

# タウンマネジメントシステムによる サービスロボットのための地図情報管理

Management System of Map Information in Town Management System

○ 田中 堅三(九大) 正 辻 徳生(九大) 正 倉爪 亮(九大) 正 長谷川 勉(九大)

Kenzo TANAKA, Kyushu University, kenzo@irvs.ait.kyushu-u.ac.jp

Tokuo TSUJI, Kyushu University

Ryo KURAZUME, Kyushu University

Tsutomu HASEGAWA, Kyushu University

This paper introduces a map management system in Town Management System (TMS), which is a data management system of a structured environment for a service robot. TMS has been being developed in "Robot Town Project" (MEXT) and "Intelligent RT Software Project" (METI/NEDO) for years as a key technology for managing a structured environment. TMS consists of a database for storing and extracting a variety of environmental information and a software library for data access. In this paper, we introduce some new functions which we appended to TMS to strengthen the map management function so that it processes metric and topological maps efficiently for a service robot.

**Key Words:** Service robot, Information structured environment, Town Management System, Metric map, Topological map

## 1 はじめに

我々はこれまで、文部科学省科学技術振興調整費「ロボットタウンの実証的研究」(2005年11月~2007年度)、経済産業省/NEDO次世代ロボット知能化技術開発プロジェクト「施設内生活支援ロボット知能の研究開発」(2007年度~2012年度)などを通して、サービスロボットが作業を行う環境側に多様なセンサ群を分散配置し、ロボットに搭載したセンサと分散センサ群を連携させ運用する「環境構造情報化」の研究を推進してきた[1-3]。その中で、分散センサから得られる膨大な環境情報や、環境内で作業する複数のロボットの状態を記録、整理、分析し、ロボットからの要求に応じて必要な情報を提供するタウンマネジメントシステム(Town Management System, 以下 TMS)を開発し、RT Middleware 化や介護施設を想定した実証実験などを通して、TMSの機能強化と機能検証を行ってきた[4,5]。

本報告では、特に TMS の環境地図の管理伝達機能の強化を目的に、ロボットによるサービス提供に必要な定量地図と位相地図を TMS に格納し、それらを結びつけてロボットに提供する機能を実装した、サービスロボット用地図情報管理システム(TMS 3.0)を紹介する。

## 2 タウンマネジメントシステム, TMS

### 2.1 タウンマネジメントシステムの概要

分散センサシステムやロボットに搭載したセンサから取得された情報は、移動体の情報(人間やロボットの種類、位置、速度等、「体」と定義)、被操作対象の情報(種別や位置等、「物」と定義)、および環境構造の情報(環境地図、IC タグ配置図等、「場」と定義)を大別できる。

TMS は、この「体」、「物」、「場」の情報を整理、保持し、ロボットからの問い合わせに応じて情報を受け渡すデータベース、およびこのデータベースにネットワークを介して簡便にアクセスするためのインタフェース用 API (Application

Program Interface) ライブラリから構成されている。API は、ロボット・TMS 間のネットワーク通信処理とデータベース操作の二つのプログラムを関数ライブラリとして共通化したものである。TMS で管理された環境で運用されるサービスロボットは、用意された種々の API 関数を用いてネットワーク経由で TMS へアクセスし、「体」、「物」、「場」の情報を取得する。具体的には、データベースには MySQL を、インタフェースの通信方式には SOAP (Simple Object Access Protocol) をベースとする Web サービスを利用して実装されている。それぞれの実装形態は API の中に隠蔽されるため、ロボット側でこれらを考慮する必要はない。また、API はロボットからの問い合わせに応じて TMS が情報を返すリクエスト/リプライ形式と、問い合わせによらず TMS からロボットへ情報を送信する割り込み形式の両方を実装している。割り込み形式はイベントの発生をロボットに通知する場合などに用いられる。イベントは来客や手招きなど環境内で起こる様々な事象を抽象化したものであり、ユーザ側の必要に応じて自由に定義できる。

