自律移動機能共通インタフェース仕様書
（第1.0 版）

NEDO 次世代ロボット知能化技術開発プロジェクト

2012 年 2 月 24 日

CC BY SA

【改版履歴】

日付	版番号	改版ページ	改版内容
2012.2.24	1.0	全ページ	新規作成

【本書の利用にあたって】

本書は、クリエイティブ・コモンズ 表示 2.1 ライセンスの下に提供される。

（http://creativecommons.org/licenses/by-sa/2.1/jp/）

【本書の策定メンバー】

（敬称略、五十音順）

五十嵐広希　（京都大学 工学研究科 機械理工学専攻機械力学講座）
小笠原哲也　（東京大学大学院 情報理工学系研究科 知能機械情報学専攻）
喜多伸之　（独立行政法人産業技術総合研究所）
齋藤俊久　（セグウェイジャパン株式会社）
清水正晴　（千葉工業大学 未来ロボット技術研究センター）
竹内栄二朗　（東北大学大学院 情報科学研究科）
中本啓之　（株式会社セック 開発本部 第四開発部）
二宮恒樹　（富士ソフト株式会社 ロボット事業グループ 商品開発ユニット）
松本吉央　（独立行政法人産業技術総合研究所）
水川真　（芝浦工業大学 工学部 電気工学科）

（所属は 2012 年 2 月 24 日現在）

目次

1 はじめに 1
1.1 対象機能の概要 1
1.2 標準システム構成 2
2 本書を読む上での注意 4
2.1 基本方針 4
2.2 フォーマットと表現方法 4
2.2.1 列挙型定義 4
2.2.2 型定義 4
2.3 本仕様書における前提条件 5
2.3.1 座標系定義について 5
3 名前空間定義 6
4 データ型定義 6
4.1 標準型 6
4.1.1 RTC::Time 6
4.1.2 RTC::Pose2D 6
4.1.3 RTC::Point2D 6
4.1.4 RTC::Velocity2D 7
4.2 型宣言 7
4.3 自律移動機能用 7
4.3.1 TimedPath2DSeq 7
4.3.2 TimedVelocity2D 8
4.3.3 TimedPose2D 8
4.3.4 TimedPose2DSeq 8
5 共通インタフェース定義 9
5.1 データポート 9
5.1.1 計画経路インタフェース 9
5.1.2 走行指令、走行情報インタフェース 10
5.1.3 確率的自己位置姿勢インタフェース 11
5.1.4 目的地インタフェース 11
5.1.5 走行状態インタフェース 12
5.2 サービスポート 12
6 共通インタフェースを利用したシステム構築例 13
6.1 人追従機能 RTC モジュール群 13
6.2 RTC-CANopen リファレンスロボット 14

7 CORBA IDL 15
7.1 IIS.idl 15
8 参考文献 16

表目次

表 4.1 RTC::Time 6
表 4.2 RTC::Pose2D 6
表 4.3 RTC::Point2D 6
表 4.4 RTC::Velocity2D 7
表 4.5 TimedPath2DSeq 7
表 4.6 TimedVelocity2D 8
表 4.7 TimedPose2D 8
表 4.8 TimedPose2DSeq 8

図目次

図 1.1 移動知能ロボット共通インタフェースの使用シーン例 1
図 1.2 自律移動機能共通インタフェースを使用したシステム例(レベル1) 2
図 1.3 自律移動機能共通インタフェースを使用したシステム例(レベル 2) 2
図 1.4 自律移動機能共通インタフェースを使用したシステム例(レベル 3) 3
図 2.1 ロボット中心座標系 5
図 2.2 地図座標系 5
図 5.1 ウェイポイント列の表現例 9
図 5.2 計画経路インタフェース 10
図 5.3 走行指令、走行情報インタフェース 10
図 5.4 確率的自己位置姿勢インタフェース 11
図 5.5 目的地インタフェース 12
図 5.6 走行状態インタフェース 12
図 6.1 人追従制御コンポーネント適用例：人追従システム 13
図 6.2 人追従機能 RTC モジュール群 13
図 6.3 RTC-CANopen リファレンスロボット 14
図 6.4 RTC-CANopen リファレンスロボット制御モジュール群 14

1　はじめに

近年、ロボットの開発を効率化するためにコンポーネントベースのミドルウェア開発が盛んになっている。コンポーネントベースのミドルウェア開発において、インタフェースの共通化は、コンポーネントの相互接続性や相互運用性を確保するうえで非常に重要である。このような背景に基づき、本書では、自律移動機能に関わるインタフェースの共通仕様を定義する。

1.1　対象機能の概要

本仕様書では、地図を保持する室内中の既知の場所から、指定された任意の場所に自律移動するモジュールの「自律移動機能共通インタフェース仕様」を規定している。

本仕様書で規定する共通インタフェースでは、移動知能ロボットの構成を複数レベルで規定しており、各レベル毎に使用するインタフェースを定義している。また、実際に移動に用いる機構については、RT コンポーネントのレベルで隠蔽されていると考え、移動機構に依存しない形でのインタフェース定義を行っている(ただし、詳細例については主に車輪移動型ロボットを例として説明を行っている)。

