

# アクティブデータベースを用いた地理情報システム

寺田 努<sup>†</sup> 塚本 昌彦<sup>†</sup> 西尾 章治郎<sup>†</sup>

近年, GPS 機器や携帯端末の普及により, 屋外で自分の位置を基点とした地理情報システムを利用したいという要求が高まっている. 屋外で, 情報サーバが無線を用いて地理情報を発信していれば, 地理情報システムではそのような受信情報を端末の地図情報と統合し, ユーザに高度なサービスを提供できる. そこで, 本研究では, データ管理基盤として AMDS (Active Mobile Database System) を用いた地理情報システム ActiveGIS を構築した. AMDS は, 移動体環境において移動体を持つデータを統合利用するために, アクティブデータベースを拡張し, 移動体の接続, 切断, データ交換などを処理するシステムである. ActiveGIS は, データの到着や特定の建物への接近などの事象に対する処理をルールとして記述しておくことで, システムにさまざまな機能を付加できる.

## A Geographic Information System Using Active Database Systems

Tsutomu Terada,<sup>†</sup> Masahiko Tsukamoto<sup>†</sup> and Shojiro Nishio<sup>†</sup>

In recent years, the increasing popularity of mobile hosts and GPS equipments creates a great demand on location-aware geographic information systems (GIS). If information servers transmit geographic information with wireless communication equipments anywhere in the real world, a GIS running on a mobile computer can provide a high level service to users by integrating the information received from servers and the map information stored in the computer. To realize this kind of service, we proposed and implemented the ActiveGIS, a geographic information system based on the AMDS (Active Mobile Database System). The AMDS is a kernel system for data management in mobile computing environments such as integrating and using the data held by mobile host. ActiveGIS enables users to add their own functionalities by describing rules which express the processes for the events such as data arrival and user's approach to a specific real-world situation.

### 1. はじめに

近年, 無線通信技術や計算機ハードウェア技術の急速な発展にともなって, 無線通信機能を持つ携帯端末を用いることにより場所を固定せずにネットワーク上のさまざまな資源を利用することが可能になった. この新しい計算環境は移動体計算環境と呼ばれる. 移動体計算環境のモデルは固定ネットワークに無線通信可能な携帯端末を含んだ形態である. このような移動体計算環境が普及し, ユーザが携帯端末を持ち歩くことが一般的になってくると, 携帯端末上で地理情報システムを利用したいという要求が生じる. 地理情報システム (GIS: Geographic Information Systems) とは, デジタル化された地図をベースに, さまざまな情報を付加して加工・分析し, 人間に分かりやすい形にビジュアル化するシステムである.

一般に GIS を利用する目的としては, 地図上の距離や, 体積などを調べる地理的計量処理, 地図要素間の空間関係を分析する近接分析, 点, 線, 面などの図形から等しい距離にある領域を確定し, 小売店の商圈などを調査するバッファリング, 属性の異なる複数のレイヤを重ね合わせて新しい主題図を作るオーバレイ, 最短経路の探索や巡回問題などを処理するネットワーク分析などがあげられる<sup>15)</sup>. 実際の商品では, エプソン社のロカティオ<sup>1)</sup>やソニー社の VAI0 PCG-C2GPS<sup>19)</sup>, エンペックス社のポケナビ<sup>14)</sup>のように, GPS を搭載した携帯端末がすでに販売されており, 位置情報を利用したサービスが行われている. また, GPS 機器の普及により, 地図ソフトも GPS と連携する機能を標準で持つようになった. さらに, OGC (Open GIS Consortium)<sup>2)</sup>の活動や, G-XML<sup>4)</sup>, POIX (Point Of Interest eXchange Language)<sup>3)</sup>, RWML (Road Web Markup Language)<sup>3)</sup>など, 地理情報記述の標準化活動がさかに行われている. このように, 屋外で携帯端末を用いて地理情報システムを利用できる環

<sup>†</sup> 大阪大学大学院工学研究科情報システム工学専攻  
Department of Information Systems Engineering,  
Graduate School of Engineering, Osaka University

境が整ってきたといえる。

しかし、従来のシステムには以下のような問題がある。

- 情報の統合利用が難しい。  
従来のシステムやアプリケーションは相互に互換性がないため、あるシステムで利用することのできる情報が他のシステムでは利用できないという問題がある。地理情報システムでは、各種のアプリケーションを連携動作させた柔軟なサービスが要求されるため、アプリケーション間の連携が必要となる。
- 多様な位置依存サービスへの要求に対応できない。  
屋外での地理情報システムを利用する場合、以下のような位置依存サービスへの要求が出てくる。
  - － 遊園地などのアミューズメント施設で、各アトラクションの待ち時間などの情報を端末で受信する。また、訪れたいアトラクション情報を端末に入力しておくことで、効率的な順番にナビゲートしてくれる。
  - － 買いたい本の情報を端末に入力しておくことで、自分が本屋の近くに来たときに、自動的にその本屋のデータベースを検索して本の在庫や価格などの情報を得る。
  - － 近くにいる人を対象に、趣味情報などを交換することで、遊び相手や人探しを行う。
  - － 観光地などで、名所に近づくと、その詳細を説明してくれたり、順路を教えてくれたりする。
 ここで示した例はいずれも、その時間、その場所にいる人にとって役に立つ情報である。たとえば、特定の場所に行ってオブジェクトを発見する実世界におけるアドベンチャーゲームを提供したり、特定の日に特定の場所に行かなければ見られないパルチヤル花火のイベントを行うといったサービスは、位置依存だから行えるサービスであり、位置依存であるから情報の価値が高まる<sup>17)</sup>。いくつかの位置依存サービスは既存の商品で実現されているが、例にあげたようなサーバとインタラクションを行う複雑なサービスは提供できていない。
- 柔軟なデータ管理機構を持っていない。  
地図上のオブジェクトデータは膨大な量であるが、携帯端末の記憶容量は限られているため、自分が今いる場所に近いオブジェクト情報だけを得ることが必要となる。ただし、携帯端末の移動により近接するオブジェクトは変化するため、ユーザの動きに合わせて近接するオブジェクトのデータを、随時受け取り、更新する必要がある。  
また、端末上で近辺の情報を取得したり、必要な

