

移動機能



インターフェース仕様書 Version 2.6



2025年10月

ロボット革命・産業IoTイニシアティブ協議会

ロボットイノベーションWG

ソフトウェアアーキテクチャ調査検討委員会

発行者 ロボット革命・産業IoTイニシアティブ協議会

〒169-0075 東京都新宿区高田馬場1丁目31-18

高田馬場センタービル 12階

TEL 03-6302-1861

E-mail office@jmfri.gr.jp

URL <https://www.jmfri.gr.jp/>

Copyright © 2025 ロボット革命・産業IoTイニシアティブ協議会 All Rights Reserved.

本文書は、著作権法および国際条約により保護されています。個人または会社（または会社に準ずるもの）内部での使用を目的として、本文書をダウンロード、印刷、または電子的に閲覧することができます。本資料の内容の全部又は一部については、私的使用又は引用等著作権法上認められた行為として、適宜の方法により出所を明示することにより、引用・転載複製を行うことができます。内容の全部又は一部について、ロボット革命・産業IoTイニシアティブ協議会に無断で改変を行うことはできません。

ロボット革命・産業IoTイニシアティブ協議会はいかなる目的においても使用可能性を保証するものではなく、本文書の内容を使用したいいかなる場合においても責任を負いません。本文書の使用者は、本文書に記載された内容の使用に関連して発生したすべての要求、請求、訴訟、損失、損害（人身事故による損害を含む）、費用、経費（弁護士費用を含む）について、ロボット革命・産業IoTイニシアティブ協議会に何らの損害も与えないことに同意するものとします。

目次

1	はじめに	2
1.1	本仕様書で提供する情報	2
1.2	本仕様の対象とするユーザ.....	2
1.3	参考文献	2
1.4	委員名簿	3
2	想定システム	4
2.1.	想定環境	4
2.2.	自律移動機能の状態遷移.....	5
3	Platform Independent Model (PIM)	6
3.1.	書式について	6
3.1.1.	クラスとインターフェース.....	6
3.1.2.	列挙型	6
3.2.	インターフェース定義	7
3.2.1.	データ型定義	11
3.2.2.	インターフェース定義(常時通信データ)	20
3.2.3.	インターフェース定義(イベントドリブンデータ)	23
4	Platform Specific Model (PSM)	29
4.1.	ROS PSM	29
4.1.1.	SysML-ROS 変換用マッピングルール	29
4.1.2.	定義ファイル.....	30
Appendix.1.	内部構成の設計例	37
A.1.1.	移動状態管理	38
A.1.2.	経路計画(Path planning)	40
A.1.3.	位置リスト	41
A.1.4.	位置	41
A.1.5.	周辺環境計測	41
A.1.6.	障害物検知(Obstacle detection).....	41
A.1.7.	自己位置推定(Localization).....	42
A.1.8.	移動制御.....	42
A.1.9.	移動機能プロパティ管理	43
A.1.10.	充電管理.....	43
A.1.11.	HW 管理	43
Appendix.2.	地図について	44
Appendix.2.1.	地図の種類について	44
Appendix.2.2.	地図の切り替えについて	46

Appendix.3. 「人協働マニピュレーションモジュール」と連携する際の動作について
47

1 はじめに

近年、様々な分野で自律移動ロボットの導入が進められている。こうしたロボットの多くは様々な企業により提供されているが、今後もさらに多くの企業が参入することが想定される。また、単体の運用だけでなく、複数台の運用というケースになってきた場合、ハードウェア開発を主とする企業だけではなく、システム開発を担当する企業の参入も想定される。こうした複数台を含むアプリケーション開発を円滑に行うためには、自律移動ロボットとそれらを運用するシステム間でのインターフェースが厳密に規定されていることが好ましい。このような仕様が作られることで、ハードウェア開発企業と SIer 間での責任分担が行いやすくなり、市場の活性化につながることを期待できる。

本仕様書では、このような運用システムと自律移動ロボット(本仕様書では移動機能と呼ぶ)の間におけるインターフェースを規定したものとなっている。主として移動機能を扱うが、移動機能として利用するに当たって必要となる周辺情報(入力・出力される情報や想定する利用方法、地図など)についても適宜示すことで、本仕様書のスコープの把握を行いやすくしている。

1.1 本仕様書で提供する情報

本仕様書では、実装に依存しない形での設計指針の提示を目的としているため、PIM(Platform Independent Model)での記述を行う。また、本仕様書では、国際標準となっているシステムモデリング言語である SysML(Systems Modeling Language)を用いており、主として以下のダイアグラムを用いている。

- ブロック定義図
- 内部ブロック図
- シーケンス図
- ステートマシン図

国際標準となっているモデリング言語を用いることで、幅広い技術者・研究者によって参照しやすい仕様としている。

1.2 本仕様の対象とするユーザ

- ロボットハードウェア開発者
- ロボットシステムインテグレータ
- ロボット要素技術開発者・研究者

1.3 参考文献

[SysML] Object Management Group, OMG Systems Modeling Language (OMG SysML), Version 1.7, OMG document number formal/24-01-07, 2024

[RLS] Object Management Group, Robotic Localization Service (RLS), Version 1.1, OMG document number formal/12-08-01, 2012

[ISO-19111-2] Geographic information - Spatial referencing by coordinates - Part 2: Extension for parametric values, 2019

1.4 委員名簿

(委員長) 名城大学 大原 賢一 (敬称略)
(副委員長) 産業技術総合研究所 安藤 慶昭

【委員メンバー】

I H I	山崎 峻一
I D E C	延廣 正毅
会津大学	成瀬 継太郎
会津大学	屋代 眞
NTT 西日本	長濱 星斗
NTT 西日本	細川 周秀
NTT 西日本	間下 有理
NTT 西日本	松村 誠明
オリエンタルモーター	近藤 大生
神奈川県立産業技術総合研究所	宮澤 以鋼
川崎重工	蓮沼 仁志
川田テクノロジーズ	長嶋 功一
キビテク	前原 賢一
国際航業	武田 浩志
国土館大学	山本 大介
サイバネット MBSE	三好 崇生
産業技術総合研究所	中坊 嘉宏
セイコーエプソン	木本 吉則
セイコーエプソン	長谷川 浩
セック	中本 啓之
大同大学	吹田 和嗣
T H K	氏家 弘貴
T H K	佐野 将人
T H K	和田 成広
東芝	貞本 敦史
東芝	森 明慧
日本品質保証機構	駒澤 香介
パナソニックホールディングス	安藤 健
パナソニックホールディングス	上松 弘幸
パナソニックホールディングス	岡本 球夫
パナソニックホールディングス	是兼 慎也
日立製作所	大塚 敏史
日立製作所	坂井 亮
富士ソフト	酒井 貴史
本田技術研究所	小川 直秀
三菱電機	山隅 允裕
ヤンマーホールディングス	空閑 融
ヤンマーホールディングス	杉浦 恒
早稲田大学	菅 佑樹

【オブザーバ】

経済産業省 ロボット政策室	山本 知生
経済産業省 ロボット政策室	梁島 拓郎
新エネルギー・産業技術総合開発機構	土井 浩史
新エネルギー・産業技術総合開発機構	小林 綾乃

2 想定システム

2.1. 想定環境

本仕様書では、以下のようなシステムを想定している。

- ・自律移動ロボットは、周囲に人が存在する環境下で動作し、上位アプリ(レストランロボットシステム、物流倉庫における自動搬送システムなど)の「経路点管理機能」からの指令に基づいて、指定された目標点まで安全かつ迅速に移動する必要がある。
- ・自律移動ロボットは、障害物を自律的に回避できる機能、もしくは障害物を発見した場合に接触せずに停止する機能を有すると想定している。単純に与えられた経路/軌道に沿って走行するだけのロボットは想定していない。(バンパによる接触検知で停止する機能のみを有するロボットは想定していない)。
- ・同一環境内で複数台の自律移動ロボットが動作しており、それぞれの自律移動ロボットに「自律移動機能」が組み込まれている
- ・システム全体に含まれる複数のロボットの運行を管理する機能は「経路点管理機能」側が持っており、自律移動ロボット間では位置情報などの各種情報を伝達するための通信は行っていない。
- ・自律移動ロボットは、基本的には2次元平面内のみを移動する。ただし、エレベータなどの手段を利用して階の移動を行う場合はある(ドローンのように3次元空間中を移動するロボットについては、十分な検討を行っていない)。
- ・地図情報については、事前に作成した地図を配信する場合のみを想定している。動作中に地図を更新する機能については考慮していない。
- ・本仕様書では、自律移動機能のソフトウェア・コンポーネント部分のアーキテクチャに着目しているため、ネットワークの通信障害などは考慮していない(「経路点管理機能」「外部アプリ」と「自律移動機能」の間では、送信したメッセージが確実に届くと仮定している)。

