

機能仕様書
タウンマネジメントシステム RTC

V e r . 0 . 4 0

2 0 1 1 年 1 2 月 1 日

九州大学

改版履歷

[illegible]

目次

改版履歴	i
目次	ii
1. はじめに	1
1. 1. 本書の適用範囲	1
1. 2. 関連文書	1
1. 3. 本書を読むにあたって	1
2. 機能仕様	2
2. 1. 機能概要	2
2. 2. モジュール構成	2
3. RTC 仕様	3
3. 1. TmsRtProvider (タウンマネジメントシステム RTC)	3
3. 1. 1. 機能概要	3
3. 1. 2. 動作環境	3
3. 1. 3. ポート情報	4
3. 1. 4. サービスポート I/F 仕様	5
3. 2. TmsRtConsumer (タウンマネジメントシステム RTC 動作確認用コンポーネント)	26
3. 2. 1. 機能概要	26
3. 2. 2. 動作環境	26
3. 2. 3. ポート情報	27
3. 2. 4. コンフィグレーション情報	28
4. 特記事項	29

1. はじめに

1. 1. 本書の適用範囲

本書はロボット向けミドルウェア OpenRTM 上で、環境情報の管理機構であるタウンマネジメントシステムの機能を利用するためのタウンマネジメントシステム RTC の仕様について記述した文書である。

1. 2. 関連文書

本書の記述する文書について記述する。

表 1-1 関連文書

No.	文書名	備考
1	タウンマネジメントシステム RTC 操作手順書	タウンマネジメントシステム RTC の操作手順について記載

1. 3. 本書を読むにあたって

本書は RT ミドルウェア、RT コンポーネント(以下、RTC)、およびタウンマネジメントシステム(以下、TMS)に関する基本知識を備えた利用者を対象としている。RT ミドルウェア、RTC、TMS については下記を参照のこと。

OpenRTM-aist Official Website:

<http://www.openrtm.org/>

タウンマネジメントシステム Website:

http://openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID121

2. 機能仕様

2. 1. 機能概要

本知能モジュールは、タウンマネジメントシステムが提供し環境情報の登録・更新・検索などを行う API を OpenRTM 上で実行する。

2. 2. モジュール構成

本知能モジュールは、

- ① TmsRtProvider
- ② TmsRtConsumer

で構成される。モジュール構成を図 1 に示す。各モジュールが実行される環境においてそれぞれ TMS Server と TMS API がインストールされている必要がある。なお、TmsRtConsumer は環境に応じて置き換えられるモジュールである。

