

空間データベースを用いた 動的近隣情報更新検索システムの試作

島根大学 総合理工学部 数理・情報システム学科

計算機科学講座 田中研究室

S053059 中山 裕太

目次

第1章 序論

- 1. 1 研究目的
- 1. 2 研究概要

第2章 システムを実現する技術・考案

- 2. 1 DBMS
 - 2. 1. 1 ORDB
 - 2. 1. 2 空間データ操作
 - 2. 1. 3 Buffer メソッド、Within メソッド
- 2. 2 Web システム
 - 2. 2. 1 Web サーバ
 - 2. 2. 2 Java サーブレット
 - 2. 2. 3 JDBC
- 2. 3 アンケートの実施

第3章 店舗情報入力・更新システム

- 3. 1 研究概要
- 3. 2 システムの流れ
- 3. 3 位置情報取得方法
- 3. 4 格納データ
 - 3. 4. 1 バス停位置テーブル
 - 3. 4. 2 バス路線テーブル
 - 3. 4. 3 バス編成テーブル
 - 3. 4. 4 ユーザデータテーブル
- 3. 5 店舗情報入力から更新まで

第4章 近隣情報検索システム

- 4. 1 研究概要
- 4. 2 システムの流れ
- 4. 3 検索方法
 - 4. 3. 1 周辺の店舗情報検索
- 4. 4 結果表示
 - 4. 4. 1 使用方法

第5章 アンケート結果と改善点

第6章 考察と今後の課題

第7章 謝辞

参考文献・資料

付録 アンケート結果

第1章 序論

1.1 研究目的

現在、Google Maps[1]のような地図サーバを用いた Web サイトが増えてきている。こういったサイトは地図上にレストランなどの店舗情報を表示し、今までは住所しか分からなかったものを地図上に表示することでより店舗などの位置を探しやすくなったといえる。しかし、時刻表などの更新の少ない組織的に整備されたデータや今日のガソリン料金、レストランの日替わりメニューといった更新の多いデータ、これらを同時に検索するシステムは少ない。

このことから、これらのデータについて必要な部分だけを抽出し、地図上に同時に追加して描画するシステムがあれば有益であろう。

本研究では先行研究[2]のベースをもとに、地図上をクリックすることでその位置の動的な地域情報を検索することが可能なシステムと店舗所有者などが地図上をクリックすることでその位置に店舗情報を挿入することができ、また、その後も Web 上で店舗情報を更新することが可能なシステムの試作を行った。

1.2 研究概要

システムの実装にあたって、店舗情報の管理や検索には DBMS を用いた。本システムでは座標データといった文字数値以外のデータを容易に扱うことができ、地図を用いた空間検索も行うことが可能なことから、先例[2]同様に ORDBMS[3]を使用した。

また、Webサーバには高い安定性と軽快な動作に加え、高いシェアを持ち、拡張性が高いオープンソースであるApache[4]を使用した。クライアントとWebサーバ間をつなぐ言語として、高速な処理が可能で、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可能なJavaサーブレットを使用し、サーブレットやJSPを実行するためのサーブレットコンテナには安定性が高く、オープンソースであることからTomcat[4]を使用した。クライアントとWebサーバ間でJavaを用いたことから、WebサーバとDBサーバ間の接続にはJDBCドライバ[5]を使用することとした。

本研究では表示ツールとしてユーザ数の多さや多角形や線やマーカーの表示など機能の豊富さから、先行研究[2]と同様に Google Maps を用いることとした。

第2章 システムを実現する技術・考案

2. 1 DBMS

DBMS (データベース管理システム)とは、コンピュータのデータベースを構築するために必要なデータベース運用、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。

2. 1. 1 ORDB

空間検索オブジェクトリレーショナルデータベース (ORDB) とは、従来のリレーショナルデータベース (RDB) [6]の延長線上に位置しており、RDBにオブジェクト指向の環境を提供するものである。また、RDBMSにおいて別々のDBに格納していた汎用データと空間データをORDBMSでは同一のDBに格納することが可能となる。

同一の DB 内で取り扱うことによりマルチメディア情報と文字数値データを属性データとして同一行内に関連付けることが出来る。更に、マルチメディア情報そのものを条件として検索することも可能となる。

本システムは、純国産 ORDBMS である日立 HiRDB Version 7[7]を用いてユーザデータ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。HiRDB には、GEOMETRY(抽象データ)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。

GEOMETRY 型とは HiRDB 空間検索プラグイン(HiRDB Spatial Search Plug-in Version 3)[8]によって定義されているユーザ定義型で、Point 型、Line String 型、Polygon 型といった空間データを格納することができる。バス停や店舗の位置情報をこの 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。

2. 1. 2 空間データ操作

ORDB では大まかに Point (点) 型、LineString (折線) 型、Polygon (多角形) 型が存在している。これらのデータを Buffer メソッドにより Point 型又は LineString 型を Polygon 型へ変換したり、Within メソッドにより検索範囲に図形が完全に含まれているかを判定することができる。

2. 1. 3 Buffer メソッド、Within メソッド

Buffer メソッドというのは Point 型に対しては指定した値を半径とする円を生成し、LineString 型のデータに対しては指定した値の幅を持つ Polygon 型を生成するという機能を持つ。

Within メソッドというのは検索する列にある Point、Line String、Polygon 型で格納されているデータと検索範囲の図形との空間関係をチェックして、前者が後者に「完全に含まれるかどうか」を判定する機能を持つ。