なお、走行に使用する地図の構成方法、地図情報の伝達方法については、本仕様書の対象範囲外である。

移動知能ロボット共通インタフェースを実装した RT コンポーネントの使用シーンの一例を以下に示す。

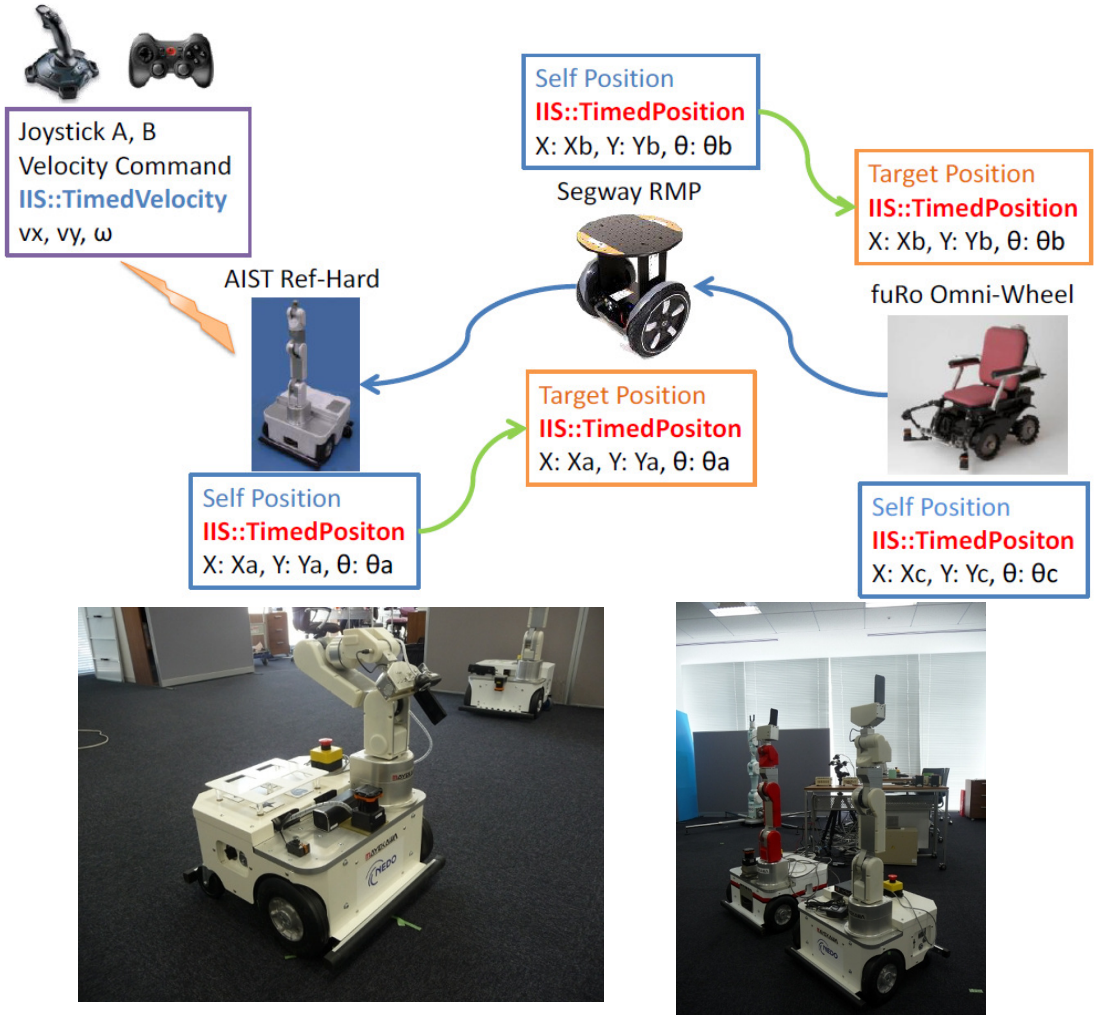


図 1.1　移動知能ロボット共通インタフェースの使用シーン例

1.2　標準システム構成

自律移動機能共通インタフェースを利用した標準的なシステム構成例を以下に示す。

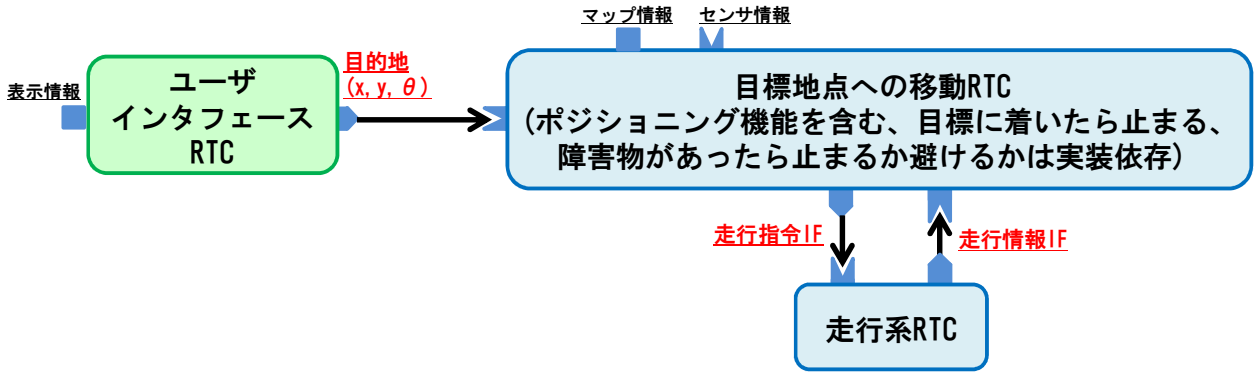


図 1.2　自律移動機能共通インタフェースを使用したシステム例(レベル1)

レベル1の構成では、実際に移動機構を制御する「走行系 RTC」とユーザからの指示を受け取る「ユーザインタフェース RTC」を独立したコンポーネントとし、それ以外の要素を全て「移動 RTC」としている。このレベルでは、「移動 RTC」への「目的地」入力部分、「移動 RTC」が「走行系 RTC」に「走行指令」を指示する部分、「移動 RTC」が「走行系 RTC」からの「走行情報」を受け取るための部分を共通インタフェースとして規定している。

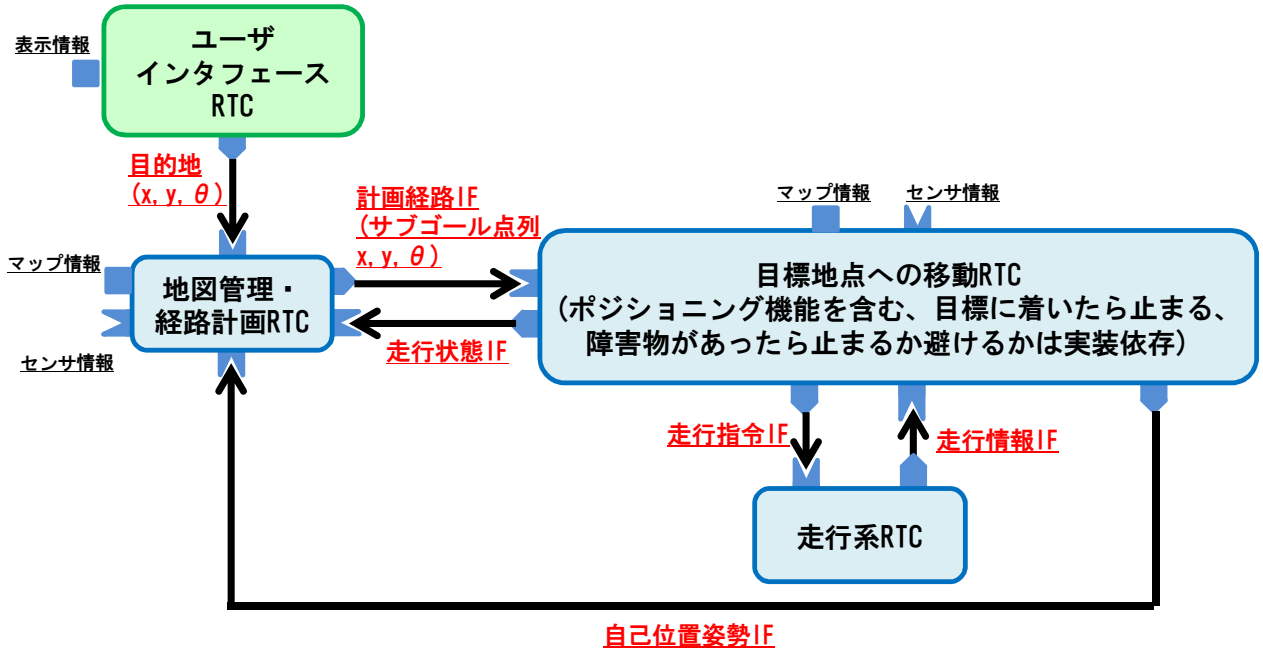


図 1.3　自律移動機能共通インタフェースを使用したシステム例(レベル 2)

