

大域経路計画コンポーネント 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 24 年 1 月 11 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	3
2	各コンポーネントについて	4
2.1	GlobalPathPlanner コンポーネント	4
2.2	GlobalMapLoader コンポーネント	5
2.3	GlobalMapView コンポーネント	6
2.4	Dummy2PosesSender コンポーネント	7
2.5	MessageReceiver コンポーネント	8
3	各データ型・インターフェースについて	8
3.1	IIS::TimedPose2D	8
3.2	IIS::TimedPose2DSeq	9
3.3	IIS::TimedPoseVel2DSeq	9
3.4	RTC::OGMapConfig	10
3.5	MRFC::TimedAbsoluteOGMapData	10
3.6	MRFC::TimedFloatAbsoluteOGMapData	10
3.7	MRFC::AbosoluteMapService	11
4	コンポーネントの実行手順	12
4.1	各プログラムの起動	12
4.2	RT System Editor 上でのコンポーネントの接続	12
4.3	コンポーネントの実行と動作の確認	14
5	連絡先	15

1 初めに

このドキュメントでは、大域地図の情報を基にロボットが移動する大域的な経路を計画する GlobalPathPlanner コンポーネント、大域地図をファイルから読み込むための GlobalMapLoader コンポーネント、大域地図と計画した経路を表示するための GlobalMapView コンポーネント、動作確認のために経路の開始点と目的地の表す位置・姿勢を出力する Dummy2PosesSender コンポーネントおよび入力された文字列データを表示する MessageReceiver 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。

- GlobalPathPlanner
ロボットが移動する経路を計画するコンポーネントの実行ファイルを含むフォルダ。
- GlobalMapLoader
大域地図データをファイルから読み込み出力するコンポーネントの実行ファイルを含むフォルダ。
- GlobalMapView
大域地図を表示するコンポーネントの実行ファイルを含むフォルダ。
- Dummy2PosesSender
動作確認用の位置・姿勢データ出力コンポーネントの実行ファイルを含むフォルダ。
- MessageReceiver
入力された文字列データを表示するコンポーネントの実行ファイルを含むフォルダ。
- 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。

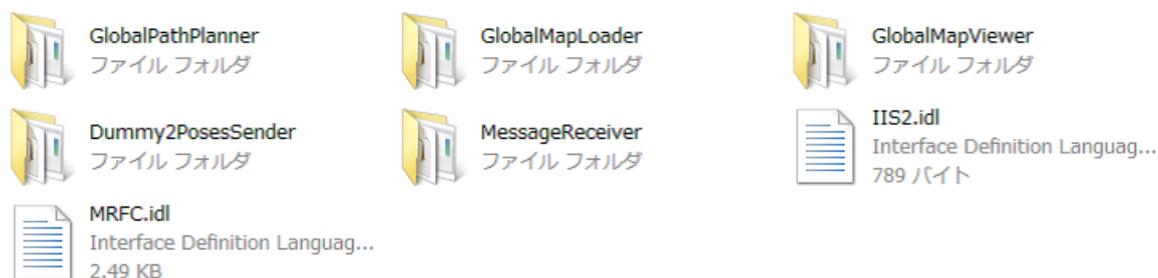


図 1: 解凍したフォルダの中身

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，GlobalMapLoader および GlobalMapViewer 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/>

GlobalMapLoader コンポーネントは CGAL 3.7 を使用しています．CGAL(Computational Geometry Algorithms Library) については下記サイトを参照して下さい．

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

2 各コンポーネントについて

ここでは付属する各 RT コンポーネントについて解説します。

2.1 GlobalPathPlanner コンポーネント

このコンポーネントは、データポートから入力された開始位置と目的地位置を結ぶロボットの移動経路を計算し、出力します。経路データは新しいデータが入力されたときに計算され、出力されます。また、出力する各中間目的地には速度情報が格納されますが、このコンポーネントではコンフィグレーションで指定した値が全ての中間目的地に設定されます。GlobalPathPlanner 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	IIS::TimedPose2D	start	経路計画の開始位置
In Port	IIS::TimedPose2DSeq	goal	経路計画の目的地位置
Out Port	IIS::TimedPoseVel2DSeq	path	ロボット移動経路を表す中間目的地系列を出力
Out Port	RTC::TimedString	message	経路探索の成否を出力

