

移動エージェントに基づくナビゲーションシステムの設計*

5P-3

長谷川誠† 大和田勇人 溝口文雄

東京理科大学 理工学部‡

1 はじめに

近年カーナビゲーションシステムは、単なる経路探索の機能を持つシステムではなく、ITS等との関係や、インターネットとの接続機能をもった統合システムへと変わりつつある。新しいシステムでは、外部と通信をし、より多くのサービスを受けることが可能となる。しかし携帯電話やPHS等の不安定な通信環境で、従来のネットワークとの接続方式をそのまま利用する事は困難である。本論文では、次世代のナビゲーションシステムを考慮した通信の方式として、移動エージェントの利用を提案する。

2 移動エージェント

移動エージェントは、実行するシステムに縛られることなく、ネットワークを介してあるシステムから別のシステムへと移動する。移動先のシステムでサービスを受けた結果はエージェントが保持し、次のシステムへと移動を繰り返す。移動エージェントを導入するメリット [1] としては、次のものが挙げられる。

1. ネットワークの負荷軽減
サービスの行なわれるホストに移動し、そこでプロセス間の通信を行わない結果のみをクライアントへ戻す事でネットワークへの負荷を軽減できる
2. 非同期通信
エージェントを送ったホストとの通信をいったん止め、後ほど結果を持ったエージェントを回収することが可能である。
3. プロトコルのカプセル化
移動エージェントが独自のプロトコルを実装することで、プロトコルの標準化を待たずに新たなサービスに対応することが可能となる。

逆に移動エージェントのデメリットとしては、ユーザとのインタラクションが必要な対話的なアプリケーションを開発する場合である。エージェントが他のシ

ステムへ移動している間は、ユーザとインタラクションすることは困難である。

以下に、これらのメリット・デメリットを考慮し、ナビゲーションシステムにどのように適用するのかを述べる。

3 ナビゲーションシステムへの適用

ナビゲーションシステムへの適用例としては、(1)サーバでの高度な経路探索、(2)広告・駐車場等の情報のナビゲーションへの配信、(3)無線LANによるリアルタイム情報の収集、などが考えられる。以下にどのようなシステムを用い、どのようにして移動エージェントを適用するのかを述べる。

3.1 ナビゲーションシステム

ナビゲーションシステム全体の構成を図1に示す。各コンポーネントの機能は次のようになる。

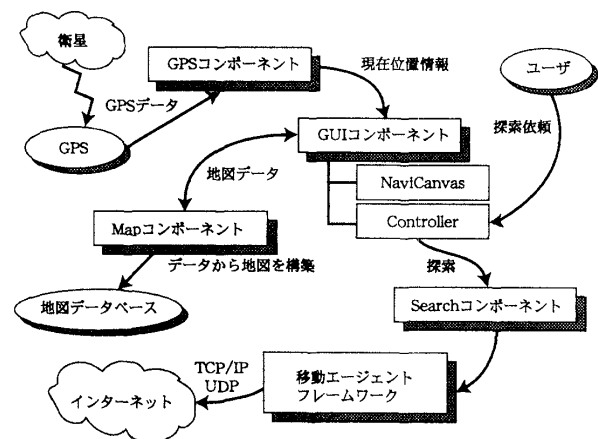


図1: ナビゲーションシステムの構成

- GPS コンポーネント
GPSからデータを取得しデータを1秒毎に他のコンポーネントへ通知する。
- GUI コンポーネント
地図のデータを描画するためのNaviCanvasやユーザとのインターフェースであるControllerを含む。Mapコンポーネントから地図情報を取得する。

*Design of a Navigation System using Mobile Agent

†Makoto HASEGAWA, Hayato OHWADA, Fumio MIZOGUCHI

‡Faculty of Sci. and Tech. Science University of Tokyo

- Search コンポーネント
A*アルゴリズムを利用して経路探索をする。計算した結果は、GUI コンポーネントの方に渡され経路を描画する。
- Map コンポーネント
カーナビゲーション用の地図データベースからデータを読み込む。経度、緯度を指定することで任意の図葉のデータを取得することが可能である。
- 移動エージェントフレームワーク
移動エージェントが動作するための実行環境を提供する。