レベル2の構成では、レベル1中の「移動 RTC」から「地図管理・経路計画 RTC」を分離している。このレベルでは、レベル1の内容に加え、「地図管理・経路計画 RTC」が「移動 RTC」に「移動経路」を指示する部分、「地図管理・経路計画 RTC」が「移動 RTC」から「走行状態」および「自己位置姿勢」を受け取る部分を共通インタフェースとして規定している。

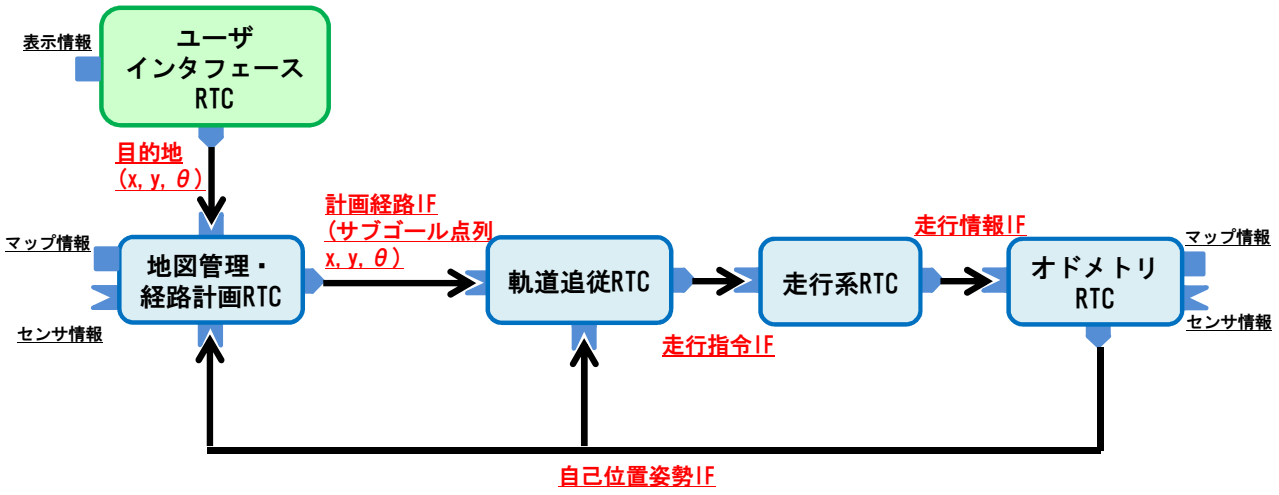


図 1.4 自律移動機能共通インタフェースを使用したシステム例(レベル 3)

レベル3の構成では、レベル2の「移動 RTC」を「軌道追従 RTC」と「オドメトリ RTC」に分割している。このレベルでは、レベル2と同等のインタフェースを使用している。

2　本書を読む上での注意

2.1　基本方針

インタフェース仕様の共通化は、仕様に合致しないコンポーネントを排除するため、時に開発内容を制限してしまうこともある。本仕様では、そのような制限を低減するために、以下のような方針で共通インタフェース仕様を定義する。

- 最低限のインタフェース仕様の定義:コンポーネントを相互接続・相互運用するために必要な最低限のインタフェース仕様のみを定義する。開発の制約となる仕様は最低限にとどめ、その他の部分は開発者が自由に拡張することができるようにする。
- 任意の機能の定義:いくつかの機能については実装を任意とする。実装された場合は、本書に書かれた仕様に準拠することを要求するが、実装をするかどうかは任意であり、それを実装していなかったからといって共通インタフェース仕様から外れるものとはしない。

2.2　フォーマットと表現方法

2.2.1 列挙型定義

本仕様書では、列挙型定義を次の表形式を用いて記述する。

表 XX 〈列挙型名〉

〈定数名〉	〈内容〉

2.2.2 型定義

本仕様書では、型定義を次の表形式を用いて記述する。

表 XX 〈型名〉

属性		
〈要素名〉	〈要素型〉	〈内容〉
・・・	・・・	・・・

2.3　本仕様書における前提条件

2.3.1 座標系定義について

本仕様書で使用する座標系定義を以下に示す。

○ロボット中心座標系

ロボット側の座標系としては、ロボット中心に原点があり、ロボット進行方向をＸ軸方向とした右手系を想定している。また、各要素の単位は、位置[m]、速度[m/s]、角度[radian]、角速度[radian/s]を想定している。

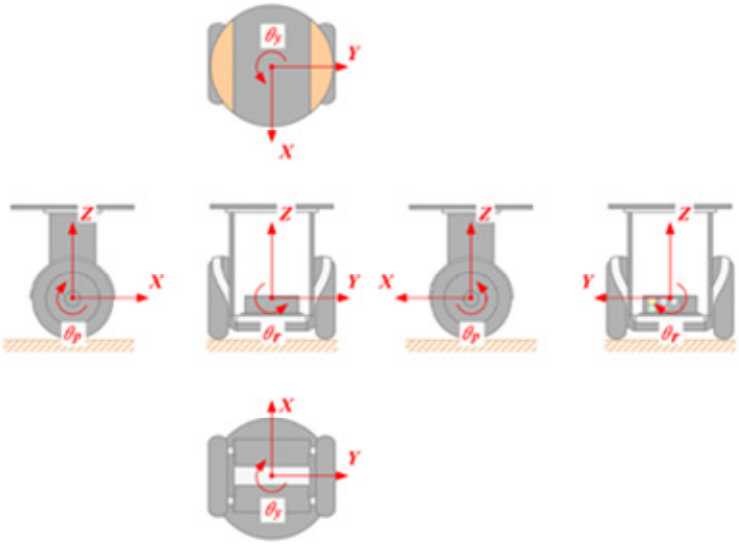


図 2.1 ロボット中心座標系

○地図座標系

地図側の座標系としては、原点、X 軸方向ともに任意の位置、方向に指定可能な右手系を想定している。また、各要素の単位は、位置[m]、速度[m/s]、角度[radian]、角速度[radian/s]を想定しており、radian の範囲は 0～2πで正規化されていると仮定している。更に、可能であれば平面直角座標系を推奨している。