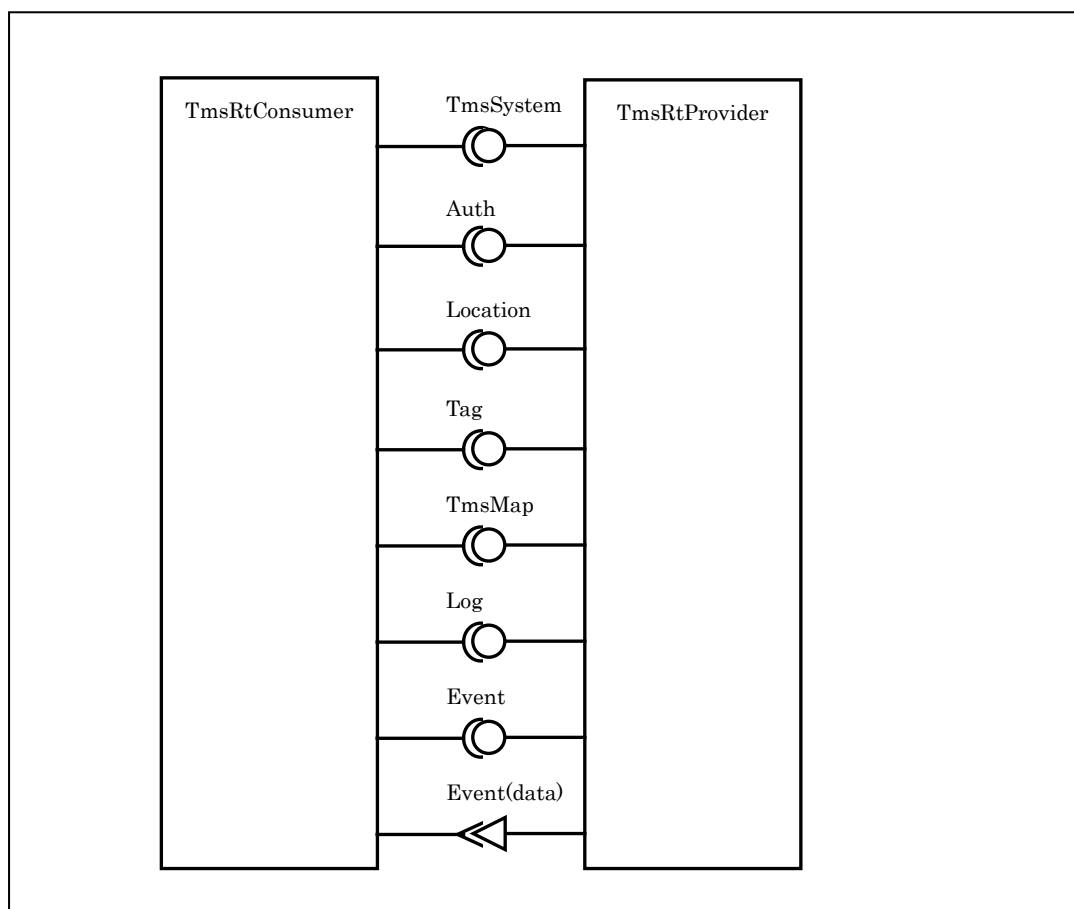


図 1 モジュール構成図

3. RTC 仕様

3. 1. TmsRtProvider（タウンマネジメントシステム RTC）

3. 1. 1. 機能概要

本コンポーネントは、TMS Server と合わせて用いられる。サービスポートで指定された API を TMS Server で実行する。

3. 1. 2. 動作環境

本知能モジュールの動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	WindowsXP
開発言語	JAVA
コンパイラ	JDK 1.6
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0（JAVA 版）
依存パッケージ	TMS Server

3. 1. 3. ポート情報

A) サービスポート (Provider)

サービス名	インターフェース名	説明
TmsSystem	TmsSystem	初期化に関する命令を提供
Auth	Auth	認証に関する命令を提供
Location	Location	位置情報に関する命令を提供
Tag	Tag	RFID タグに関する命令を提供
TmsMap	TmsMap	地図に関する命令を提供
Log	Log	ログに関する命令を提供
Event	Event	イベントに関する命令を提供

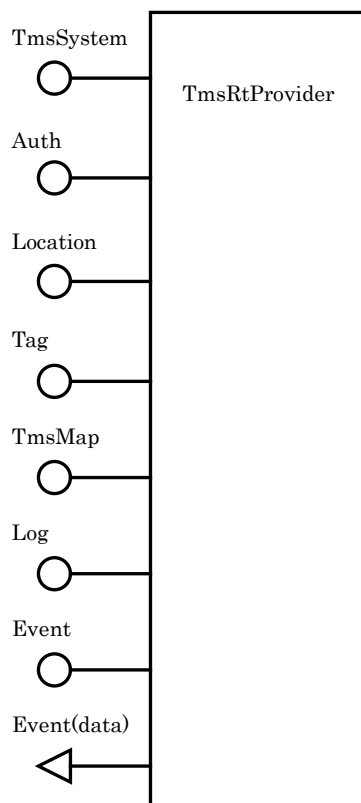


図 3-1 TmsRtProvider コンポーネント

3. 1. 4. サービスポート I/F 仕様

3. 1. 4. 1. TmsSystem

関数名	initialize			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	API の初期化処理を行う			

関数名	destroy			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	API の終了処理を行う			

関数名	isInitialized			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	true			初期化されている
	false			初期化されていない
説明	API が初期化されているかを確認する			

3. 1. 4. 2. Auth

関数名	getInstance			
引数	名称	型	I/O	説明
	deviceid	String	入力	デバイス ID
	名称	型	I/O	説明
	userid	String	入力	ユーザ ID
	名称	型	I/O	説明
	password	String	入力	パスワード
戻り値	値			説明
	Auth			インスタンス
説明	Auth クラスのインスタンスを取得する			

関数名	login			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	Token			ログイン管理用の識別子
説明	ユーザ認証を行い TMS へ接続する			

関数名	logout			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	ユーザの開放を行い TMS との接続を終了する			

3. 1. 4. 3. Location

関数名	getInstance			
引数	名称	型	I/O	説明
	token	Auth.Token	入力	ログイン時に取得した認証用の識別子
戻り値	値			説明
	インスタンス			Location クラスのインスタンス
説明	Location クラスのインスタンスを取得する			