データを記録したりするといったサービスへの要求が出てくるが、そのようなサービスを実現するためには柔軟なデータ管理機構が必要となる。

したがって、本研究ではこれらの問題点を解決する地理情報システムの基盤システムを構築することを目的とする。基盤システム上でアプリケーションを動作させることで、データの統合利用を容易にすることを考える。

多様な位置依存サービスを実現するためには、位置依存の情報や、周囲のオブジェクトデータを提供する移動体サーバを各地に設置したうえで、移動体から、または移動体上でデータを収集する必要がある。また、提案するシステムは、ユーザのオブジェクトへの接近やデータの到着といった突発的な事象を取り扱う必要がある。そこで、本研究ではイベント駆動型データベースであるアクティブデータベースを用いて地理情報システムを構築する。構築したシステムを ActiveGIS と呼ぶ。

以下、2章では ActiveGIS の概要について述べ、3章では ActiveGIS の設計について述べる。4章でシステムの実装について述べ、5章で考察および関連研究について述べる。最後に6章で本研究のまとめと今後の課題について述べる。

## 2. ActiveGIS

ActiveGIS は GPS 機器と無線通信機能および電子地図を備えた携帯端末にアクティブデータベースを組み込んだ地理情報システムである。アクティブデータベースは、データベースの内界・外界で起こる事象の発生に対して、あらかじめ定義された処理を行うデータベースである<sup>5)</sup>。その動作は、発生する事象(イベント)、ルールの発火条件(コンディション)、実行される操作(アクション)の3つの組で表される ECA ルールで記述される。アクティブデータベースはイベント駆動型であるため、データの受信やオブジェクトへの接近などの突発的な事象を処理するのに適している。

ActiveGIS には、従来のアクティブデータベースで提供されているデータベースに関するイベント(挿入、削除、更新、検索)に加えて、移動体の動作に関するイベント(移動体のセルへの接続・切断)、ActiveGIS 間や移動体サーバとの通信に関するイベント(データの受信)、および、地図オブジェクトに対するイベント(オブジェクトへの近接・離脱)などを用意する。アクションには ActiveGIS 間のデータ送信関数、データベースに対する問合せ関数などを提供する。

ActiveGIS はシステム内にデータベースを持ち、データベース内のオブジェクトテーブル内に存在するオブジェクトを地図上に表示する機能を持つ。画面上には現在地を中心とした地図を表示し、表示しているオブジェクトをクリックすることで詳細データを得ることができるようにする。さらに、システム内に、ECA ルールを格納することで、さまざまな機能を実現できる。以下に典型的な ECA ルールの利用例をあげる。

#### (1) 目的別地図作成

建物のデータを配信するサーバからデータを受信したとき、建物の種類などの条件によってローカルに蓄積するかどうかを決めるようなルールを用いることで、レストラン地図やショッピング地図など、ユーザが要求するものだけを表示できる。

#### (2) ローカル情報の取得

建物に近づいたとき、特定種類の建物であったらサーバから情報を得るルールを用いることで、たとえば観光地において名所に近づいたときにその詳細を説明したり、観光のための順路を地図上に表示したりするサービスが実現できる。

#### (3) ローカルサービスの利用

ローカル情報の取得だけでなく、サーバとやりとりを行うことで、サービスを受けることができる。たとえば、本屋に近づいたときに、ルールにより自動的に個人情報から買いたい本の情報を検索して本屋のサーバに送信し、サーバ側のルールにより、検索結果を返してもらうといった複雑なサービスが実現できる。このようなルールを用いると、商店街を歩くだけで自動的に自分が欲しい商品の特價情報が得られ、安く売っている店の前に来たときに教えてくれるようなサービスが実現できる。

#### (4) 行動履歴の蓄積

特定の場所に来たときに、自分のデータベースの内容を更新するようなルールを用いることで、スタンプラリーやオリエンテーリングなどのように、さまざまな場所をまわってアイテムを集めてまわるようなアミューズメントサービスが実現できる。また、通行証のようなデータを特定の場所を通るたびに蓄積していけば、自分のたどった道筋をあとから知ることができる。

#### (5) 情報の共有

ローカルデータベースが更新されたとき、その内容によって、サーバに更新内容を送信するようなルールを用いることで、情報の共有が可能になる。たとえば、ローカルデータベースには自分だけが利用する建物情報や経路情報を自由に追加し、ローカル

データベースの更新に対して、更新データを移動体サーバに送信するようなルールを用いることで、更新を移動体サーバに伝播し、複数の人で経路などの情報が共有できるようになる。

#### (6) 動的なシステム拡張

ECA ルールは、システム間で送受信することが可能であり、その作業自体も ECA ルールで行うことができる。したがって、ユーザはあらかじめ ECA ルールを格納しておかなくても、移動体サーバから送られる ECA ルールを格納することで、即座にアプリケーションを更新できる。このような機構を用いて、施設の入口で必要な ECA ルールを受け取ることで、遊園地では遊園地の、水族館では水族館の提供する独自のアプリケーションが利用できる。