### 2.2 API ライブラリの実行環境

移植性を高めるため API はまず Java で開発され、それをラッピングする形で他のプログラム言語用の API の開発が行われている。現在、Microsoft Windows XP Professional 上で動作するライブラリとして、Java(TM) SE Runtime Environment 6 用と MySQL 5.1.45, Microsoft Visual C++ 2008 用の API ライブラリが提供されている。

## 3 地図管理機能の強化

本章では、新たに追加した地図管理機能強化のためのデータベース構成および追加した API 群について説明する。

Table 2 Topological map class

| 項目名               | 項目内容                               |
|-------------------|------------------------------------|
| graphmapid        | 位相地図 ID                            |
| type              | 種類 (建物, 階, 部屋, 収納庫)                |
| status            | 状態                                 |
| hgraphmapid       | 上位の位相地図 ID                         |
| coordid           | 座標 ID                              |
| grid_mapid        | 関連する定量地図 ID                        |
| linked_graphmapid | リンクしている位相地図 ID<br>(建物, 階, 部屋, 収納庫) |
| x(yz)_graphmapid  | リンクしている位相地図 ID の<br>x(yz) 座標       |

### 3.1 地図データベース構造

TMS が管理する地図には、壁の位置などの具体的な座標値を表す定量地図と、部屋間の関係をグラフ構造で示した位相地図がある。以下、それぞれの地図のためのデータベース構造を示す。

#### 3.1.1 定量地図

定量地図のために追加したデータベース構造を表 1 に示す。定量地図データベースでは、coordid で関連する位相地図固有の座標系での原点座標を保持している。type は int 型で管理されており、また content において、倉爪らの手法 [6] 等で得られた 実際の定量地図データを、グリッドの高さを示す Multi-layer DEM に変換して保持している。

Table 1 Metric map class

| 項目名           | 項目内容             |
|---------------|------------------|
| mapid         | 定量地図 ID          |
| coordid       | 座標系 ID (原点位置)    |
| scale         | 縮尺 (scale)       |
| type          | 種類               |
| coord_x(yz)1  | 地図左上起点 x(yz) 座標  |
| coord_x(yz)2  | 地図右下終点 x(yz) 座標  |
| content       | バイナリデータ          |
| grid_x_width  | グリッド x 幅         |
| grid_y_width  | グリッド y 幅         |
| grid_x_counts | x 方向のグリッド数       |
| grid_y_counts | y 方向のグリッド数       |
| datasize      | 1 グリッドあたりのデータサイズ |
| graphmapid    | 位相地図 ID          |

#### 3.1.2 位相地図

位相地図のために追加したデータベース構造を表 2 に示す。位相地図データベースでは、grid\_mapid として関連する複数の定量地図 ID を保持しており、位相地図から定量地図の参照が可能となっている。また隣り合う位相地図の ID は、linked\_graphmapid で保持しており、さらに建物、階、部屋などの階層構造を表すために hgraphmapid に上位の位相地図 ID を保持する。

### 3.2 地図データベース管理用 API

位相地図、および位相地図データベースへのアクセス用 API の一部を表 3、表 4 に示す。

Table 3 APIs for metric map class

| メソッド名       | 機能                 |
|-------------|--------------------|
| getInstance | 定量地図クラスのインスタンスを取得  |
| getMapInfo  | 定量地図 ID で定量地図情報を取得 |
| setMapInfo  | 定量地図 ID で定量地図情報を更新 |

Table 4 APIs for topological map class

| メソッド名            | 機能                  |
|------------------|---------------------|
| getInstance      | 位相地図クラスのインスタンスを取得   |
| getGraphMapInfo  | 位相地図 ID で位相地図情報を取得  |
| setGraphMapInfo  | 位相地図 ID で位相地図情報を更新  |
| getTransformInfo | 座標 ID で座標系パラメタ情報を取得 |
| setTransformInfo | 座標 ID で座標系パラメタ情報を更新 |

