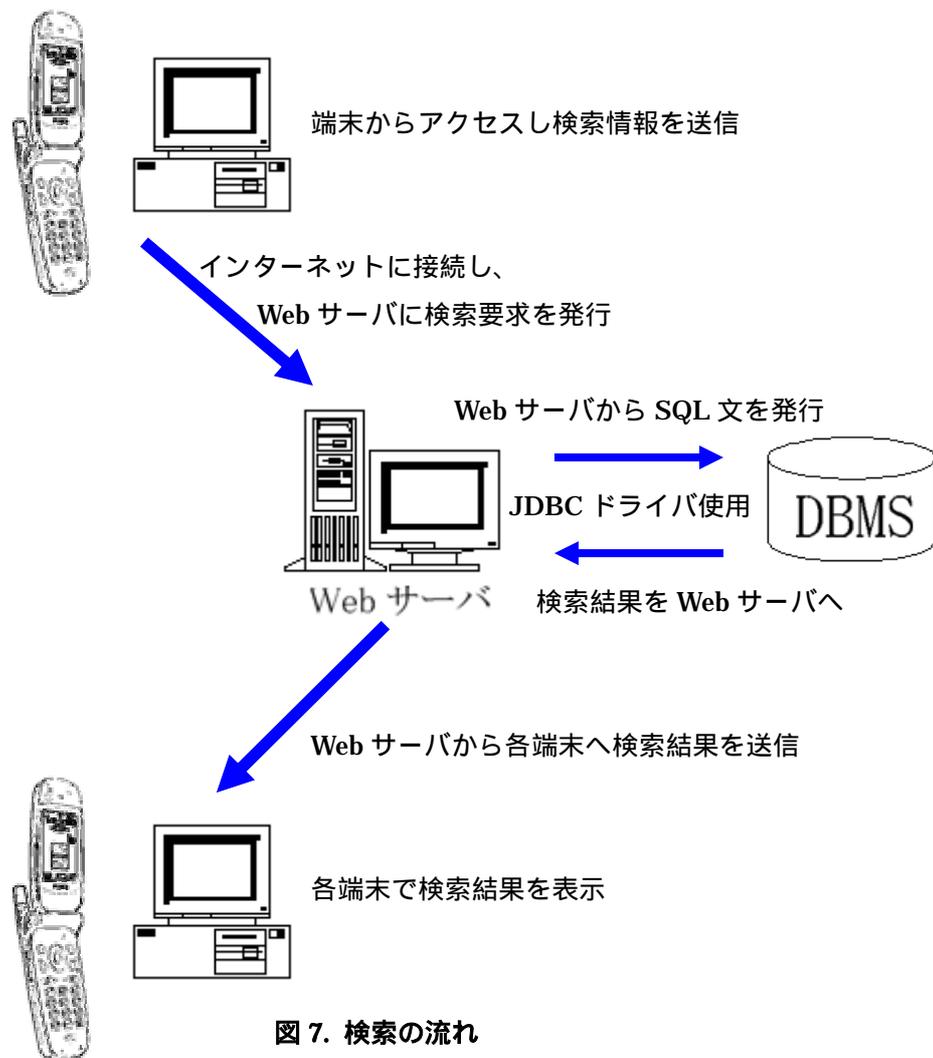


図 7. 検索の流れ

・ Java サブレット(埋め込み SQL)と DBMS の動作

本システムの検索の基本は各端末からサブレットにアクセスした際のパラメータを受け取り、そのパラメータを用いて Java サブレットから JDBC ドライバを使用して、DBMS へ SQL 文を発行し、DBMS で検索した結果を Java サブレットで受け取り、各端末へ検索結果を出力する形態となっている。

つまり、バス停名の一部(または全部)を入力して、その文字列を含むバス停を検索するサブレット(bus\_name.class)を例にすると、

(例)bus\_name へアクセスした場合

pc.html で以下を入力

バス停名を入力して時刻表を検索する。

出発バス停名	<input type="text" value="島根大学"/>	←	出発バス停名に"島根大学"
目的バス停名	<input type="text"/>	←	目的バス停名は空白
出発の時刻	<input type="text" value="10"/>	←	出発の時刻は"10"時を選択
出発の曜日	<input type="text" value="平日"/>	←	出発の曜日は"平日"を選択

バス停検索をクリック

バス停検索をクリックすると以下の URL が作成されそこへジャンプする。

```
http://rena.cis.shimane-u.ac.jp/bus/servlet/bus_name?start_bus=%93%87%8D%A  
A%91%E5%8Aw&goal_bus=&time_bus=10&day_bus=%95%BD%93%FA
```

ここで、bus\_name はサブレット名、start\_bus は出発バス停名、goal\_bus は目的バス停名、time\_bus は時刻、day\_bus は曜日を表す引数名である。

この引数を bus\_name が受け取ると、以下のような SQL 文を作成する。そして、この SQL 文を JDBC ドライバを使用して、DBMS へ発行する。

```
SELECT バス停名 FROM 名前 WHERE バス停名 LIKE '%島根大学%';
```

ここで、"名前"は表の名前、"バス停名"は列名、%は0文字以上の文字列を表す。そして、DBMS 側で検索し、"島根大学前"というバス停名が検索される。そして、JDBC ドライバを用いて、Java サブレットで受け取り、ユーザの端末へ検索結果を送信する。

バス停名	<input type="text" value="島根大学"/> <input type="text" value="島根大学前"/> <input type="text" value="部分一致検索"/>	←	検索された結果を返す
目的バス停	<input type="text"/>		
出発時刻	<input type="text" value="10"/>	←	選択したものをそのまま表示
出発曜日	<input type="text" value="平日"/>		

上記がこのシステムの基本的な Java サーブレットと DBMS との動作である。しかし、この方法では、HiRDB の JDBC ドライバの動作が非常に重たいため、検索自体は高速であるにもかかわらず、時刻表を検索する場合には、データの受け渡しの数が多いために、DBMS から Java サーブレットへのデータの受け渡しで時間がかかってしまい、DBMS の高速な検索の性能を發揮できない。(受け渡すデータが少なければ Web サーバの用途であれば十分高速である。)

そこで、時刻表の検索についてのみ JDBC ドライバを使用せずに C 言語で作成し、埋め込み SQL を使用したアプリケーションによって DBMS へ SQL 文を発行、DBMS で検索し、検索した結果を Java サーブレットで受け取る形態をとっている。