これらのメソッドを組み合わせることで Buffer メソッドにより、現在位置から 100m 以内の距離にある円を生成し、Within メソッドにより、その円の中に Point 型で格納されている飲食店の座標が含まれているどうか判定することで現在位置から 100m 以内にある飲食店といった検索を行うことが可能となる。

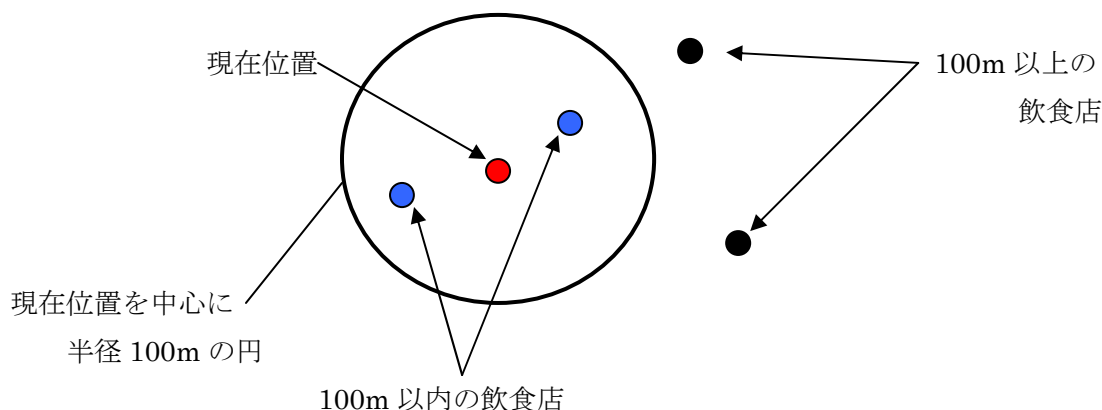


図 2.1.3.1 半径 100m 以内の飲食店検索

2. 2 Web システム

2. 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。

本システムでは Web サーバとして Apache2.2[4]を用い、Web サーバ・サーブレットコンテナとして Tomcat6.0[4]を用いる。Apache と Tomcat を連携させることにより、Java サーブレットや JDBC を用いた Web システムの構築をおこなっている。具体的には静的コン

テンツを Apache が処理し、動的コンテンツを Tomcat が処理するといった方法をとっている。Tomcat にも Web サーバの機能は存在するが、Apache に比べ処理が遅く、細かな設定が出来ないという理由から本システムでは使用せず、Apache を用いる。

2. 2. 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る。また、サブレットと似た技術に CGI がある。CGI に対するサブレットの利点を次に示す。

- 効率的

CGI の場合、HTTP リクエスト毎に新たなプロセスが生成される。それに対しサブレットは、Java 仮想マシンが唯一のプロセスとして常に動作しており、個々のリクエストはスレッドによって処理される。そのため、プロセス起動のためのオーバーヘッドが生じず、また、処理終了後もメモリ上に存在するため、複数のリクエストから簡単に利用できる。

- 開発しやすさ

サブレットは HTML フォームデータの構文解析、HTTP ヘッダの読み取りと設定、クッキーの処理、セッションの管理、といった多くの基礎的機能を豊富に備えている。また、信頼性も高く再利用性にも富む。

- 強力

複数サブレットでのデータベース接続の共有など、資源を共有する最適化を容易に実装することができる。また、サブレットは複数のリクエストにまたがって情報を保持するため、セッション管理や、前の処理結果のキャッシュなどの技術も簡単に実現できる。

- 可搬性

サブレットは Java の標準 API を使用するため、一般的にプラットフォーム非依存である。

2. 2. 3 JDBC

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ので、このドライバを使用して、Java サブレットから 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。

2. 3 アンケートの実施

本システムの試作を行い、実際に出来上がったシステムを一般ユーザにも使ってもらい、そのアンケート結果から改善点を見つけ出す。ユーザが求めていることを分析することでシステムの再構成を行い、より使いやすいシステムへと改良する。

本研究ではアンケートを一回実施しており、論文中の第3章、第4章のシステムはアンケート結果によって再構成を行ったシステムについてである。

アンケート結果の詳細は第5章にて記述する。

第3章 店舗情報入力・更新システム

3.1 システム概要

本システムは表示された Google Maps 上をクリックすることで、その位置に店舗所有者などが情報を入力することができ、また、ここで入力された情報を用いることで店舗情報を更新をすることができるシステムの試作を行う。Web サーバや DB サーバ、ブラウザを用いて行い、クライアント側からのデータ挿入と更新を可能とする。

ユーザー名:

パスワード:

職種:

緯度:

経度:

