

LocalMap コンポーネント 取扱説明書

豊橋技術科学大学 行動知能システム学研究室

平成 23 年 7 月 19 日

目 次

1	初めに	2
1.1	ファイルの展開	2
1.2	開発・動作環境	2
2	LocalMap コンポーネントについて	3
3	LocalMapView コンポーネントについて	4
4	各データ型・インターフェースについて	4
4.1	SensorRTC::LaserRangeSensor::idl::MeasuredData	4
4.2	SensorRTC::LaserRangeSensor::idl::TimedMeasuredData	5
4.3	IIS::TimedPose2D	5
4.4	RTC::OGMapConfig	5
4.5	MRFC::TimedRelativeOGMapData	6
4.6	MRFC::TimedFloatRelativeOGMapData	6
4.7	MRFC::RelativeMapService	6
5	コンポーネントの実行手順	7
5.1	各プログラムの起動	7
5.2	RT System Editor 上でのコンポーネントの接続	8
5.3	コンフィギュレーションの設定	9
5.4	コンポーネントの実行と動作の確認	9
6	連絡先	10

1 初めに

このドキュメントでは、移動ロボットがレーザーレンジセンサの距離データとロボットの移動量を基に局所地図を生成するための RT コンポーネントである LocalMap コンポーネントおよび局所地図を表示するための LocalMapViewComp コンポーネントについて解説し、その使い方を説明します。

1.1 ファイルの展開

ファイルの中身は図 1 のようになっています。

- IIS2.idl および MRFC.idl
このコンポーネント群独自のデータ型・サービスを定義した IDL ファイル。
- rtc.conf
- LocalMapComp.exe
局所地図を生成するコンポーネントの実行ファイル。
- LocalMapViewComp.exe
局所地図を表示するコンポーネントの実行ファイル。

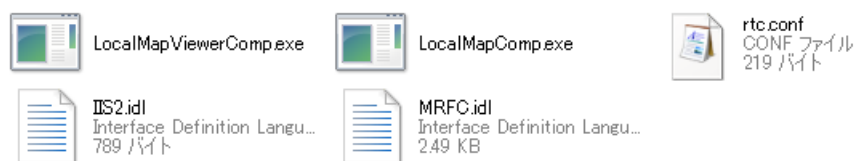


図 1: 解凍したフォルダの中身

1.2 開発・動作環境

各コンポーネントは以下の環境で開発し、動作確認を行っています。

- Windows XP Pro SP3
- Open-rtm-aist 1.0.0(C++版)
- Visual studio 2008

また、LocalMapViewComp コンポーネントは OpenCV 2.1 を使用しています。OpenCV については下記サイトを参照して下さい。

<http://sourceforge.net/projects/opencvlibrary/>

2 LocalMap コンポーネントについて

このコンポーネントは，データポートから入力された距離データとロボットの移動量を用いてロボットの周囲の障害物存在確率地図を生成します．LocalMap コンポーネントが持つデータポートの一覧を表 1 に，サービスポートの一覧を表 2 に，コンフィギュレーションの一覧を表 3 に示します．

表 1: 入出力データポート

Port Type	Data Type	Port Name	備考
In Port	SensorRTC::LaserRangeSensor::idl::idl::TimedMeasuredData	rangedata	LRF からのデータ入力
In Port	IIS::TimedPose2D	pose	ロボット移動量入力

表 2: サービスポート

Port Type	Data Type	Port Name	備考
Service Provider	MRFC::RelativeMapService	RelativeMapService	ロボット座標系地図の取得

表 3: コンフィギュレーション

型	変数名	単位	備考
int	LocalMapWidth	cell	地図の X 方向の大きさ
int	LocalMapHeight	cell	地図の Y 方向の大きさ
double	LocalMapScale	m/cell	1 グリッドのサイズ
double	SensorPositionX_m	m	センサ取り付け位置
double	SensorPositionY_m	m	センサ取り付け位置
double	SensorPositionTheta_deg	degree	センサ取り付け角度
double	SensorMaxDistance_mm	mm	センサの最大測定距離
double	SensorMinDistance_mm	mm	センサの最小測定距離
long	Error_code	-	対象が遠すぎて値が取得できなかった際のデータ値 classicURG の場合は 19 , TopURG の場合は 1 とする .
long	Error_mode	-	値に応じて URG のエラーコードの範囲を決定する . classicURG の場合は 1 , TopURG の場合は 2 とする .