## 4 使用例

本章では、TMS を用いた環境地図の使用例について説明する。特に、格納する具体的なデータを例を用いて示す。

### 4.1 地図データの分割

建物・フロアなど広大な空間を 1 つの定量地図データにすると、データサイズが大きくなってしまふ。ロボットのメモリや無線通信の伝送速度の制限から、地図データは出来る限り小さいことが望ましい。そこで、まず手動で地図データの分割を行う。

今回の実験では、建物のフロア (40m × 80m) のデータを用いた。図 1 に分割の例を示す。枠線で囲む領域を、1 つの定量地図データとして登録する。全体を一つの地図データとした時、データサイズはバイナリ形式で 4.57MB であった。これを 19 分割した時、定量地図データのサイズは、バ

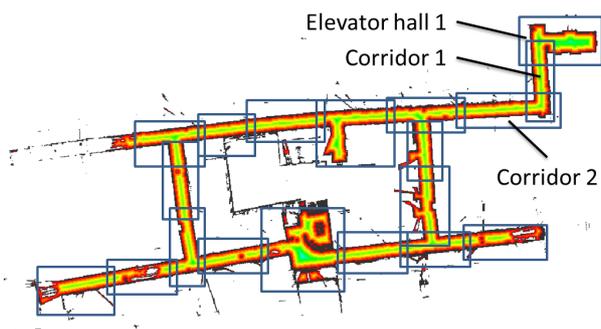


Fig.1 地図データの分割

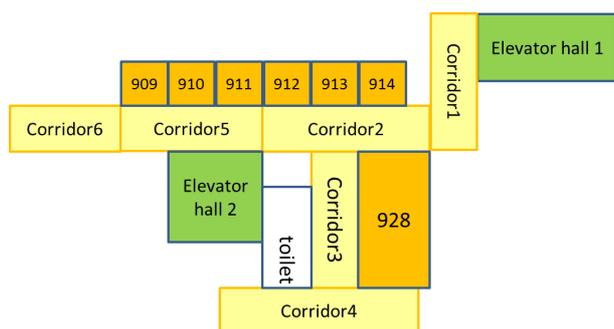


Fig.2 定量地図

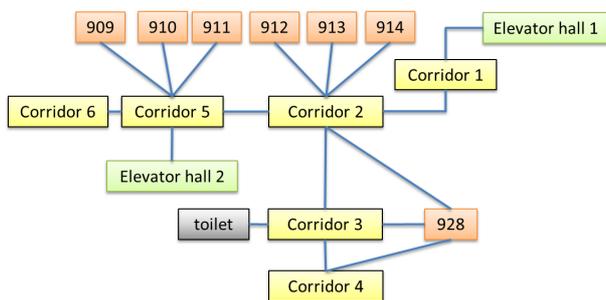


Fig.3 位相地図

イナリ形式で一つ当たり 130~670KB であった。

#### 4.2 データベース登録

分割した地図を定量地図として登録する方法について述べる。例として図 2 のような, Elevator hall・Corridor・Room・Toilet の空間から成る場合を考える。各空間は接続関係があり, 例えば”Elevator hall 1”は, ”Corridor 1”と繋がっている。位相地図は, このような空間の繋がりを図 3 のように表現したものである。経路設計には, 空間の連結関係情報が必要であり, この情報は位相地図の linked graphmapid, および hgraphmapid に格納される。また隣接する位相地図への接続位置 (リンク位置) は, x(yz) graphmapid に格納されている。表 5, 表 6 に”Elevator hall 1”の定量地図と位相地図の例を示す。なお, 以下の例では位相地図と定量地図は一対一に対応しているが, 実際には複数の grid\_mapid を持つことで, 一つに位相地図に複数の定量地図が定義されていても構わない。