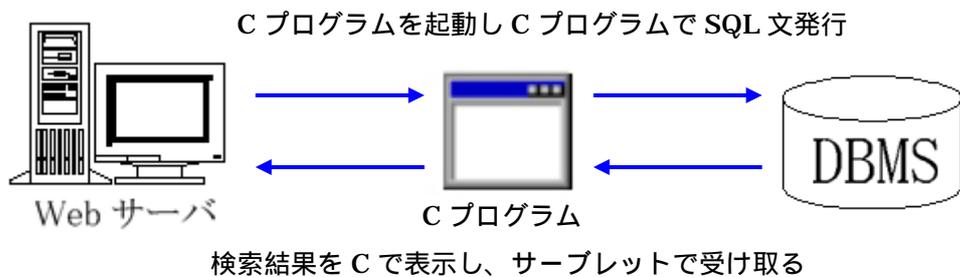


図 8. 時刻表検索の場合

・本システムの構成

本システムは以下のような html と Java サーブレットによって構成されている。

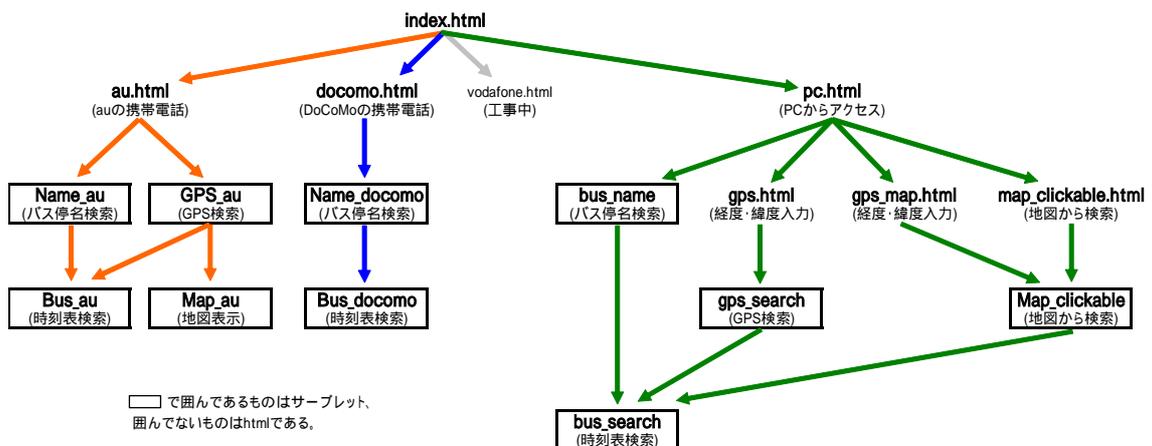


図 9. 本システムの構成の全体図

### 4.3.2 GPS による検索システム

次に GPS を利用して検索する場合、現在位置の位置情報と各バス停の位置情報を HiRDB の空間検索プラグイン(Spatial Search Plug-in)を利用して DBMS で検索を行っている。まず、各バス停の位置情報(経度・緯度それぞれ日本測地系)を変換したもの(4.4 格納データで説明)をジオメトリ型で DBMS へ格納している。そこで、GPS を用いて取得した位置情報(経度・緯度それぞれ日本測地系)を格納された各バス停のデータと同じ手順で変換し、その変換したデータを元に HiRDB(DBMS)で半径 100m 以内のバス停の検索を行う。

HiRDB の空間検索プラグインでは、ジオメトリ型で定義されたデータを様々な形で検索することが可能である。例えば、円形、矩形、多角形などである。また、それらを組み合わせてドーナツ状の検索も行うことができるが、本システムでは半径 100m 以内の検索であるので、円形の検索を行っている。実際の HiRDB へ発行する SQL 文は、以下のようなものである。

```
SELECT バス停名 FROM バス停位置 WHERE WITHIN(座標, RegionFromText('CIRCLE(経度 緯度, 半径[100m]')) IS TRUE;
```

”バス停名”は列名。”バス停位置”は表名。”WHERE WITHIN”以下が検索の条件。”RegionFromText( )”は HiRDB Spatial Search Plug-in の独自のもので空間検索する場合に( )内に条件を指定する(ジオメトリ型を指定)。”CIRCLE( )”も Spatial Search Plug-in 独自のものであり、円形を表し、( )内は x 座標、スペース y 座標、カンマ、半径の順で入力する。

経度・緯度を変換したものをこの SQL 文内に代入し、SQL 文を発行することで、ユーザの情報を元に検索することができる。

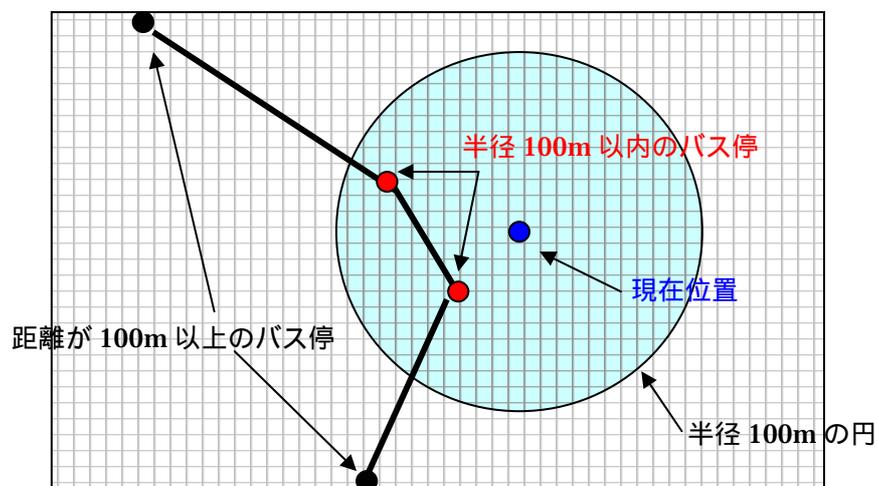


図 10. 空間検索のイメージ

### 4.3.3 au の GPS 携帯端末による検索システム

au の携帯端末で表示することができる Web ページの作成において、当初は au 独自の仕様である HDML という形式に対応していたが、現在は HTML を拡張し、携帯端末向けにした XHTML Basic という形式に対応しており、HTML に似た記述であること、au のサイト作成において、これからのスタンダードであると思われることなどから、本システムでは XHTML Basic で記述した。

au の GPS 携帯端末の GPS 機能を使用する場合、URL の記述に注意する必要がある。まず、au の GPS 機能(eznavigation)を使用するサイトの作成の方法は、残念なことに au の公式ホームページには掲載されていない。GPS 機能を持っていない機種でも使用することが出来る GPS を使用せず、基地局で位置情報を取得する簡易位置情報を使用する方法については、au の公式ページ (<http://www.au.kddi.com/ezfactory/tec/spec/eznavi.html>) に掲載があった。

そのため、google(<http://www.google.co.jp>)などの検索ページを利用して、検索を行ったところ、いくつかのサイトに GPS 機能を使ったサイトの作成方法が掲載されていた。au の公式ページに掲載されていなかったため、正しい記述であるかは検証することは出来なかったが、本システムでは今のところ正しく動作している。

なお、GPS を使用する記述は、ジャンプ先(サブレット)の URL の前に device:gpsone?を付加することで、GPS 携帯端末の GPS を使用することができる。また、その際に URL の後ろにはいくつかの決まったパラメータを記述しなければならない。また、独自のパラメータは使用することができないという制約がある。そして、5000 系と 3000 系の携帯端末では URL の後ろに記述するパラメータが異なる。

実際に XHTML Basic で記述する場合、位置情報からバス停を検索するサブレットである [http://rena.cis.shimane-u.ac.jp/bus/servlet/GPS\\_au](http://rena.cis.shimane-u.ac.jp/bus/servlet/GPS_au) に GPS で位置情報を取得してから、ジャンプさせるには以下のように記述する。

#### ・型番が 5000 系の携帯端末の場合

```
<a href="device:gpsone?url=http://rena.cis.shimane-u.ac.jp/bus/servlet/GPS_au?ver=1&datum=1&unit=0&acry=0&number=0">位置情報取得</a>
```

ここで、<a href="url">位置情報取得</a>は""で囲まれた URL へのリンクを表すタグであり、"位置情報取得"が実際の画面に表示される。

・型番が 3000 系の携帯端末の場合

```
<a href="device:gpsone?url=http://rena.cis.shimane-u.ac.jp/bus/servlet/GPS_au?ver=1&datum=1&unit=0">位置情報取得</a>
```

という記述となる。また、これらに記述しているパラメータについて、

- ・ **ver** : 不明(GPS のバージョンか?)
- ・ **datum** : 測地系(0 で世界測地系 WGS84、1 で日本測地系 TOKYO)
- ・ **unit** : 単位(0 で dms(度分秒)表示、1 で dd(度)表示)

の 3 つのパラメータに関しては、5000 系、3000 系の両方の型番の携帯端末で必要である。ver については情報が無かったため 1 を、本システムに関しては datum について日本測地系での測定を行う方が都合がいいため 1 を、unit についても dms(度分秒)表示の方が都合がいいため 0 を入力値としてある。また、以下の

- ・ **arcy** : 不明
- ・ **number** : 不明

の 2 つのパラメータは型番が 5000 系の携帯端末で必要となったパラメータである。arcy、number とともに不明であるため 0 を入力値としてある。また、実際にこの URL を記述したページにアクセスし、位置情報を取得すると、以下のようなパラメータを付加した上で、その url= から? で囲まれた URL のページへジャンプする。

```
http://rena.cis.shimane-u.ac.jp/bus/servlet/GPS_au?ver=1&datum=1&unit=0&lat=+35.28.52.82&lon=+133.04.12.85&alt=51&time=20031219154915&smaj=15&smin=10&vert=25&majaa=14&fm=0
```

ここで、付加されたパラメータについて

- ・ **lat** : 緯度
- ・ **lon** : 経度
- ・ **time** : 取得日時
- ・ **fm** : 測位方法(0 で GPS、1 で GPS + 基地局、2 で基地局か?)