### 3. ActiveGIS の設計

本章では、ActiveGIS の設計について述べる。まず ActiveGIS で用いる ECA ルールの言語仕様について述べる。次に ActiveGIS のシステム構成について述べ、最後に、ActiveGIS のアクティブデータベース部として利用する AMDS について説明する。

#### 3.1 ECA ルール言語仕様

ActiveGIS は一般のアクティブデータベースと同様に ECA ルールで動作を記述する。本節では、ActiveGIS における ECA ルールの言語仕様について説明する。提供する ECA ルールの記述構文を図 1 に示す。‘ルール名’にはルールの内容を表す名前を付け、‘イベント名’にはルールが対象とするイベントの名前を記述する。‘対象テーブル’にはイベントが対象とするテーブルを指定し、‘変数型宣言’では ECA ルール中で使用するローカル変数を定義する。宣言する変数は、文字列変数、タプル変数またはタプルの組を表す変数のいずれかである。‘コンディション’にはルールの発火条件を‘<左辺> <オペレータ> <右辺>’の形式を繰り返すことで記述する。両辺には、データベース属性、受信パケット内容や定数、<オペレータ> としては、=, !=, <=, >=, >, < が記述できる。‘アクション’にはルールが発火したときに行う動作について記述する。ActiveGIS で取り扱うことのできるイベントおよびアクションを表 1、表 2 に示す。

また、ECA ルールを記述する場合「挿入されたタプルに対して～という処理を行う」であるとか「削除されたタプルが～に関するものだった場合、関連するタプルもすべて削除する」といったルールを書きたい場合がある。このような ECA ルールを記述するためには、挿入されたタプルや削除されたタプル

```
CREATE RULE ルール名 ON イベント名
[ TO 対象テーブル ]
[ 変数型宣言 ]
[ WHERE コンディション ]
THEN DO アクション
```

図 1 ECA ルールの記述構文  
Fig. 1 Syntax of ECA-Rule.

表 1 ActiveGIS のイベント  
Table 1 Events of the ActiveGIS.

名称	内容
SELECT	テーブルに対するデータ参照
INSERT	テーブルに対するタプルの挿入
DELETE	テーブルのタプル削除
UPDATE	テーブルのタプル更新
RECEIVE	データパケットの受信
CONNECT	移動体のセルへの接続
DISCONNECT	移動体のセルからの退出
TIMER	設定したタイマの発火
CLOSE	オブジェクトへの接近
AWAY	オブジェクトから離れた

表 2 ActiveGIS のアクション  
Table 2 Actions of the ActiveGIS.

名称	内容
QUERY([クエリ内容])	データベース操作
SEND([宛先], [送信内容])	データの送信
INSERT_ECA([ルール内容])	ECA ルール格納
DELETE_ECA([ルール識別子])	ECA ルール削除
ENABLE_ECA([ルール抽出条件])	ECA ルール有効化
DISABLE_ECA([ルール抽出条件])	ECA ルール無効化
SET_TIMER([タイマ条件])	新たなタイマの設定
KILL_TIMER([タイマ識別子])	タイマの削除
DISPLAY([表示タイプ], [表示内容])	画面表示
XML_PARSE([XML 表現])	XML を処理

に関する情報が必要になる。そこで、NEW データおよび OLD データと呼ぶシステム変数を用意する。イベントが発生したときにはこれらの変数に必要な情報が格納される。コンディション部やアクション部において、これらの変数は自由に利用できる。各イベントが発生したときの NEW データ、OLD データの内容を表 3 に示す。

### 3.2 ECA ルール記述例

ActiveGIS における ECA ルールの記述例を図 2 に示す。例に示した接続ルールは、移動体が接続してきたときに、その移動体に対してデータ要求を行う移動体サーバ用のルールである、返信ルールは、移動体サーバからの要求パケットを受信したとき、個人情報を送り返す移動体用ルールである。

移動体サーバでは、移動体が接続してきたときに接続ルールが起動し、接続ルールのアクションにより、

表 3 NEW データと OLD データの内容  
Table 3 Data contained in the NEW-Data and the OLD-Data.

イベント	NEW	OLD
SELECT	参照タプル	-
INSERT	挿入タプル	-
DELETE	-	削除タプル
UPDATE	更新後タプル	更新前タプル
RECEIVE	到着パケット内容	-
CONNECT	接続移動体情報	-
DISCONNECT	-	切断移動体情報
TIMER	タイマ識別子	-
CLOSE	近接オブジェクト情報	-

```
CREATE RULE 接続 on CONNECT
THEN DO
SEND( NEW.FROM, "Request" );

CREATE RULE 返信 on RECEIVE
RECORDSET DATA;
WHERE NEW.HEADER = 'Request'
THEN DO DATA = QUERY("select s.*
                        from Profile s");
SEND( NEW.FROM, "result", DATA );
```

図 2 ECA ルール例  
Fig. 2 An example of ECA-Rules.