表 2: サービスポート

Port Type	Data Type	Port Name	備考
Service Consumer	AreaInfoService	AreaInfoService	経路計画用のグラフ情報を取得

表 3: コンフィギュレーション

型	変数名	単位	備考
double	RobotSafetyRadius	m	障害物までの距離がこの値より近い点には中間目的地を生成しない(狭い場所を通る場合には無視されます)
double	RobotVelocity	m/s	中間目的地に設定する速度

2.2 GlobalMapLoader コンポーネント

このコンポーネントは大域地図データを画像ファイルから読み込み、それを基にサービスポートから地図を出力します。また、起動時に地図データを自動的に解析して経路探索用のグラフを構築します。グラフ情報はサービスポートから出力され、これを基に GlobalPathPlanner は大域経路を計算します。

GlobalMapLoader コンポーネントが持つサービスポートの一覧を表 4 に、コンフィギュレーションの一覧を表 5 に示します。データポートはありません。

表 4: サービスポート

Port Type	Data Type	Port Name	備考
Service Provider	MRFC::AbsoluteMapService	AbsoluteMapService	絶対座標系地図を出力
Service Provider	TUT::AreaInfoService	AreaInfoService	経路計画用のグラフ情報を出力

表 5: コンフィギュレーション

型	変数名	単位	備考
string	MapData	-	読みこむ地図画像のファイル名を指定
double	InputMapScale	m/pixel	読みこんだ地図画像のスケール
double	RobotSafetyRadius	m	指定した直径（この値の倍）以下の隙間は通れないものとしてグラフを構築する

読み込んだ大域地図画像データはまずグレースケール化され、そのときの画素値が 0 に近いほど障害物の存在確率が低く（ロボットが通れる）、255 に近いほど高い（ロボットが通れない）ことを表すものとします。大域地図画像データの例を図 2 に示します。画像データは PNG といった OpenCV が読み込むことができる形式でなければなりません。



図 2: 大域地図画像の例

2.3 GlobalMapViewer コンポーネント

このコンポーネントは、大域地図およびロボットの経路を表示するために用います。GlobalMapViewer コンポーネントが持つデータポートの一覧を表 6 に、サービスポートの一覧を表 7 に、コンフィギュレーションの一覧を表 8 に示します。

表 6: 入出力データポート

Port Type	Data Type	Port Name	備考
In Port	IIS::TimedPoseVel2DSeq	path	経路データの取得
In Port	IIS::TimedPose2D	robot_pose	ロボット位置入力

表 7: サービスポート

Port Type	Data Type	Port Name	備考
Service Consumer	MRFC::AbsoluteMapService	AbsoluteMapService	絶対座標系地図の取得

表 8: 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	map_type	-	値が 0 のときは octed 型の地図を表示に用い、 値が 0 以外のときは float 型の地図を表示に用いる

MRFC::AbsoluteMapService では地図の各セルの値を整数で扱う octed 型の地図と実数で扱う float 型の 2 通りが実装されており、map_type の値によってどちらを用いるかを設定することができます。octed 型の地図 (MRFC::TimedAbsoluteOGMapData) については 3.5 節を、float 型の地図 (MRFC::TimedFloatAbsoluteOGMapData) については 3.6 節を、MRFC::AbsoluteMapService については 3.7 節を参照して下さい。

2.4 Dummy2PosesSender コンポーネント

このコンポーネントは、経路計画を行う開始位置および目的位置をを表す位置・姿勢をデータポートから出力する動作確認用のコンポーネントです。Dummy2PosesSender コンポーネントが持つデータポートの一覧を表 9 に、コンフィギュレーションの一覧を表 10 に示します。サービスポートはありません。