であると思われる。また、その他のパラメータは不明である。測位して得られた緯度と経度のデータをサブレットで受け取り、それらの値を格納したデータベースの値と同じ方法で変換し、検索を行っている。

#### 4.3.4 地図の表示システム

地図の表示は、1つの Java サブレット(Map\_circle.class)で行っている。それは複数のサブレットで画像を扱うとメモリなどの問題で、画像を読み込めなくなったりする可能性があるからである。また、複数にすることでシステムが煩雑になることを避けるためでもある。

このシステムの地図表示は、大きな地図画像(JPEG 画像)を読み込み、その地図画像から現在位置の場所を中心に 640×480 ピクセルのサイズで切り取って JPEG 画像を出力する Java サブレット(Map\_circle.class)を別のサブレット内から読み込んで表示を行っている。

Tomcat で Web サーバを構築する場合、画像のサイズが大きくなると、Java サブレットに割り当てられるメモリ量の関係で、画像を読み込むことができない。また、1枚の地図画像だけでは松江市全体をカバーすることはできなかった。そこで、地図画像をいくつかの画像に分割し、別の場所の画像が必要になった場合にその画像を読み込むことで、メモリの問題を解決し、松江市全体の地図に対応させている。

地図の画像と経度・緯度の情報は、ゼンリンの電子地図帳 Z6 を用いて作成を行った。松江市の地図画像を切り出してそれらを用いて、まず、1枚の大きな地図画像(横 15040 ピクセル×縦 14480 ピクセル)を作成し、その画像を 42枚(縦 7枚×横 6枚)の分割した画像(横 3040 ピクセル×縦 2480 ピクセル)を作成した。

分割する際、そのまま分割してしまうと、分割した場所付近を切り取る場合にもう一度その隣の画像と合成しなければ分割した場所の付近を中心にして、画像を切り取ることができなくなってしまう。そこで、この検索システムの場合、地図画像の最大サイズは、横 640 ピクセル×縦 480 ピクセルであるので、本来切り取る場所からそれぞれ横は 320 ピクセル分ずつ、縦は 240 ピクセル分ずつ大きく切り取った。したがって、横は 640 ピクセル分、縦は 480 ピクセル分ずつ重複している部分を作ることで、切り取る部分の問題を解決した。

GPS 携帯端末を使用した場合には、半径 100m 以内にバス停がなかった場合に、順に半径 200m、半径 300m と検索を行う。また、QVGA(320×240 ピクセル)の表示ができる比較的最新の携帯端末であれば、画像を見ることができが、従来の 120×120 ピクセルくらいの表示の画面では見ることができない。そこで、この地図画像だけでは、不十分であるので、縮尺の違う画像も同じように作成して、ユーザのリクエストに応じて、読み込む画像を変更して表示を行うサブレットを作成した。

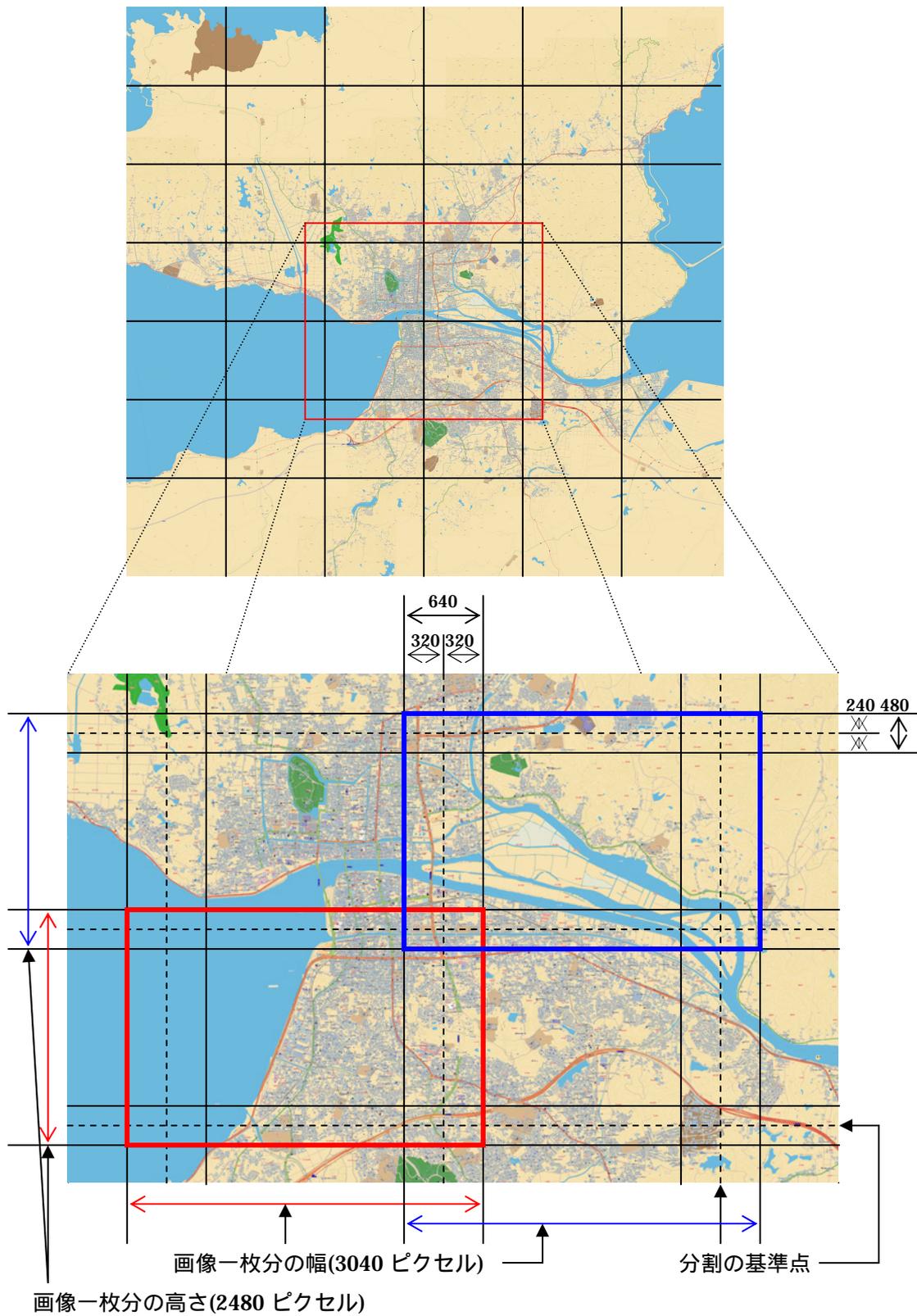


図 11. 地図の分割のイメージ

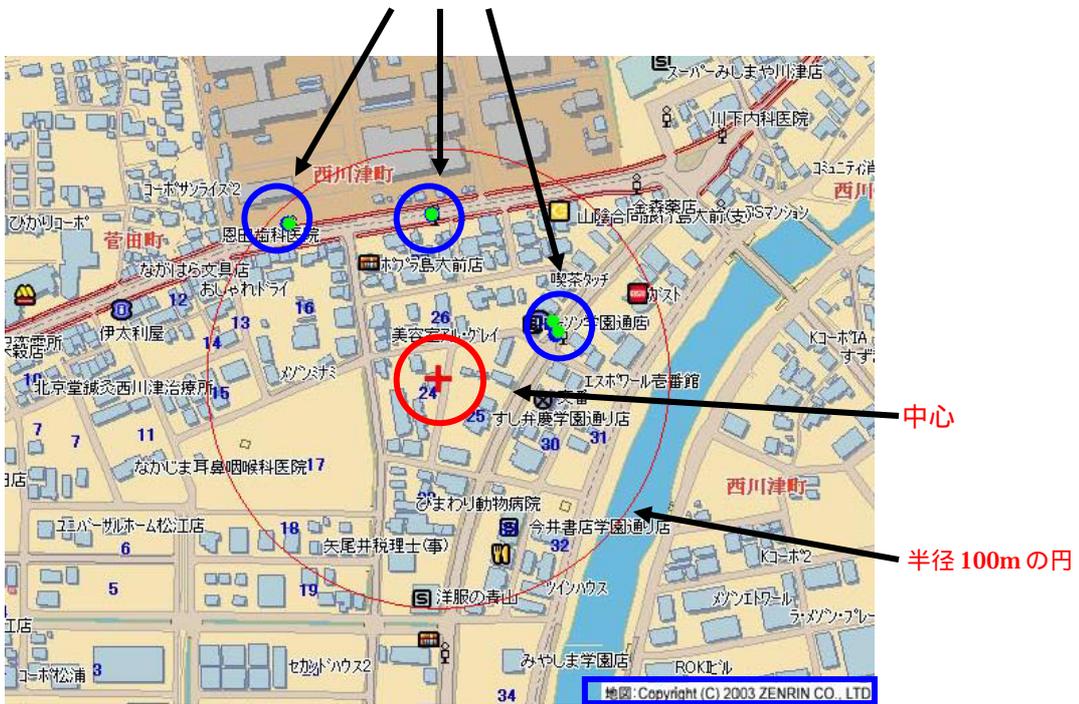
このように、地図を分割した理由は前述したようにプログラムに割り当てられるメモリの制限のためである。Java サブレットで画像を読み込む場合にその画像のサイズ(画像のピクセル数)が割り当てられるメモリ量よりも大きくなる(このシステムでは JPEG 画像を使用しているが、圧縮率を高くしても読み込めないため、恐らくは画像を展開した場合のファイルサイズだと思われる)と読み込むことができない。そこで、画像を分割し、必要に応じて読み込んだ画像をプログラム内で上書きすることで、松江市全体をカバーする仕組みとなっている。また、このプログラム内で、画像を上書きする場合には、読み込める画像の最大サイズで読み込むと、新しい画像を読み込む時にメモリの制限で読み込むことができなくない、また、別のサブレットでも読み込む事はできない。そこで、分割する画像のサイズを読み込める画像の最大サイズの半分とすることでこの制限を解決している。