なお、本仕様書では、原則SI単位系を使用することを想定している。

2.2. 自律移動機能の状態遷移

本仕様書で想定している自律移動機能の状態遷移を以下に示す。

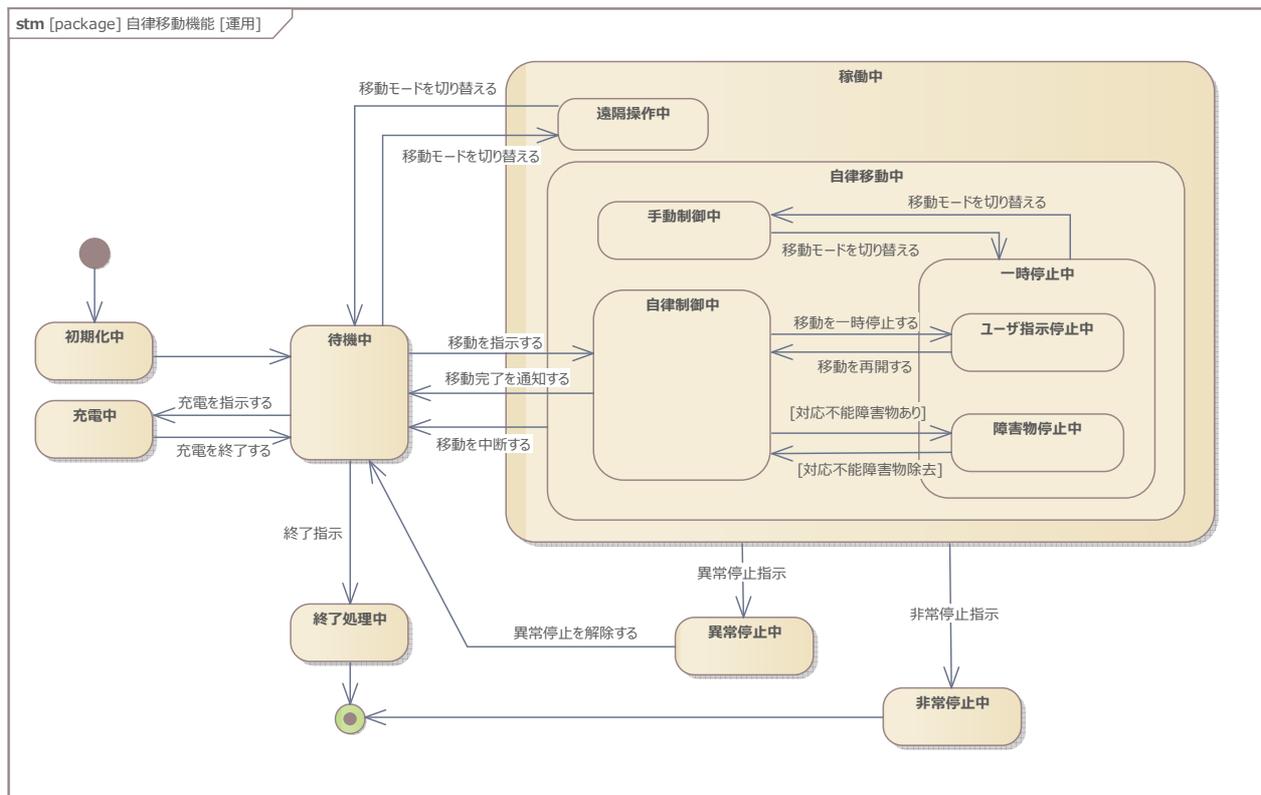


図 1 自律移動機能の状態遷移

各状態の意味を以下に示す。

状態	概要
初期化中	電源 ON を行った後の状態。各種初期化処理を行った後、自動で「待機中」に遷移する
待機中	経路点管理機能からの指示を待っている状態。
稼働中	経路点管理機能からの指示に基づき、何らかの動作を行っている状態。
自律移動中	指示された動作計画に基づき、動作している状態。
自律制御中	指示された動作計画に基づき、自律的に制御している状態。経路計画や障害物回避などを実行する。
手動制御中	動作計画を実行している途中で、外部システムから手動で操作されている状態。経路計画や障害物回避などは実行されない。
一時停止中	何らかの理由で、動作を一時的に停止している状態。
ユーザ指示停止中	外部アプリからの指令で、動作を一時的に停止している状態。
障害物停止中	周辺障害物の影響で、動作が継続できず、一時停止している状態。
遠隔操作中	動作計画が指示されていない段階で、外部システムから手動で操作されている状態。
充電中	充電を実行している状態
異常停止中	外部から緊急停止された状態。適切なりカバー処理を実行する事で、電源 OFF を行わずに動作を継続可能な状態。
非常停止中	外部から非常停止され、手動のリセットが必要な状態であり、ソフトウェアは認識できた場合に終了へ遷移する事が望ましい。JIS B8445:2016 (ISO13482:2014) の「6.2.2.2 非常停止」が発生している状態。
終了処理中	電源 OFF を行う前の状態。各種終了処理を行った後、電源が OFF となる。

3 Platform Independent Model (PIM)

汎用的な自律移動機能のソフトウェア・コンポーネントのPlatform Independent Model(PIM)を以下に示す.

3.1. 書式について

3.1.1. クラスとインターフェース

クラスとインターフェース定義で使用している表の書式を以下に示す.

Description: <対象要素の概要説明>				
DerivedFrom: <対象要素の親要素>				
Attributes:				
<属性名>	<属性の型>	<省略可否>	<多重度>	<対象属性の概要説明>
...
Operations:				
<操作名>		<省略可否>		
<方向>	<パラメータ名>	<パラメータ型>	<省略可否>	<対象パラメータの概要説明>
...

<省略可否>で使用している記号の意味を以下に示す.

M(mandatory) : 必須(省略不可)属性, 引数, 操作

O(optional) : 省略可能属性, 引数, 操作

C(conditional) : ある条件の下で必須となる属性, 引数, 操作. 条件の詳細は, 概要説明部に記述.

<多重度>で使用している記号の意味を以下に示す.

N : 属性, 引数の多重度の上限なし.

Ord : 属性, 引数が順序付きのリスト

3.1.2. 列挙型

列挙型定義で使用している表の書式を以下に示す.

Description: <対象列挙型の概要説明>	
<定数名>	<定数の概要説明>
...	...

3.2. インターフェース定義

自律移動機能の外部インターフェース定義を以下に示す.

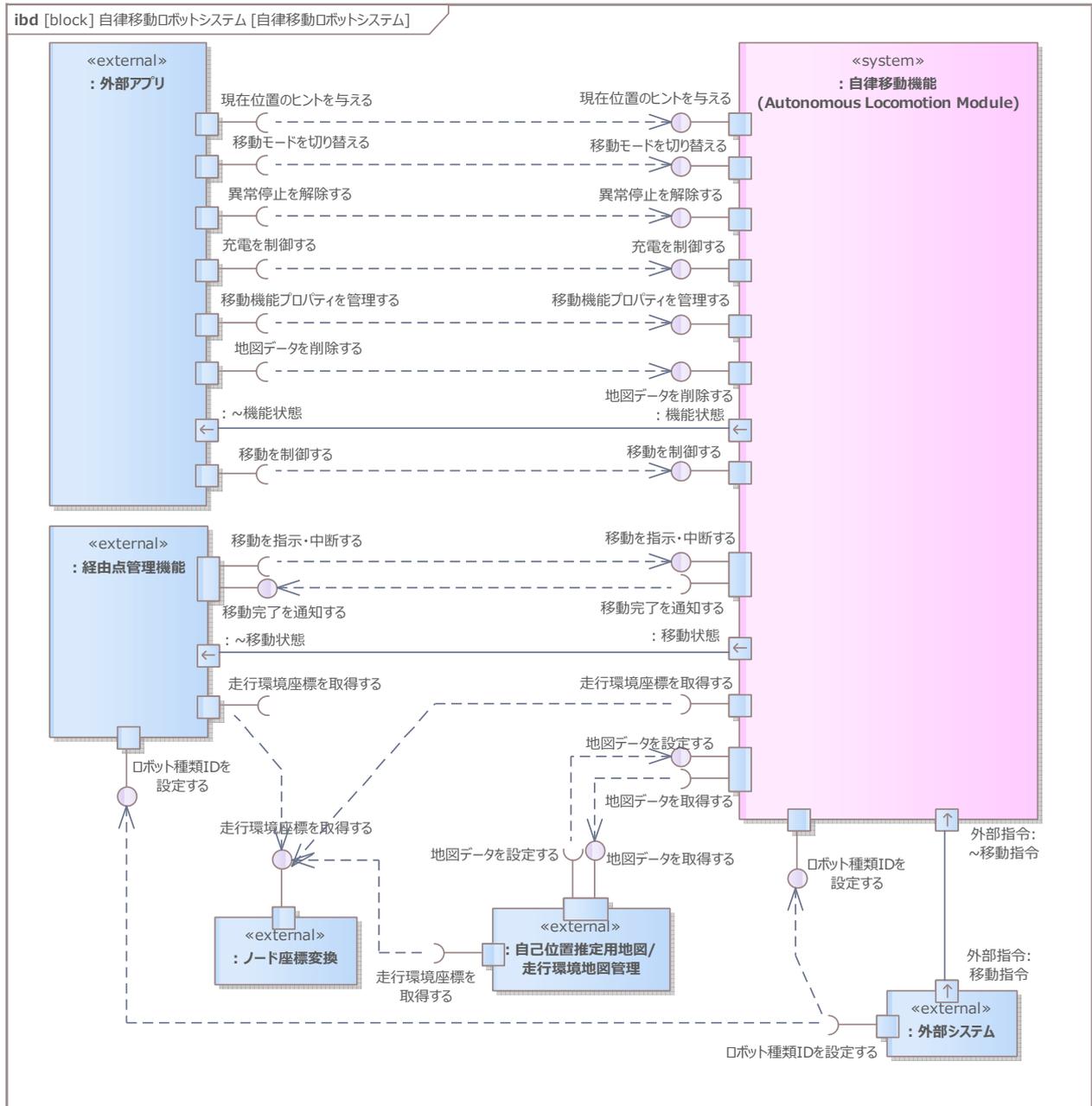
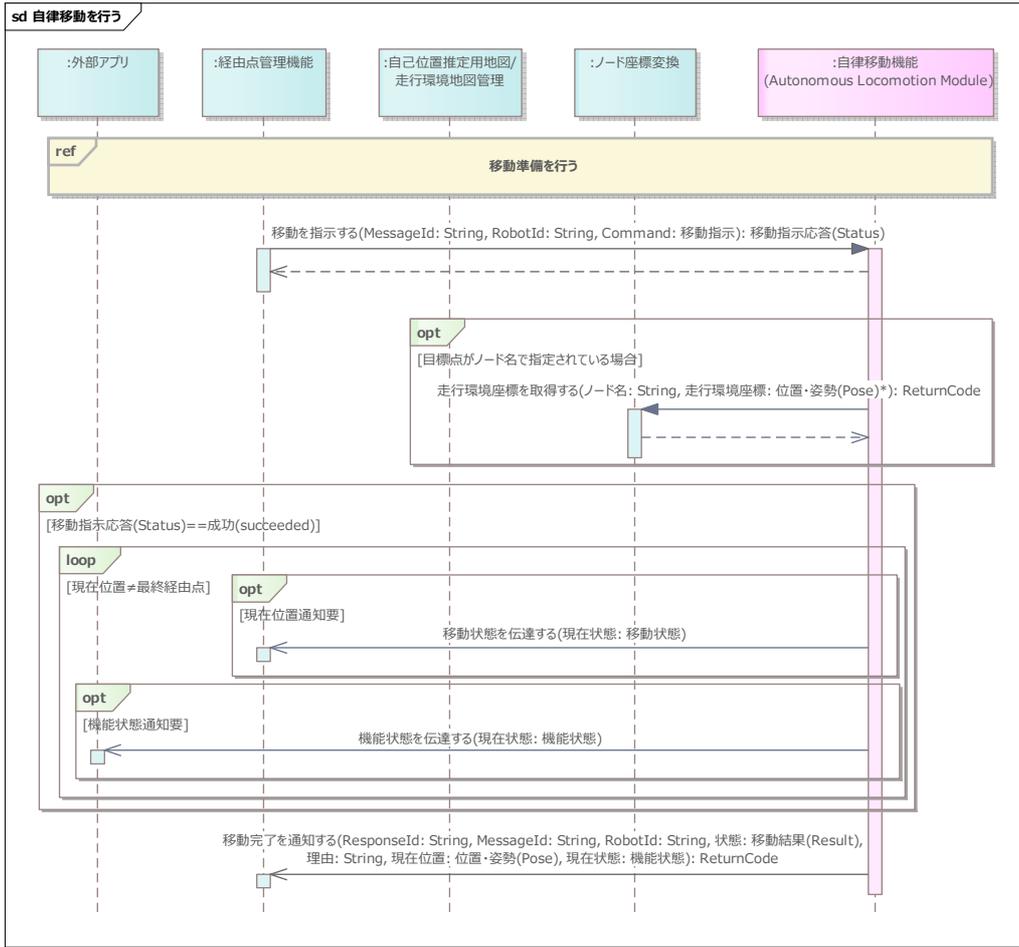


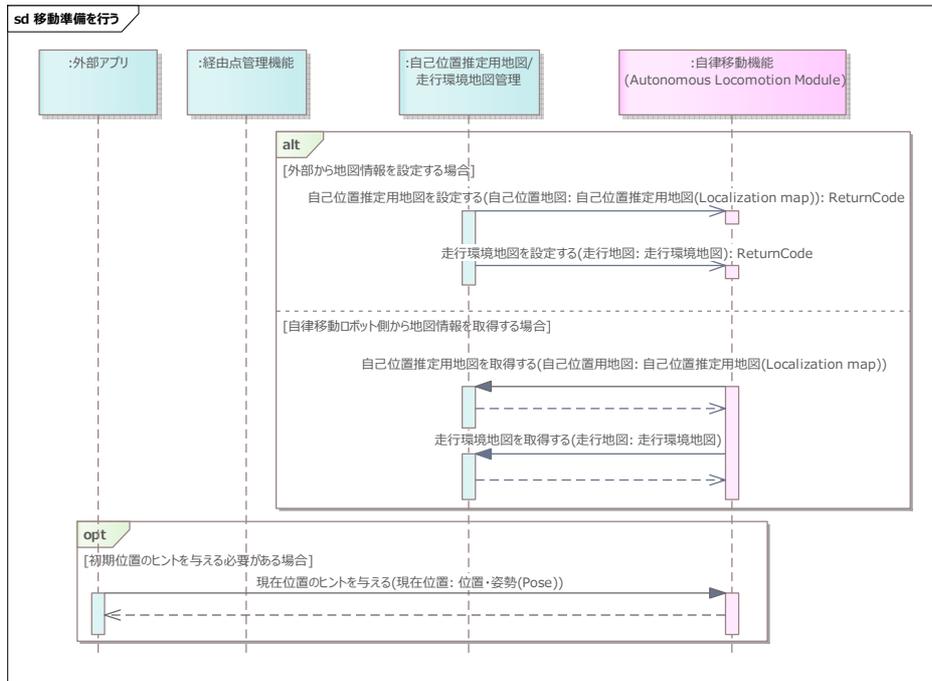
図2 外部インターフェース

なお、各インターフェースには複数のAPIが定義されている(例えば、「移動を一時停止する」は「移動を制御する」I/Fに含まれる)。各インターフェース定義の詳細については、「2.4.3インターフェース定義(イベントドリブンデータ)」を参照。