表 9: 入出力データポート

Port Type	Data Type	Port Name	備考
Out Port	IIS::TimedPose2D	start_pose	位置・姿勢の出力
Out Port	IIS::TimedPose2DSeq	goal_pose	位置・姿勢の出力

表 10: コンフィギュレーション

型	変数名	単位	備考
double	start_x	m	start_pose の X 座標
double	start_y	m	start_pose の Y 座標
double	start_heading	radian	start_pose の向き
double	goal_x	m	goal_pose の X 座標
double	goal_y	m	goal_pose の Y 座標
double	goal_heading	radian	goal_pose の向き
int	continuous_output	-	値が 0 のときは起動直後に 1 度だけ出力して自動的にディアクティベートする
int	deactivate_to_stop	-	値が 0 以外の場合はディアクティベート時に停止を意味する空のゴールを出力する

continuous_output の値が 0 以外の時は周期的にデータを出力し続けます。continuous_output の値が 0 の時は位置・姿勢データを起動直後に 1 度だけ出力して自動的にディアクティベートしますが、この場合は deactivate_to_stop の値に関わらず空のゴールは出力されません。

2.5 MessageReceiver コンポーネント

このコンポーネントは受け取った文字列データをコンソールの画面に表示する動作確認用のコンポーネントです。Dummy2PosesSender コンポーネントが持つデータポートの一覧を表 11 に示します。サービスポートおよびコンフィギュレーションはありません。

表 11: 入出力データポート

Port Type	Data Type	Port Name	備考
In Port	RTC::TimedString	message	

3 各データ型・インターフェースについて

ここでは各コンポーネントで使用されているデータ型・インターフェースについて説明します。

3.1 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 (X 座標 [m]), y (Y 座標 [m]), heading (向き [radian]) を格納。
- error: 誤差分散を格納する。
- id: 使用しない
- RTC::Time tm: タイムスタンプ, 使用しない

なお, ロボットの位置・姿勢には図 3 に示す座標系を用います。

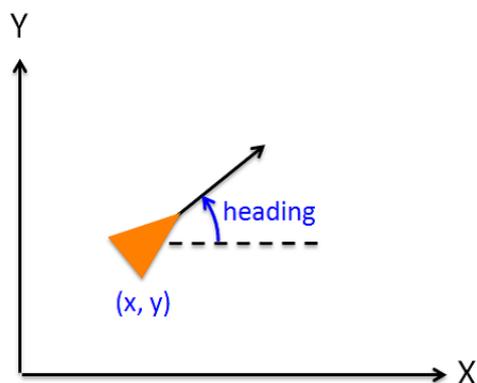


図 3: ロボット位置・姿勢の座標系

3.2 IIS::TimedPose2DSeq

IIS::TimedPose2DSeq はロボットの位置・姿勢の系列を格納するデータ型です。ロボットの目的地として用いる場合、要素数は1で通常使用します。また、要素数0の場合は停止要求とみなし目的地を破棄して停止することを表します。

- Pose2D data: x (X 座標 [m]), y (Y 座標 [m]), heading (向き [radian]) の系列を格納。
- error: 誤差分散を格納する。
- id: 使用しない
- RTC::Time tm: タイムスタンプ, 使用しない

3.3 IIS::TimedPoseVel2DSeq

IIS::TimedPoseVel2DSeq はロボットの移動経路を指示するための位置・姿勢 (中間目的地) と速度の系列を表すデータ型です。中間目的地での最大速度を (v_x , v_y , v_a) で表現します。経路の速度制限に利用可能です。ただし各パラメータは必ず守らなければならないものではなく、局所経路計画の実装次第では無視されます。また、要素数0の場合は停止要求とみなし目的地を破棄して停止することを表します。