当然、読み込む画像の上書きがなかった場合(画像を読み込まなかった場合)の方が画像の出力は高速である。従って、できるだけ大きな画像を読み込んだ方が、一度読み込んだ場所の近くを検索する場合(読み込んでいる画像と同じ画像を使用する場合はレスポンスが良くなる。また、分割する画像の数が少ない(つまり、分割された画像の一枚あたりの大きさが大きい)場合の方が管理が容易である。一方、画像の更新があった場合は、画像を読み込むための時間がかかってしまうため、レスポンスが悪くなってしまふ。これは、読み込む画像の大きさを小さくすることで、レスポンスを向上させることができる。本システムでは、管理の容易さを優先しており、読み込める画像の最大サイズの半分以下の大きさに、かつ、重複部分があるように分割を行ったため、前述したような画像の大きさになっている。

また、画像を出力する場合に、画像の著作権(ゼンリンの地図ソフトから作成したため)保護のため、地図画像の右下の部分に”Copyright(c) 2003 ZENRIN CO., LTD”と言う記述を画像に描画して出力している。実際には、左記の記述をした画像を JPEG 画像ファイルとして保存しておき、それを同じ地図表示サブレット(Map\_circle.class)で読み込み、出力サイズに切り取った地図画像の右下の部分(次ページ図 10.の上の PC 用の画像の右下を参照)に上書きして出力している。

また、画像をユーザの各端末毎にサイズを変えて切り取ることで、1つのタブレットで、このシステムの地図全てを出力している。それぞれの端末毎のサイズに切り取った後に、中心に赤で中心を表す十字の印と半径100m(または200m または 300m)の円を描画している。その後、DBMS で検索した半径100m(または200m または 300m)以内にあるバス停を青または緑の丸い点で地図上のバス停の位置に描画して、その画像を JPEG 画像として出力している。

検索されたバス停(青点もあるが緑点のために見えにくい)



地図に上書きした著作権保護用の画像

検索されたバス停



図 12. Map\_circle.class の表示結果(上が PC、下が QVGA 表示の携帯端末)

携帯端末の地図画像のバス停の位置を表す点は、図 10.の下のように青と緑だけではなく、9色ほどの色分けを行っている。これは携帯端末の場合、読み込めるページの容量や画面のサイズの制限があるためパソコンよりも表示できる情報量が遙かに少なく、さらに、パソコンの地図検索機能の様なバス停の上にマウスが重なった場合にそのバス停名を表示するという機能が現在の携帯端末では使えないため、2色で色分けしてしまうと、複数のバス停が検索された場合に、どの地図上の点とバス停名が関連づけられているのかが分かりづらいためである。

そこで、地図上の点と対応するバス停の名前の色とを同じ色で表示することで、地図上の点と対応するバス停名を関連づけし、一目でユーザが分かるようにしている。また、あまり色の違いがない場合、人間の目では区別が付かなくなってしまうことも考えられ、携帯端末によっても微妙に異なってしまうため、Java のプログラムや XHTML でも名前が付いている色 (RED や BLUE、GREEN といった色名で指定できるもの) を利用して、色分けを行った。Java のプログラムと XHTML で共通して名前を付けられている色で地図上に表示した際に見分けることができる色を選択すると9色となった(ディスプレイや人によっても色の見え方が異なるので、必ずしも正しい選択であると断言はできない)。場所によっては9つ以上のバス停が検索されることもあるが、実際に使用する場合、それほど問題にはならないと判断した。

#### 4.3.5 地図画像による検索システム

地図による検索は、JavaScript と Java サブレットを組み合わせることで実現している。Java Script を用いて、画像上の x 座標と y 座標を取得し、その座標を Java サブレットで受け取っている。

まず、地図上のバス停(青または緑の点)をクリックすることで、そのバス停の時刻表を検索する機能は、クリッカブルマップと JavaScript の組み合わせで実現している。地図画像全体をクリッカブルマップとして、地図上の青または緑の点の部分をクリッカブルマップにして、その部分をクリックした場合は、JavaScript を用いて、クリックしたバス停名(start\_bus)、目的バス停名(goal\_bus)、出発時刻(time\_bus)、出発曜日(day\_bus)を主なパラメータとして、時刻表検索サブレット(bus\_search.class)を呼び出して、時刻表の検索を行っている。この青または緑の点の座標を得るために、このサブレット(Map\_clickable.class)も地図を出力するサブレット(Map\_circle.class)と同じ SQL 文を DBMS へ発行して、座標値を取得することでクリッカブルマ

ップを作成している。また、ユーザが直感的に分かりやすいように青または緑の点(バス停)の上にマウスカーソルが重なった時だけ、地図画像の下にあるテキストフィールドにそのバス停名を表示する機能を JavaScript を用いて追加している。また、バス停を検索する半径を 200m にするボタンを追加し、切り替えることができるようになっている。

また、地図の青または緑の点以外の部分(バス停以外の地図上)をクリックするとその場所を中心に地図が表示される(地図上を移動する)機能もクリックブルマップと JavaScript を用いて実現している。上に書いたバス停以外の地図画像全体もクリックブルマップとして、そのバス停以外の地図上で、クリックした場合、その画像の(x, y)座標を Internet Explorer の JavaScript を用いて取得し、その座標と現在表示している場所の座標とを計算することで、クリックした場所の座標を導き出し、その座標をパラメータとして自分自身であるサブレット(Map\_clickable.class)を呼び出して、更新することで、画像を更新する(ユーザから見た場合に地図上を移動したように見える)機能を実現している。

また、地図画像の著作権保護のための方法として、地図画像の右下のメッセージの他に、地図画像上の右クリックを禁止にしている。ページ全体を右クリック禁止にしてしまうと操作がしづらくなったり、人によっては嫌悪感さえ感じてしまうため地図上のみ禁止にしている。ただし、Web ページで表示をする以上、完璧な画像の保護方法は存在しないため、ブラウザの JavaScript を OFF にしたりすることで、コピーできてしまうことが問題である。ただし、作成したオリジナルの地図画像(サブレットで読み込む画像)は公開していないため、直接コピーすることは不可能である。Web で公開していない場所のファイルをサブレットが読み込むため外部からは見ることができないのである。

地図からの検索システムは、JavaScript を多用しているため、スクリプトの設定は ON にしておく必要があり、また、Internet Explorer と Netscape Navigator では画像の座標を取得する関数などで互換性がないため、現在のシステムでは Internet Explorer のみしか動作しない。また、このシステムに使用している一部の機能(地図画像上での右クリック禁止の機能)が Internet Explorer 5.x 以上でないと動作しないため、このシステムが動作する環境を、Internet Explorer 5.x 以上としている。