ユーザー名、パスワード、職種、店舗位置を入力し、送信をクリックしてください。

図 3.1.1 店舗情報入力フォーム

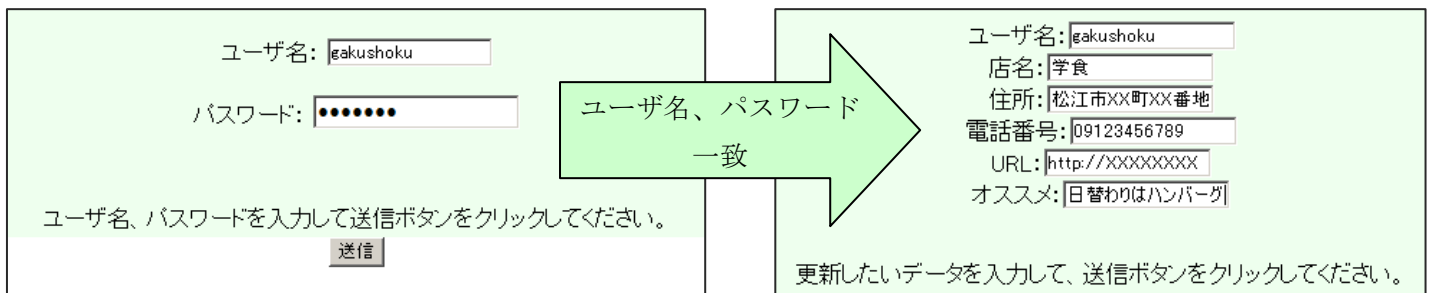


図 3.1.2 店舗情報更新フォーム

3. 2 システムの流れ

店舗情報入力フォームの使用方法は、地図上から店舗位置をクリックし、店舗情報更新時に用いるユーザ名とパスワードを個人で決めて入力、最後に飲食店やスーパーと行った店舗情報を選択する(①)。全項目を入力後、送信ボタンをクリックすることで店舗情報をWebサーバに送信する(②)。Webサーバは送信されてきたデータを元にJDBCを用いて挿入・更新に必要なSQL文を発行し、DBMSに接続し、店舗情報の挿入、もしくは更新を行う(③④⑤)。

店舗情報挿入時には、入力したユーザ名がすでにDB内で使用されていないかを検索し、使用されていない場合は今、入力された店舗情報をDBに挿入する(③④⑤)。もし使用されていた場合は(③④⑤⑥)、入力されたユーザ名がすでに登録されていることを表示する(⑦⑧)、再入力時にはユーザ名を変更して送信する必要がある(⑨⑩)。

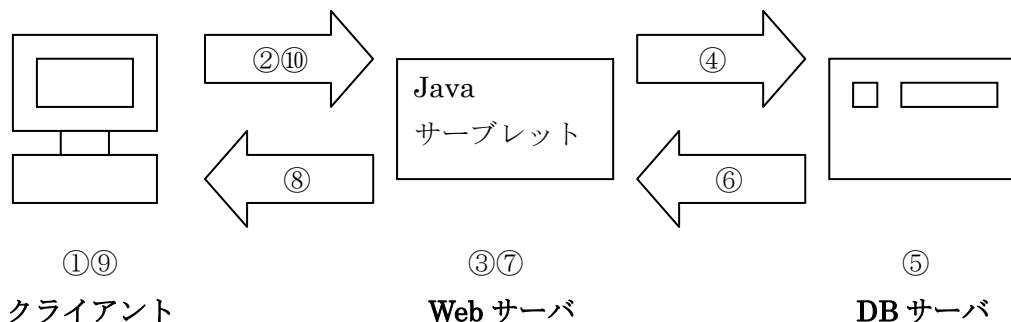


図 3.2.1 店舗情報入力の流れ

店舗情報更新時には、ユーザ名とパスワードを入力することで(①)、そのデータを送信し(②)、DB内に挿入されているユーザ名とパスワードが一致するデータを検索する(③④⑤)。一致するデータがあった場合、店舗情報更新フォームへと移動する(⑥⑦⑧⑨)。もし一致するデータがなかった場合は、ユーザ名、もしくはパスワードが一致しなかったことを表示する(⑥⑦⑧)。更新したいデータを入力後(⑨)、送信することで(⑩)、店舗情報を更新することができる(⑪⑫⑬)。

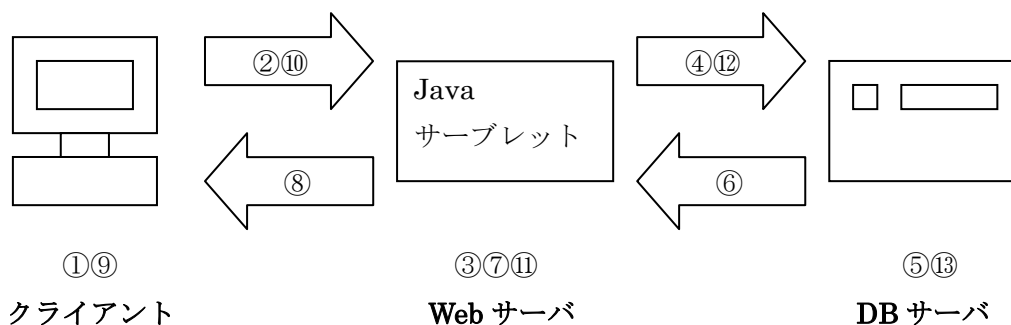


図 3.2.2 店舗情報更新の流れ

3. 3 位置情報取得方法

Google Maps API を用いてクリックした位置の情報を取得することで店舗の位置情報を取得することができる。以下に具体的な方法としてプログラムを記述する。

```
function onLoad(){
  //発場所の地図作成
  map = new GMap2(document.getElementById('map'));
  map.addControl(new GScaleControl());
  map.setCenter(center, 13);
  //クリックした位置を取得
  GEvent.addListener(map, 'click', onsMapClick);
}
//クリック処理
function onsMapClick(overlay, point){
  y=point.y; //y に緯度格納
  x=point.x; //x に経度格納
}
```