- PoseVel2D data : x (位置 X), y (位置 Y), heading (姿勢), v_x (速度 x 成分), v_y (速度 y 成分), v_a (角速度) を格納。
- error: 使用しない
- id: 使用しない
- RTC::Time tm: タイムスタンプ, 使用しない

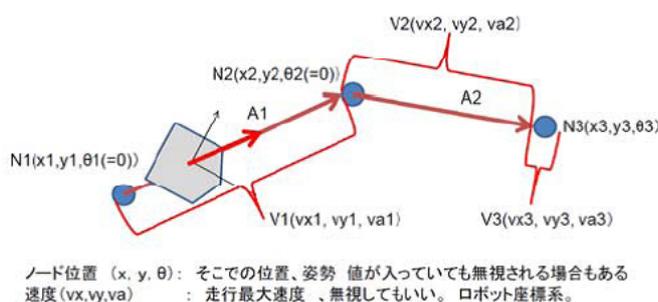


図 4: TimedPoseVel2DSeq について

3.4 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。なお、大域地図を扱う場合の origin の姿勢情報には常に 0 が格納されます。

- 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

3.5 MRFC::TimedAbsoluteOGMapData

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 5 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::Time tm: タイムスタンプ

3.6 MRFC::TimedFloatAbsoluteOGMapData

地図を float 型の系列で表現したデータ型です。ここで、float 型は単精度浮動小数であり、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 5 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::Time tm: タイムスタンプ

3.7 MRFC::AbosoluteMapService

MRFC::AbosoluteMapService は大域地図 (絶対座標系地図) 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は以下の通りです。

- RTC::OGMapConfig getAbsoluteOGMapConfig();
地図全体の情報を取得。
- TimedAbsoluteOGMapData getAbsoluteOGMap(in double x, in double y, in unsigned long width, in unsigned long height);
指定した範囲の地図を取得。セル (0,0) の絶対座標 x, y とセルサイズを指定する。
- TimedFloatAbsoluteOGMapData getFloatAbsoluteOGMap(in double x, in double y, in unsigned long width, in unsigned long height);
指定した範囲の地図を取得。セル (0,0) の絶対座標 x, y とセルサイズを指定する。

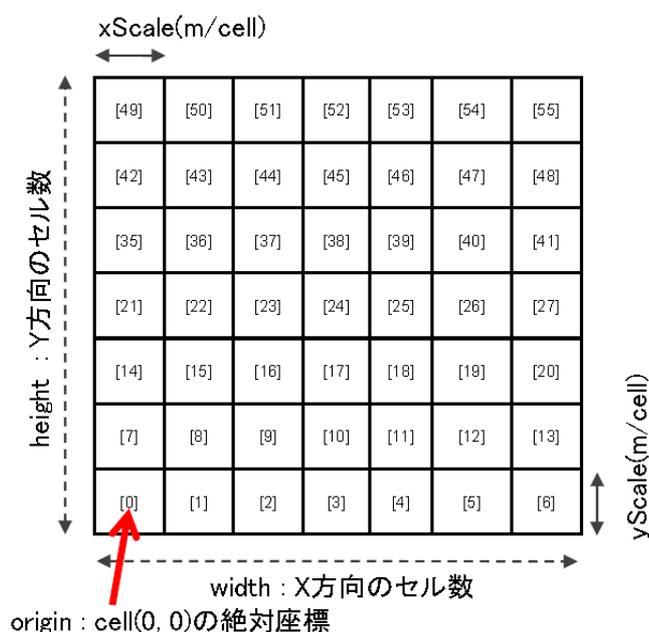


図 5: 大域地図の仕様

4 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。

4.1 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『GlobalPathPlannerComp.exe』、『GlobalMapLoaderComp.exe』、『GlobalMapViewComp.exe』、『Dummy2PosesSenderComp.exe』、『MessageReceiverComp.exe』を実行して下さい。