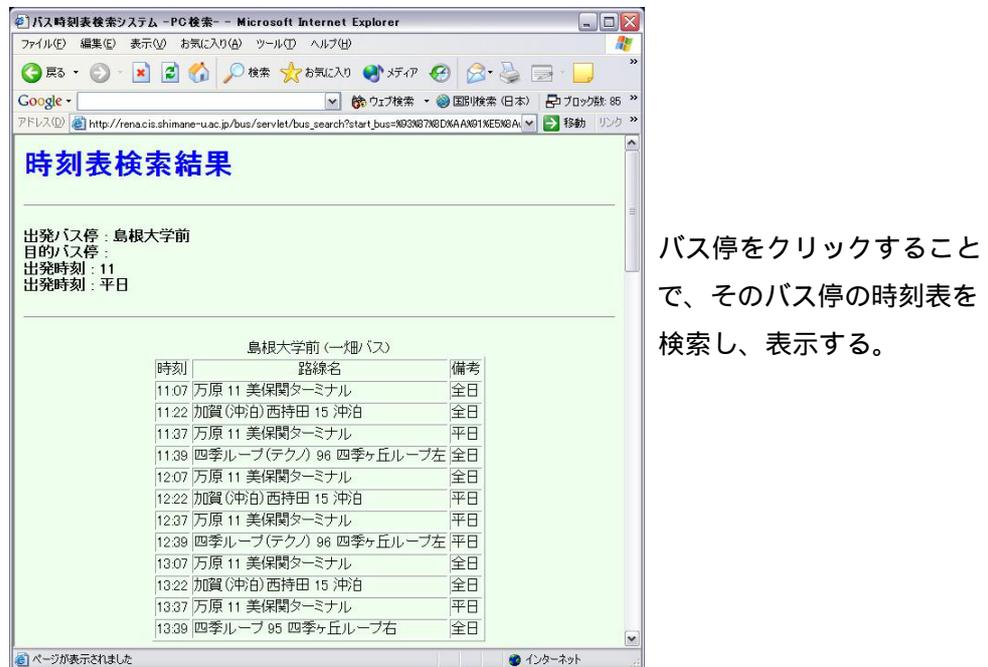
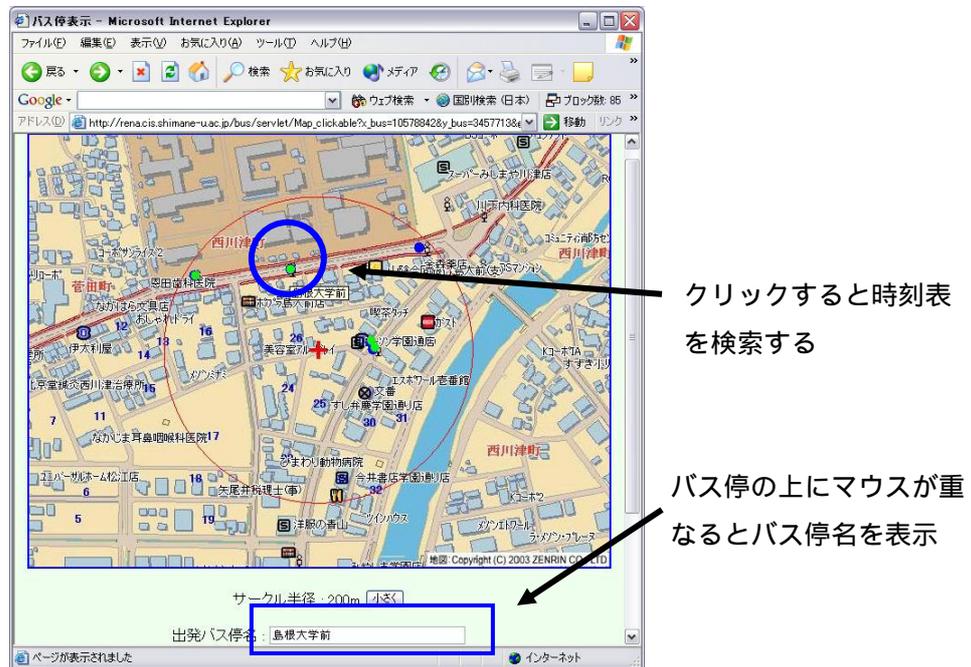


図 13. 地図からの時刻表検索

## 4.4 DB 格納データ

格納データに関して、バス停や路線のデータ、時刻表といったデータは自分でランダムに作成したものを使用しても良かったのであるが、今回このシステムを作成するにあたり、バス停や路線のデータ、時刻表のデータを松江市の方から期限付きで提供をして頂くことができたため、松江市の実際のデータを格納している。ただし、時刻表の変更などで現在の時刻表とは合っていない部分も存在すると思われる。

また、このシステムでは、一畑バスと松江市営バスの 2 つのバスの時刻表に対応している。

### 4.4.1 バス停位置テーブル

各バス停の情報を格納した表である。バス停を検索する場合、この表を利用することになる。

このバス停位置テーブルには、バス停 ID、バス停名、バス停の経度・緯度のデータを格納している。バス停 ID は各バス停毎にユニークに付けた ID を格納している。このユニークな ID は提供して頂いたデータに付けてあった ID とほぼ同じものを付けている。

次にバス停名は、提供して頂いたデータをそのまま使用させて頂いている。

そして、バス停の経度・緯度に関しては、提供して頂いたデータは、日本測地系で計測したデータを 1/1000 秒単位の 10 進整数で記述してあった(例えば経度が 133 度 12 分 34.567 秒であったとすると度と分を秒に直し、1000 倍したもので、つまり  $(133 \times 3600 + 12 \times 60 + 34.567) \times 1000$  を計算した値)。そこで、地図からの検索をする上で都合がいいように(経度・緯度を直接格納した場合、経度と緯度の 1 度あたりの長さが異なっているため、円形で検索をすると誤差が出てしまう。また、地図上を移動する場合に計算が複雑になる。) 地図からの経度・緯度の情報を元に変換を行ってから格納している。具体的には、以下のような計算を行った。

#### ・経度

地図画像の左端の座標 : 478689130 (132 ° 58' 09.13")

地図画像の右端の座標 : 479370210 (133 ° 09' 30.21")

地図画像の幅 : 15040 ピクセル

したがって、

$$(479370210 - 478689130) / 15040 = 45.2845\dots$$

つまり経度は 1 ピクセルあたり 0.0452845...秒である。

・緯度

地図画像の上端の緯度：127960080 (35 ° 32' 40.08")

地図画像の下端の緯度：127425180 (35 ° 23' 45.18")

地図画像の高さ：14480 ピクセル

したがって

$$(127960080 - 127425180) / 14480 = 36.9406\dots$$

つまり緯度は1ピクセルあたり0.0369406...秒である。

よって、これらの値(経度が45.2845...、緯度が36.9406...)で割ることで、地図画像の検索の際にJavaScriptで取得した座標を変換することなく加減算することができるようになる。よって、バス停の位置座標は、経度と緯度をそれぞれの値で割った値を整数に直して格納している。また、GPSを用いて経度と緯度を取得した場合もこれらの値で割って使用している。また、これらの値で割ることで、経度・緯度をピクセル単位(m)に変形したことで、円形の検索もできるようになった。

また、この経度・緯度はHiRDB Spatial Search Plug-inを使用して検索ができるようにジオメトリ型(抽象データ型)で定義し、格納している。このプラグインを使用することで、比較的簡単なSQL文を発行することで円形などの検索が可能となる。

実際に格納した表は以下のようなものである。下表は松江市営バスの場合の表の一部である。一畑バスの表も同じく格納している。

バス停ID (VARCHAR型)	バス停名 (VARCHAR型)	座標 (抽象データ(ジオメトリ)型)
10012	上乃木	POINT(10578466 3454132)
10023	相生町	POINT(10578172 3455164)
10041	朝日町	POINT(10578136 3455835)
10053	上谷南	POINT(10580079 3454326)
10093	石橋町	POINT(10578074 3457482)
10103	石橋三丁目	POINT(10577854 3457336)
10112	石橋四丁目	POINT(10578023 3457420)
⋮	⋮	⋮
10383	県庁前	POINT(10577469 3456544)
⋮	⋮	⋮
10731	島根大学前	POINT(10578706 3457796)
⋮	⋮	⋮
11220	松江駅	POINT(10578367 3455811)
⋮	⋮	⋮
14241	フォーゲルパーク	POINT(10568827 3456901)

表 2. バス停位置テーブル

表 2.のようにユニークに付けたバス停 ID、バス停名は VARCHAR 型、座標を抽象データ型で定義し、格納を行った。実際に格納したバス停数は、松江市営バスの表で 475 カ所、一畑バスの表で 622 カ所であった。これらを DBMS などを使用せずに管理するのは、不可能と言ってよいと思われる。

#### 4.4.2 バス路線テーブル

バス路線テーブルは、時刻表を検索する場合に必要となる、バスの路線の情報を格納した表である。出発バス停と目的バス停を使った検索(両方のバス停を通る路線の検索)または出発バス停を通る路線の検索を行うためのデータを格納してある。