ここでは、`GEvent.addListener(map, 'click', onsMapClick);`によりクリックした位置の緯度経度をそれぞれ `point.y`、`point.x` に格納する。その緯度経度を `x` と `y` に格納することにより、緯度経度を取得することが出来る。この方法により、クリックした位置から緯度経度を取得している。

3. 4 格納データ

格納データに関して近傍バス停時刻検索システムに関するデータは先例[2]同様に松江市の実際のバス停や路線、時刻表のデータを格納しており、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が、一畑バスと松江市営バスの2つのバス停時刻表に対応している。

店舗情報検索システムに関するデータは試作段階ということで、実際に存在する店舗情報を使用しているのではなく、個人で考えられた実際にある店舗情報とは関係のない店舗情報を使用している。

3. 4. 1 バス停位置テーブル

バス停位置テーブルは、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)。そのため、Google Maps で扱えるようにするために世界測地系に変換する必要がある。

世界測地系への変換方法はまず、日本測地系で計測されたデータを 3600000 で割り、度で表す。次に、その座標の整数部分から 1 次メッシュコードを導く。さらに、その座標の整数部分をひいたものからそれぞれ緯度を 12 倍、経度を 8 倍し、その整数部分を 2 次メッシュコードとする。最後に、2 次メッシュコードを導いた座標から整数部分を引いたものからそれぞれ緯度を 120 倍し、経度を 80 倍し、その整数部分を 3 次メッシュコードとする。

導いた 3 次メッシュコードでは地図からの検索をする上で、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。この問題に対して、3 次メッシュがほぼ 1 km ということから、緯度経度をそれぞれ 1200 倍、800 倍することで、1 が 100m となり、この値を格納している。[2]

バス停 ID (VARCHAR 型)	バス停名 (VARCHAR 型)	X 座標 (VARCHAR 型)	Y 座標 (VARCHAR 型)	座標 (ユーザ定義(ジオメトリ)型)
10012	上乃木	35.4469	133.0644	point(42536.3443 106451.5507)
10024	相生町	35.4567	133.0605	point(42548.1166 106448.4793)
10033	相生町入口	35.4594	133.0607	point(42551.3966 106448.5927)
:	:	:	:	:
10045	朝日町	35.4644	133.0602	point(42557.3156 106448.2234)
10054	上谷南	35.4489	133.0846	point(42538.7177 106467.6855)
10062	三中前	35.4623	133.0687	point(42554.4490 106454.9923)
:	:	:	:	:
10082	朝酌公民館前	35.4646	133.1017	point(42557.5303 106481.3724)
:	:	:	:	:
10094	石橋町	35.4815	133.0594	point(42577.8928 106447.5393)

表 3.4.1.1 バス停位置テーブル

表 3.4.1.1 のようにユニークに付けたバス停 ID、バス停名は VARCHAR 型、表示に使用する座標である X 座標、Y 座標を VARCHAR 型、検索に使用する座標である座標をユーザ定義型で定義し、格納を行った。プライマリーキーはバス停 ID とする。空間 INDEX は秒単位での分割を行っている。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。

3. 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)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、通過するバス停 ID の列で構成されており、通過するバス停 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、通過するバス停 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、検索がすることが出来る。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 堅町～大橋	11612 11463 12552 10944 10952 10414 … 10274
10120101	平成車庫－川津 桧山～駅～くにびき	12321 12331 12341 12351 12454 12464 … 10274
10140101	平成車庫－川津 八重垣・作橋・駅・大	12321 12331 12341 12351 12454 12464 … 10274
10190102	松江駅－平成車庫 桧山・八重垣	11224 10173 10493 11181 13053 10983 … 12322
10400101	馬淵－県庁前松江駅 経由せず	11192 11302 11372 11292 10932 11202 … 10384
10510102	川津－竹矢 大橋～駅～作橋	10273 10732 10262 10093 10112 10103 … 10820
10700101	松江駅－女子高前 くにびき	11228 10374 10684 10691 11241 11251 … 10541
:	:	:
17120101	北循環(内回り)福祉センター経由せず	11221 10374 10684 11514 10674 11164 … 11220

表 3.4.2.1 バス路線テーブル

表 3.4.2.1 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。プライマリーキーは路線 ID とする。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。

3. 4. 3 バス編成テーブル

バス編成テーブルは、各バス停の時刻表を格納している表である。バス停位置テーブルよりバス停 ID を検索し、その ID より、路線テーブルからどの路線の何番目に通るバス停 ID かを検索した後で、時刻表を検索する。

具体的には、路線 ID、時刻、バス 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 ID については同じ路線でも何本もバスがあるので何本目のバスなのかを決定するために用いる。時刻については通過するバス停と同形式で時刻が格納されている。そのため、何番目に通過するバス停かがわかれば、時刻を特定することが出来る。このデータに関しては、提供していただいたデータの状態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。時刻の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)	備考 (VARCHAR 型)	バス ID (INTEGER 型)	時刻 1 (TIME 型)	...	時刻 18 (TIME 型)	...
10020101	平日	1	7:40:00	...	8:03:00	...
10120101	平日	2	6:25:00	...	6:41:00	...
10120101	休日	2	16:55:00	...	17:15:00	...
10510102	休日	3	18:50:00	...	19:14:00	...
:	:	:	:	:	:	:
10610102	平日	2	15:30:00	...	0:00:00	...
11390101	休日	22	15:45:00	...	16:06:00	...
11400102	平日	1	7:25:00	...	7:50:00	...
11850101	平日	6	16:00:00	...	0:00:00	...
17040101	平日	2	17:45:00	...	18:08:00	...
:	:	:	:	:	:	:
17080101	平日	3	19:00:00	...	19:18:00	...
17140101	1、3、5 土曜	2	20:00:00	...	20:19:00	...