自律移動を行う際の「経路点管理機能」と「自律移動機能」間のやり取りを以下に示す。



(A)自律移動を行う



(B)移動準備を行う

図 3 自律移動実行時のシーケンス図

自律移動を実行する際には、まず「自己位置推定用地図/走行環境地図管理」から「自律移動機能」に対して走行に必要な地図情報の設定を行う。なお、地図情報は、外部要素(「自己位置推定用地図/走行環境地図管理」)から「自律移動機能」に設定する場合と、「自律移動機能」が外部要素から取得する場合の両方を想定している。

次に「自律移動機能」に対して初期位置を教えるために、「現在位置のヒントを与える」を呼び出し、「外部アプリ」から現在位置を通知する。なお、この処理は自己位置推定の精度を向上させるために、起動直後の「待機中」状態で一度だけ実行する事を想定している。また、「自律移動機能」側で初期位置の設定が不要な場合には、本処理を実行する必要はない。

そして、「経路点管理機能」から「自律移動機能」に対して、「移動を指示する」というメッセージで、移動対象のロボット、移動時の目標点を通知する。ここで、目標点が位置・姿勢ではなく、ノード名で与えられた場合には、「ノード・座標変換」に問い合わせ、指定されたノード名に対応する位置・姿勢を取得する。なお、何らかの理由により「自律移動機能」が移動指示を受けられる状態でない場合には、「移動指示応答(Status)」として「拒否(rejected)」を返し、処理は終了する。移動可能な場合には「移動指示応答(Status)」として「成功(succeeded)」を返し、移動を開始する。

自律移動ロボットが目標点まで移動している間は、「自律移動機能」側で設定した一定周期で「移動状態」を「経路点管理機能」に、「機能状態」を「外部アプリ」にそれぞれ伝達する。なお、「移動状態」「機能状態」は、「経路点管理機能」「外部アプリ」で必要とする情報のみを送信することや、それぞれ異なる送信周期を設定することも可能である。

自律移動ロボットの移動が完了(目標点に到着、もしくは何らかの理由により移動を継続できなくなる)すると、「移動完了を通知する」というメッセージで「経路点管理機能」に通知する。この際、どの移動指示に対する応答なのかを「経路点管理機能」側で識別するため、「移動を指示する」で通知されたMessageIdを「移動完了を通知する」のMessageIdに設定する。

次に「経路点管理機能」側から移動を中断する際のシーケンス図を以下に示す。

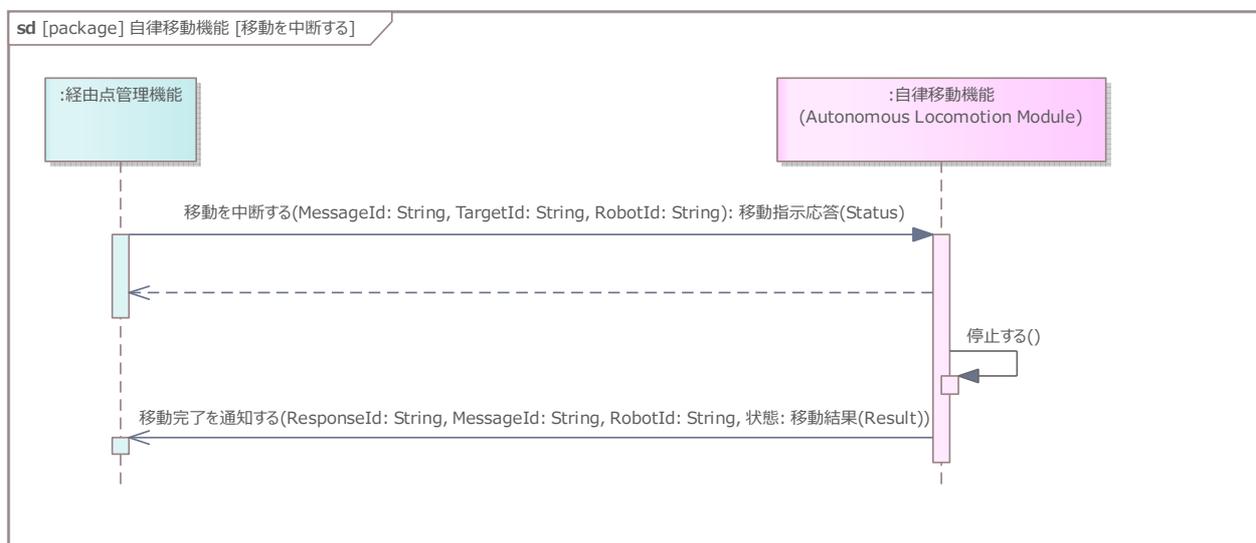


図4 自律移動中断時のシーケンス図

「経路点管理機能」側から移動を中断する場合には、「移動を中断する」を呼び出す。この際、「自律移動機能」側で、どの移動指令に対する中断指令なのかを識別するために、対応する移動指示で指定したMessageIdを「移動を中断する」のTargetIdに指定する。

「自律移動機能」側は、移動を中断し、停止した後、中断指示で指定されたMessageIdを「移動完了を通知する」のMessageIdに設定して「経路点管理機能」側に通知する。この際、移動結果には「中断(preempted)」が設定される。

次に「外部アプリ」側から自律移動を一時停止した後、手動制御で操作を行い、再度、自律移動を再開する際のシーケンス図を以下に示す。

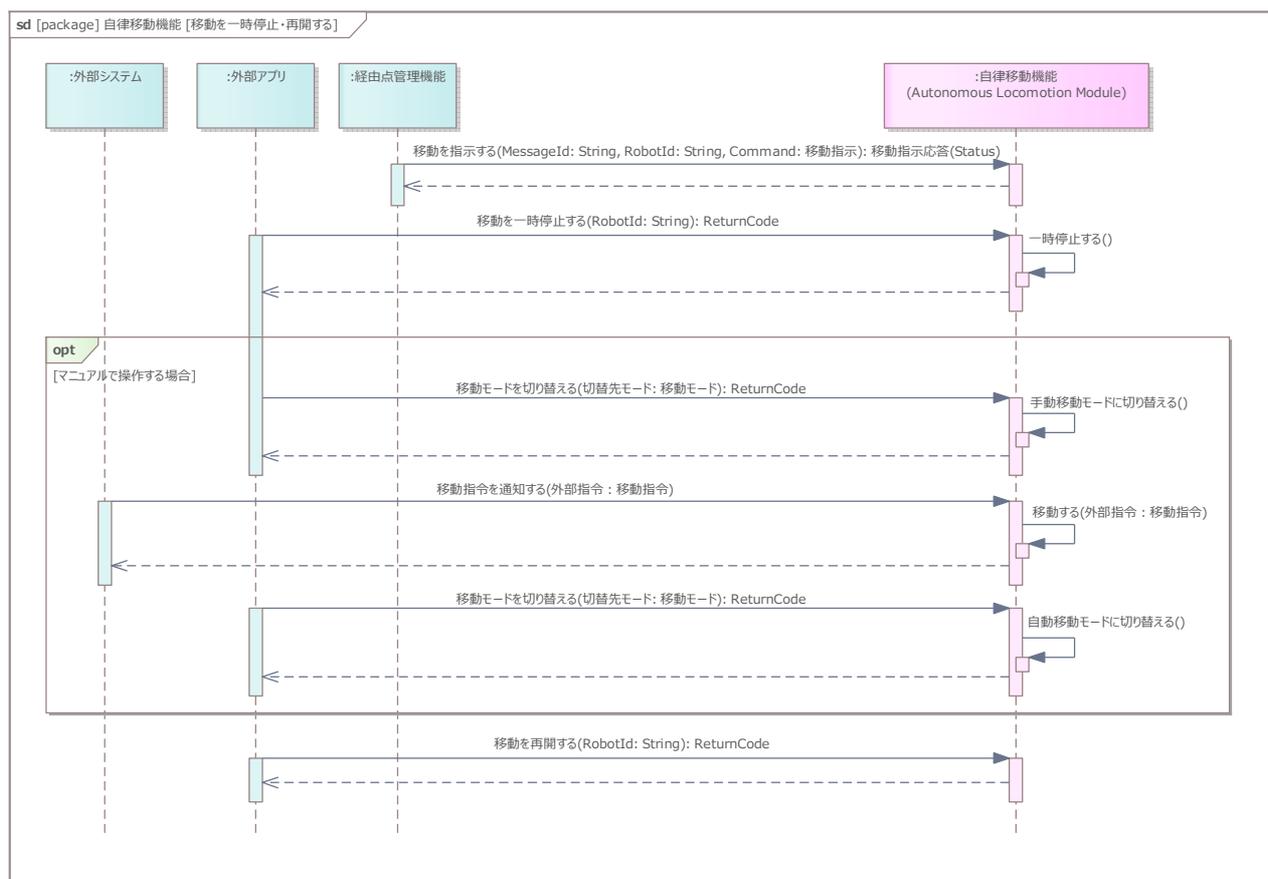


図5 自律移動一時停止・再開のシーケンス図

「外部アプリ」側から自律移動を一時停止する場合には、「移動を一時停止する」を呼び出す。

その後、「外部アプリ」側から「切替先モード」に「手動移動モード」を設定して、「移動モードを切り替える」を呼び出し、手動制御に切り替える。そして、「外部システム」から「移動指令」を直接「自律移動機能」に送信し、手動で位置の調整を行う。

位置の調整が完了した後、「切替先モード」に「自律移動モード」を設定して、「移動モードを切り替える」を呼び出し、自律制御に切り替える。その後、「外部アプリ」側から「移動を再開する」を呼び出し、一時停止前に指令されていた移動を再開する。