位置情報：

```
class LocationInfo {
    double coordinateX; // 座標 X
    double coordinateY; // 座標 Y
    double coordinateZ; // 座標 Z
    double orientP; // 向き P
    double orientR; // 向き R
    double orientY; // 向き Y
    double velocityX; // 速度 X
    double velocityY; // 速度 Y
    double velocityZ; // 速度 Z
    string additionalInfo; // 追加情報
    string time; // 計測時刻
};
```

ビジョン情報（移動体情報）：

```
class VisionInfo : public LocationInfo {
    int property; // 属性
    int status; // 状態
    BinaryInfo image; // 画像
    string name; // 名前
    string action; // 動作
    string place; // 場所
};
```

関数名	getLocationInfoByVisionID			
引数	名称	型	I/O	説明
	visionid	String	入力	移動体 ID
	名称	型	I/O	説明
	info	LocationInfo	出力	移動体位置情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した移動体 ID の位置を取得する			

関数名	getLocationInfoByRobotID			
引数	名称	型	I/O	説明
	robotid	String	入力	ロボット ID
	名称	型	I/O	説明
	info	LocationInfo	出力	移動体位置情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定したロボット ID の位置を取得する			

関数名	getVisionInfoByVisionID			
引数	名称	型	I/O	説明
	visionid	String	入力	移動体 ID
	名称	型	I/O	説明
	info	VisionInfo	出力	ビジョン情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した移動体 ID の情報を取得する			

関数名	setLocationInfoByRobotID			
引数	名称	型	I/O	説明
	robotid	String	入力	ロボット ID
	名称	型	I/O	説明
	info	LocationInfo	入力	位置情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定したロボットの位置情報を TMS へ送信する			

関数名	getRobotIDByVisionID			
引数	名称	型	I/O	説明
	visionid	String	入力	移動体 ID
	名称	型	I/O	説明
	robotid	String	出力	ロボット ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	移動体 ID よりロボット ID を取得する			

関数名	getVisionIDByRobotID			
引数	名称	型	I/O	説明
	robotid	String	入力	ロボット ID
	名称	型	I/O	説明
	visionid	String	出力	移動体 ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	ロボット ID より移動体 ID を取得する			

関数名	getVisionIDByVisionInfo			
引数	名称	型	I/O	説明
	infos	VisionInfo_Key	入力	ビジョン情報検索キー
	名称	型	I/O	説明
	visionids	String	出力	移動体 ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	ビジョン情報より移動体 ID を取得する			

関数名	setVisionInfoByVisionID			
引数	名称	型	I/O	説明
	infos	VisionInfo	入力	移動体 ID、ビジョン情報（複数設定可）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した移動体 ID の情報を更新する			

関数名	getSurroundingInfo			
引数	名称	型	I/O	説明
	radius	double	入力	範囲を半径で指定
	名称	型	I/O	説明
	infos	LocationInfo	出力	移動体 ID、ビジョン情報（複数有り）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定範囲内の移動体情報を取得する			

関数名	createVisionID			
引数	名称	型	I/O	説明
	info	VisionInfo	入力	ビジョン情報
	名称	型	I/O	説明
	enableExistCheck	boolean	入力	二重登録チェックフラグ true=二重チェックする false=二重チェックしない
	名称	型	I/O	説明
	visionid	String	出力	新規移動体 ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	移動体 ID を新規に発行する			

関数名	deleteVisionID			
引数	名称	型	I/O	説明
	visionid	String	入力	開放する移動体 ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	移動体 ID を開放する			

3. 1. 4. 4. Tag

関数名	getInstance			
引数	名称	型	I/O	説明
	token	Auth.Token	入力	ログイン時に取得した認証用の識別子
戻り値	値			説明
	インスタンス			Tag クラスのインスタンス
説明	Tag クラスのインスタンスを取得する			

タグ情報：

```
class TagInfo {
    TagObjectInfo* pObj; // 汎用部品情報
    TagLocationInfo* pLoc; // 汎用位置情報
    TagTaskInfo* pTask; // 汎用タスク情報
};
```

タグ汎用位置情報：

```
class TagLocationInfo : public LocationInfo
{
};
```

タグ汎用品物情報：

```
class TagObjectInfo {
    string name; // 名前
    int color; // 色
    string size; // サイズ
};
```