Table 5 Metric map class (Elevator hall 1)

| 項目名           | 項目内容                      |
|---------------|---------------------------|
| mapid         | Elevator hall 1-01        |
| coordid       | 9floor-coord              |
| scale         | 1                         |
| type          | 2 (0:研究室,1:廊下,2:エレベータホール) |
| coord_x(yz)1  | 0(0,0)                    |
| coord_x(yz)2  | 5000(5000,0)              |
| content       | Multi-layer DEM           |
| grid_x_width  | 100 (mm)                  |
| grid_y_width  | 100 (mm)                  |
| grid_x_counts | 50                        |
| grid_y_counts | 50                        |
| datasize      | 4 (byte)                  |
| graphmapid    | Elevator hall 1           |

Table 6 Topological map class (Elevator hall 1)

| 項目名               | 項目内容               |
|-------------------|--------------------|
| graphmapid        | Elevator hall 1    |
| type              | 部屋 (建物, フロア, 部屋)   |
| status            | 0 (0:立入り可,1:立入り不可) |
| hgraphmapid       | 9Floor             |
| coordid           | 9floor_cood        |
| grid_mapid        | Elevator hall 1-01 |
| linked_graphmapid | Corridor 1         |
| x(yz)_graphmapid  | 0(1000,0)          |

#### 4.3 TMS とロボットの連携

TMS を用いたロボットの移動の実行例を示す。以下に処理の流れを示す。

1. 位相地図を用いた大域経路設計
2. TMS から現在地を含む定量地図を取得【図 4(a),(b)】
3. 定量地図を用いた局所経路設計【図 4(c)】
4. ロボットの移動【図 4(d)】
5. 次の定量地図を TMS から取得【図 4(e)】
6. 目的地に到着するまで 2~6 を繰り返す【図 4(f)】

各処理についての詳細な説明を以下に示す。

##### 4.3.1 位相地図を用いた大域経路設計

ロボットの現在地と目的地の位置情報から, 位相地図を用いて経路設計を行う。現在地は, ロボット搭載センサや外部のセンサから取得し, TMS に格納されている。目的地

Table 7 Object class

| 項目名         | 項目内容         |
|-------------|--------------|
| tagid       | タグ ID        |
| type        | 収納物種別        |
| name        | 名前           |
| coord_x(yz) | 位置情報座標 x(yz) |

は、ロボットに物品を取得させる場合、物品位置が目的地となる。TMS の物品クラスの一部を表 7 に示す。この探索には位相地図を用い、グラフ探索により目的の位相地図の ID を探索する。表 6 の linked\_graphmapid には、接続している位相地図の ID が、また x(yz)\_graphmapid には次の位相地図の接続部の座標が格納されている。目的地の位相地図が見つかるまでリンクを辿り、経路を見つける。

例) Elevatorhall 1 Corridor 1 Corridor 2 Room 5

#### 4.3.2 TMS から現在地を含む定量地図を取得【図 4(a),(b)】

ロボットの実際の移動には、ロボットの現在地を含んだ定量地図を TMS から取得する必要がある。そこで、ロボットの現在地と、定量地図の範囲を示す表 5 の coord\_x(yz)1,2 の座標を比較する。この座標内に現在位置が含まれていれば、その定量地図の Multi-layer DEM を取得する。もし、取得した地図に次の位相地図の接続部が含まれない場合、接続部が含まれるまで同じ位相地図内の周囲の定量地図を取得する。

#### 4.3.3 定量地図を用いた局所経路設計【図 4(c)】

実際にロボットが移動する経路を作成する。経路は、ロボットの現在地から次の位相地図の接続部までである。Elevatorhall 1 から Corridor 1 に移動する場合、表 6 の linked\_graphmapid と x(yz)\_graphmapid を参照することで Corridor 1 の連結座標が分かる。