表 3.4.3.1 バス編成テーブル

表 3.4.3.1 のように、路線 ID、備考は VARCHAR 型で、バス ID は INTEGER 型、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 1238 個、一畑バスは 818 個であった。

3. 4. 4 ユーザデータテーブル

ユーザデータテーブルは、店舗情報を格納した表である。店舗情報挿入や更新、検索する場合、この表を利用することになる。

このテーブルには、店舗情報更新時を用いるユーザ名、パスワードや検索時に用いる飲食店やスーパーといった店舗情報、経度・緯度、また検索結果に表示される店舗の店名、住所、電話番号、オススメ情報が格納されている。

店舗情報更新時には、入力されたユーザ名、パスワードから格納されているユーザ名、パスワードが一致するデータを検索する。一致した場合は、現在格納されているデータをユーザに表示し、更新することが可能となる。

店舗情報検索時には、格納されている店舗情報と座標を用いることで、検索した範囲内にある目的の店舗情報を検索することができる。

実際に格納した表は以下のようなものである。

店名 (VARCHAR 型)	住所 (VARCHAR 型)	電話番号 (VARCHAR 型)	オススメ (VARCHAR 型)
スーパー	松江市 OO 町 OO 番地	09011111111	アイスが全品20%割引
学食	松江市 XX 町 XX 番地	09123456789	日替わりはハンバーグ
:	:	:	:

ユーザ名 (VARCHAR 型)	パスワード (VARCHAR 型)	店舗情報 (INTEGER 型)	店舗データ	X 座標 (VARCHAR 型)	Y 座標 (VARCHAR 型)	座標 (ユーザ定義(ジオメトリ)型)
nakayama	s053059	1	:	35.4644	133.0634	point(42557.3340 106450.7904)
gakushoku	gakkou	4	:	35.4840	133.0701	point(42580.8199 106456.1462)
yuta	asdf	2	:	35.4373	133.0640	point(42524.7798 106451.2024)
:	:	:	:	:	:	:

表 3.4.4.1 ユーザデータテーブル

表 3.4.4.1 のようにプライマリーキーであるユーザ名を VARCHAR 型、パスワードは VARCHAR 型、店舗情報検索時に用いる店舗情報を INTEGER 型、結果表示に使用する店舗データの店名、住所、電話番号、オススメを VARCHAR 型、座標である X 座標、Y 座標を VARCHAR 型、検索に使用する座標である座標をユーザ定義型で定義し、格納を行った。

3. 5 店舗情報入力から更新まで

店舗情報入力時

- ① 入力されたユーザ名が DB 内にすでに格納されていないか検索。すでに格納されていた場合は、再入力画面へ移動。
- ② 入力されたユーザ情報を格納。

店舗情報更新時

- ③ 入力されたユーザ名、パスワードが一致するユーザデータを検索。一致するユーザデータがなかった場合は、再入力画面へ移動。
- ④ 入力された店舗情報に更新。

第4章 近隣情報検索システム

4.1 システム概要


本システムは第3章で格納されたデータを検索するシステムである。表示された Google Maps 上から出発場所と目的場所を選択し、検索したい店舗情報を選択することで、近隣の地域情報の検索を Web サーバや DB サーバ、ブラウザを用いて行い、検索結果を Google Maps 上に表示するというシステムの試作を行う。

近隣情報検索システム

①出発場所をクリックしてください
変更したい場合は「出発変更」を押してください
出発場所選択



②目的場所をクリックしてください
変更したい場合は「目的変更」を押してください
目的場所選択



③検索したい店舗情報、出発時刻を選択し、送信をクリックしてください

店舗情報: 飲食店 出発時刻: 16時 平日 休日 送信

店舗位置	店名	電話番号	住所	URL	オススメ
<input checked="" type="checkbox"/> 店舗位置表示	第二	09987654321	松江市〇〇町〇〇番地	http://〇〇〇〇〇〇〇〇	カレーライス20円引
<input checked="" type="checkbox"/> 店舗位置表示	学食	09123456789	松江市〇〇町〇〇番地	http://〇〇〇〇〇〇XX	日替わりはハンバーグ

図 4.1.1 近隣情報検索システム結果表示

4. 2 システムの流れ

近隣情報検索システムの使用方法は、地図上から出発場所と目的場所をクリックし、その後検索したい店舗情報、日時を選択する(①)。送信ボタンをクリックすることで、その情報を Web サーバに送信する(②)。Web サーバは送信されてきたデータを元に JDBC を用いて検索に必要な SQL 文を発行し、DBMS に接続する(③④)。まず目的場所周辺に求めている店舗情報がないかを検索する(⑤)。なかった場合はそこで全検索を終了する(⑥⑦⑧)。あった場合は、出発場所周辺のバス停検索と目的場所周辺のバス停の検索し、その検索結果のバス停から路線を検索する(⑤⑨)。取得した情報をクライアントへ送信する(⑩⑪)。クライアントはその情報から店舗情報と時刻表、地図上には出発のバス停と目的のバス停、店舗位置を表示する(⑫)。