タグ汎用タスク情報：

```
class TagTaskInfo {
    string task; // タスク内容
};
```

収納物情報（タグ用）：

```
class ContentObjectInfo : public TagObjectInfo {
    int type; // 種別
    string manufacturer; // メーカー
    string registration; // 登録日（使用開始日）
    string comment; // 説明
    double width; // 幅
    double depth; // 奥行き
    double height; // 高さ
    string history; // 使用履歴（最終使用日）
    int status; // 状態（使用可／不可）
    string schedule; // 使用可となる予定・実績
    int cabinetType; // 収納棚種別
    string cabinetId; // 収納棚 ID
};
```

収納棚情報（タグ用）：

```
class CabinetObjectInfo : public TagObjectInfo {
    int cabinetType; // 収納棚種別
    string manufacturer; // メーカー
    string registration; // 登録日（使用開始日）
    string additionalInfo; // 追加情報
    double width; // 幅
    double depth; // 奥行き
    double height; // 高さ
    int capacity; // 容量（個数、枚数）
    int space; // 空き容量（個数、枚数）
    double sens_start_x; // センシング開始位置（座標 X）
    double sens_start_y; // センシング開始位置（座標 Y）
    double sens_start_z; // センシング開始位置（座標 Z）
    double sens_width; // センシング幅
    vector<string> contentIds; // 収納物 ID 一覧
};
```

関数名	getTagInfo			
引数	名称	型	I/O	説明
	tagids	string	入力	タグ ID （複数設定可）
	名称	型	I/O	説明
	taginfos	TagInfo	出力	品物情報、位置情報、タスク情報（複数有り）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定したタグ ID の物品情報を取得する			

関数名	setTagInfo			
引数	名称	型	I/O	説明
	taginfos	Taginfo	入力	タグ ID、品物情報、位置情報、タスク情報（複数設定可）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定したタグ ID の物品情報を更新する			

関数名	getTagIDByObjectInfo			
引数	名称	型	I/O	説明
	infos	TagObjectInfo	入力	検索キー（品物情報）情報（複数設定可）
	名称	型	I/O	説明
	tagids	string	出力	タグ ID （複数有り）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した品物情報を持つタグ ID を取得する			

関数名	getTagIDByObjectInfoWithBM			
引数	名称	型	I/O	説明
	infos	TagObjectInfo_Key	入力	検索キー（品物情報）情報（複数設定可）
	名称	型	I/O	説明
	flag	int	入力	0=複数条件 AND 検索、1=複数条件 OR 検索
	名称	型	I/O	説明
	tagids	string	出力	タグ ID （複数有り）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	品物情報より部分一致検索でタグ ID を取得する			

関数名	getTagIDByLocationInfo			
引数	名称	型	I/O	説明
	infos	TagLocationInfo	入力	検索キー（位置情報）情報（複数設定可）
	名称	型	I/O	説明
	tagids	string	出力	タグ ID （複数有り）
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した品物情報を持つタグ ID を取得する			

物体形状情報

```
class ObjectShapeInfo {
    int object_id;
    mediumblob shape;    // 物体形状
};
```

関数名	getObjectShapeInfo			
引数	名称	型	I/O	説明
	tagid	string	入力	タグ ID
	名称	型	I/O	説明
	info	ObjectShapeInfo	出力	物体形状情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	物体形状情報を取得する			

物体テクスチャ情報

```
class ObjectTextureInfo {
    int object_id;
    medumblob images;
};
```

関数名	getObjectTextureInfo			
引数	名称	型	I/O	説明
	tagid	string	入力	タグ ID
	名称	型	I/O	説明
	info	ObjectTextureInfo	出力	物体テクスチャ情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	物体テクスチャ情報を取得する			

物体 JAN コード情報

```
class ObjectJanInfo {
    string jan;           // JAN コード
    string maker_kanji;   // ブランド名(漢字)
    string maker_kana;    // ブランド名(カナ)
    string name_kanji;    // 商品名(漢字)
    string name_kana;     // 商品名(カナ)
    string type_kanji;    // 規格／型番(漢字)
    string type_kana;     // 規格／型番(カナ)
    string comment;       // 簡易商品説明詳細商品説明
```

```

double price;           // 価格
double weight;          // 単品重量
int class;              // 商品区分
int ex_im;              // 国産 or 輸入
datetime sale_start_date; // 販売開始日
double width;           // 商品サイズ(幅)
double height;          // 商品サイズ(高さ)
double depth;           // 商品サイズ(奥行き)
string unit;            // 内容量単位
datetime term_validity; // 有効期間（賞味期間）
int nation; // 原産国コード
};