4.2 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. RT System Editor を起動する。
2. NameServiceView にホスト名が表示されていない場合、図 6 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。そして、図 6 のように『ネームサーバに接続』の Adress Port に『localhost』と入力して OK を選択する。
3. 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。
4. File > Open New System Editor を選択する。または File の下にある ON と書かれたアイコンを選択する。
5. 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。
6. 図 7 のようにモジュールを接続する。

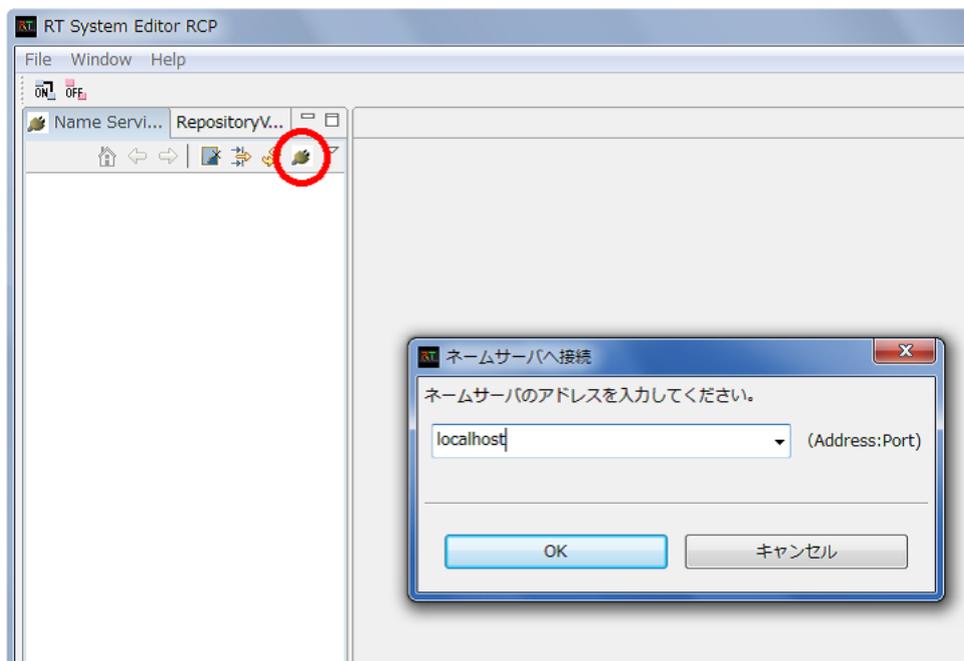


図 6: RT System Editor の画面

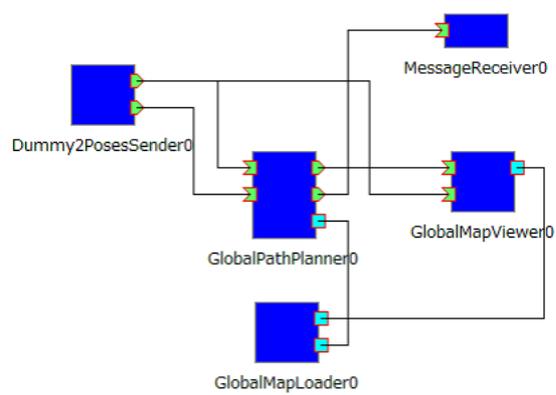


図 7: 各 RT コンポーネントの接続

4.3 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大域経路計画を開始することができます。実行結果の一例（大域地図および計画された経路）を図 8 に示します。この図では赤い点がロボットの位置を、水色の点が生成された中間目的地および目的地を表しています。

なお、経路を計算するための開始地点および目的地は Dummy2PosesSender コンポーネントで指定します。この図の例では Dummy1Heading=0.0, Dummy1X=1.0, Dummy1Y=1.0, Dummy2Heading=0.0, Dummy2X=23.0, Dummy2Y=23.0 を設定しました。