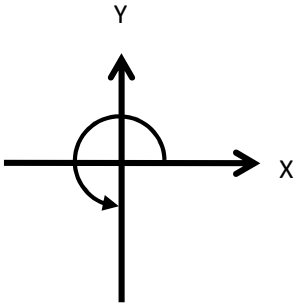


図 2.2 地図座標系

3　名前空間定義

自律移動機能用共通インタフェースでは、固有の名前空間として「IIS」を定義している。

本仕様書において規定するデータ型およびインタフェース定義において、名前空間の記載がない場合は、名前空間 IIS に属しているものとする（ただし、Primitive 型は除く）。

4　データ型定義

移動知能ロボット用共通インタフェースで使用するデータ型を以下に示す。

4.1　標準型

4.1.1 RTC::Time

時刻情報を格納するための型。OpenRTM-aist の標準型として BasicDataType.idl 内で定義されている。

表 4.1 RTC::Time

属性		
sec	unsigned long	秒単位の時刻情報
nsec	unsigned long	ナノ秒単位の時刻情報

4.1.2 RTC::Pose2D

2次元直交座標系上での位置・姿勢情報を表現するための型。OpenRTM-aist の拡張型として ExtendedDataTypes.idl 内で定義されている。

表 4.2 RTC::Pose2D

属性		
position	RTC::Point2D	2次元直交座標系上での位置情報
heading	double	2次元直交座標系上での姿勢情報。 X 軸から反時計回りに 0[rad]～2π[rad]の範囲。 負の値、多回転は無いものとしている。

4.1.3 RTC::Point2D

2次元直交座標系上での位置情報を表現するための型。OpenRTM-aist の拡張型として ExtendedDataTypes.idl 内で定義されている。

表 4.3 RTC::Point2D

属性		
x	double	2次元直交座標系上での X 方向位置情報。単位:m
y	double	2次元直交座標系上での Y 方向位置情報。単位:m

4.1.4 RTC::Velocity2D

2次元直交座標系上での速度情報を表現するための型。OpenRTM-aist の拡張型として ExtendedDataTypes.idl 内で定義されている。

表 4.4 RTC::Velocity2D

属性		
vx	double	2次元直交座標系上での X 方向速度情報。単位:m/s
vy	double	2次元直交座標系上での Y 方向速度情報。単位:m/s
va	double	2次元直交座標系上での Z 軸周りの角速度情報。 単位:rad/s

4.2 型宣言

本仕様では固有の型宣言は行っていない。

4.3 自律移動機能用

4.3.1 TimedPath2DSeq

移動知能ロボットの目標経路を格納するための型。目標経路はウェイポイント列(x,y, θ)で表現する。

表 4.5 TimedPath2DSeq

属性		
tm	RTC::Tine	時刻情報。目標経路を生成した時刻やデータ送受信時刻を格納するために利用。
id	sequence<long>	計画経路に対する ID 情報
pose	sequence<RTC::Pose2D>	ウェイポイント列情報。経路を構成する経由点列の情報を格納する。
velocity	Sequence<RTC::Velocity2D>	各ウェイポイントにおける速度情報を格納する。
error	sequence<double>	各ウェイポイントにおける誤差情報

※RTC::Time 型、RTC::Pose2D、RTC::Velocity2D 型は OpenRTM-aist で用意している標準型。

※id に格納する ID 情報の付番方法、コード体系については未定義。

※error に格納する誤差情報の表現方法については未定義。

4.3.2 TimedVelocity2D

移動知能ロボットの走行情報を格納するための型。ロボット中心座標系における速度を保持する。

表 4.6 TimedVelocity2D

属性		
tm	RTC::Tine	時刻情報。走行情報を生成した時刻を格納するために利用。
id	sequence<long>	走行情報に対する ID 情報
data	RTC::Velocity2D	ロボット中心座標系における速度情報
error	sequence<double>	各ウェイポイントにおける誤差情報

※RTC::Time 型、RTC::Velocity2D 型は OpenRTM-aist で用意している標準型。

※id に格納する ID 情報の付番方法、コード体系については未定義。

※error に格納する誤差情報の表現方法については未定義。

4.3.3 TimedPose2D

2次元直交座標上での確率的自己位置姿勢の推定結果を格納するための型。位置の推定結果に加えて、ID 情報、推定誤差などを保持することができる。

表 4.7 TimedPose2D

属性		
tm	RTC::Time	時刻情報。位置推定を行った時刻やデータ送受信時刻を格納するために利用。
id	sequence<long>	位置推定結果に対する ID 情報
data	RTC::Pose2D	二次元位置情報
error	sequence<double>	自己位置姿勢推定時の誤差情報

※RTC::Time 型、RTC::Pose2D 型は OpenRTM-aist にて事前定義されている標準型。

※id に格納する ID 情報の付番方法、コード体系については未定義。

※error に格納する誤差情報の表現方法については未定義。

4.3.4 TimedPose2DSeq

2次元直交座標上での目的地情報を格納するための型。位置情報に加えて、ID 情報、誤差情報などを保持することができる。

表 4.8 TimedPose2DSeq

属性		
tm	RTC::Time	時刻情報。目的地情報の送受信時刻を格納するために利用。
id	sequence<long>	目的地に対する ID 情報
data	sequence<RTC::Pose2D>	二次元位置情報
error	sequence<doublet>	誤差情報