具体的にはユニークに付けた路線 ID、路線名、バス会社、通る順のバス停 ID の列で構成されており、バス停 ID の列を検索することで、路線の検索を行っている。路線の検索において、バス停 ID の列はスペースで区切った文字列として、格納している。SQL 文発行の際には、SQL 文の条件文で、“%出発バス停 ID%目的バス停 ID%”とすることで、検索が簡単になる。ここで、%は 1 文字以上の文字列を表している。このデータに関しては、提供頂いたデータを元に作成を行った。

実際に格納した表は以下のようなものである。松江市営バスの表の一部である。通過するバス停 ID のバス停数は各路線によって、異なっている。また、一畑バスも同じように格納している。

路線ID (VARCHAR型)	路線名 (VARCHAR型)	通過するバス停ID (VARCHAR型)
10020101	県合同庁舎 - 川津 豎町 ~ 大橋	11612 11463 12552 10944 ... 10394
10060101	県合同庁舎 - 川津 駅 ~ 大輪町	11612 11463 12552 10944 ... 10334
10080101	県合同庁舎 - 川津 桧山 ~ 駅 ~ くにびき	11612 11463 12552 10944 ... 10274
10120101	平成車庫 - 川津 桧山 ~ 駅 ~ くにびき	12321 12331 12341 12351 ... 10731
10130102	川津 - 平成車庫 くにびき ~ 駅 ~ 桧山	10273 10732 10262 11163 ... 12342
10140101	平成車庫 - 川津 八重垣・作橋・駅・大	12321 12331 12341 12351 ... 12574
10150102	川津 - 平成車庫 大橋・駅・作橋・八重	10273 10732 10262 10093 ... 10861
10180101	平成車庫 - 松江駅 八重垣・桧山	12321 12331 12341 12351 ... 10174
10190102	松江駅 - 平成車庫 桧山・八重垣	11224 10173 10493 11181 ... 12342
10200101	竹矢 - 平成車庫 桧山 ~ 八重垣	10820 10722 10882 11292 ... 11182
10210101	平成車庫 - 川津 大庭・作橋・駅・大橋	12321 12331 12341 12351 ... 12444
10220202	川津 - 平成車庫 大橋・駅・作橋・大庭	10273 10732 10262 10093 ... 10861
⋮	⋮	⋮
17140101	南循環(外回り) 駅発・女子短大止め	11221 11091 10873 11573 ... 12291

表 3. バス路線テーブル

表 3.のようにユニークに付けた路線 ID、路線名、路線情報の列は全て VARCHAR 型で定義し、格納を行った。格納したバス路線の数は、松江市営バスが 102 路線、一畑バスが 159 路線であった。格納した行数はバス停位置テーブルより少ないが、こちらの表の方が時刻表の改訂で変更される可能性が高く、大幅な変更があった場合など、DBMS を使用せずに更新することは不可能である。しかし、DBMS を用いると、表の一部であっても、全体であっても高速に更新することが可能である。

#### 4.4.3 時刻表テーブル

時刻表テーブルは、各バス停の時刻表を格納している表である。バス停の ID を検索し、その ID より、路線 ID を検索した後で、時刻表を検索する。

具体的には、路線 ID、時刻、バス停 ID、備考の列で構成されている。路線 ID はバス路線テーブルの路線 ID と同じもの(その時刻の路線名の ID)が格納されており、バス停 ID についてもバス停位置テーブルのバス停 ID(その時刻のバス停の ID)と同じものが格納されている。

出発バス停だけの場合は、バス停 ID を使用して、出発バス停と目的バス停を使った検索の場合は、バス停 ID の検索結果とバス路線 ID の検索結果を元にしてこの時刻表を検索をする。この表の検索だけは他の検索と違い、JDBC ドライバを使用せずに、埋め込み SQL を用いた C プログラムを Java サーブレットから呼び出して検索を行っている。JDBC ドライバを使用した場合に検索結果が遅くなってしまうのは、DBMS の検索が遅いわけではなく(DBMS 内の検索自体は高速である)、ODBC をマッピングしたデータの受け渡し方法がボトルネックとなっているためである。

この時刻表のデータに関しては、提供していただいたデータの状態では、このシステムに関しては不都合であったので、このシステムに合うように変換している。具体的には、提供していただいたデータは、バス路線のバス停の順番に並んで時刻が記述してある状態のデータであったが、今回のシステムでは、バス停別の時刻表としたかったので、上記のような形の表に変換をしてから格納してある。

バス停ID (VARCHAR型)	時刻 (TIME型)	バス路線ID (VARCHAR型)	備考 (VARCHAR型)
12321	6:15:00	10120101	平日
12331	6:16:00	10120101	平日
12341	6:16:00	10120101	平日
12351	6:16:00	10120101	平日
12454	6:17:00	10120101	平日
12464	6:18:00	10120101	平日
11612	6:25:00	11550101	1、3、5土曜+休
12321	6:25:00	10120101	平日
11222	6:26:00	10120101	平日
11463	6:26:00	11550101	平日
11463	6:26:00	11550101	1、3、5土曜+休
⋮	⋮	⋮	⋮
12352	22:00:00	10220202	平日
12352	22:00:00	10220202	1、3、5土曜+休
12342	22:01:00	10220202	平日
12342	22:01:00	10220202	1、3、5土曜+休
12332	22:02:00	10220202	平日
12332	22:02:00	10220202	1、3、5土曜+休

表 4. 時刻表テーブル

表 4.のように、バス停 ID、バス路線 ID、備考は VARCHAR 型で、時刻は TIME 型で定義し、格納した。格納した時刻表の数は、松江市営バスが 25095 個、一畑バスは 27544 個であった。これだけの量のデータの管理は不可能である。

また、本システムで時刻表を検索するときのプロセスは以下のように行われている。

・バス停名からの検索の場合

まず、クライアント側から入力されたバス停名(出発バス停名と入力してあれば目的バス停名も)と一致(完全一致または部分一致)するバス停の”バス停 ID”と”バス停名”をバス停位置テーブルから検索する。次に、その”バス停 ID”を使用して、バス路線テーブルから、”出発バス停 ID”(もし入力してあれば、”出発バス停 ID”と”目的バス停 ID”の両方)を”路線情報”に含む”バス路線 ID”と”路線名”を検索を行う。そして、最後に検索された出発”バス停 ID”と”バス路線 ID”が両方とも一致するものを時刻表テーブルから検索を行っている。

(例)出発バス停名に”大学”、目的バス停名に”水道局前”と入力  
 以下のような SQL 文が発行されバス停を検索する。

```
SELECT * FROM バス停位置 13 WHERE バス停名 LIKE "%大学%";
```

バス停ID	バス停名	座標
10012	上乃木	POINT(10578466 3454132)
10023	相生町	POINT(10578172 3455164)
...	...	...
110731	島根大学前	POINT(10578706 3457796)
110732	島根大学前	POINT(10578811 3457803)
...	...	...
14241	フォーゲルパーク	POINT(10568827 3456901)

“大学”を含むバス停のバス停 ID とバス停名を取得。

バス停ID	バス停名	座標
10012	上乃木	POINT(10578466 3454132)
10023	相生町	POINT(10578172 3455164)
...	...	...
10673	水道局前	POINT(10578408 3457030)
10674	水道局前	POINT(10578394 3457042)
...	...	...
14241	フォーゲルパーク	POINT(10568827 3456901)

“水道局前”を含むバス停のバス停 ID とバス停名を取得。

路線ID	路線名	通過するバス停ID
10020101	県合同庁舎 - 川津 豎町 ~ 大橋	11612 11463 ... 12552 ... 10944 ... 10394
10060101	県合同庁舎 - 川津 駅 ~ 大輪町	11612 11463 ... 12552 ... 10944 ... 10334
...	...	...
10080101	県合同庁舎 - 川津 松山 ~ 駅 ~ くにびき	11612 11463 ... 10731 ... 10674 ... 10274
...	...	...
10120101	平成車庫 - 川津 松山 ~ 駅 ~ くにびき	12321 12331 ... 10731 ... 10674 ... 10274
...	...	...
11270101	松江駅 - あじさい団地 くにびき ~ 大学	11222 10374 ... 10731 ... 10674 ... 11841
...	...	...
171 0101	南循環(外回り) 駅発・女子短大止め	11221 11091 ... 10873 ... 11573 ... 12291