移動体の返信ルールが起動する。このように、ECA ルールでは、アクションの実行が新たなイベントを発生させられるため、1 つのイベントの発生によって複数の ECA ルールを連鎖的に実行できる。そのため、ECA ルールを連鎖的に実行させることで、複雑な動作が実現できる。

### 3.3 システム構成

ActiveGIS のシステム構成を図 3 に示す。ActiveGIS はアクティブデータベースを拡張するものであるが、筆者らはこれまで、アクティブデータベースを移動体計算環境に適応させた AMDS (Active Mobile Database System) に関する研究を行ってきた<sup>7),9)</sup>。AMDS は移動体のネットワークへの接続・切断などをイベントとして処理できるため、ActiveGIS は AMDS モジュールを用いて実現した。AMDS モジュールでは、ルール処理やデータの送受信などの処理を行っている。地図表示部では、GPS から現在地の情報を受け取り、自分の位置を中心とした地図を表示する。情報表示部では、地図上のオブジェクトがポイントされたときにその情報を表示するとともに、ルールの DISPLAY アクションによって指定された内容を画面に表示する役割を持つ。ユーザインタフェース部では、各種パラメータを入力させたり、地図の拡大縮小など、

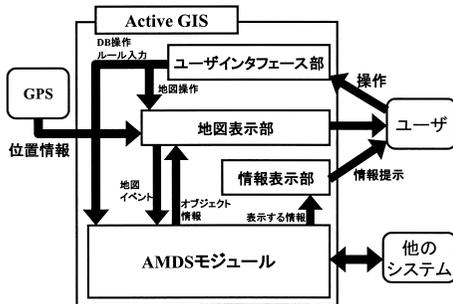


図3 ActiveGISのシステム構成

Fig. 3 The system structure of the ActiveGIS.

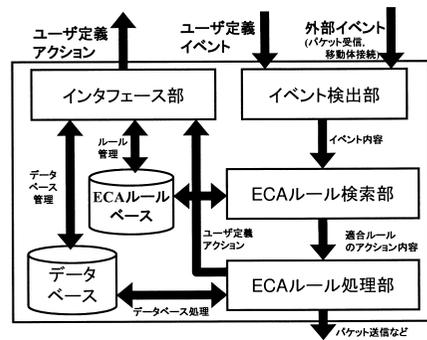


図5 AMDSモジュールのシステム構成

Fig. 5 The system structure of the AMDS module.

```

<G-XML version="0.1">
<Feature Fid="1" Theme="本屋">
<Polygon>
<Coordinate>153000,163500</Coordinate>
<Coordinate>153800,163500</Coordinate>
<Coordinate>153800,164000</Coordinate>
<Coordinate>153000,164000</Coordinate>
</Polygon>
</Feature>
<Feature Fid="2" Theme="道路">
<LineString>
<Coordinate>148000,165000</Coordinate>
<Coordinate>151000,165000</Coordinate>
<Coordinate>151000,162000</Coordinate>
</LineString>
</Feature>
</G-XML>
    
```

図4 G-XMLのインスタンス記述例

Fig. 4 An example of G-XML instance.

ユーザからの入力を受け取って処理する。

ActiveGISで想定する環境では、実世界上に多くの移動体サーバが存在し、付近のオブジェクト情報を放送していると看做す。オブジェクト情報の形式は、現在標準化が行われている、建物や空間情報、略地図などの情報をXML記法で表したG-XMLプロトコルのα版<sup>4)</sup>を参考にした。記述例を図4に示す。

### 3.4 AMDSモジュール

データの送受信やルール処理などのアクティブなメカニズムは、すべてAMDSモジュールで実現している。AMDSモジュールは、WindowsのDLL(Dynamic Link Library)として実装を行っており、Visual BasicやVisual C++などで作成するアプリケーションに含めるだけで、その機能が利用できる。AMDSは1つのクラスとして動作するため、クラスのプロパティを設定したり、提供するメソッド(関数)を呼び出すことで各機能を利用する。プロパティおよびメソッドの仕様を付録に示す。

AMDSモジュールのシステム構成を図5に示す。以下、システムの各部について説明する。

- イベント検出部  
保持しているデータベースや外部からのパケット受信など、システム内外に起こったイベントを検出する。その際、NEW, OLDデータの作成などを行う。
- ECAルール検索部  
イベント検出部から受け取ったイベントを受け取り、ルールベースから対応するECAルールを検索する。存在すれば、それらのルールすべてを取り出し、コンディションの評価を行い、実行されるべきルールをECAルール処理部に渡す。
- ECAルール処理部  
受け取ったECAルールを実行する。
- インタフェース部  
アプリケーションからの要求を処理する。AMDSモジュールが提供するさまざまな機能の呼び出しは、すべてインタフェース部を介して行われる。また、ECAルールビューアや、イベントログビューアなどのGUIの管理も行っている。

### 3.5 ユーザ定義イベントとアクション

アプリケーションからAMDSモジュールを利用する場合、用意されたイベントやアクションだけでは十分ではない場合がある。たとえば、ActiveGISにおいて、CLOSE イベントや、DISPLAY アクションは、AMDSが標準で持っていないイベントとアクションである。そこで、AMDSモジュールのユーザイベント定義機能およびユーザアクション定義機能を利用する。ユーザ定義イベントでは、あらかじめイベント名やイベント発生時に作成されるNEWデータ、OLDデータの形式を与えておけば、アプリケーションからイベントが起こせる。また、ユーザ定義アクションでは、あらかじめアクション名と引数のとり方を与えておけば、自由にルールからアプリケーションにデータ

```
CREATE RULE 近接 on Close(50m)
WHERE NEW.TYPE = 'Library'
THEN DO
  Display("NormalMessage",
    "図書館が近くにあります。");
```

図 6 ECA ルール例 1

Fig. 6 An example of ECA-Rules set (1).

```
CREATE RULE データ受信 on RECEIVE
STRING DATA;
WHERE NEW.DATATYPE = 'ObjectData'
      NEW.TYPE = 'BookStore'
THEN DO
  DATA = QUERY("SELECT BookName
                FROM RequestBookTbl");
  SEND( NEW.FROM, "BookRequest",
        "NAME", DATA );

CREATE RULE データ返送 on RECEIVE
STRING DATA;
WHERE NEW.DATATYPE = 'BookRequest'
THEN DO
  DATA = QUERY("SELECT Price FROM BookTbl
                WHERE BookName = '%s'", NEW.NAME );
  SEND( NEW.FROM, "Result",
        "PRICE", DATA );

CREATE RULE 結果表示 on RECEIVE
WHERE NEW.DATATYPE = 'Result'
THEN DO
  DISPLAY( "BookPrice", NEW.PRICE );
```