3.2 移動エージェントによる経路計画

より高度な経路計画を行なうためには、複数移動体の情報をモニタし予測する方法が必要となる [2]。しかし複数移動体の経路計画を行なうためには、スタートからゴールまでの予測経路を送り、ノードの情報を取得するというサーバとの通信を繰り返し行なう必要がある。経路情報はデータ量は多く、転送速度の遅い移動体通信で利用することは困難である。また、繰り返しサーバとのセッションを確立することもネットワークへの負荷がかかる。

しかし、移動エージェントの非同期通信の特徴を活かす事で、ネットワークへの負荷を減らすことが可能である。

3.3 広告等の情報の配信

経路計画の方法では、ナビゲーションシステムからサーバへエージェントが移動し計算をする方法であった。これとは逆にあるサーバ上から広告や駐車場の利用状況といった情報をナビゲーションシステムへ配信するために移動エージェントを利用する方法が考えられる。

移動エージェントを利用しない場合、サーバとクライアントの両方のシステムにプロトコルを実装する必要があり、新しいサービスの標準化を待つことが必要であった。しかし、移動エージェントを利用する場合、プロトコルをカプセル化したコードを持つことが可能であり、ナビゲーションシステムを変更することなく新しいサービスを受ける事を可能にする。

3.4 移動エージェントの実装

サーバ上で経路探索を行う移動エージェントのプログラムの実装例を示す。この場合、サーバにエージェントを送った後、ネットワークを切断する非同期通信によりエージェントの送受信を実現する。

```
01: import agent.*;
```

```
02:
03: public class Search extends MobileAgent {
04:     SearchPlugin hal;
05:
06:     public void arrive() {
07:         hal = (SearchPlugin)getPlugIn();
08:         hal.setSearch(this);
09:     }
10:
11:     public static void main(String[] args) {
12:         try {
13:             Search s = new Search();
14:             s.go('agent://hostname/HAL');
15:         } catch (Exception e) {}
16:     }
17: }
```

移動エージェントは、MobileAgent クラスのサブクラスとして作成する。go メソッドで任意の実行環境上に移動することが可能である。エージェントが実行環境を移動した後、新しい実行環境で arrive メソッドが呼び出される。次に PlugIn として検索サービスを利用する。

移動エージェントを利用した経路計画を行なった場合に必要なネットワークの転送量と転送時間を表 1 に示す。表の送信はエージェントをサーバ上に移動を表し、受信はエージェントがクライアントに戻ってくる場合を指す。距離の短い探索 (10km) と距離の長い場合 (100km) について結果を示す。

表 1: エージェントの転送量

	送信	受信
オブジェクト	10km 圏内	3kbyte
	100km 圏内	27kbyte
クラス	5kbyte	
転送量 (合計)	7.6kbyte	8~33kbyte
通信時間 (9600bps)	5 秒	7~22 秒

移動エージェントの実行環境を含んだ移動エージェントフレームワークは、49 個のクラス 3800 行から構成され、その大きさは 100KB 程度である。

4 まとめ

本研究では移動エージェントを用いたナビゲーションシステムを提案した。移動エージェントの非同期通信を利用する事で、不安定な移動体通信において有効である。またプロトコルをカプセル化する特徴を利用することで、全く新しいサービスの実現も可能となる。

参考文献

- [1] Danny B. Lange, "Mobile Objects and Mobile Agents: The Future of Distributed Computing?", In Proceedings of ECOOP'98, LNCS 1445, Springer, pp.1-13, 1998.
- [2] 木戸, 大和田, 溝口, "複数移動体のための経路計画法", 日本ソフトウェア科学会第 14 回大会論文集, 1997.