両方とも通るバス路線の路線 ID と路線名を取得。

バス停ID	時刻	バス路線ID	備考
10731	6:34:00	10120101	平日
10731	6:44:00	10120101	平日
10731	7:24:00	10120101	平日
10731	16:49:00	10120101	平日
10731	16:54:00	10120101	平日
10731	16:54:00	10120101	1、3、4土曜 + 休
10731	17:19:00	10120101	平日
10731	17:19:00	10120101	1、3、6土曜 + 休

バス停 ID と路線 ID より時刻表の検索

・GPS (空間)検索の場合

GPS(空間)検索の場合は、まず、得られた座標を用いて、バス停位置テーブル内で空間検索を行い、半径 100m または、200m 以内にあるバス停の”バス停 ID”と”バス停名”を取得する。そして、後はバス停名からの検索と同じ方法で、もし入力されていれば目的バス停の”バス停 ID”と”バス停名”を検索する。そして”バス路線 ID”、”路線名”を取得し、その結果から時刻表の検索を行っている。

(例)東経 133 度 04 分 12.85 秒、北緯 35 度 28 分 52.82 秒の地点で計測

以下のような SQL 文が発行されバス停を検索する。

```
SELECT * FROM バス停位置13 WHERE WITHIN(座標,RegionFromText('CIRCLE
(10578720 3457788, 85)')) IS TRUE;
```

バス停ID	バス停名	座標
10731	島根大学前	POINT(10578706 3457796)
10732	島根大学前	POINT(10578811 3457803)

上記の経度・緯度の地点から半径 100m 以内のバス停を検索

検索されたバス停 ID を使用して、バス停名からの検索と同様にして時刻表の検索を行っている。

検索のプロセスや格納したデータを見て分かるように、松江市のバス停の情報だけでも、データ量はかなり多い。従って、人間の手で管理していくのは、もはや不可能な量である。また、一部の変更であったとしても大変な作業量となってしまう。しかし、これだけの量のデータであっても、HiRDB を用いて、検索を行うと数秒程度(条件によっては 1 秒未満)での検索が可能である。本システムでの時刻表の検索時間が、かかっているのは、検索の条件によっては、SQL 文を複数回発行しなくてはならないため、その都度起こる C プログラムの起動やデータの受け渡しに時間がかかってしまうためである。DBMS を使用することで、データの管理は勿論のこと、そのデータベースを別のアプリケーションで使用してのシステムを運用することも可能であり、DBMS を使用するメリットは大きいと言える。

## 第 5 章 検証実験及び考察

### 5.1 検証実験

#### 5.1.1 検証方法

検証方法は、GPS 携帯端末(A5501T)を用いて、島根大学前、県庁前、松江駅前の 3 カ所で 10 回ずつの計測を行い、その時の誤差を測定する。

なお、3 カ所の基準座標として、電子地図帳 Z6 を使用した。また、GPS 携帯端末との比較のため、コンパクトフラッシュ型 GPS レシーバ(CFGPS2)を使用して、同じ地点で測定を行った。

#### ・実験機器

A5501T (au の GPS 携帯端末、TOSHIBA)

CFGPS2(コンパクトフラッシュ型 GPS レシーバ、I-O DATA)

VAIO C1(ノート型パソコン、SONY)

電子地図帳 Z6(地図ソフト、ゼンリン)

計測は、まず本システム(au.html)にアクセスし、その Web ページ中の”現在位置取得”というリンクにアクセスすることで、GPS 測位を行い、その時に得られた経度と緯度の情報である。また、CFGPS2 は電子地図帳 Z6 で使用することができ、CFGPS2 で経度・緯度を取得し、現在位置を表示することが可能である。また、IO-DATA からユーティリティソフトが配布されており、CFGPS2 を使用して、経度と緯度のみを表示する事ができる。そこで、電子地図帳 Z6 を使用した場合とユーティリティソフトをした場合の経度・緯度を取得したものを比較のために記載した。

### 5.1.2 実験結果

・検証実験の結果は以下の通りである。

	経度	緯度	誤差			天候
			経度(秒)	緯度(秒)	距離(m)	
<b>馬根大学前</b>	<b>133° 04' 15.16"</b>	<b>35° 28' 53.04"</b>				
電子地図帳Z	133° 04' 15.29"	35° 28' 53.05"	0.13	0.01	3.29	晴れ
CFGPS2	133° 04' 15.30"	35° 28' 53.04"	0.14	0.00	3.53	晴れ
GPS 携帯電話	1回目	133° 04' 15.80"	0.64	-0.08	16.32	雪
	2回目	133° 04' 15.41"	0.25	0.98	30.85	雪
	3回目	133° 04' 14.40"	-0.76	0.08	19.32	曇り
	4回目	133° 04' 14.98"	-0.18	-0.37	12.27	曇り
	5回目	133° 04' 15.23"	0.07	0.29	9.11	晴れ
	6回目	133° 04' 14.66"	-0.50	0.96	32.16	晴れ
	7回目	133° 04' 14.70"	-0.46	0.27	14.27	晴れ
	8回目	133° 04' 15.62"	0.46	0.29	14.64	曇り
	9回目	133° 04' 15.38"	0.22	0.14	7.03	曇り
	10回目	133° 04' 14.81"	-0.35	0.32	13.23	曇り
平均	133° 04' 15.10"	35° 28' 53.33"	-0.06	0.29	16.92	
分散			0.19	0.16	63.87	
<b>栗庁前</b>	<b>133° 03' 15.92"</b>	<b>35° 28' 08.81"</b>				
電子地図帳Z	133° 03' 15.99"	35° 28' 08.33"	0.07	-0.48	14.90	晴れ
CFGPS2	133° 03' 15.88"	35° 28' 08.79"	-0.04	-0.02	1.18	晴れ
GPS 携帯電話	1回目	133° 03' 15.60"	-0.32	-0.42	15.25	雪
	2回目	133° 03' 16.01"	0.09	-0.71	22.00	雪
	3回目	133° 03' 15.98"	0.06	-1.11	34.24	曇り
	4回目	133° 03' 15.67"	-0.25	0.67	21.59	曇り
	5回目	133° 03' 16.05"	0.13	0.12	4.94	晴れ
	6回目	133° 03' 15.76"	-0.16	0.11	5.27	晴れ
	7回目	133° 03' 16.37"	0.45	0.24	13.54	晴れ
	8回目	133° 03' 15.99"	0.07	-0.29	9.11	曇り
	9回目	133° 03' 16.51"	0.59	-0.50	21.42	曇り
	10回目	133° 03' 15.54"	-0.38	0.24	12.10	曇り
平均	133° 03' 15.95"	35° 28' 08.65"	0.03	-0.17	15.95	
分散			0.09	0.26	73.60	
<b>松江駅前</b>	<b>133° 03' 52.91"</b>	<b>35° 27' 40.89"</b>				
電子地図帳Z	133° 03' 54.20"	35° 27' 41.03"	1.29	0.14	32.80	晴れ
CFGPS2	133° 03' 53.54"	35° 27' 40.21"	0.63	-0.68	26.29	晴れ
GPS 携帯電話	1回目	133° 03' 53.13"	0.22	0.04	5.68	雪
	2回目	133° 03' 51.68"	-1.23	0.10	31.16	雪
	3回目	133° 03' 52.42"	-0.49	-1.08	35.50	曇り
	4回目	133° 03' 52.00"	-0.91	-0.70	31.49	曇り
	5回目	133° 03' 53.71"	0.80	0.20	21.09	晴れ
	6回目	133° 03' 52.98"	0.07	-0.38	11.84	晴れ
	7回目	133° 03' 53.53"	0.62	0.14	16.21	晴れ
	8回目	133° 03' 52.83"	-0.08	1.32	40.73	曇り
	9回目	133° 03' 52.85"	-0.06	0.80	24.70	曇り
	10回目	133° 03' 52.86"	-0.05	0.96	29.61	曇り
平均	133° 03' 52.80"	35° 27' 41.03"	-0.11	0.14	24.80	
分散			0.35	0.50	109.76	

表 5. 検証実験結果

## 5.2 考察

検証実験の結果から、au の GPS 携帯端末(A5501T)を使用して、経度 / 緯度を計測した場合、概ね十数 m の誤差で検索することができた。これは本システムで使用する場合には、許容できるくらいの誤差であると考えられると言える。また、松江駅の計測した時の誤差が大きいのはおそらく、計測した場所のそばに建物があったため、誤差が大きくなったものだと思う。実験の結果からは分かりづらいが天候によっても誤差が生じた。検証実験を行った日とは別の日の雪がひどい場合に計測した時に測定できなかつたり、離れた場所になってしまう場合もあった。