※RTC::Time 型、RTC::Pose2D 型は OpenRTM-aist にて事前定義されている標準型。

※id に格納する ID 情報の付番方法、コード体系については未定義。

※error に格納する誤差情報の表現方法については未定義。

5　共通インタフェース定義

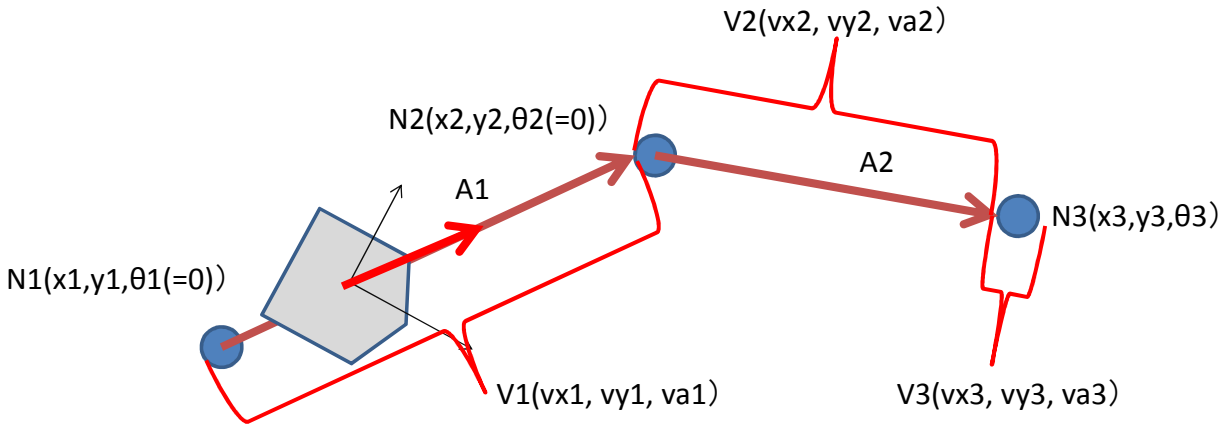
以下に自律移動機能共通インタフェースで使用する共通インタフェースの定義を示す。

5.1　データポート

5.1.1 計画経路インタフェース

地図管理モジュールから移動モジュール(レベル2)もしくは軌道追従モジュール(レベル3)へ目標経路を伝達するためのインタフェースである。

目標経路は、TimedPath2DSeq 型を用いてウェイポイント列(x,y, θ)で表現されている。ただし、ウェイポイントの間隔および個数は任意である。ウェイポイント列の表現例を以下に示す。



ノード位置（x, y, θ): そこでの位置、姿勢　値が入っていても無視される場合もある
速度(vx,vy,va)　　: 走行最大速度　、無視してもいい。 ロボット座標系。

図 5.1 ウェイポイント列の表現例

対象ロボットをバックさせる場合には、vx が負数、vy がゼロの指定となる。また、ウェイポイントの角度指定については、最後のポイント以外は無視しても良い。更に、ウェイポイントの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、ロボットの構造や走行制御系によってはθ や vy、va 等は無視する場合もある。

新しいウェイポイントを受信した場合、既に受け取っている旧ウェイポイント情報は破棄し、新たに受け取ったウェイポイント列を採用する。ただし、ウェイポイント切り替え処理時にロボットを停止させるかどうかについては実装依存である。また、ウェイポイント列のデータ長がゼロである場合は、その場で停止する事を意味しており、旧ウェイポイント情報は破棄する。

ウェイポイント列の1番目のデータは、スタート地点を表現する。このため、本インタフェースを用いて伝達するデータには少なくとも2つのウェイポイント情報が必要となる。

ウェイポイント列1番目のデータで表現しているスタート点と自己位置にズレが存在する場合の修正動作については未定義である。

計画のための経路地図のフォーマットやプランニングについては、使用するシナリオに依存するので、実装依存となる。

本仕様自体は最低限のナビゲーションのための経路情報の仕様であり、充電や制御しながらの移動等、動作に特殊な制御が入るような場合は、専用のモジュールに制御を移す。

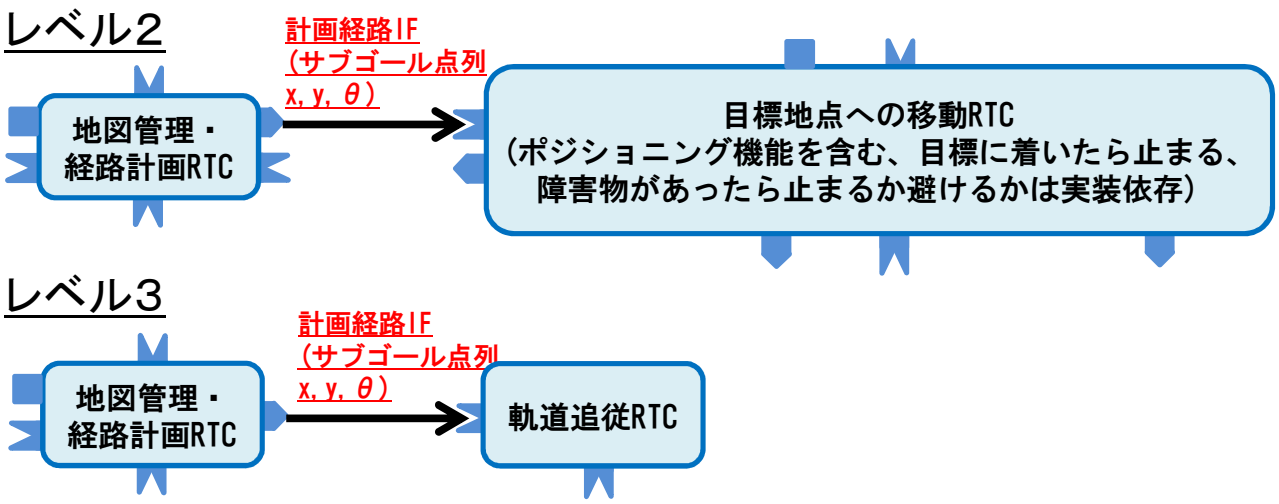


図 5.2 計画経路インタフェース

5.1.2 走行指令、走行情報インタフェース

走行指令インタフェースは、移動モジュールから走行系モジュール(レベル1, レベル2)もしくは軌道追従モジュールから走行系モジュール(レベル3)へ走行指令情報を伝達するためのインタフェースである。また、走行情報インタフェースは、走行系モジュールから移動モジュール(レベル1, レベル2)もしくは走行系モジュールからオドメトリ(位置推定)モジュールへ現状の走行情報を伝達するためのインタフェースである。

走行指令、走行情報ともに TimedVelocity2D 型を用いており、ロボット中心座標系における速度(Vx,Vy,Va)を設定する。

走行情報インタフェースでは、時間が重要な意味を持つので、RTC::Time tm の情報を付加することを推奨する。また、10ms 程度以下の周期で出力することも推奨する。