図 7 ECA ルール例 2

Fig. 7 An example of ECA-Rules set (2).

```
CREATE RULE メニュー要求 ON CLOSE(30m)
WHERE
  NEW.TYPE = "レストラン"
THEN DO
  SEND( MAPSERVER, "RequestMenu", "SQL",
        "SELECT * FROM Menutable
          WHERE Price < 1000" );

CREATE RULE データ返送 on RECEIVE
RECORDSET DATA;
WHERE NEW.DATATYPE = 'RequestMenu'
THEN DO
  DATA = QUERY( NEW.SQL );
  SEND( NEW.FROM, "MenuData", DATA );

CREATE RULE 結果蓄積 ON RECEIVE
WHERE
  NEW.TYPE = "MenuData"
THEN DO
  QUERY( "INSERT INTO collectedData
        VALUES( %s, %s, %s)", NEW.Shopname,
        NEW.productname, NEW.price );
```

図 8 ECA ルール例 3

Fig. 8 An example of ECA-Rules set (3).

- (1) 自分から一定範囲内に指定した種類のオブジェクトを見つけたときは、ユーザに通知する。
- (2) 自分の一定範囲内に本屋があったとき、自分の探している本があるかどうかを本屋のデータベースから検索して結果を表示する。
- (3) レストランの前を通るとき、自動的に千円以下のメニューを検索し、結果を蓄積しておく。街を歩きまわれば自動的にデータが収集される。

これらの機能を実現するためのルール例は図 6, 図 7, 図 8 に示すとおりである。ルール例 1 は、図書館が 50 メートル以内に入ったときにユーザに知らせるためのルールである。ルール例 2 は、本屋の建物データを受け取ったときに、自分のデータベースから欲しい本の情報を検索して本屋に送信し(データ受信ルール)、本屋においてデータベースを検索して本の値段を返し(データ返送ルール)、受け取った値段情報をユーザに通知する(結果表示ルール)ルール群である。また、ルール例 3 は、レストランに近づいたときにサーバに千円以下のメニューを問合せし(メニュー要求ルール)、サーバは問合せを実行して結果を返信し(データ返送ルール)、結果を collectedData テーブルに蓄積する(結果蓄積ルール)ルール群である。

このように、アプリケーションの機能は ECA ルールの組で表現でき、ECA ルールを追加することで、システムにさまざまな機能を追加できる。

が渡せる。

ActiveGIS においては、まず、CLOSE イベントを定義し、ActiveGIS の地図表示において自分とオブジェクトとの距離を計算しておき、一定範囲内であればイベントが発火するようにした。また、アクションに関しては DISPLAY アクションと XML\_PARSE アクションを追加した。ルールが発火したとき、そのアクションが DISPLAY であったら、あらかじめ定義しておいたコールバック関数が呼び出され、画面表示を行えるようにし、XML\_PARSE アクションであれば G-XML 記述を処理して地図上に表示できるようにした。

### 3.6 ECA ルールによる機能実現例

本節では、ECA ルールによる機能追加の例を示す。ActiveGIS に以下のような機能を設定する場合を考える。



図9 プロトタイプ稼働図  
Fig. 9 Snapshot of the prototype system.

## 4. 実装

ActiveGISのプロトタイプシステムを実装した。実装はWindowsPC上で、Visual Basic6.0およびVisual C++6.0を用いて行い、無線通信には赤外線通信を用いた。システムの稼働図を図9に示す。

## 5. 考察

### 5.1 ECA ルールの言語仕様について

ActiveGISで用いるECAルールにおいて記述できるイベントおよびアクションは表1、表2に示したが、地図上のオブジェクトに対してのイベントは接近と離脱しか用意していない。現時点においても、2章で述べたようなさまざまなサービスを提供できる。しかし、地理情報システムのアプリケーションには現状のECAルールでは実現できないサービスもある。たとえば、ネットワーク分析やオーバーレイの制御をECAルールで行うことができない。また、定期的な処理や災害調査などにおいては、柔軟なタイム処理を記述できる必要がある。したがって、新たなイベントおよびアクションの追加を含め、言語仕様を再考することが今後の課題となる。

### 5.2 ECA ルールの一貫性維持について

ActiveGISでは、複数のアプリケーションがECAルールの組として混在する。また、ECAルールの送受信や、有効化、無効化を自由に行うことができる。そのため、有効なECAルール群の構成は動的に変化し、矛盾や無限連鎖を起こす可能性がある。アクティブデータベースの研究分野では、ECAルールの無限連鎖や矛盾の検出についてさまざまなアプローチがなされているが<sup>(8),18)</sup>、本システムでは、移動体が自由に動きまわるため、システム内のルールだけでなく、ネットワークトポロジの変化も考慮する必要がある。

筆者らの研究グループでは、移動体環境におけるECAルールの矛盾や無限連鎖を静的に検出する機構<sup>(8)</sup>および動的に検出する機構<sup>(18)</sup>に関する研究を行っており、ActiveGISについても同様の手法が有効であると考えられる。

### 5.3 システムの規模増大について

システムの規模が増大しても、移動体サーバの数を比例して増加させれば、スケールの問題は起こらない。しかし、遊園地などで動作させる場合など、1つの移動体サーバに対して移動体の数が非常に多くなるような環境では、イベント発生の集中およびトラフィックの増大が起こる可能性がある。現在のシステムでは、移動体はすべて移動体サーバとの相互作用によってサービスを受けるため、移動体の集中度合いに応じて移動体サーバに大きな負荷がかかる。さまざまなサービスを想定し、どの程度の処理能力が必要とされるかを実測評価することが今後の課題となる。