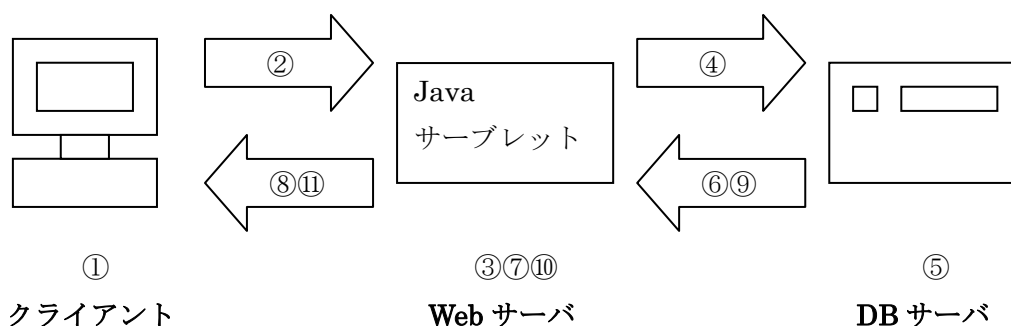


図 4.2.1 近隣情報検索の流れ

4. 3 検索方法

- ① 目的地の半径 500m 以内に目的の店舗情報があるか判定する。なければ検索終了。
- ② 半径 500m 以内の目的の店舗情報を取得。
- ③ 出発地と目的地で共に半径 500m 以内にバス停があるか判定する。なければ時刻表検索のみ終了。
- ④ 受け取った出発位置からまず、半径 100m 以内の周辺のバス停を検索する。なければ、半径 100m 毎で 500m まで行い、1 件も見つからなければ、時刻表検索終了。
- ⑤ 目的位置からも出発位置と同様にまず、半径 100m 以内の周辺のバス停を検索する。なければ、半径 200m 以内の周辺のバス停を検索する。これを 100m 毎に 500m まで行い、1 件も見つからなければ、時刻表検索終了。
- ⑥ 周辺のバス停が見つかった場合、そのバス停と出発のバス停との路線を検索する。路線があれば、時刻表を検索し、なければ、次の目的位置周辺のバス停を検索する。それでもない場合は④へ戻る。
- ⑦ 路線があった場合、この路線の何番目のバス停かを取得する。
- ⑧ 何番目のバス停かと日時を元に時刻を検索する。

4. 3. 1 周辺の店舗情報検索

周辺の店舗情報の検索には、取得した目的地情報と各店舗の位置情報を HiRDB の空間検索プラグイン(Spatial Search Plug-in)利用して DBMS で検索を行っている。まず、取得した位置情報を変換して、各店舗の位置情報を用いた検索を出来るようにする。ここでは例として半径 100m 以内の飲食店の検索を行う。

実際の HiRDB に発行する SQL 文は、以下のようになる。

```
SELECT 店舗データ FROM ユーザデータ WHERE 店舗情報 = "飲食店" AND  
ITHIN (座標 , RegionFromText('CIRCLE(経度 緯度, 半径 100m)')) IS TRUE;
```

"ユーザ情報"は列名。"ユーザデータ"は表名。"WHERE"以下が検索の条件、今回は飲食店を検索した場合なので、列"店舗情報"から飲食店の検索となる。"RegionFromText()"は HiRDB Spatial Search Plugin 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。

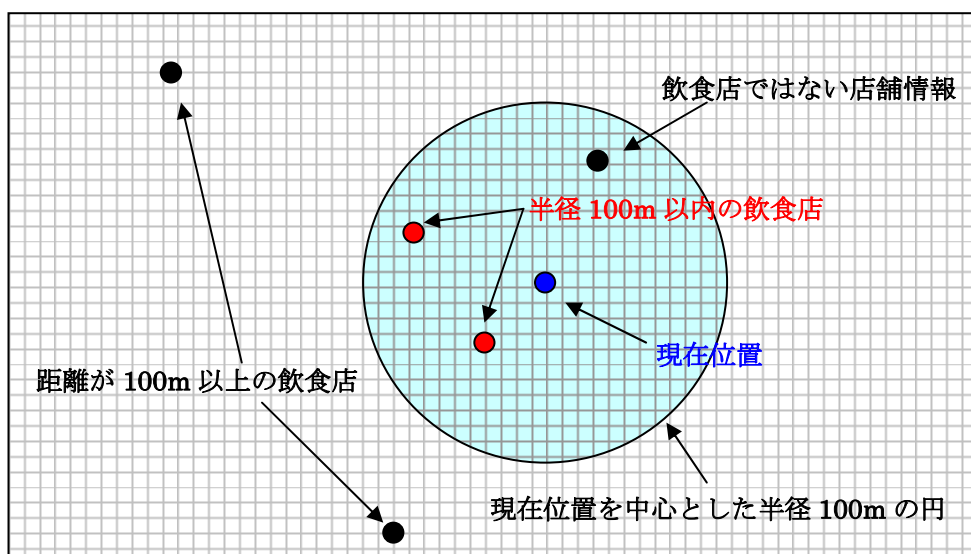


図 4.3.1.1 空間検索

4. 4 結果表示

PC では主に Google Maps API を用いて、表示を行っている。検索によって得られた位置情報から Google Maps 上にマーカーを設置する。そのマーカーにマウスのポインタが重なったとき、クリックしたときにイベントを発生させることで地図上から店舗情報を取得する処理を行う。