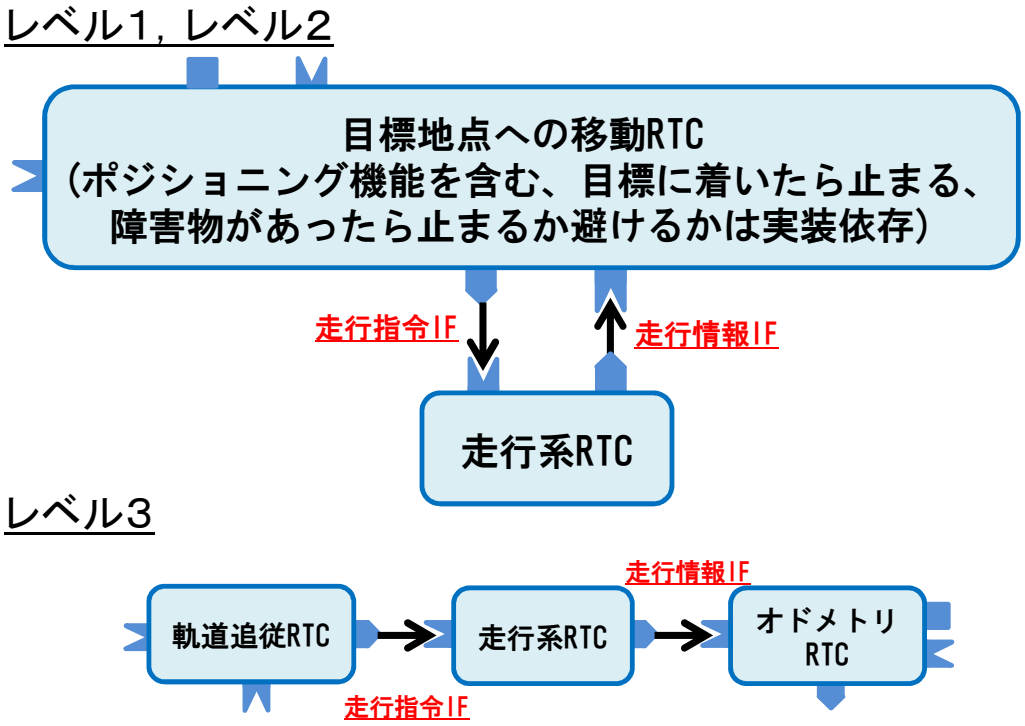


図 5.3 走行指令、走行情報インタフェース

5.1.3 確率的自己位置姿勢インタフェース

移動モジュールから地図管理・経路計画モジュール(レベル2)もしくはオドメトリ(位置推定)モジュールから軌道追従モジュールと地図管理・経路計画モジュール(レベル3)に自己位置姿勢を伝達するためのインタフェースである。

TimedPose2D 型を用いて、地図座標系における位置と姿勢(x,y, θ)を設定する。

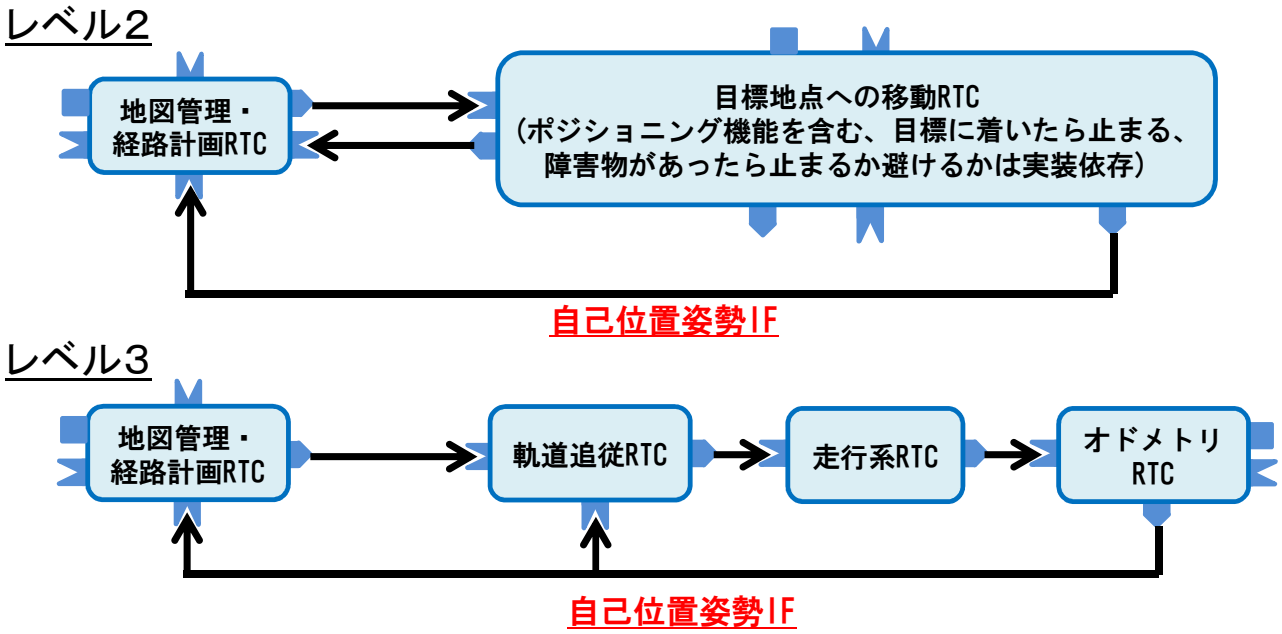


図 5.4 確率的自己位置姿勢インタフェース

5.1.4 目的地インタフェース

UI モジュールから移動モジュール(レベル1)もしくは UI モジュールから地図管理・経路計画モジュール(レベル 2 およびレベル 3)に目的地情報を伝達するためのインタフェースである。

TimedPose2DSeq 型を用いて、地図座標系における位置と姿勢(x,y, θ)を伝達する。型定義としては sequence 型であるため、複数のデータを設定可能であるが、1番目のデータのみが有効であり、その他のデータについては無視する。

新しい目的地位置姿勢を受信した場合、過去に受け取った旧目的地位置姿勢情報は破棄し、新しい目的地位置姿勢情報を採用する。

受け取った目的地位置姿勢情報のデータ長がゼロだった場合は、目的地情報のクリアを意味しており、旧目的地位置姿勢情報は破棄する。目的地情報がクリアされた場合の経路計画モジュールの動作については実装依存であるが、その場で停止することを推奨する。