### 5.4 実現可能性について

本システムを実環境で動作させるためには、街中に移動体サーバを設置する必要がある。また、すべての移動端末と移動体サーバにActiveGISを搭載する必要がある。移動体サーバの設置に関しては、既存のPHSや携帯電話の基地局にActiveGISのシステムを組み込むことで実現できる可能性がある。また、移動体サーバはネットワークにつながってなくても位置依存サービスを提供できるため、各店舗が無線通信機能を備えた移動体サーバを独自に設置することで、すぐにサービスが開始できる。移動体サーバは一般のパソコンにActiveGISを組み込んだものでよいので、コスト面でも十分実現可能である。移動体に関しては、携帯端末にActiveGISを組み込んでおけばよい。PDA(Personal Digital Assistant)や携帯電話などディスク容量やCPUパワーに制限がある場合は、ECAルールの手入力による追加機能など、互換性に問題のない部分を削除するなど、機能を絞って搭載することになる。

### 5.5 位置の特定方法について

現在のシステムでは位置を特定する方法としてGPS機器を用いている。近年、GPS精度の向上により誤差は数メートル程度に抑えられているため、位置の特定に関してはGPS機器で十分であると考えられる。また、GPSに加えて、他のデバイスと組み合わせることで、より柔軟な位置管理を行える。たとえば、移動体サーバから移動体に対して、今どのセルに入っているかという情報を送信することで、どの移動体サーバに属しているかといった情報を用いてサービスを受け

ることができる。

#### 5.6 移動体サーバの設置間隔について

本システムでは移動体サーバの規模や設置間隔については規定していない。すべてのサーバが対等に存在し、一定間隔ごとに移動体サーバを設置する方法のほかに、セル範囲が広く、基本的なオブジェクトすべてを管理する移動体サーバを広い間隔で設置し、詳細情報や位置依存サービスを提供するサーバを特定の位置ごとに設置するような階層型の設置方法も考えられる。階層型にすることで、大まかな情報は遠くからでも取得でき、位置依存の情報は特定サーバの近くに行かないと得られないようにできる。

#### 5.7 蓄積するオブジェクトの量について

一般に移動体が保持できるデータ量には制限がある。ActiveGIS ではデータ取得の際に ECA ルールを用いたフィルタリングを行うことで、必要なデータのみを蓄積できる。一方、すでに大量のデータが蓄積されている場合、不要なデータを削除する操作が必要となる。ECA ルールのイベント部分に蓄積データ量の制限オーバーを記述できるようにするなど、キャッシュを管理するルールを記述できるようにすることが今後の課題となる。

#### 5.8 ECA ルールの記述者について

ECA ルールの記述は一般にアプリケーション作成者が行う。しかし、一般ユーザがルールのカスタマイズを行ったり、ユーザ固有のルールを記述したりしたいという要求もある。ECA ルールはシンプルな言語であるが、一般ユーザがすべてを記述することは現実的でないため、ECA ルールのオーサリングツールなどを提供する必要がある。

#### 5.9 関連研究

本研究と同様に、GIS にアクティブデータベースの機構を取り入れたものに、GIS ユーザインタフェースの動的カスタマイズに関する研究<sup>11)</sup>がある。この研究は、ユーザインタフェースを ECA ルールを用いて動的に変化させる研究<sup>2),10)</sup>を GIS に適用したものである。Oliveira ら<sup>11)</sup>は、ActiveGIS と同様、GIS にアクティブルールのメカニズムを盛り込んでいるが、ECA ルールはユーザインタフェースをカスタマイズすることに用いている。データ内容によってインタフェースを変化させることで、ユーザにとって分かりやすいデータ処理が行えるが、追加されているイベントやアクションも、ユーザインタフェース構築用のものであり、アプリケーションの機能を制御することはできない。一方、ActiveGIS では、ユーザインタフェースをルールのみで制御することはできないが、アプリケー

ションの機能そのものが ECA ルールで表現されるように拡張を行っているため、ECA ルールを追加したりカスタマイズしたりすることで、アプリケーション自体をカスタマイズできる。

モバイル環境で GIS を利用するための研究としては伊藤らのシステム<sup>6)</sup>があげられる。文献で述べられているモバイル GIS に求められる機能としては、図形情報の入力や更新、データの整理や分析、出力などがあげられているが、その目的は現地調査における各種のデータ入力をその場で行うことである。一方、ActiveGIS は、ECA ルールによるサーバとのインタラクションを用いて位置依存サービスを受けることを主な目的としている。

位置依存サービスの研究例としては、SpaceTag<sup>17)</sup>や、モバイルインフォサーチ<sup>16)</sup>があげられる。SpaceTag は、時空間上にオブジェクトを配置し、その場、その時間にはないとオブジェクトにアクセスできないように制限を加えることで位置依存サービスを実現する機構である。SpaceTag は、それ自体が時空間上の場所にあるのではなく、サーバにまとめて蓄積されている位置情報を持ったオブジェクトである。したがって、端末はデータにアクセスする際には、ネットワーク上に存在するサーバにアクセスするための機能を持つ必要がある。また、悪意を持ったユーザによって、離れた場所からネットワーク越しにデータを見られてしまう可能性がある。一方、ActiveGIS における移動体サーバは、自分の周りにある情報を保持し、それを無線によって送信しているため、近くにいなければ受信できない。したがって、より強固な位置依存サービスを提供できるが、移動体サーバが各所に存在するという前提が必要となる。モバイルインフォサーチでは、モバイルユーザの現在位置に対して、時刻表や最寄の店などの情報を提供することで、実世界とネットワークの両方から情報を得ることができるようにしたシステムである。移動体上でデータ収集を行う点で ActiveGIS との共通点があるが、モバイルインフォサーチでは、データ収集をエージェントによって実現しようとしているのに対し、ActiveGIS では ECA ルールによってデータ収集を行っている。