3.2.1. データ型定義

自律移動機能で使用するデータ型を以下に示す。

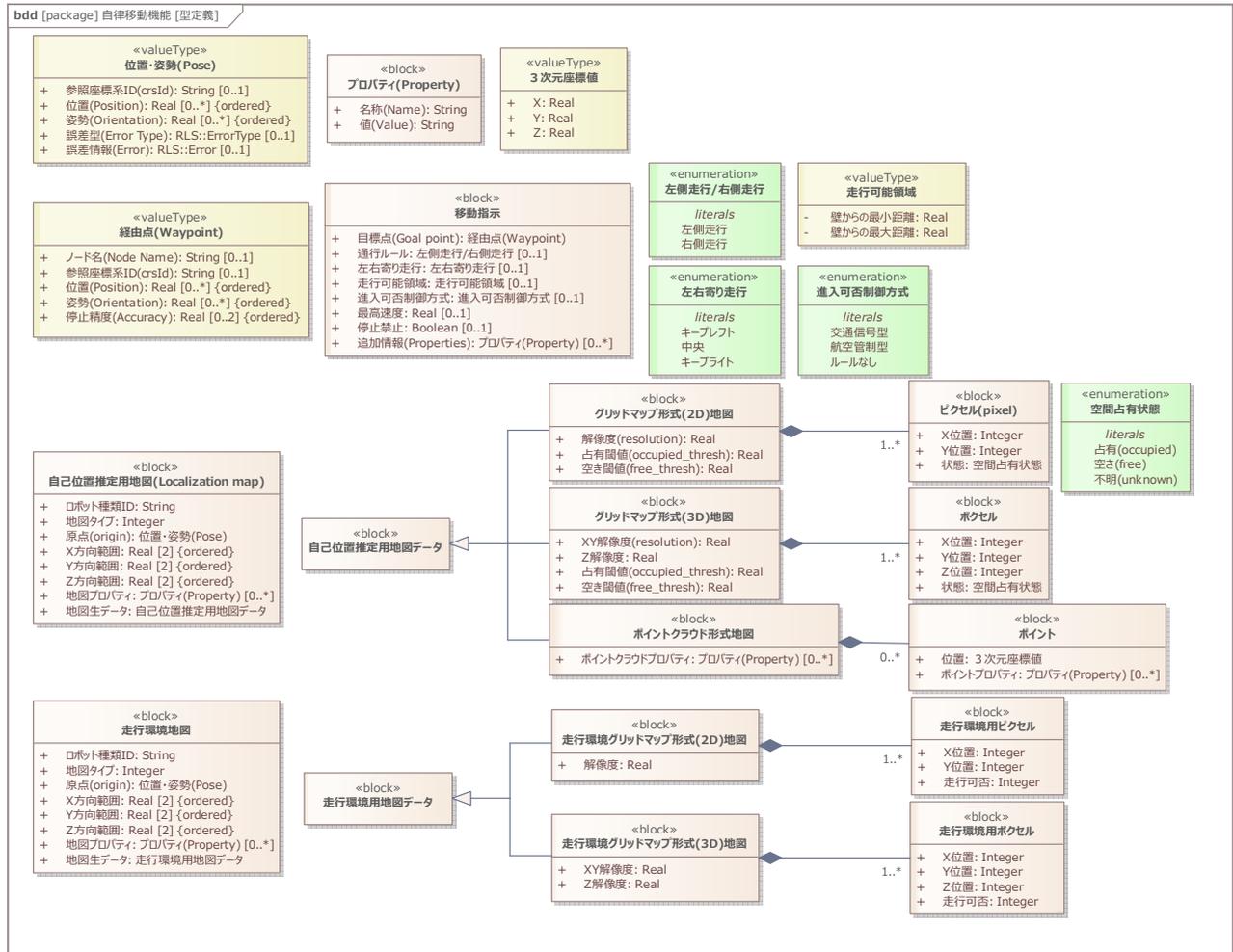


図6 データ型定義

3.2.1.1. 位置・姿勢(Pose)

Description: 設定された地図内での位置・姿勢を示す ValueType. 座標系の原点については, 事前に地図を設定する際に SIer が決定する				
Attributes				
参照座標系 ID (crsId)	String	O	1	位置, 姿勢の表現方法を定義している参照座標系 (Coordinate Reference System) の Id. 位置の表現方法としてはデカルト座標系, 極座標系, 地理座標系および相対座標系など, 姿勢の表現方法としては, オイラー角, クォータニオンなどがあるが, どの表現方法で値を保持しているのかを示す Id. 位置および姿勢で使用する単位も定義する. 関係者間で, 事前に使用する表現方法が規定されている場合には省略可能.
位置 (Position)	Real	M	0..N Ord	位置情報を表現する順序付き数値列. 各値が何を表現しているかは参照座標系 Id で定義される
姿勢 (Orientation)	Real	M	0..N Ord	姿勢情報を表現する順序付き数値列. 各値が何を表現しているかは参照座標系 Id で定義される
誤差型 (Error Type)	RLS::ErrorType	O	1	位置・姿勢の誤差の表現形式を示す. OMG Robotic Localization Service で定義されている以下の形式を指定可能. Reliability, CovarianceMatrix, Gaussian, UniformGaussian, MixtureOfGaussian, ParticleSet, LinearMixtureModel
誤差情報 (Error)	RLS::Error	O	1	位置・姿勢の誤差情報. 誤差型で指定した形式で表現

3.2.1.2. 経路点(Waypoint)

Description: 自律移動ロボットの目標位置・姿勢を示す ValueType				
Attributes				
ノード名(Node Name)	String	C	1	目標ノードの名称. 経路点をノード名で指定する場合に使用する.
参照座標系 ID (crsId)	String	O	1	位置, 姿勢の表現方法を定義している参照座標系 (Coordinate Reference System) の Id. 経路点の座標を数値で指定する場合に使用する. 事前に使用する表現方法が規定されている場合には省略可能.
位置(Position)	Real	C	0..N Ord	位置情報を表現する順序付き数値列. 経路点の座標を数値で指定する場合に使用する. 各値が何を表現しているかは参照座標系 Id で定義される
姿勢(Orientation)	Real	C	0..N Ord	姿勢情報を表現する順序付き数値列. 経路点の座標を数値で指定する場合に使用する. 各値が何を表現しているかは参照座標系 Id で定義される
停止精度(Accuracy)	Real	O	0..2 Ord	経路点に対する停止/通過精度. 位置・姿勢毎の許容範囲を設定可能. ただし, ここで設定された内容に従うかどうかは, それぞれの自律移動ロボットに依存する.

3.2.1.3 プロパティ(Property)

Description: 各種データに追加情報を設定するため型.				
Attributes				
名称(Name)	String	M	1	追加情報を識別するための名称. 使用可能な名称については, 事前に定義する必要がある.
値(Value)	String	M	1	追加情報の値.

3.2.1.4 移動指示

Description: 自律移動機能に対する移動命令を示す型				
Attributes				
目標点(Goal point)	経由点(Waypoint)	M	1	自律移動ロボットが, 到達すべき目標点.
通行ルール	左側走行/右側走行	O	1	走行時の基本ルールとして通路のどちら側を走行するかを指定.
左右寄り走行	左右寄り走行	O	1	通路の指定された「走行可能領域」内のどこを走行するかを指定.
走行可能領域	走行可能領域	O	1	自律移動ロボットが走行可能な範囲を壁からの距離で規定.
進入可否制御方式	進入可否制御方式	O	1	指定された移動目標点に進入可能かどうかを判断する方式.
最高速度	Real	O	1	指定された移動目標点まで走行する際に, 許容される最高速度.
停止禁止	Boolean	O	1	指定された移動目標点に到達する前に, 停止することが許容されているかを指定.
追加情報 (Properties)	プロパティ (Property)	O	0..N	移動指示を行う際に, 追加で指定する情報. 搬送ロボットであれば搬送対象の「荷物 Id」, 接客・給仕ロボットであれば「案内客追従確認 ON/OFF」など.

3.2.1.5 左側走行/右側走行

Description : 走行時の基本ルールとして, 通路の右側を走行するか, 左側を走行するかを設定する Enum 型	
左側走行	基本的には通路の左側を走行
右側走行	基本的には通路の右側を走行

3.2.1.6. 左右寄り走行

Description : 通路に設定された「走行可能領域」内のどちら側を走行するかを設定する Enum 型	
キープレフト	「走行可能領域」内のなるべく左側を走行
中央	「走行可能領域」内のなるべく中央を走行
キープライト	「走行可能領域」内のなるべく右側を走行

3.2.1.7. 進入可否制御方式

Description : 通路や交差点への進入可否制御をどのように行うかを表す Enum 型	
交通信号型	ロボット側の状況, 現在位置は考慮せず, 上位側で進入可能な方向を時間などで制御している方式.
航空管制型	各ロボットの状況, 現在位置を考慮した上で, 上位側で進入可能なロボットを個別に制御する方式.
ルールなし	上位側では特に進入可否制御を行わず, 各ロボットの判断に任せている方式.

3.2.1.8. 走行可能領域

Description: 現在地から指定された移動目標地点に向かう場合に、自律移動ロボットが走行可能なエリアを壁からの距離で指定するための型。同時に指定される「通行ルール」の指定に応じて、以下のように基準となる壁が変化する。

「右側走行」が指定されている場合、左側の壁からの距離

「左側走行」が指定されている場合、右側の壁からの距離

なお、「通行ルール」が指定されていない場合には、左側の壁からの距離をなる。

Attributes

壁からの最小距離	Real	M	1	基準となる壁との最小距離。自律移動ロボットが、どれだけ壁に近づいて走行しても良いかを指定する。
壁からの最大距離	Real	M	1	基準となる壁との最大距離。自律移動ロボットが、どれだけ壁から離れても走行しても良いかを指定する。

「左側走行/右側走行」の指定内容と、「走行可能距離」の関係を以下に示す。

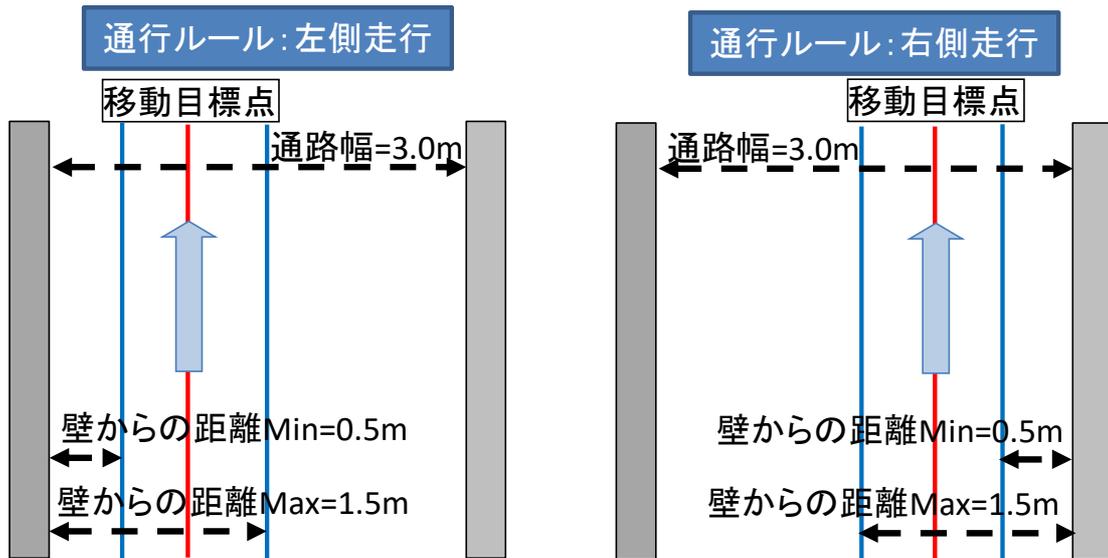


図7 「左側走行/右側走行」と「走行可能距離」の関係

3.2.1.9. 3次元座標値

Description: ポイントクラウド形式地図において、個々の点(ポイント)の位置を示す型。

Attributes

X	Real	M	1	各点の X 座標の位置
Y	Real	M	1	各点の Y 座標の位置
Z	Real	M	1	各点の Z 座標の位置

3.2.1.10. 自己位置推定用地図(Localization map)

Description: 自律移動機能が走行時に自己位置を推定するために使用する地図。SLAMなどの技術を用いて事前に作成を行う。

本地図は、各自律移動ロボットの位置を表示するために使用する場合もある。また、新たに自律移動ロボットを追加した際に、既存の地図情報を利用するために配信する場合もある。更に、自律移動機能は利用せずに、別の手法で構築される場合もある。これらの理由から、本地図は「自己位置推定用地図/走行環境地図管理」から「自律移動機能」に対して配信する形となる。グリッドマップ形式(2D)地図を使用した自己位置推定用地図のサンプルを図8に示す。