3 LocalMapView コンポーネントについて

このコンポーネントは、生成された局所地図を表示するために用います。表示する必要がない場合には必ずしも必要ではありません。LocalMapView コンポーネントが持つデータポートの一覧を表 4 に、サービスポートの一覧を表 5 に、コンフィギュレーションの一覧を表 6 に示します。

表 4: 入出力データポート

Port Type	Data Type	Port Name	備考
In Port	MRFC::TimedPeopleTrackingData	peopledata	人物データの取得（不使用）

表 5: サービスポート

Port Type	Data Type	Port Name	備考
Service Consumer	MRFC::RelativeMapService	RelativeMapService	ロボット座標系地図の取得

表 6: コンフィギュレーション

型	変数名	単位	備考
int	color_reverse	-	値が 0 のときは障害物の存在確率が高いセルを白で表示する 値が 0 以外のときは障害物の存在確率が高いセルを黒で表示する
int	robot_radius	pixel	ロボット位置（原点）を示す円の表示サイズ
int	person_radius	pixel	人を示す円の表示サイズ

4 各データ型・インターフェースについて

ここでは LocalMap コンポーネントおよび LocalMapView コンポーネントで使用されている各データ型・インターフェースについて説明します。

4.1 SensorRTC::LaserRangeSensor::idl::MeasuredData

SensorRTC::LaserRangeSensor::idl::MeasuredData は株式会社セックが開発した北陽電機社 URG シリーズ用のコンポーネントで使用されているデータ型です。レーザ距離センサから距離データを取得するために用いられます。

- float startPosition: distance に最初に格納されているデータの方向 [degree]
- float endPosition: distance に最後に格納されているデータの方向 [degree]

- long scanInterval: スキャン間引き数
- long dataGroupingNumber: まとめるステップ数
- sequence<long> distance: 各方向に対する距離データ [mm]
- float dataInterval: 各データの間の間隔 [degree]
- string sensorState: センサの状態 (例 : "NORMAL", "UPDATED")

4.2 SensorRTC::LaserRangeSensor::idl::TimedMeasuredData

SensorRTC::LaserRangeSensor::idl::TimedMeasuredData は株式会社セックが開発した北陽電機社 URG シリーズ用のコンポーネントで使用されているデータ型です。レーザ距離センサから距離データを取得するために用いられます。

- MeasuredData data: 取得したデータ
- RTC::Time tm: タイムスタンプ

4.3 IIS::TimedPose2D

IIS::TimedPose2D はロボットの位置・姿勢を格納するデータ型です。

- Pose2D data: x (位置の X 座標), y (位置の Y 座標), heading (姿勢) を格納。
- error: 誤差分散を格納する。ただし, LocalMap コンポーネントではこの値は使用していない。
- id: 使用しない
- RTC::Time tm: タイムスタンプ

4.4 RTC::OGMapConfig

大きさやスケールといった地図の情報を格納するデータ型です。

- double xScale: X 軸方向の地図のスケール [m/cell]
- double yScale: Y 軸方向の地図のスケール [m/cell]
- double width: X 軸方向の地図の大きさ [cell]
- double height: Y 軸方向の地図の大きさ [cell]
- RTC::Pose2D origin: ロボット中心から見た cell(0,0) の絶対座標

4.5 MRFC::TimedRelativeOGMapData

地図を octed 型の系列で表現したデータ型です．ここで，octed 型は 8bit の符号付整数 (-128 ~ 127) であり，各セル毎の障害物の存在確率を 0 から 100 の値で格納しています．また，そのセルが未観測の場合（未知領域の場合）は-1 が格納されます．セルの並びといった地図の仕様については図 2 も参照してください．

- RTC::OGMapConfig mapconfig: 地図の大きさやスケール
- RTC::OGMapCells cells: octed 型の系列，各セルの値を格納
- RTC::Pose2D pose: ロボット中心の絶対座標
- RTC::Time tm: タイムスタンプ

4.6 MRFC::TimedFloatRelativeOGMapData

地図を float 型の系列で表現したデータ型です．ここで，float 型は単精度浮動少数であり，各セル毎の障害物の存在確率を 0.0 から 1.0 の値で格納しています．また，そのセルが未観測の場合（未知領域の場合）は負の値が格納されます．セルの並びといった地図の仕様については図 2 も参照してください．

- RTC::OGMapConfig mapconfig: 地図の大きさやスケール
- RTC::OGMapFloatCells cells: float 型の系列，各セルの値を格納
- RTC::Pose2D pose: ロボット中心の絶対座標
- RTC::Time tm: タイムスタンプ

4.7 MRFC::RelativeMapService

MRFC::RelativeMapService は局所地図（ロボット座標系地図）を扱うためのインターフェースです．インターフェースに含まれるサービスの一覧を表 7 に示します．

表 7: MRFC::RelativeMapService

サービス名	引数	戻り値	備考
getRelativeOGMapConfig	なし	RTC::OGMapConfig	
getRelativeOGMap	なし	TimedRelativeOGMapData	
getFloatRelativeOGMap	なし	TimedFloatRelativeOGMapData	