## 6. おわりに

本稿では、ECA ルールによる拡張が可能な地理情報システム ActiveGIS の構築について述べた。ActiveGIS を用いることで、さまざまな位置依存サービスが利用可能となる。今後は、さらにイベントやアクションを追加し、機能拡張を行っていく予定である。ま

た、G-XML 仕様はまだ検討段階であるが、現時点の  $\alpha$  版においても多彩な表現が可能であり、今回実現したシステムではその一部にしか対応していない。そこで、G-XML 仕様のより多くの部分に対応し、ActiveGIS の有効性を検証する予定である。

謝辞 本研究を進めるにあたり、G-XML 標準化検討委員会のプロトコル検討小委員会、プロトタイプ開発小委員会での議論が大変参考になった。ここに、東京大学の有川正俊助教授、中央大学の久保田光一助教授、財団法人データベース振興センターの天野清憲氏をはじめとする関係者の皆様に感謝の意を示す。なお本研究は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究( Project No.JSPS-RFTF97P00501 )」の研究助成によるものである。

### 参 考 文 献

- 1) CLUB Locatio:  
<http://www.i-love-epson.co.jp/locatio/>
- 2) Diaz, O., Jaime, A. and al Qaimari, G.: Supporting Dynamic Displays Using Active Rules, *ACM Sigmod Record*, Vol.23, No.1, pp.21-26 (1994).
- 3) 道路用 Web 記述言語 RWML の提案:  
<http://www2.ceri.go.jp/its-win/RWML.htm>
- 4) G-XML ホームページ:  
<http://gisclh.dpc.or.jp/gxml/contents/>
- 5) 石川 博: アクティブデータベース, 情報処理, Vol.35, No.2, pp.120-129 (1994).
- 6) 伊藤 悟, 岡部篤行, 奥貫圭一, 東明佐久良, 秋田義一, 小坪宏則, 大喜多祐司, 後藤 寛, 金子忠明, 足立俊雅, エリックパーズリー: 都市計画基礎調査におけるモバイル GIS 利用の試み(その1), 地理情報システム学会講演論文集, No.7, pp.137-140 (1998).
- 7) 村瀬 亨, 塚本昌彦, 西尾章治郎: アクティブデータベースシステムによる移動体計算環境におけるデータ統合, 電子情報通信学会データ工学研究会, Vol.95, No.287, pp.41-48 (1995).
- 8) 村瀬 亨, 塚本昌彦, 西尾章治郎: 移動体環境におけるアクティブデータベースの安全性について, 情報処理学会研究報告, 96-DBS-106, Vol.96, No.11, pp.33-40 (1996).
- 9) Murase, T., Tsukamoto, M. and Nishio, S.: A system Platform for Mobile Computing base on Active Database, *International Symposium on Cooperative Database Systems*, Vol.2, pp.424-427 (1996).
- 10) Oliveira, J.L., Cunha, C.Q. and Magalhaes, G.C.: Object Model for Dynamic Construction of Visual Interfaces, *9th Brazilian Symposium on Software Engineering*, pp.143-158 (1995).
- 11) Oliveira, J.L., Medeiros, C.B. and Cilia, M.A.: Active Customization of GIS User Interfaces, *13th International Conference on DATA ENGINEERING*, pp.487-496 (1997).
- 12) Open GIS Consortium, Inc:  
<http://www.opengis.org/>
- 13) Point Of Interest eXchange Language:  
<http://www.w3.org/TR/poix/>
- 14) ポケナビ:  
[http://www.empex.co.jp/sp-gps/s\\_gps.html](http://www.empex.co.jp/sp-gps/s_gps.html)
- 15) スター, J., エステス, J. (著), 岡部篤行, 貞広幸雄, 今井 修 (共訳): 入門地理情報システム, 共立出版 (1992).
- 16) 高橋克巳: モバイル環境下での情報収集を支援するエージェント, 人工知能学会誌, Vol.14, No.4, pp.14-21 (1999).
- 17) 垂水浩幸, 森下 健, 中尾 恵, 上林弥彦: 時間限定型オブジェクトシステム: SpaceTag, 日本ソフトウェア科学会 WISS'98, pp.1-9 (1998).
- 18) 寺田 努, 莫 君, 村瀬 亨, 塚本昌彦, 西尾章治郎: 移動体計算環境におけるアクティブデータベースの ECA ルール実行監視機構の設計と実装, 情報処理学会研究報告, 99-DBS-119, Vol.99, No.203, pp.55-60 (1999).
- 19) VAIO PCG-C2GPS:  
<http://www.sony.co.jp/sd/ProductsPark/Consumer/PCOM/PCG-C2GPS/>

### 付録 AMDS モジュールのアプリケーションインタフェース

Visual Basic または Visual C++ において, AMDS モジュールをプログラムから参照設定すれば, AMDS クラスが使用できるようになる. AMDS モジュールではさまざまなプロパティとメソッドを提供し, アプリケーションプログラマが容易に AMDS を利用できるようにしている. 以下, 実装した AMDS クラスのプロパティとメソッドを示す.

#### プロパティ

- portNum

AMDS の赤外線通信に使うポート番号を指定する.

- disconnectTime

AMDS が移動体サーバのとき専用のプロパティ. 移動体からのハローパケットが, このプロパティに設定した値以上の間受信しなければ, その移動体が切断したと判断する.