Attributes

ロボット種類 ID	String	M	1	ロボットの種類を識別するための ID。同じ地図を共用できるかどうかの判断に使用する。同一機種のロボットであっても、センサ配置や全体寸法が異なる場合には、別のロボット種類 ID となる。
地図タイプ	Integer	M	1	自己位置推定用地図として使用している地図の種類
原点(origin)	位置・姿勢(Pose)	M	1	自己位置推定用地図の原点の座標値。
X 方向範囲	Real	M	2 Ord	自己位置推定用地図が表している X 方向の範囲の最小値と最大値
Y 方向範囲	Real	M	2 Ord	自己位置推定用地図が表している Y 方向の範囲の最小値と最大値
Z 方向範囲	Real	M	2 Ord	自己位置推定用地図が表している Z 方向の範囲の最小値と最大値
地図プロパティ	プロパティ	O	0..N	各地図の追加情報。階層情報やエリア情報などを設定する
地図生データ	自己位置推定用地図データ	M	1	実際の地図データ。「地図タイプ」で指定した地図で定義された情報が格納される。

3.2.1.11. 自己位置推定用地図データ

Description: 自己位置推定用地図に設定されている地図の実データ。

Attributes: なし

3.2.1.12. グリッドマップ形式(2D)地図

Description: 2次元のグリッドマップ形式の地図。

DerivedFrom: 自己位置推定用地図データ

Attributes

解像度(resolution)	Real	M	1	地図の解像度。(m/pixel)
占有閾値(occupied_thresh)	Real	M	1	ピクセルの状態を「占有」と判断する閾値。占有確率がこの値より大きい場合、対象ピクセルは「占有」と判断される。
空き閾値(free_thresh)	Real	M	1	ピクセルの状態を「空き」と判断する閾値。占有確率がこの値より小さい場合、対象ピクセルは「空き」と判断される。
ピクセルリスト	ピクセル(pixel)	M	1..N	個々のエリアの状態を示すリスト

3.2.1.13. ピクセル(pixel)

Description: グリッドマップ形式(2D)地図の個々のエリアの状態

Attributes

状態	空間占有状態	M	1	対象エリアの状態。
X 位置	Integer	M	1	グリッドマップ形式(2D)地図内における対象エリアの X 方向(横方向)の位置。地図の左下からの相対値。
Y 位置	Integer	M	1	グリッドマップ形式(2D)地図内における対象エリアの Y 方向(縦方向)の位置。地図の左下からの相対値。

3.2.1.14. グリッドマップ形式(3D)地図

Description: 3次元のグリッドマップ形式の地図.				
DerivedFrom: 自己位置推定用地図データ				
Attributes				
XY 解像度	Real	M	1	地図の XY 方向の解像度. (m/pixel)
Z 解像度	Real	M	1	地図の Z 方向の解像度. (m/pixel)
占有閾値 (occupied_thresh)	Real	M	1	ピクセルの状態を「占有」と判断する閾値. 占有確率がこの値より大きい場合, 対象ピクセルは「占有」と判断される.
空き閾値 (free_thresh)	Real	M	1	ピクセルの状態を「空き」と判断する閾値. 占有確率がこの値より小さい場合, 対象ピクセルは「空き」と判断される.
ボクセルリスト	ボクセル	M	1..N	個々のエリアの状態を示すリスト

3.2.1.15. ボクセル

Description: グリッドマップ形式(3D)地図の個々のエリアの状態				
Attributes				
状態	空間占有状態	M	1	対象エリアの状態.
X 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの X 方向(横方向)の位置.
Y 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの Y 方向(縦方向)の位置.
Z 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの Z 方向(高さ方向)の位置.

3.2.1.16. 空間占有状態

Description: グリッドマップ形式地図の各エリアの状態を表す Enum 型	
占有(occupied)	対象エリアが占有されている状態. 障害物などが存在し, 自律移動ロボットが移動できないことを表す.
空き(free)	対象エリアが空いている状態. 自律移動ロボットが移動可能なことを表す.
不明(unknown)	対象エリアの状況を判断できなかった状態. 計測時の欠損値や手前に他の物体が存在する場合など, 対象ピクセルの状態を計測できなかったことを表す.

3.2.1.17. ポイントクラウド形式地図

Description: 周辺に存在する物体を点で表現した地図.				
DerivedFrom: 自己位置推定用地図データ				
Attributes				
ポイントクラウド プロパティ	プロパティ	O	0..N	各ポイントの「ポイントプロパティ」にどんな情報が格納されているかを設定.

3.2.1.18. ポイント

Description: ポイントクラウド形式地図の個々の点の情報				
Attributes				
位置	3次元座標値	M	1	個々の点の座標値
ポイントプロパティ	プロパティ	O	0..N	色情報(R,G,B 値)や, 反射強度など, 使用するセンサに応じた情報を格納する

3.2.1.19. 走行環境地図

Description: 自律移動機能が走行可能なエリアを判断するために使用するグリッドマップ形式の地図。自己位置推定用地図の情報を基に、移動対象エリア全体に対して走行禁止エリアをユーザが設定して作成する。自律移動機能の「経路計画(Path Planning)」が移動時に利用する。「経路計画(Path Planning)」では各種センサで計測した情報も付加した上で、移動方向の判断などを行う。なお、自律移動ロボットが移動した際に、「経路計画(Path Planning)」で使用する地図の範囲を切り替えるタイミングについては「外部アプリ」側で判断を行う(フロア切替を含む)。

本地図は「自律移動機能」側のみが保持する。ただし、自己位置推定用地図と同様に、新たに自律移動ロボットを追加した際に、既存の地図情報を利用する場合もあるため「自己位置推定用地図/走行環境地図管理」から配信する形となる。

グリッドマップ形式(2D)の走行環境地図のサンプルを図 9 に示す。

Attributes				
ロボット種類 ID	String	M	1	ロボットの種類を識別するための ID。同じ地図を共用できるかどうかの判断に使用する。同一種類のロボットであっても、センサ配置や全体寸法が異なる場合には、別のロボット種類 ID となる。
地図タイプ	Integer	M	1	走行環境地図として使用している地図の種類
原点(origin)	位置・姿勢(Pose)	M	1	走行環境地図の原点の座標値。
X 方向範囲	Real	M	2 Ord	走行環境地図が表している X 方向の範囲の最小値と最大値
Y 方向範囲	Real	M	2 Ord	走行環境地図が表している Y 方向の範囲の最小値と最大値
Z 方向範囲	Real	M	2 Ord	走行環境地図が表している Z 方向の範囲の最小値と最大値
地図生データ	走行環境地図データ	M	1	実際の地図データ。地図タイプで指定した地図で定義された情報が格納される。

3.2.1.20. 走行環境用地図データ

Description: 走行環境地図に設定されている地図の実データ。

Attributes: なし

3.2.1.21. 走行環境グリッドマップ形式(2D)地図

Description: 2次元のグリッドマップ形式の地図。

DerivedFrom: 走行環境用地図データ

Attributes

解像度(resolution)	Real	M	1	地図の解像度。 m/pixel
ピクセルリスト	ピクセル(pixel)	M	1..N	個々のエリアの状態を示すリスト

3.2.1.22. 走行環境用ピクセル(pixel)

Description : グリッドマップ形式(2D)地図の個々のエリアの状態				
Attributes				
X 位置	Integer	M	1	グリッドマップ形式(2D)地図内における対象エリアの X 方向(横方向)の位置. 地図の左下からの相対値.
Y 位置	Integer	M	1	グリッドマップ形式(2D)地図内における対象エリアの Y 方向(縦方向)の位置. 地図の左下からの相対値.
走行可否	Integer	M	1	対象領域の状態が設定される. 障害物が存在せず, 走行可能な場合には「0」が設定される. それ以外の場合には, 「1」以上の値が設定される. ただし, 使用する値, 範囲については Sier が決定する. 値の設定例を以下に示す. ・静的障害物が存在するため常に走行不可(100) ・動的障害物を検出したため一時的に走行不可(50) ・通行ルール上は走行不可だが, 障害物を回避する場合などやむを得ない場合には走行可能(25)

3.2.1.23. 走行環境グリッドマップ形式(3D)地図

Description : 3次元のグリッドマップ形式の地図.				
DerivedFrom : 走行環境用地図データ				
Attributes				
XY 解像度	Real	M	1	地図の XY 方向の解像度. m/pixel
Z 解像度	Real	M	1	地図の Z 方向の解像度. m/pixel
ボクセルリスト	ボクセル	M	1..N	個々のエリアの状態を示すリスト

3.2.1.24. 走行環境用ボクセル

Description : グリッドマップ形式(3D)地図の個々のエリアの状態				
Attributes				
X 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの X 方向(横方向)の位置.
Y 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの Y 方向(縦方向)の位置.
Z 位置	Integer	M	1	グリッドマップ形式(3D)地図内における対象エリアの Z 方向(高さ方向)の位置.
走行可否	Integer	M	1	対象領域の状態が設定される. 障害物が存在せず, 走行可能な場合には「0」が設定される. それ以外の場合には, 「1」以上の値が設定される. ただし, 使用する値, 範囲については Sier が決定する. 値の設定例を以下に示す. ・静的障害物が存在するため常に走行不可(100) ・動的障害物を検出したため一時的に走行不可(50) ・通行ルール上は走行不可だが, 障害物を回避する場合などやむを得ない場合には走行可能(25)