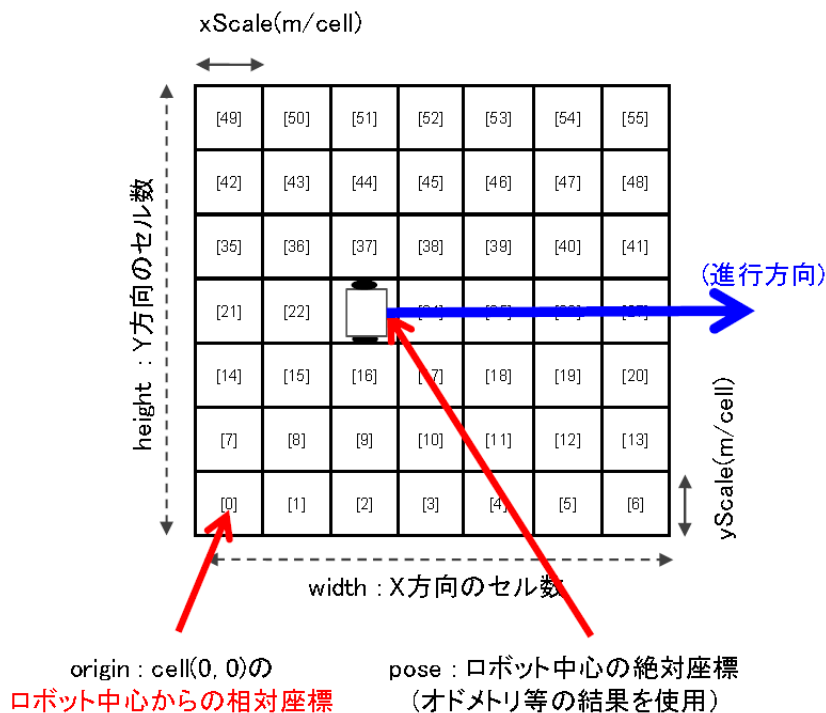


図 2: 局所地図の仕様

5 コンポーネントの実行手順

この章では、使用するための手順について説明します。

5.1 各プログラムの起動

まず初めにネームサーバ、RT System Editor および各コンポーネントを起動する必要があります。ネームサーバは、

スタート > すべてのプログラム > OpenRTM-aist > C++ > tools > Start Naming Service

を選択することで起動することができます。RT System Editor も同様に、

スタート > すべてのプログラム > OpenRTM-aist > C++ > tools > RT System Editor

を選択することで起動することができます。

次に各 RT コンポーネントを起動します。展開したフォルダの下にある『LocalMapComp.exe』，『LocalMapViewComp.exe』を実行して下さい。また、LocalMap コンポーネントにデータを入力するために用いる各コンポーネントを起動して下さい。ここでは例として URGDataFlowComp（北陽電機社製レーザセンサのデータを取得するモジュール）および MobileRobotController コンポーネント（Mobile Robot 社ロボット用制御コンポーネント）を用います。

5.2 RT System Editor 上でのコンポーネントの接続

RT System Editor の起動とコンポーネントの接続は次のような手順で行うことができます。

1. eclipse を起動し、パースペクティブで RT System Editor を選択する。
2. 図 3 の赤い丸で囲んだアイコン『ネームサーバを追加』を選択する。
3. 図 3 のように『ネームサーバに接続』の Address Port に『localhost』と入力して OK を選択する。
4. NameServiceView に起動したモジュールが表示されていることを確認する。
5. ファイル > Open New System Editor を選択する。
6. NameServiceView 上のモジュールを選択して、System Editor 上にドラッグしてモジュールのアイコンを表示させる。
7. 図 4 のようにモジュールを接続する。