- helloInterval

AMDS が移動体のとき専用のプロパティ. 移動体のハローパケット射出間隔を定義する.

## メソッド

- **OpenDB**( [対象データベース], [対象テーブル名], [ホストのタイプ], [自分のホスト名] ) 戻り値: エラー情報 (整数型)  
AMDSの初期化を行う。‘ホストの種類’には, MH (移動体), MHS (移動体サーバ), SH (固定ホスト) のいずれかを指定する。
- **RiseDBGrid**() 戻り値: 成功/失敗  
AMDSが現在対象としているテーブル内容をグリッド形式で表示。ユーザはグリッド上のデータを自由に書き換えできる。グリッド上での書き換えに対してもイベントは発生する。
- **HideDBGrid**() 戻り値: なし  
テーブル内容を非表示にする。
- **SqlProc**( [SQL 文字列] ) 戻り値: SELECT文の場合, その結果 (配列型)  
SQL 文を実行する。‘SQL 文字列’に SELECT 文を指定した場合, その結果を Variant 型の配列で返す。それ以外の SQL を指定した場合は単に内容を実行する。すべての場合において, 対応するイベントを発生させる。
- **RiseEventLog**() 戻り値: 成功/失敗  
イベントログビューアを表示する。イベントログビューアには, イベントの発生やその処理のログを, 発生時間とともに記録する。
- **HideEventLog**() 戻り値: なし  
イベントログビューアを閉じる。
- **GetMHInfo**() 戻り値: 接続移動体名  
AMDSが移動体サーバのとき, 自分に接続している移動体の一覧を得る。戻り値は文字列型配列。
- **EnterECA**( [ECA ルール定義] ) 戻り値: エラー情報 (整数型)  
ECA ルールをシステムに格納する。‘ECA ルール定義’には, ECA ルールの定義を文字変数に格納してそのまま渡す。
- **GetECA**( [抽出条件] ) 戻り値: ルール文字列  
ECA ルール定義文を (表示するために) 得る。
- **EnableECA**( [抽出条件] ) 戻り値: なし  
ECA ルール有効化。‘抽出条件’にあったルールの実行可能性を ‘有効’ にする。
- **DisableECA**( [抽出条件] ) 戻り値: なし  
ECA ルール無効化。‘抽出条件’に合ったルールの実行可能性を ‘無効’ にする。無効化されたルールは, システム上には存在するが, 有効化されるまで発火することはない。

- **RiseVersion**() 戻り値: なし  
バージョン情報の表示。
- **RiseEcaRuleViewer**() 戻り値: なし  
組込みの ECA ルールビューアを表示する。ルールビューア上では, ECA ルールの閲覧のほか, その場で修正が行える。また, ルール数が多くなったときなどのために, 条件を指定して ECA ルールをサーチする機能を備えている。
- **SaveECA**( [ファイル名] ) 戻り値: なし  
現在システムに格納されている ECA ルールを記録する。ECA ルールの記録形式 (.eca) は, 単純にテキストファイルに ECA ルールの定義を記述したもので, テキストエディタなどで自由に書き換えることができる。
- **LoadECA**( [ファイル名] ) 戻り値: なし  
ECA ルールを読み出す。
- **DefineNewEvent**( [イベント名], [NEW データラベル], [OLD データラベル] ) 戻り値: 成功/失敗  
ユーザ定義イベントを定義する。‘NEW データラベル’, ‘OLD データラベル’には, それぞれ, イベントが発火したときに NEW データ, OLD データにセットする値の識別名を記述する。
- **DefineNewAction**( [アクション名], [引数ラベル], [コールバック関数名] ) 戻り値: 成功/失敗  
ユーザ定義アクションを定義する。‘引数ラベル’には, ルールがアクション内の引数で返す値にアクセスするための識別名を記述する。‘コールバック関数名’には, アクションが発火する際に呼び出してほしいアプリケーション内の関数名を指定する。
- **RiseEvent**( [イベント名], [NEW データ], [OLD データ] ) 戻り値: なし  
ユーザ定義イベントを発火させる。‘NEW データ’, ‘OLD データ’には, それぞれルール中で用いる NEW データ, OLD データを記述する。  
(平成 12 年 4 月 26 日受付)  
(平成 12 年 9 月 7 日採録)



寺田 努 (学生会員)

平成 9 年大阪大学工学部情報システム工学科卒業。現在同大学大学院情報システム工学専攻博士後期課程在学中。アクティブデータベース, データ放送の研究に興味を持つ。



塚本 昌彦 (正会員)

昭和 62 年京都大学工学部数理工学科卒業。平成元年同大学大学院工学研究科修士課程修了。同年シャープ(株)に入社,同社研究員。平成 7 年大阪大学工学部情報システム工学教室講師。平成 8 年より同教室助教授,現在に至る。工学博士。モバイルコンピューティング,分散データベースシステムの研究開発に従事。ACM, IEEE 等 7 学会会員。



西尾章治郎 (正会員)

昭和 50 年京都大学工学部数理工学科卒業。昭和 55 年同大学大学院工学研究科博士課程修了。工学博士。京都大学工学部助手,大阪大学基礎工学部および情報処理教育センター助教授を経て,平成 4 年より大阪大学大学院工学研究科情報システム工学専攻教授となり,現在に至る。この間,カナダ・ウォータールー大学,ビクトリア大学客員。データベース,知識ベース,分散システムの研究に従事。現在, *Data & Knowledge Engineering*, *Data Mining and Knowledge Discovery*, *VLDB Journal* 等の論文誌編集委員。ACM, IEEE 等 8 学会会員。