また、より実用的なシステム構築のため、先例[2]同様に時刻表の表示を非同期通信で行った。非同期通信とは、クライアントとサーバ間で同期を取らずに通信することである。

同期的な通信の場合、現在ブラウザに表示しているページから他のページへ移動するにはリロードを行う必要があるのに対し、非同期的な通信の場合は、他のページへ移動する際にも絶えずサーバとの通信を行うため、ページ遷移が発生しない。さらに、部分的に更新も行えるので軽い処理となり、高速化にもつながる。このように、非同期通信はシステム利用者にサーバの存在を意識させず、ブラウザの制御を奪うことなしに HTTP 通信を行うことができる。

実行結果

①出発場所、目的場所、出発時刻、店舗情報を選択し、送信をクリックする。

③検索したい店舗情報、出発時刻を選択し、送信をクリックしてください

店舗情報: 出発時刻: 平日 休日

②店舗検索結果が図のように表示される。店舗位置表示のチェックボックスを選択することで、地図上に店舗位置が表示される。

③検索したい店舗情報、出発時刻を選択し、送信をクリックしてください

店舗情報: 出発時刻: 平日 休日

店舗位置	店名	電話番号	住所	URL	オススメ
<input checked="" type="checkbox"/> 店舗位置表示	第二	09987654321	松江市00町00番地	http://00000000	カレーライス20円引
<input checked="" type="checkbox"/> 店舗位置表示	学食	09123456789	松江市00町0X番地	http://0000000X	日替わりまんばん

- ③バス停検索結果が図のように表示される。バス停位置表示のチェックボックスを選択することで、地図上に店舗位置が表示される。

③検索したい店舗情報、出発時刻を選択し、送信をクリックしてください

店舗情報: 出発時刻: 平日 休日

店舗位置	店名	電話番号	住所	URL	オススメ
<input checked="" type="checkbox"/> 店舗位置表示	第二	09987654321	松江市00町00番地	http://00000000	カレーライス20円引
<input checked="" type="checkbox"/> 店舗位置表示	学食	09123456789	松江市00町0X番地	http://000000XX	日替わりはハンバーグ

バス停位置表示

松江駅2番のりば→西川津

時間	経路
16:50 - 17:08	万原11美保関ターミナル

<所要時間>18分

- ④地図上の店舗位置にポインタを移動させることで店舗情報が表示される。

③検索したい店舗情報、出発時刻を選択し、送信をクリックしてください

店舗情報: 出発時刻: 平日 休日

店舗位置	店名	電話番号	住所	URL	オススメ
<input checked="" type="checkbox"/> 店舗位置表示	第二	09987654321	松江市00町00番地	http://00000000	カレーライス20円引
<input checked="" type="checkbox"/> 店舗位置表示	学食	09123456789	松江市00町0X番地	http://000000XX	日替わりはハンバーグ

バス停位置表示

松江駅2番のりば→西川津

時間	経路
16:50 - 17:08	万原11美保関ターミナル

第5章 アンケート結果と改善点

付録のアンケート結果よりシステムの再構成を行った。

問題点としては、全体的にシステムの説明が不足していたため、こういった目的で作られ、どのように使われるのかが分からない。詳しく説明するためにも、ヘルプといった項目もつけた方が分かりやすい。またバス停検索と店舗情報検索が別々の検索になっているため、両方の検索を行う場合には二回の検索をしなければならない。

検索結果の表示では検索項目の少なさから目的としている検索結果がでなかった。検索結果が一つしかでないため、結果が見にくく、検索し辛い、などのいくつかの問題点が挙げられた。以下に改善点について記述する。

改善点

- ・ システムの目的、使用方法についての説明
- ・ バス停検索と店舗情報検索の同時検索
- ・ 地図を用いた店舗情報表示
- ・ 結果項目の増加

第6章 考察と今後の課題

本研究では Google Maps といった表示ツールから ORDBMS を用いて店舗情報の格納を行い、その検索結果を表示する機能と Buffer メソッドと Within メソッドを用いて地図上から選択した位置を中心にした円検索を行い近隣情報を検索する機能、これらを組み合わせたシステムの試作を行いました。

また、アンケートを行うことで、ユーザが求めていることを分析し、より使いやすいシステムへと再構成を行いました。(付録参照)

今後の課題として、先例である近傍バス停時刻検索システムと組み合わせることでバス停検索のみではなく、バスのルート検索を可能とすることで、ルート上にある近隣情報を検索できるシステムの実装を行っていきたいと考えている。

第7章 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、同じ研究室の清水洋志さん、坂口隼さん、熊谷恵理さん、森瀧昌志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。

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

[1] 「Google Maps」

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

[2] 木村 眞吾, 「目的別空間データベースサーバを用いた Web システムの設計と実装」
島根大学 総合理工学部 数理・情報システム学科 卒業論文 (2007 年)

[3] Michael Stonebreaker 著 「オブジェクトリレーショナル DBMSs」

太田佳伸 訳

株式会社ビー・エヌ・エヌ

International Thomson Publishing Japan (1996 年)