```

関数名	getObjectJanInfoByTagID			
引数	名称	型	I/O	説明
	tagid	string	入力	タグ ID
	名称	型	I/O	説明
	info	ObjectJanInfo	出力	JAN コード物品情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定したタグ ID に関連付けられた JAN コード関連情報を取得する			

関数名	getTagIDByObjectJanInfo			
引数	名称	型	I/O	説明
	infos	ObjectJanInfo	入力	検索キー（JAN コード物品情報）情報
	名称	型	I/O	説明
	tagid	string	出力	タグ ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定した JAN コード関連情報を持つタグ ID を取得する			

関数名	getTagIDByObjectJanInfoWithBM			
引数	名称	型	I/O	説明
	infos	ObjectJanInfo_Key	入力	検索キー (JAN コード物品情報) 情報
	名称	型	I/O	説明
	flag	int	入力	0=複数条件 AND 検索、1=複数条件 OR 検索
	名称	型	I/O	説明
	tagid	string	出力	タグ ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	JAN コード物品情報より部分一致検索でタグ ID を取得する			

3. 1. 4. 5. TmsMap

関数名	getInstance			
引数	名称	型	I/O	説明
	token	Auth.Token	入力	ログイン時に取得した認証用の識別子
戻り値	値			説明
	インスタンス			TmsMap クラスのインスタンス
説明	TmsMap クラスのインスタンスを取得する			

関数名	getMapInfo			
引数	名称	型	I/O	説明
	mapid	String	入力	地図 ID
	名称	型	I/O	説明
	info	MapInfo	出力	地図情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	地図の詳細情報を取得する			

地図情報：

```
class MapInfo {
    int    mid;           // 内部管理 ID
    string mapid;         // 地図 ID
    int    coordid;       // 座標系 ID
    double scale;         // 縮尺
    int    type;          // 種類
    double startX;        // 地図左上起点 x 座標
    double startY;        // 地図左上起点 y 座標
    double startZ;        // 地図左上起点 z 座標
    double endX;          // 地図右下終点 x 座標
    double endY;          // 地図右下終点 y 座標
    double endZ;          // 地図右下終点 z 座標
    byte[] content;       // バイナリデータ(サイズは最大 16M バイト)
    double grid_X_width;  // グリッド x 幅
    double grid_Y_width;  // グリッド y 幅
    int    grid_X_counts; // x 方向のグリッド数
}
```

```
int    grid_Y_counts; // y 方向のグリッド数
int    datasize;      // グリッドひとつあたりのデータサイズ
string nodeid;        // ノード ID(グラフ地図 ID)
};
```

関数名	getMapList			
引数	名称	型	I/O	説明
	infos	MapInfo	出力	地図 ID (複数)
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	地図一覧の取得			

関数名	setMapInfo			
引数	名称	型	I/O	説明
	info	MapInfo	入力	地図情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	地図の詳細情報を TMS へ送信する			

3. 1. 4. 6. Log

関数名	getInstance			
引数	名称	型	I/O	説明
	token	Auth.Token	入力	ログイン時に取得した認証用の識別子
戻り値	値			説明
	インスタンス			Log クラスのインスタンス
説明	Log クラスのインスタンスを取得する			

関数名	setActionInfo			
引数	名称	型	I/O	説明
	rid	String	入力	ロボット ID
	名称	型	I/O	説明
	info	ActionInfo	入力	作業情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定するロボット ID の作業内容を追加する			

作業情報：

```
class ActionInfo {
    string      action;    //作業内容
    Date        time;      //日時
    LocationInfo location; //作業位置
    string      additional; //附加情報
};
```

関数名	getActionInfo			
引数	名称	型	I/O	説明
	keyinfos	LogActionInfo_Key	入力	検索条件（複数指定可）
	名称	型	I/O	説明
	num	int	入力	検索件数
	名称	型	I/O	説明
	rids	String	入力	ロボット ID(複数指定可)
	名称	型	I/O	説明
	infos	ActionInfo	出力	作業情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	ロボット ID、作業内容、作業日時、作業位置を検索キーとして、 検索条件を満たす作業情報を指定した件数だけ取得する			

3. 1. 4. 7. Event

関数名	getInstance			
引数	名称	型	I/O	説明
	token	Auth.Token	入力	ログイン時に取得した認証用の識別子
戻り値	値			説明
	インスタンス			Event クラスのインスタンス
説明	Event クラスのインスタンスを取得する			