当然ながら、GPS を使用する場合、その時の天候といった気象条件、測定する場所の周りの建物などの障害物によって誤差が生じてくるため、同じ場所であっても測定したデータが異なってしまう。しかし、実際に GPS 携帯端末を使用して、松江市内で本システムを使い地図を表示させると、ほぼ正しい計測ができていますので、検索結果も正しく表示することができた。携帯端末での文字入力はやりにくいので、現在位置の近くのバス停を検索する場合は、文字入力をする必要が無いので、計測に少し時間がかかるがキー入力は少なくすむため、検索が楽である。

## 第6章 終論

GPS 携帯端末と本システムを用いることで、簡単に現在位置の近くのバス停とそのバス停の時刻表を検索することが可能になった。また、パソコンを使用して、バス停の時刻表を検索する場合にも今までのような文字だけの検索だけでなく、地図画像をクリックするだけでバス停と時刻表の検索を行うことができ、直感的で分かりやすい検索が可能となった。ただし、GPS を使用するため、屋外でしか使用できないという問題がある。

また、検索に DBMS を用いることで、高速かつ確実な検索を行うことができるようになるため、管理者側にもクライアント側にも便利なシステムとなった。また、管理者側では、Web サーバが稼働中であってもデータベースの更新も可能となり、さらに、表へのアクセス権限を変更することで、各バス会社のデータベースを各バス会社によって変更することも可能となる。また、セキュリティの面でも優れており、Java サブレットを使用することで、ユーザ側からはデータベースの操作が見えないばかりでなく、データベースを変更することも不可能であるので、さらにセキュリティが高いシステムとなった。

また、クライアント側は検索された結果のみが返ってくることになるため、クライアント側の端末には、本システム専用のシステムやプログラムといったものが一切必要ないので、一般的なブラウザや携帯端末から検索することが可能である。従って、クライアント側は、一般的な機器さえあれば良いということであり、誰もが使用できるシステムである。

本システムは Java サブレットとデータベースを使用して、動的に検索を行うシステムであるので、松江市だけでなく、全国どこであってもシステムの構築が可能である。バス停の位置情報、バス路線の情報、バス停の時刻表などのデータがあれば(どのバス会社にもあると思われる)構築が可能である。また、地図画像を作成し、プログラムの値を変更することで各地図にも対応することができる。もちろん、バス停だけでなく、駅や主な建物などやそれらを組み合わせたものにも対応することができるため、様々なものの検索の用途に応用することが可能である。

今後の課題としては、時間的な問題で、DoCoMo の GPS 携帯端末に対応することができなかったため、対応させる必要がある。また、GPS 携帯端末は au 以外の携帯端末会社からはほとんど発売されていない(2004年2月13日現在)ので普及率がそれほど高くないため、誰もが使用できる環境には無いと言う問題もまだある。また、課金のためのログ採取機能を追加し、トランザクション管理の徹底、表のアクセス権限の細かな設定とクライアントからの表内容管理更新ツール作成といった課題がある。スキーマ形態別の性能評価も必要であろう。検索機能をさらに洗練し、より使いやすいシステムにする必要もある。

## 謝辞

本研究にあたり、最後まで熱心な御指導をいただきました田中章司郎教授には心より御礼申し上げます。また、田中研究室の高木さん、長田さん、喜代吉さん、鷲見さん、樋上さん、本村さん、和田さんには本研究に関しまして、数々の御協力と御助言を頂きました。厚く御礼申し上げます。本研究におきまして、たびねっと松江の時刻表・バス停データを提供して頂きました松浦正敬松江市長、松江市役所企画財政部、また、地図データの使用許可を頂きました株式会社ゼンリンに、心より感謝申し上げます。

なお、本論文、本研究で作成したプログラム及びデータ、並びに関連する発表資料等の全ての知的財産権を本研究の指導教官である田中章司郎教授に譲渡致します。

## 参考文献・資料・サイトなど

### DBMS・Java(サーブレット・画像)・埋め込み SQL 関連

- [1]マニュアル 『HiRDB システム導入・設計ガイド』  
HITACHI
  
- [2]マニュアル 『HiRDB Spatial Search Plug-in Version3』  
HITACHI
  
- [3]マニュアル 『HiRDB UAP 開発ガイド』  
HITACHI
  
- [4]卒業論文 2001 年度 島根大学総合理工学部数理・情報システム学科 情報分野  
「対話型意思決定システムの試作 - 国道と人口メッシュを例として - 」  
荒木俊太郎
  
- [5]書籍 『Java、PostgreSQL、VirtualTeam で構成するマルチメディアサーバの作成』  
一條博 著、テクノプレス、2002 年
  
- [6]書籍 『完全オープンソースによる(+Java2 SDK)基礎からの Web サービス』  
藤田泰徳 著、セレンディップ/小学館、2003 年
  
- [7]書籍 『ここからはじめる！データベース基礎の基礎 SQL Server 2000 データベース編』  
松崎為裕 著、ソフトバンクパブリッシング、2002 年
  
- [8]書籍 『JAVA2 プログラミング講座』  
アスキー書籍編集部 著、ASCII、2002 年
  
- [9]サイト 『Java-Java3d-Swing-Game-Servlet-J2ME-Java2d』 (Java サーブレット全般)  
<http://www.javadrive.jp/index.html>
  
- [10]サイト 『JAVA 入門』 (Java を使用した画像関係)  
<http://black.sakura.ne.jp/~third/system/java/javax.html>

## JavaScript 関連

- [11]書籍 『詳解 JavaScript 辞典』  
半場方人 著、秀和システム、2002 年
- [12]書籍 『ホームページ作りにそのまま使える JavaScript 例文活用辞典』  
古籾一浩 著、技術評論社、2001 年
- [13]サイト 『DynamicHTML/SimpleSamples』 (JavaScript のサンプル)  
<http://www.fureai.or.jp/~tato/DHTML/simple/content2.htm>

## 携帯サイト・GPS 関連

- [14]書籍 『ケータイサイト構築完全ガイド』  
インフォシエル 著、MYCOM、2002 年
- [15]サイト 『i モード』 (DoCoMo の公式ページ)  
<http://www.nttdocomo.co.jp/mc-user/i/index.html>
- [16]サイト 『技術情報』 (au の公式ページ)  
<http://www.au.kddi.com/ezfactory/tec/index.html>
- [17]サイト 『ボータフォン』 (vodafone の公式ページ)  
[http://www.dp.j-phone.com/dp/tech\\_svc/web/](http://www.dp.j-phone.com/dp/tech_svc/web/)
- [18]サイト 『DoCoMo Net - i モード-』 (DoCoMo の GPS 関連)  
[http://www.nttdocomo.co.jp/p\\_s/imode/](http://www.nttdocomo.co.jp/p_s/imode/)
- [19]サイト 『簡易位置情報』 (au の公式ページ)  
<http://www.au.kddi.com/ezfactory/tec/spec/eznavi.html>
- [20]サイト 『GSI HOME PAGE』 (国土地理院のページ)  
<http://www.gsi.go.jp/>

[21]サイト 『GPS 携帯から位置情報を取得する ( CGI 編 )』 ( au の GPS 関連)

[http://www.surveytec.com/groom/landmark/au\\_cgi.html](http://www.surveytec.com/groom/landmark/au_cgi.html)

[22]サイト 『au の GPS ケータイで位置情報を表示させよう』 ( au の GPS 関連)

<http://orienteing.hp.infoseek.co.jp/gps/gpsone.html>

[23]サイト 『GPS 携帯を使う』 ( au の GPS 関連)

<http://siisise.net/gps.html>

[24]サイト 『au の GPS 携帯端末からの位置情報を CGI で受け取る』 ( au の GPS 関連)

<http://www.wireless-net.jp/gpskeitai/eznavi.html>

[25]サイト 『位置取得』 ( au の GPS 関連)

<http://kokogiko.net/lpw/genre002/>

## その他

[26]書籍 『2003-'04 年版 最新パソコン用語辞典』

岡本茂 / 大島邦夫 / 堀本勝久 著、技術評論社、2003 年

[27]サイト 『google』 ( 検索)

<http://www.google.co.jp/>

[28]サイト 『IT 用語辞典』

<http://www.itmedia.co.jp/dict/>