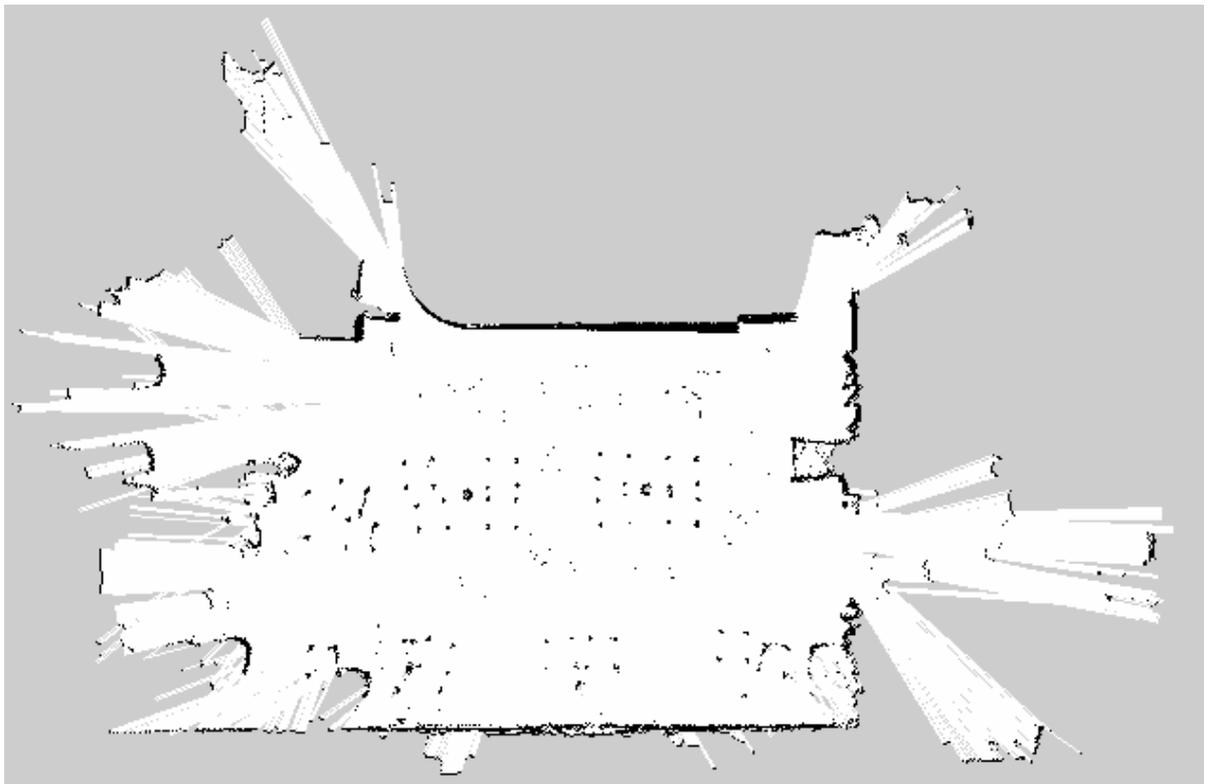


図8 自己位置推定用地図の例

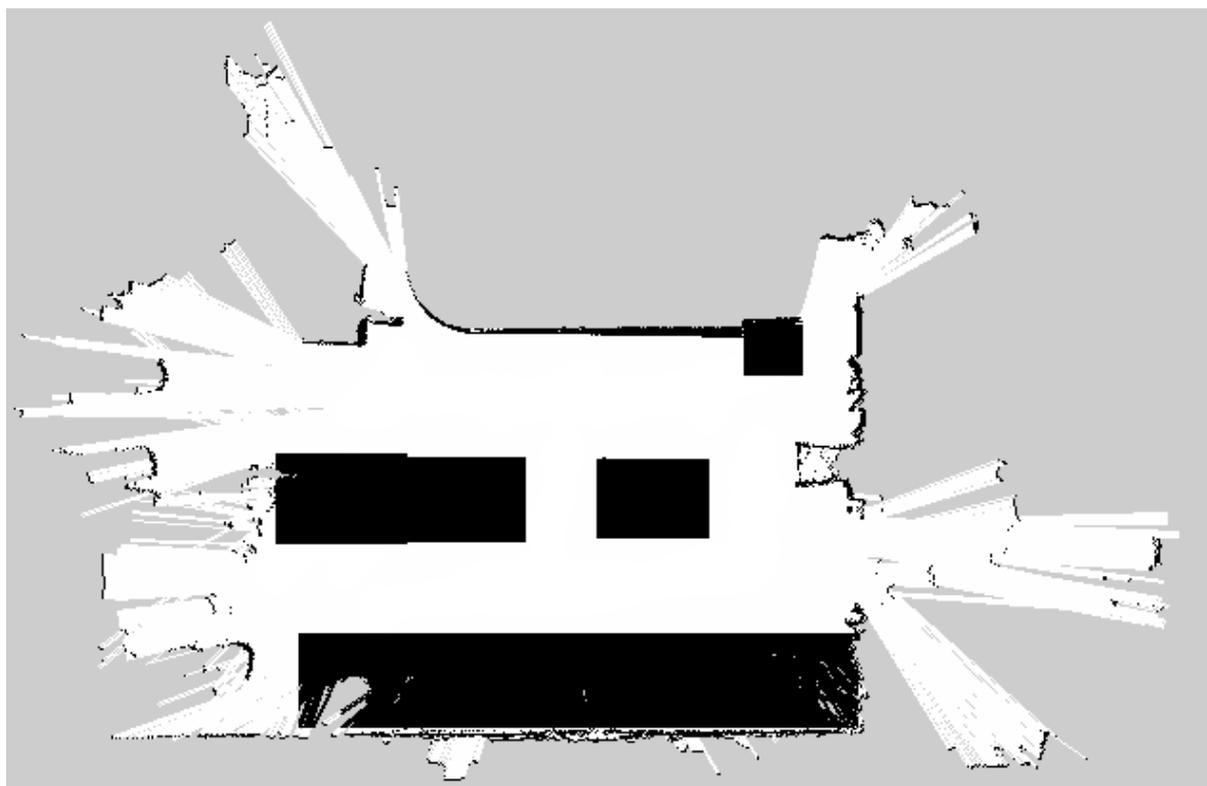


図9 走行環境地図の例

3.2.2. インターフェース定義(常時通信データ)

「自律移動機能」と「外部アプリ」「経路点管理機能」との間で、連続的にやり取りするデータの定義を以下に示す。

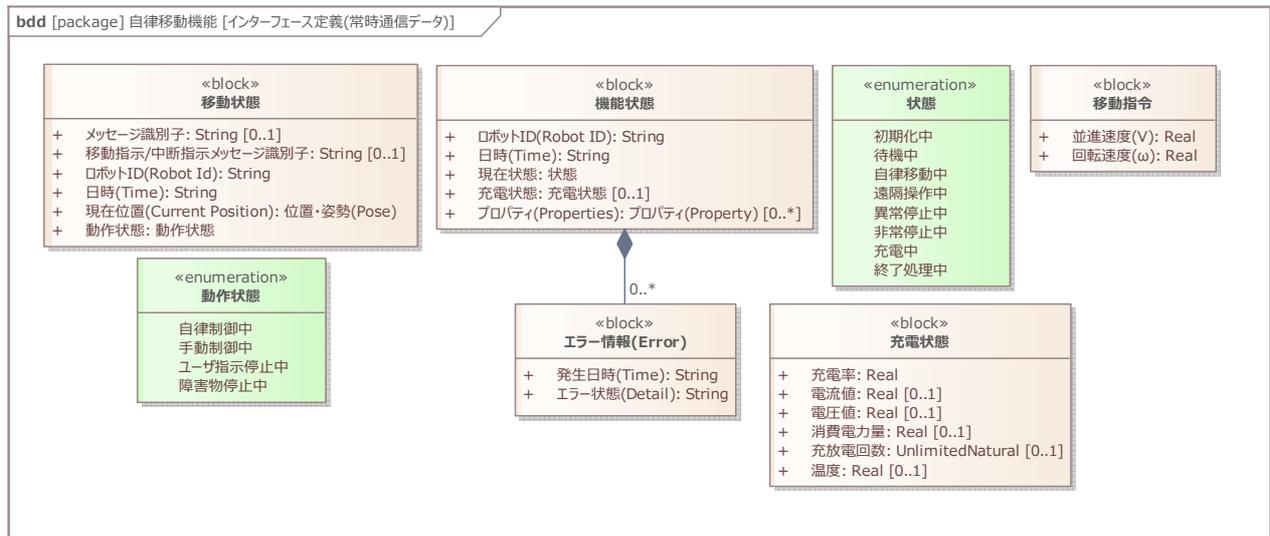


図 10 インターフェース定義(常時通信データ)

3.2.2.1 移動状態

Description:自律移動実行時の状態。「自律移動機能」の状態が「自律移動中」の場合に、定期的に「経路点管理機能」に対して送信される。

Attributes

属性名	型	必須性	範囲	説明
メッセージ識別子	String	O	0..1	メッセージ識別子。「経路点管理機能」側で「移動状態」を識別するために使用される。「経路点管理機能」側でメッセージの識別を行わない場合には、省略することができる。
移動指示/中断指示メッセージ識別子	String	O	0..1	「経路点管理機能」から受信した移動指示の識別子。現在の動作を実行する際に受け取ったメッセージの識別子。「経路点管理機能」側で、動作状態を変更したい場合に使用する。
ロボット ID	String	M	1	送信元のロボットを一意に識別するための識別子
日時	String	M	1	対象ロボットが情報を送信した日時。 ISO8601 拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ) の書式で設定
現在位置	位置・姿勢	M	1	対象ロボットの現在位置・姿勢
動作状態	動作状態	M	1	現在の「自律移動機能」の状態。「自律移動中」状態内のサブ状態を設定する。

3.2.2.2 動作状態

Description:「自律移動中」状態内のサブ状態を示す Enum 型。

自律制御中	指示された動作計画に基づき、自律的に制御している状態。
手動制御中	動作計画を実行している途中で、外部システムから手動で操作されている状態。
ユーザ指示停止中	「外部アプリ」からの指令で、動作を一時的に停止している状態。
障害物停止中	周辺障害物の影響で、動作が継続できず、一時停止している状態。

3.2.2.3. 機能状態

Description: 「外部アプリ」に通知する「自律移動機能」の内部状態を示す型。「自律移動機能」から定期的に「外部アプリ」に対して送信される。				
Attributes				
ロボット ID	String	M	1	送信元のロボットを一意に識別するための識別子。
日時	String	M	1	対象ロボットが情報を送信した日時。 ISO8601 拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ) の書式で設定
現在状態	状態	M	1	現在の「自律移動機能」の状態。
充電状態	充電状態	O	1	使用しているバッテリーの現在の充電状況。
プロパティ	プロパティ	O	0..N	「外部アプリ」に定期的に通知したい情報。 搬送ロボットであれば「搬送重量」など

3.2.2.4. エラー情報(Error)

Description: 自律移動機能内で発生したエラー情報を示す型				
Attributes				
発生日時	String	M	1	当該エラーが発生した日時。 ISO8601 拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ) の書式で設定
エラー状態	String	M	1	対象ロボットに発生したエラーの詳細情報

3.2.2.5. 状態

Description: 「自律移動機能」の現在の状態を示す Enum 型。	
初期化中	電源 ON を行った後の状態。各種初期化処理を行った後、自動で「待機中」に遷移する
待機中	「経路点管理機能」からの指示を待っている状態。
自律移動中	指示された動作計画に基づき、動作している状態。
遠隔操作中	動作計画が指示されていない段階で、外部システムから手動で操作されている状態。
異常停止中	外部から緊急停止された状態。適切なりカバー処理を実行する事で、電源 OFF を行わずに動作を継続可能な状態。
非常停止中	外部から緊急停止された状態で、電源 OFF を行わないと動作を継続できない状態。
充電中	充電を実行している状態
終了処理中	電源 OFF を行う前の状態。各種終了処理を行った後、電源が OFF となる。

3.2.2.6. 充電状態

Description: 自律移動ロボットが使用しているバッテリーの状態を示す型				
Attributes				
充電率	Real	M	1	バッテリーが完全充電された状態から放電した電気量を除いた残りの割合
電流値	Real	O	1	バッテリーが出力している電流値。
電圧値	Real	O	1	バッテリーの現在の電圧値
消費電力量	Real	O	1	バッテリーが消費した電気エネルギーの量。
充放電回数	UnlimitedNatural	O	1	バッテリーが充電され、放電するまでの流れを 1 サイクルとした場合の回数。
温度	Real	O	1	バッテリー本体の温度。

3.2.2.7. 移動指令

Description: 自律移動ロボットに対する移動方向を指定するための型。ゲーム・パッドを使用して自律移動ロボットをマニュアルで操作したい場合や、外部のセンサで計測した値を基に直接移動方向を指示したい場合など、「外部システム」から移動を指令したい場合に使用する。なお、本インターフェースは、直接自律移動ロボットを操作する場合を想定しているため、本インターフェースを用いて指令した場合は、経路計画や障害物回避などは実行されない。

Attributes

並進速度(V)	Real	M	1	自律移動ロボットの並進方向に対する速度指令
回転速度(ω)	Real	M	1	自律移動ロボットの回転方向に対する速度指令

3.2.3. インターフェース定義(イベントドリブンデータ)