関数名	getEventList			
引数	名称	型	I/O	説明
	infos	EventInfo	出力	イベント一覧
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	イベント一覧の取得			

イベント情報：

```
class EventInfo {
    int    id;        // イベント ID
    string comment;   // 備考
    int    status;    // 状態
};
```

関数名	notifyEvent			
引数	名称	型	I/O	説明
	eventid	int	入力	イベント ID
	名称	型	I/O	説明
	content	String	入力	通知情報
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	イベントを購読者に通知する			

関数名	registerEvent			
引数	名称	型	I/O	説明
	eventid	int	入力	イベント ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	イベントを登録する			

関数名	subscribeEvent			
引数	名称	型	I/O	説明
	eventid	int	入力	イベント ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定イベントを購読する ※イベントは NotifyEventType 型でデータポートに通知される			

通知イベント情報：

```
struct NotifyEventType {
    string  eventid;    // イベント ID
    string  deviceid;   // イベント発行者のデバイス ID
    string  userid;     // イベント発行者のユーザ ID
    string  content;    // 通知情報
};
```

関数名	unregisterEvent			
引数	名称	型	I/O	説明
	eventid	int	入力	イベント ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	イベントの登録を取り消す			

関数名	unsubscribeEvent			
引数	名称	型	I/O	説明
	eventid	int	入力	イベント ID
戻り値	値			説明
	0			正常終了
	0 以外			エラー
説明	指定イベントの購読を取り消す			

3. 2. TmsRtConsumer（タウンマネジメントシステム RTC 動作確認用コンポーネント）

3. 2. 1. 機能概要

タウンマネジメントシステム RTC の動作確認を行うためのサンプルモジュールであり、TMS API と合わせて用いられる。実行する API をサービスポートで指定する。

3. 2. 2. 動作環境

本知能モジュールの動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	WindowsXP
開発言語	JAVA
コンパイラ	JDK 1.6
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE（JAVA 版）
依存パッケージ	TMS API

3. 2. 3. ポート情報

A) サービスポート (Consumer)

サービス名	インターフェース名	説明
TmsSystem	TmsSystem	初期化に関する命令を実行
Auth	Auth	認証に関する命令を実行
Location	Location	位置情報に関する命令を実行
Tag	Tag	RFID タグに関する命令を実行
TmsMap	TmsMap	地図に関する命令を実行
Log	Log	ログに関する命令を実行
Event	Event	イベントに関する命令を実行

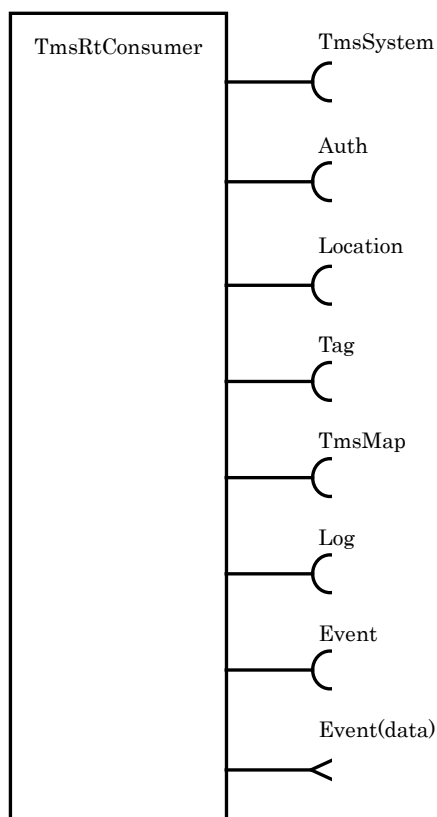


図 3-2 TmsRtConsumer コンポーネント

3. 2. 4. コンフィグレーション情報

TmsRtConsumer がタウンマネジメントシステムへログインする際のデバイス ID、ユーザ ID、パスワードを設定する。ログインするためには設定した ID がタウンマネジメントシステムに登録されている必要がある。

名称	型	デフォルト値	説明
deviceid	string	robot1	デバイス ID
userid	string	user1	ユーザ ID
password	string	abc	パスワード

4. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

九州大学大学院システム情報科学研究院情報知能工学部門 長谷川研究室

〒819-0395 福岡県福岡市西区元岡 744 ウエスト 2 号館 912 室

Tel : 092-802-3598

Fax : 092-802-3607

E-Mail : tms@irvs.ait.kyushu-u.ac.jp