[4] 「Apache Tomcat」

<http://tomcat.apache.org/>

[5] Marty Hall 著 「コア・サーブレット&JSP Java サーバ技術による Web 開発」

岩谷博 訳

ソフトバンク パブリッシング (2001 年)

[6] 斉藤 孝 著 「リレーショナルデータベース教科書」

ソフト・リサーチセンター (1999 年)

[7] HiRDB Version 7 CD-ROM マニュアル (2004 年)

[8] HiRDB マニュアル 「HiRDB Spatial Search Plug-in Version 3 解説・手引・文法・操作書」 (株) 日和出版センター (2001 年)

付録 アンケート結果

店舗情報検索フォームについてのアンケート

Q 1 : ウェブサイトに対して好印象を抱きましたか？

満足 : 2 (9.7%) やや満足 : 5 (21.7%) やや不満足 : 12 (52.2%) 不満足 : 4 (17.4%)

Q 2 : ウェブサイトの視覚的な見やすさは十分でしたか？

満足 : 4 (17.4%) やや満足 : 7 (30.4%) やや不満足 : 8 (34.8%) 不満足 : 4 (17.4%)

Q 3 : ウェブサイトを利用しようとするときの操作や手順はわかりやすかったですか？

満足 : 1 (4.3%) やや満足 : 3 (13.0%) やや不満足 : 14 (60.9%) 不満足 : 5 (21.7%)

Q 4 : ウェブサイトに対して「これは使える」「役に立つ」という感覚を抱きましたか？

満足 : 3 (13.0%) やや満足 : 9 (39.1%) やや不満足 : 8 (34.8%) 不満足 : 3 (13.0%)

Q 5 : 本システムを使用してどう思いましたか？

満足 : 1 (4.3%) やや満足 : 9 (39.1%) やや不満足 : 11 (47.8%) 不満足 : 2 (9.7%)

Q 6 : 検索サイトにおいて必要なものはなんですか？

使いやすさ 6 (26.1%)

見やすさ 7 (30.4%)

分かりやすさ 3 (13.0%)

検索結果の情報の量 7 (30.4%)

Q 7 : Q 5 で「やや不満足・不満足」と答えられた方は、

その理由を以下よりお選びください。(複数回答可能)

目的検索の業種数が少ない。 : 9 (69.2%)

求めている結果がでなかった。 : 4 (38.5%)

結果が見にくい。 : 6 (46.2%)

デザインが地味だ。 : 3 (23.1%)

検索項目が少ない。 : 11 (92.3%)

その他 : 2 (15.4%)

Q 8 : 新たに追加すると便利だと思う機能があれば選択してください。

検索項目の増加。(例、店名検索)	: 6 (30.4%)
検索結果項目の増加。	: 10 (43.5%)
地図上に店舗情報の表示。	: 3 (13.0%)
業種によって表示アイコンが変わる。	: 2 (9.7%)
特になし	: 0 (0.0%)
その他	: 1 (4.3%)

※その他の解答を選んだ方、またはお気づきの点がありましたら以下にご自由にお書きください。

- ・ 全体的にシンプルすぎる。
- ・ 説明文が少なく、初めての人にはわかりにくい。
- ・ もっと詳しい検索結果がほしい。
- ・ 店舗情報とバス停検索を同時にしたい。
- ・ 地図上に店舗位置以外の情報もほしい。
- ・ 松江以外の検索できないのか。
- ・ 地図が小さい。

店舗情報入力フォームについてのアンケート

Q 9 : ウェブサイトに対して好印象を抱きましたか？

満足 : 0 (0.0%) やや満足 : 2 (15.4%) やや不満足 : 8 (61.5%) 不満足 : 3 (23.1%)

Q 10 : ウェブサイトの視覚的な見やすさは十分でしたか？

満足 : 2 (15.4%) やや満足 : 5 (38.5%) やや不満足 : 4 (30.8%) 不満足 : 2 (15.4%)

Q 11 : ウェブサイトを利用しようとするときの操作や手順はわかりやすかったですか？

満足 : 1 (7.7%) やや満足 : 4 (30.8%) やや不満足 : 6 (46.2%) 不満足 : 2 (15.4%)

Q 12 : ウェブサイトに対して「これは使える」「役に立つ」という感覚を抱きましたか？

満足 : 0 (0.0%) やや満足 : 3 (23.1%) やや不満足 : 7 (53.4%) 不満足 : 3 (23.1%)

Q 1 3 : 本システムを使用してどう思いましたか？

満足 : 0 (0.0%) やや満足 : 3 (23.1%) やや不満足 : 8 (61.5%) 不満足 : 2 (15.4%)

Q 1 4 : Q 1 3 で「やや不満足・不満足」と答えられた方は、

その理由を以下よりお選びください。(複数回答可能)

入力できる情報が少ない。	8 (80.0%)
画像、カラーなどを用いて編集がしたい。	4 (40.0%)
入力項目の説明が少なく、分かりにくい。	5 (50/0%)
独自の入力項目が欲しい。	4 (40.0%)
どのように表示されるのかが分からない。	2 (20.0%)
その他	1 (10.0%)

※その他の解答を選んだ方、またはお気づきの点がありましたら以下にご自由にお書きください。

- ・ 全体的にシンプルすぎる。
- ・ 入力する内容に特徴がない。
- ・ 自分の HP とのリンクができない。
- ・ 動画や音楽も流せるようにしたい。