「自律移動機能」と「外部アプリ」「経路点管理機能」との間で使用するインターフェースの定義を以下に示す。

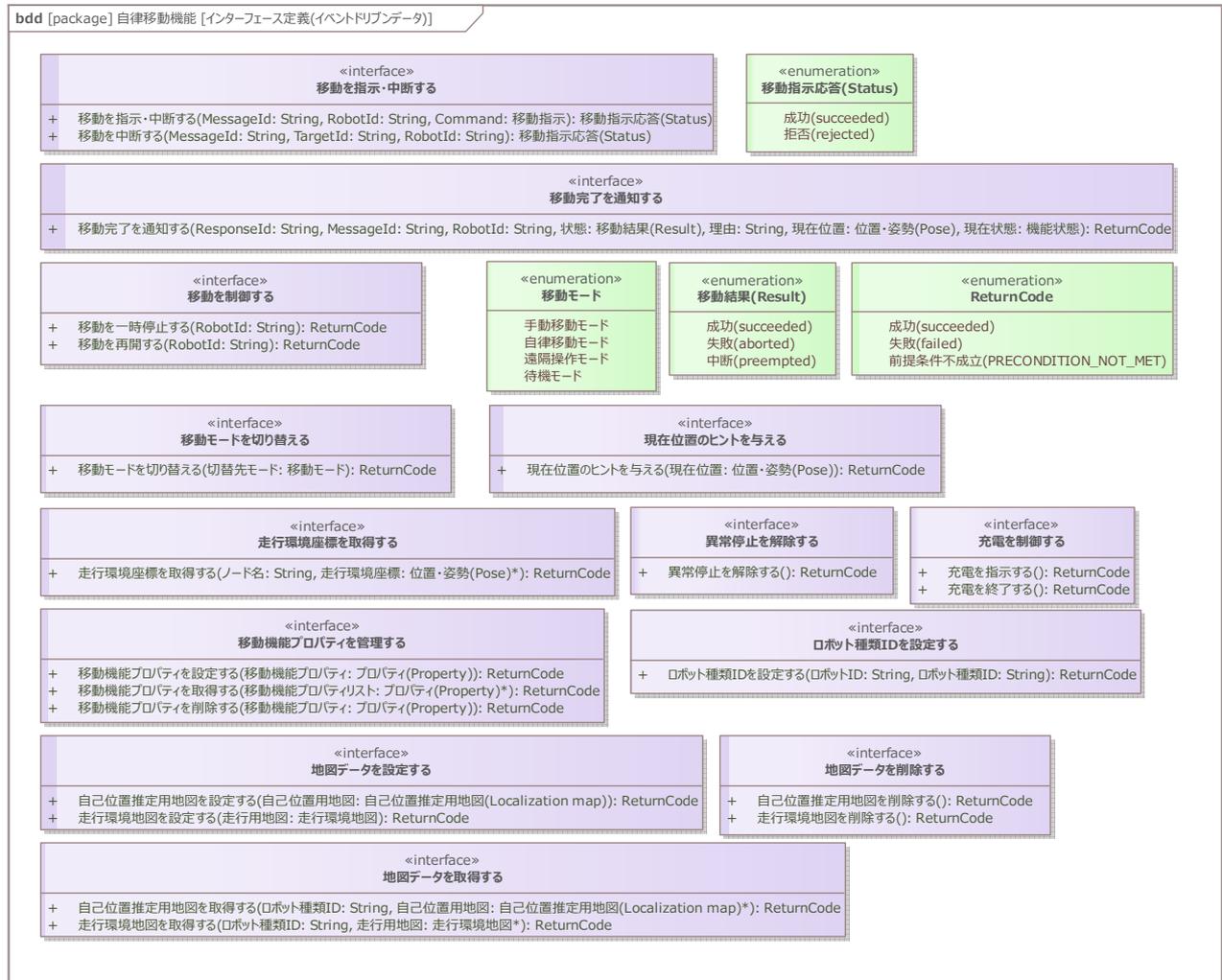


図 1 1 インターフェース定義(イベントドリブンデータ)

3.2.3.1. ReturnCode

Description: 各インターフェースの実行結果を示す Enum 型.	
成功(succeeded)	指令された動作の実行に成功.
失敗(failed)	指令された動作の実行に失敗.
前提条件不成立(PRECONDITION_NOT_MET)	前提条件が満たされていない. 指令されたコマンドを実行できない状態だった場合などに使用.

3.2.3.2. 移動を指示・中断する

Description: 「経路点管理機能」から「自律移動機能」に対して、移動の開始および中断を通知するためのインターフェース。				
Operations				
移動を指示する		M		
「経路点管理機能」から「自律移動機能」に移動先を指示する。「自律移動機能」側は、受信した後すぐに「ステータス (Status)」を「経路点管理機能」に返す (移動が完了するまで待たない)。「自律移動機能」の状態が「待機中」もしくは「自律制御中」の場合のみ発行可能。				
in	MessageId	String	M	メッセージ識別子。「自律移動機能」側で移動指示を識別するために使用される。「経路点管理機能」側では移動完了が通知された際に、どの移動指示に対する応答なのかを識別するために使用する。
in	RobotId	String	O	指示対象ロボット識別子。移動指示を送信するロボットを識別するために使用する。
in	Command	移動指示	M	「経路点管理機能」から指定された移動情報
ret		移動指示応答(Status)	M	移動指示の受信状態。「自律移動機能」の移動状態ではない
移動を中断する		M		
「経路点管理機能」から「自律移動機能」に対して、移動の中断を指示する。「自律移動機能」側は、受信した後すぐに「ステータス (Status)」を「経路点管理機能」に返す (中断が完了するまで待たない)。「自律移動機能」の状態が「自律移動中」の場合のみ発行可能。				
in	MessageId	String	M	メッセージ識別子。「自律移動機能」側で中断指示を識別するために使用される。「経路点管理機能」側では中断完了が通知された際に、どの中断指示に対する応答なのかを識別するために使用する。
in	TargetId	String	M	移動指令識別子。「自律移動機能」側で、どの移動指示に対する中断指示なのかを識別するために、「移動を指示する」で設定した MessageId を設定する。現在実行中の移動指示以外の識別子が設定されていた場合には、移動の中断は実行されない。
in	RobotId	String	O	中断対象ロボット識別子。中断指示を送信するロボットを識別するために使用する。
ret		移動指示応答(Status)	M	中断指示の受信状態。自律移動ロボットの移動状態ではない。

3.2.3.3. 移動指示応答(Status)

Description: 「経路点管理機能」「外部アプリ」からの移動指示/中断指示の受信状態を示す Enum 型	
成功(succeeded)	移動指示/中断指示を正常に受信。
拒否(rejected)	移動指示/中断指示を受けられる状態ではない

3.2.3.4. 移動完了を通知する

Description: 自律移動ロボットが指定された場所に到達した場合や、何らかの理由により移動を継続できなくなった場合、または「経路点管理機能」から移動中断指示が送られた場合に、「経路点管理機能」に「自律移動機能」の状態を通知するためのインターフェース				
Operations				
移動完了を通知する		M		
in	ResponseId	String	M	メッセージ識別子。「経路点管理機能」側で移動完了通知を識別するために使用する。
in	MessageId	String	M	移動指示/中断指示メッセージ識別子。「移動を指示する」もしくは「移動を中断する」で設定されていた MessageId を設定する。どの指示に対する応答なのかを識別するために使用する。
in	RobotId	String	O	ロボット識別子。どのロボットからの応答なのかを識別するために使用する。
in	状態	移動結果(Result)	M	自律移動ロボットの動作結果
in	理由	String	O	移動が失敗した場合や中断した場合に、その理由を設定する。
in	現在位置	位置・姿勢(Pose)	M	移動が完了した時点での現在位置・姿勢。
in	現在状態	機能状態	O	移動が完了した時点の状態
ret		ReturnCode	M	「経路点管理機能」側の通知受信結果。

3.2.3.5. 移動結果(Result)

Description: 自律移動ロボットの動作結果	
成功(succeeded)	「経路点管理機能」から指定された場所まで移動が完了。
失敗(aborted)	何らかの理由により「経路点管理機能」から指定された場所まで移動ができなかった
中断(preempted)	「経路点管理機能」からの指示で移動を途中で終了

3.2.3.6. 移動を制御する

Description: 「外部アプリ」から「自律移動機能」に対して、移動の中断/再開を指示するためのインターフェース。				
Operations				
移動を一時停止する		M		
「外部アプリ」側から移動の一時停止を指示する。「自律移動機能」側は、実際に動作を一時停止した後、「ReturnCode」を「外部アプリ」に返す。「自律移動機能」の状態が「自律制御中」の場合のみ発行可能。				
in	RobotId	String	O	指示対象ロボット識別子。一時停止指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M	一時停止の実行結果。
移動を再開する		M		
「外部アプリ」側から移動の再開を指示する。「自律移動機能」側は、実際に動作を再開した後、「ReturnCode」を「外部アプリ」に返す。「自律移動機能」の状態が「ユーザ指示停止中」の場合のみ発行可能。				
in	RobotId	String	O	指示対象ロボット識別子。再開指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M	移動再開の実行結果。

3.2.3.7. 移動モードを切り替える

Description:「外部アプリ」側から「自律移動機能」に対して、移動モードの変更を通知するためのインターフェース。指定された移動モードに遷移できたかどうかを返す。

Operations				
移動モードを切り替える		M		
in	切替先モード	移動モード	M	「自律移動機能」の遷移先のモード。
ret		ReturnCode	M	指定された移動モードへの変更結果。

3.2.3.8. 移動モード

Description: 「外部アプリ」が「自律移動機能」の移動モードを切り替える際に使用する Enum 型	
手動移動モード	外部システムから手動で操作可能なモード。「自律移動機能」の状態が「一時停止中」の場合にのみ発行可能で、「手動制御中」に遷移する。
自律移動モード	指示された動作計画に基づき自律的に動作するモード。「自律移動機能」の状態が「一時停止中」の場合にのみ発行可能で、「一時停止中」に遷移する。
遠隔操作モード	動作計画が指示されていない段階で、外部システムから手動で操作するモード。「自律移動機能」の状態が「待機中」の場合にのみ発行可能で、「遠隔操作中」に遷移する。
待機モード	「経由点管理機能」からの指示を待つモード。「自律移動機能」の状態が「遠隔操作中」の場合にのみ発行可能で、「待機中」に遷移する。

3.2.3.9. 現在位置のヒントを与える

Description:「外部アプリ」側から「自律移動機能」に対して、現在位置の情報を与えるためのインターフェース。起動時に自己位置推定のための初期位置を設定する際や地図を切り替えた際などに使用される。「自律移動機能」の状態が「待機中」の場合のみ発行可能。

Operations				
現在位置のヒントを与える		M		
in	現在位置(Current Pose)	位置・姿勢(Pose)	M	自律移動ロボットの現在位置・姿勢情報。
ret		ReturnCode	M	現在位置ヒントの受信結果。

3.2.3.10. 走行環境座標を取得する

Description:「自律移動機能」が「ノード・座標変換」から、指定されたノードの位置・姿勢情報を取得するためのインターフェース。「経由点管理機能」からノード名を使用して目標値が与えられた場合に使用する。

Operations				
走行環境座標を取得する		M		
in	ノード名	String	M	目標値として指定されたノードの名称。
out	走行環境座標	位置・姿勢(Pose)	M	指定されたノード名に対応する座標値。
ret		ReturnCode	M	ノード-座標変換の結果。

3.2.3.11. 異常停止を解除する

Description:「外部アプリ」側から「自律移動機能」の異常停止を解除するためのインターフェース。「自律移動機能」の状態が「異常停止中」の場合のみ発行可能。