#### 4.3.4 ロボットの移動【図 4(d)】

局所経路設計で作成した経路に沿い、TMS から他の移動体情報等も用いて、ロボットを移動させる。

#### 4.3.5 次の定量地図を TMS から取得【図 4(e)】

空間の連結座標まで到着すると、隣の位相地図の情報を読み込む。表 6 の linked\_graphmapid を参照し、隣の位相地図の ID を取得する。この位相地図 ID を用いて、処理手順 2~5 を同様に実行する。

## 5 まとめ

本報告では、我々がこれまでに開発してきたタウンマネジメントシステム (Town Management System, TMS) に対し、特に環境地図の管理伝達機能の強化を目指した、新たな

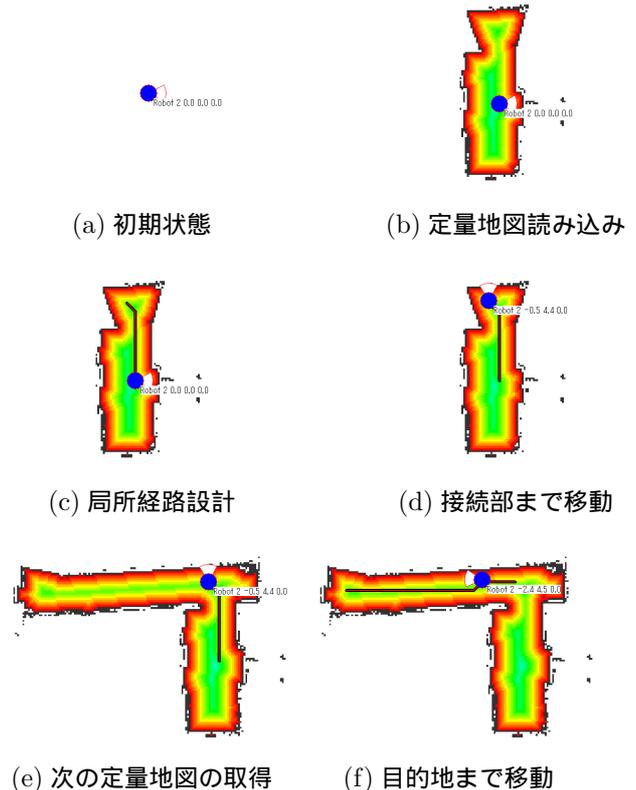


Fig.4 処理の流れ

サービスロボット用地図情報管理システムを紹介した。定量地図と位相地図を TMS のデータベースに格納し、それらを結びつけてロボットに提供する機能を実装した。

謝辞 本研究の一部は文部科学省科学研究費補助金 基盤研究 (B) (課題番号 23360115) の支援を受けた。

## 文献

- [1] 長谷川, 環境プラットフォーム「ロボットタウン」ム, 日本ロボット学会誌, Vol.26 No.05, pp.23-26, 2008.
- [2] 長谷川, 人間共生広域環境の情報構造化, 計測自動制御学会誌 計測と制御, Vol.49, No.6, pp.361-366, 2010.
- [3] 倉爪 亮, ロボット共生環境のための環境情報計測技術, 計測自動制御学会誌 計測と制御, Vol.49, No.6, pp.348-353, 2010.
- [4] 村上, 長谷川, 木室, 千田, 家永, 有田, 倉爪:”情報構造化環境における情報管理の一手法”, 日本ロボット学会誌, vol.26, no.2, pp.192-199,2008.
- [5] 倉爪, 村上, 木室, 家永, 馬場 殷, ロボットタウンの共通プラットフォーム技術のメカニズム, 日本ロボット学会誌, Vol.26 No.05, pp.27-31, 2008.
- [6] 倉爪, 戸畑, 村上, 長谷川:”Cps-slam の研究-大規模建造物の高精度 3 次元幾何形状レーザ計測システム-”, 日本ロボット学会誌, vol.25, no.8, pp.1234-1242, 2007.