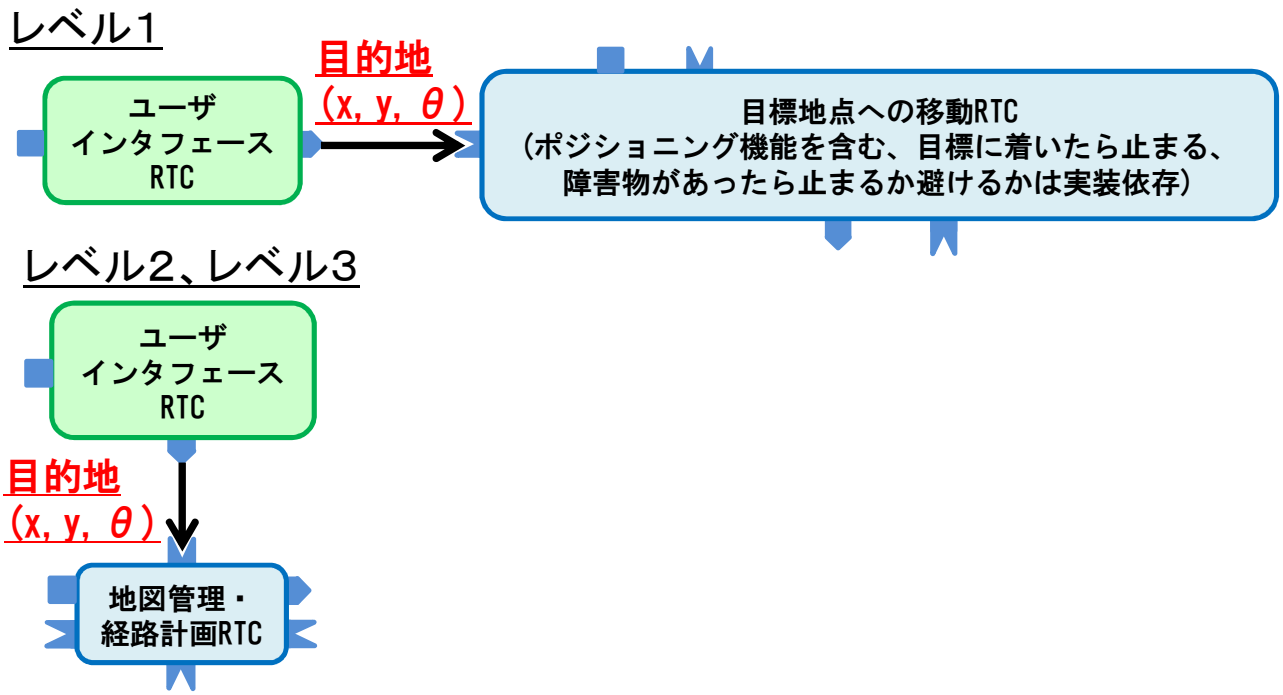


図 5.5 目的地インタフェース

5.1.5 走行状態インタフェース

移動モジュールから地図管理・経路計画モジュールへロボットの走行状態を伝達するためのインタフェースである。

TimedPose2D 型を用いて、ウェイポイント番号と走行状態(走行中 or 停止中)を伝達する。

本インタフェースは、与えられたウェイポイント列の中から、ウェイポイントを通過する度に、通過したウェイポイント番号を送信する。また、停止した場合には、フラグを立てるとともに、最後に通過したウェイポイントの情報を送信する。