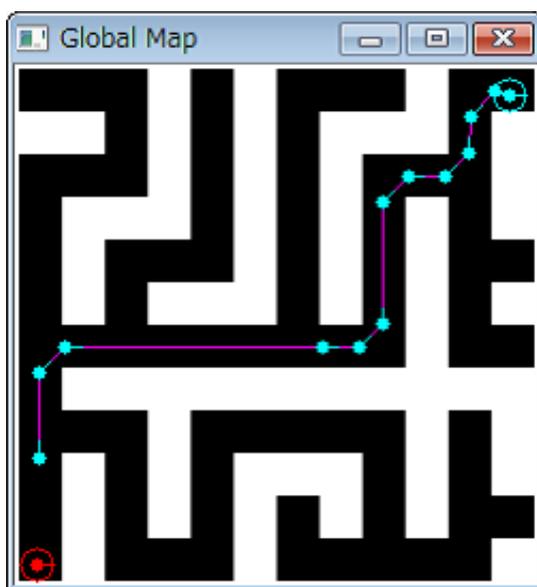
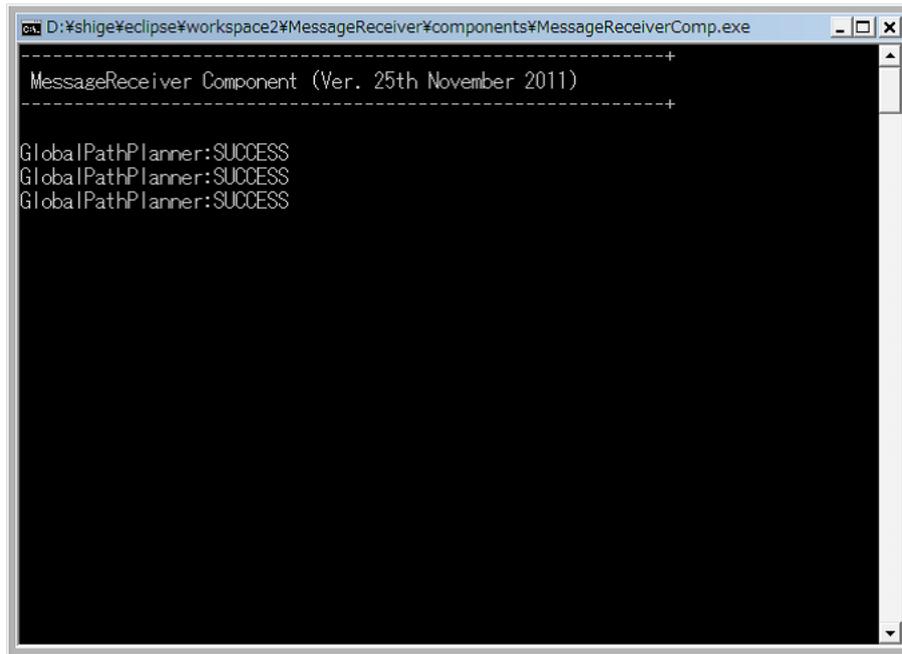


図 8: 実行結果の表示

図 9 は GlobalPathPlanner コンポーネントが出力したメッセージを MessageReceiver コンポーネントが表示したものです。GlobalPathPlanner コンポーネントが経路を計算する度にその成否を Timed-String 型の文字列データとして出力します。これによって経路計画が成功したかどうかを他のコンポーネントに通知することができます。経路計画に成功した場合は「GlobalPathPlanner:SUCCESS」が、失敗した場合は「GlobalPathPlanner:FAILED」が出力されます。



```
D:\shige\ eclipse\ workspace2\ MessageReceiver\ components\ MessageReceiverComp.exe
-----+
MessageReceiver Component (Ver. 25th November 2011)
-----+
GlobalPathPlanner:SUCCESS
GlobalPathPlanner:SUCCESS
GlobalPathPlanner:SUCCESS
```

図 9: GlobalPathPlanner が出力するメッセージの表示例

5 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.cs.tut.ac.jp まで連絡をお願いします。