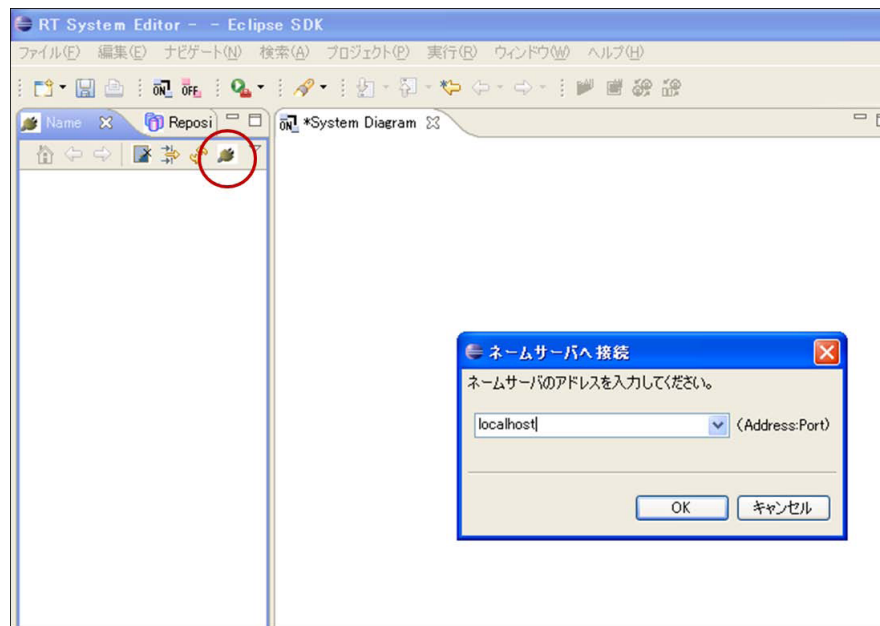


図 3: RT System Editor の画面

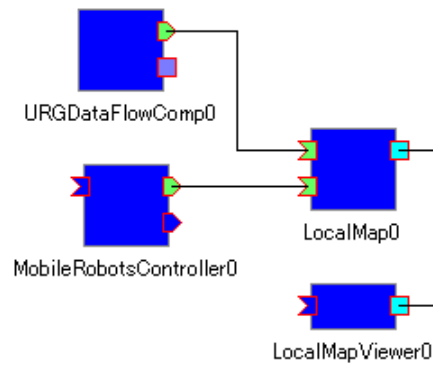


図 4: 各 RT コンポーネントの接続

5.3 コンフィギュレーションの設定

RT System Editor 上で LocalMap コンポーネントを選択すると、ConfigurationView に図 5 のように表示されます。ここで必要に応じてコンフィギュレーションの Value を設定し、『適用』ボタンを押すことで値を変更できます。

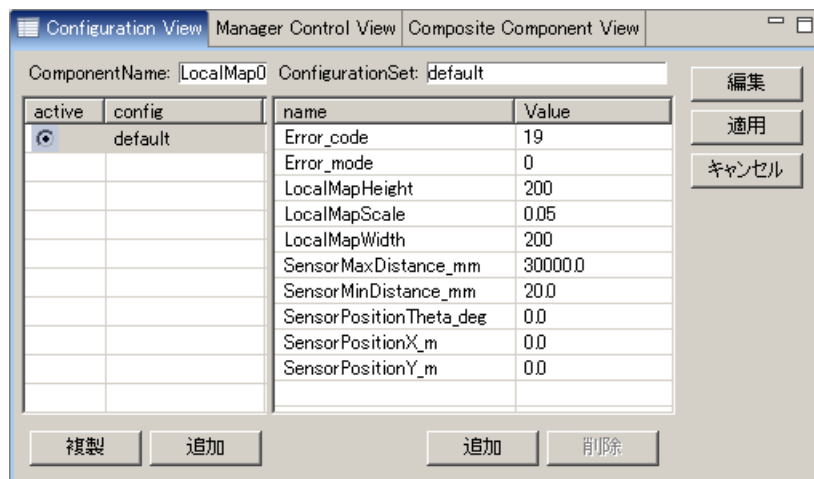


図 5: ConfigurationView

5.4 コンポーネントの実行と動作の確認

使用する RT コンポーネントの接続が完了し準備が整えば、全てのコンポーネントをアクティベートすることで局所地図生成を開始することができます。地図が生成されると図 6 のように局所地図が表示されます。

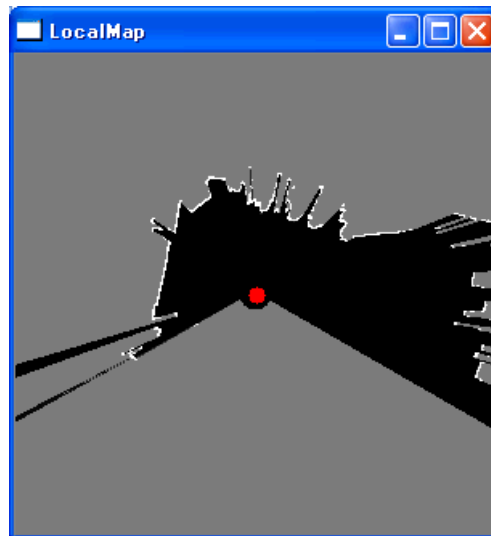


図 6: 生成された局所地図の例

6 連絡先

豊橋技術科学大学 行動知能システム学研究室

〒 441-8580

愛知県豊橋市天伯町雲雀ヶ丘 1-1

豊橋技術科学大学 情報・知能工学系

行動知能システム学研究室

TEL: 0532-44-6826

URL: <http://www.aisl.cs.tut.ac.jp/>

不明な点がある場合は rtc@aisl.ics.tut.ac.jp まで連絡をお願いします。