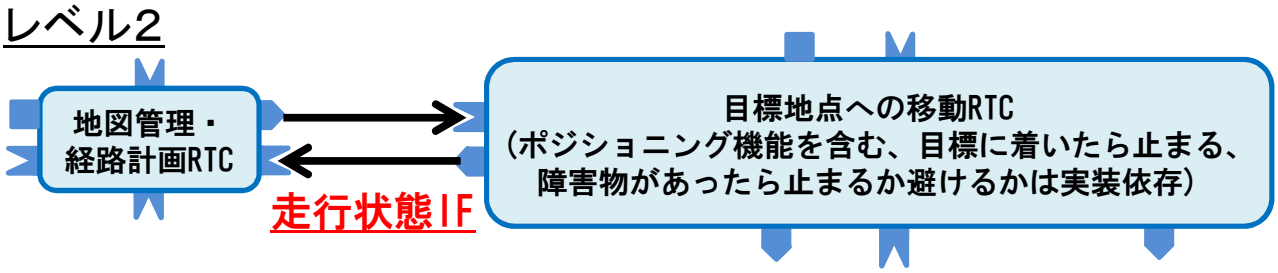


図 5.6 走行状態インタフェース

5.2 サービスポート

本仕様では固有のサービスポートは定義していない。

6　共通インタフェースを利用したシステム構築例

6.1　人追従機能 RTC モジュール群

○開発者：セグウェイジャパン株式会社

○詳細 URL：http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID053

○概要

カメラから画像データを取得し、対象人物の色ヒストグラム情報を記憶して、その人を追従する移動体向けの速度指示値を生成するモジュール群。

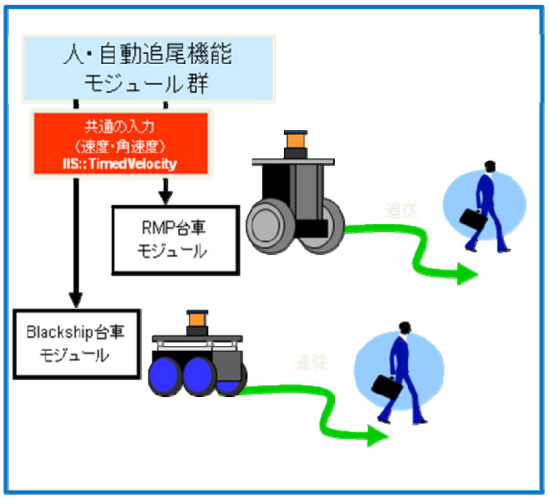


図 6.1 人追従制御コンポーネント適用例：人追従システム

本コンポーネントを利用したシステム構成例を以下に示す。図中の赤字の部分が、本仕様書で規定している共通インタフェースを使用している部分である。

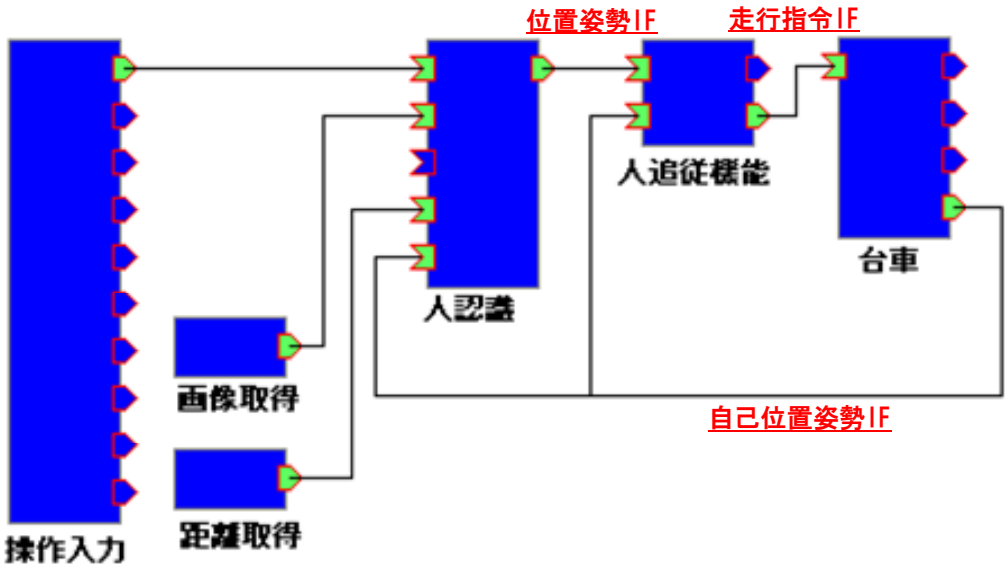


図 6.2 人追従機能 RTC モジュール群

6.2　RTC-CANopen リファレンスロボット

○開発者:芝浦工業大学　水川研究室

○詳細 URL:http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID366

○概要

RTC-CANopen を利用する上でのリファレンスとするために開発されたロボット(リファレンスロボット)を操作するためのモジュール群である。

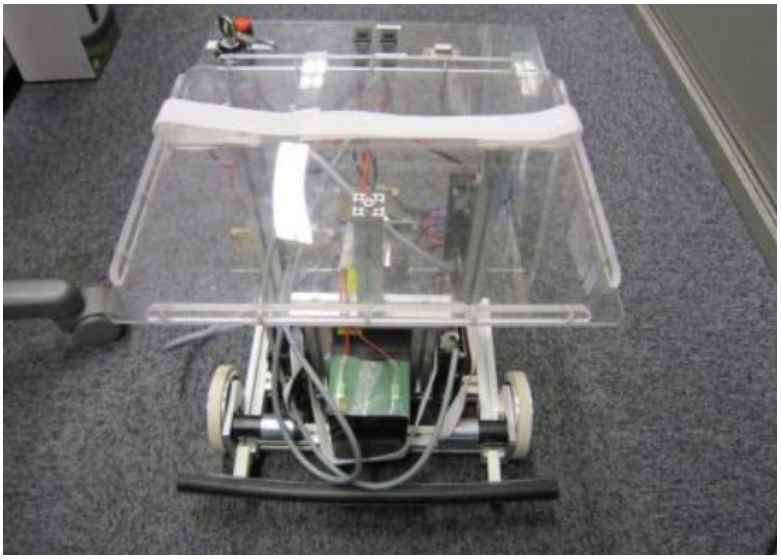


図 6.3 RTC-CANopen リファレンスロボット

本モジュール群の構成例を以下に示す。図中の赤字の部分が、本仕様書で規定している共通インタフェースを使用している部分である。

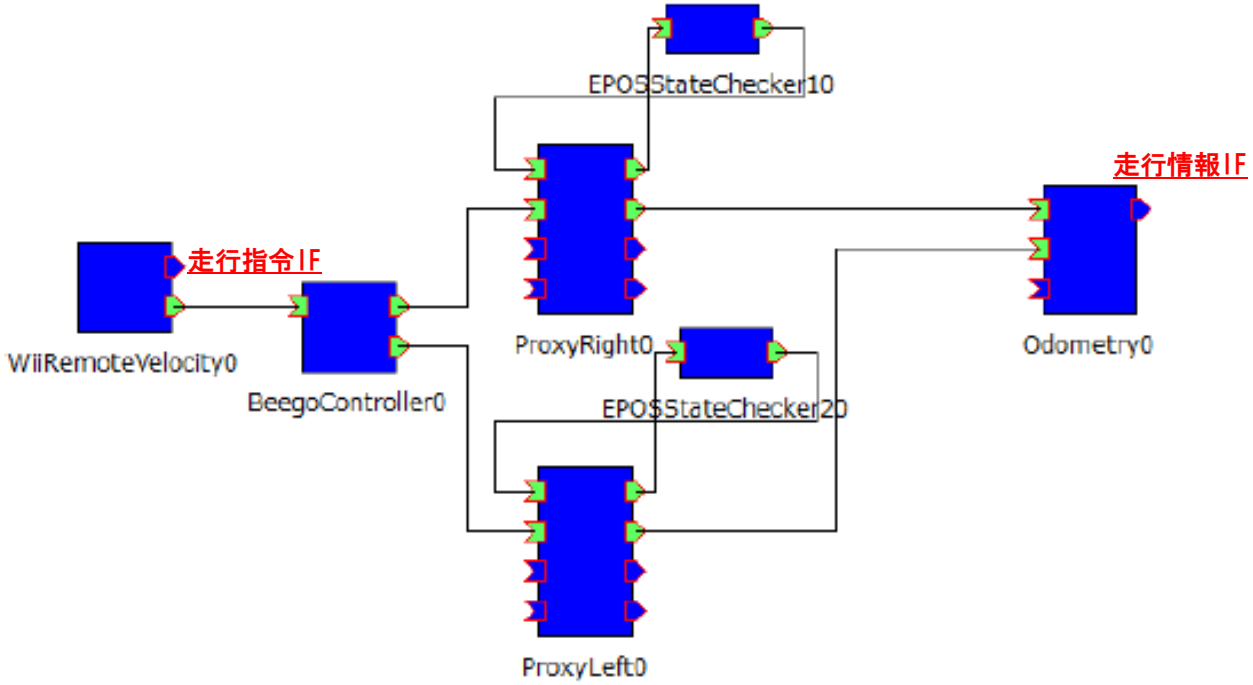


図 6.4 RTC-CANopen リファレンスロボット制御モジュール群

7 CORBA IDL

自律移動機能用共通インタフェースの IDL 定義を以下に示す。

7.1 IIS.idl

```
/* version 1.0 */

#ifndef IIS_IDL
#define IIS_IDL

#include “BasicDataType.idl”
#include "ExtendedDataTypes.idl"

module IIS {
	struct TimedPose2D {				//推定位置 IF
		RTC::Time tm;
		sequence<long> id;
		RTC::Pose2D data;
		sequence<double> error;
	};

	struct TimedPath2DSeq {				//計画経路 IF
		RTC::Time tm;
		sequence<long> id;
		sequence<RTC::Pose2D> pose;
		sequence<RTC::Velocity2D>velocity;	//進行方向
		sequence<double> error;
	};

	struct TimedVelocity2D {			//走行指令・走行情報 IF
		RTC::Time tm;
		sequence<long> id;
		RTC::Velocity2D data;
		sequence<double> error;
	};

	struct TimedPose2DSeq {				//目的地 IF
		RTC::Time tm;
		sequence<long> id;
		sequence<RTC::Pose2D> data;
		sequence<double> error;
	};
};
#endif /* IIS_IDL */
```

8　参考文献

[1] 移動ロボット用 RT コンポーネントの共通インタフェースー次世代ロボット知能化技術開発プロジェクトにおける移動 1 サブ WG 活動報告ー:清水正晴,喜多伸之,齋藤俊久,竹内栄二朗,中島裕介,武川直史,林原靖男,大和秀彰,戸田健吾,古田貴之,水川真:第 27 回日本ロボット学会学術講演会予稿集 RSJ2009AC3D1-03(2009)

[2] 移動ロボット用 RT コンポーネントの共通インタフェースー次世代ロボット知能化技術開発プロジェクトにおける移動 1 サブ WG 活動報告(第2報)ー:清水正晴,喜多伸之,齋藤俊久,竹内栄二朗,中島裕介,武川直史,五十嵐広希, 林原靖男,大和秀彰,戸田健吾,古田貴之,水川真: 第 10 回システムインテグレーション部門講演会(SI2009)予稿集(2009)