Operations				
異常停止を解除する		O		
ret		ReturnCode	M	異常停止の解除結果。

3.2.3.17. 充電を制御する

Description: 「外部アプリ」側から「自律移動機能」の充電を指示するためのインターフェース。本インターフェースの実装はオプション。				
Operations				
充電を指示する		O		
充電を開始することを指示する。「自律移動機能」側は、実際に充電を開始した後、「ReturnCode」を「外部アプリ」に返す。「自律移動機能」の状態が「待機中」の場合のみ発行可能。				
ret		ReturnCode	M	充電開始の実行結果。
充電を終了する		O		
充電を終了することを指示する。「自律移動機能」側は、実際に充電を終了した後、「ReturnCode」を「外部アプリ」に返す。「自律移動機能」の状態が「充電中」の場合のみ発行可能。				
ret		ReturnCode	M	充電開始の実行結果。

3.2.3.13. 移動機能プロパティを管理する

Description: 「自律移動機能」が内部に保持する移動機能プロパティの取得/設定/削除を行うためのインターフェース。本インターフェースの実装はオプション。				
Operations				
移動機能プロパティを設定する		O		
「外部アプリ」側から移動機能プロパティを与える。				
in	移動機能プロパティ	プロパティ	M	設定対象の移動機能プロパティ。
ret		ReturnCode	M	移動機能プロパティの設定結果。
移動機能プロパティを取得する		O		
「自律移動機能」が保持する移動機能プロパティを全て取得する。				
out	移動機能プロパティリスト	プロパティ	0..N	「自律移動機能」が保持している移動機能プロパティのリスト。
ret		ReturnCode	M	移動機能プロパティの取得結果。
移動機能プロパティを削除する		O		
指定された移動機能プロパティを「自律移動機能」から削除する。				
in	移動機能プロパティ	プロパティ	M	削除対象の移動機能プロパティ。
ret		ReturnCode	M	移動機能プロパティの削除結果。

3.2.3.14. ロボット種類 ID を設定する

Description: 「外部システム」から「自律移動機能」に対して、ロボット種類 ID を設定するためのインターフェース。同一機種 of ロボットであっても、全体寸法が異なる場合には、別のロボット種類 ID となるため、例えば積載物の形状が変化した場合などは、ロボット種類 ID が変わることになる。このような場合に、新たなロボット種類 ID を通知するために使用する。				
Operations				
ロボット種類 ID を設定する		M		
in	ロボット ID	String	M	各ロボットを一意に識別するための識別子
in	ロボット種類 ID	String	M	ロボットの状況を基に、使用する地図を指定するための ID
ret		ReturnCode	M	ロボット種類 ID の設定結果。

3.2.3.15. 地図データを設定する

Description:「自己位置推定用地図/走行環境地図管理」から「自律移動機能」に対して、地図データを設定するためのインターフェース。「地図データを取得する」インターフェースが実装されている場合には、本インターフェースは不要。

Operations

自己位置推定用地図を設定する				C
in	自己位置用地図	自己位置推定用地図	M	「自律移動機能」が走行時に自己位置を推定するために使用するグリッドマップ形式の地図
ret		ReturnCode	M	自己位置用地図の設定結果.
走行環境地図を設定する				C
in	走行用地図	走行環境地図	M	「自律移動機能」が走行可能なエリアを判断するために使用するグリッドマップ形式の地図.
ret		ReturnCode	M	走行環境地図の設定結果.

3.2.3.16. 地図データを取得する

Description:「自律移動機能」が「自己位置推定用地図/走行環境地図管理」から、地図データを取得するためのインターフェース。「地図データを設定する」インターフェースが実装されている場合には、本インターフェースは不要。

Operations

自己位置推定用地図を取得する				C
in	ロボット種類ID	String	M	ロボットの種類を識別するための ID.
out	自己位置用地図	自己位置推定用地図	M	「自律移動機能」が走行時に自己位置を推定するために使用するグリッドマップ形式の地図
ret		ReturnCode	M	自己位置用地図の取得結果.
走行環境地図を取得する				C
in	ロボット種類ID	String	M	ロボットの種類を識別するための ID.
out	走行用地図	走行環境地図	M	自律移動機能が走行可能なエリアを判断するために使用するグリッドマップ形式の地図.
ret		ReturnCode	M	走行環境地図の取得結果.

3.2.3.17. 地図データを削除する

Description:「外部アプリ」側から「自律移動機能」内部の地図データを削除するためのインターフェース.

Operations

自己位置推定用地図を削除する				O
ret		ReturnCode	M	自己位置推定地図の削除結果.
走行環境地図を削除する				O
ret		ReturnCode	M	走行環境地図の削除結果.

4 Platform Specific Model (PSM)

4.1. ROS PSM

3章のPIM(Platform Independent Model:プラットフォーム独立モデル)で定義したインターフェース仕様に基づいたROS(Robot Operating System:ROS1)向けのPSM(Platform Specific Model:プラットフォーム特化モデル)を示す。

4.1.1. SysML-ROS 変換用マッピングルール

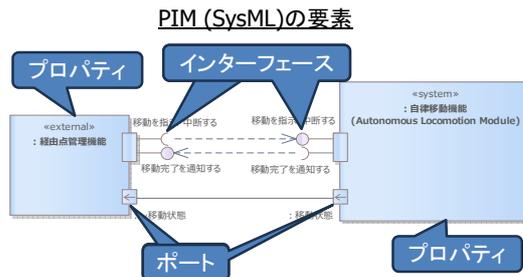
PIMで使用している SysML の要素やデータ型を ROS で使用可能な要素に変換するためにマッピングルール(変換ルール)を以下のように定義した。

4.1.1.1.要素間のマッピングルール

SysML の要素と ROS 要素は、以下の対応関係で変換した。

PIM (SysML)	PSM (ROS)
プロパティ	ノード
ポート	トピック
インターフェース	サービス

※PIM(SysML)の「ポート」は、「3.2.2.インターフェース 定義(常時通信データ)」で定義している内容。



4.1.1.2.データ型のマッピングルール

SysML のプリミティブ型と ROS のデータ型は、以下の対応関係で変換した。

PIM (SysML)	PSM (ROS)
UnlimitedNatural	uint16
Integer	int16
Real	float32
Boolean	bool
String	string
Enum	int16

※UnlimitedNatural, Integer については、データの格納個数から 16bit で設定した。

プリミティブ型以外のデータ定義については、ROS で広く使用されているメッセージ型をなるべく使用する形とした。このため「5.2.1.データ型定義」で定義しているデータ定義の一部は以下の対応関係で変換した。

PIM (SysML)	PSM (ROS)	参考 URL
位置	geometry_msgs/Point	https://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Point.html
姿勢	geometry_msgs/Quaternion	https://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Quaternion.html
位置・姿勢	geometry_msgs/Pose	https://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Pose.html

4.1.1.3.その他

ROS の制約に対応するため、以下のような変換ルールも用いた。

- ・インターフェース定義において、引数がない場合は `std_msgs/Empty` を使用する。
(参考 URL:https://docs.ros.org/en/noetic/api/std_msgs/html/msg/Empty.html)
- ・ROS のメッセージでは、汎化関係を定義できないため、汎化を使用している部分については、具象クラスの全てのデータを持つ型を定義する。
(状況に応じて、使用しないデータについては `None` を指定する運用とする)
- ・メッセージ定義ファイルの名称、各パラメータの名称については ROS で規定されているコーディング・ルールに従って命名を行う。
(参考 ROS wiki msg <https://wiki.ros.org/msg>, ROS wiki srv <https://wiki.ros.org/srv>)

4.1.2. 定義ファイル

4.1.1 のマッピングルールを用いて変換した ROS のメッセージファイルを以下に示す.

4.1.2.1. データ型定義

○Pose.msg(3.2.1.1. 位置・姿勢)

#位置・姿勢	
string crs_id	#参照座標系 ID
geometry_msgs/Point position	#位置
geometry_msgs/Quaternion orientation	#姿勢
string error_type	#誤差型
string error	#誤差情報

○Waypoint.msg(3.2.1.2. 経由点)

#経由点	
string node_name	#ノード名
string crs_id	#参照座標系 ID
geometry_msgs/Point position	#位置
geometry_msgs/Quaternion orientation	#姿勢
float32[] accuracy	#停止精度

○Property.msg(3.2.1.3. プロパティ)

#プロパティ	
string name	#名称
string value	#値

○MoveCommand.msg(3.2.1.4. 移動指示)

#移動指示	
Waypoint goal_point	#目標点
int16 traffic_rules	#通行ルール
int16 drive_side	#左右寄り走行
DrivingArea driving_area	#走行可能領域
int16 entry_control_method	#進入可否制御方式
float32 max_speed	#最高速度
bool no_stop_allowed	#停止禁止
Property[] properties	#追加情報

○DrivingArea.msg(3.2.1.8. 走行可能領域)

#走行可能領域	
float32 min_distance	#壁からの最小距離
float32 max_distance	#壁からの最大距離

○Coordinate3D.msg(3.2.1.9. 3次元座標値)

#3次元座標値	
float32 x	#X
float32 y	#Y
float32 z	#Z

○LocalizationMap.msg(3.2.1.10.自己位置推定用地図)

```
#自己位置推定用地図
string robot_type_id          #ロボット種類 ID
int16 map_type                #地図タイプ
Pose origin                   #原点
float32 x_range               # X 方向範囲
float32 y_range               # Y 方向範囲
float32 z_range               # Z 方向範囲
Property[] map_properties    #地図プロパティ
LocalizationMapData localization_map_data #地図生データ
```

○LocalizationMapData.msg(3.2.1.11.自己位置推定用地図データ)

```
#走行環境用地図データ
GridMap2D grid_map_2D        #グリッドマップ形式(2D)地図
GridMap3D grid_map_3D        #グリッドマップ形式(3D)地図
PointCloudMap point_cloud_map #ポイントクラウド形式地図
```

○GridMap2D.msg(3.2.1.12.グリッドマップ形式(2D)地図)

```
#グリッドマップ形式(2D)地図
float32 resolution          #解像度
float32 occupied_thresh     #占有閾値
float32 free_thresh         #空き閾値
Pixel pixel_list            #ピクセルリスト
```

○Pixel.msg(3.2.1.13.ピクセル)

```
#ピクセル
int16 state                 #状態
int16 x_position            # X 位置
int16 y_position            # Y 位置
```

○GridMap3D.msg(3.2.1.14.グリッドマップ形式(3D)地図)

```
#グリッドマップ形式(3D)地図
float32 xy_resolution       # XY 解像度
float32 z_resolution        # Z 解像度
float32 occupied_thresh     #占有閾値
float32 free_thresh         #空き閾値
Voxel voxel_list            #ボクセルリスト
```

○Voxel.msg(3.2.1.15.ボクセル)

```
#ボクセル
int16 state                 #状態
int16 x_position            # X 位置
int16 y_position            # Y 位置
int16 z_position            # Z 位置
```

○PointCloudMap.msg(3.2.1.17.ポイントクラウド形式地図)

```
#ポイントクラウド形式地図
Property[] point_cloud_properties #ポイントクラウドプロパティ
```

○Point.msg(3.2.1.18.ポイント)

```
#ポイント
Coordinate3D point          #位置
Property[] point_properties #ポイントプロパティ
```

○NavigationMap.msg(3.2.1.19. 走行環境地図)

# 走行環境地図	
string robot_type_id	# ロボット種類 ID
int16 map_type	# 地図タイプ
Pose origin	# 原点
float32 x_range	# X 方向範囲
float32 y_range	# Y 方向範囲
float32 z_range	# Z 方向範囲
NavigationMapData navigation_map_data	# 地図生データ

○NavigationMapData.msg(3.2.1.20. 走行環境用地図データ)

# 走行環境用地図データ	
NavigationGridMap2D navigation_grid_map_2D	# 走行環境グリッドマップ形式(2D)地図
NavigationGridMap3D navigation_grid_map_3D	# 走行環境グリッドマップ形式(3D)地図

○NavigationGridMap2D.msg(3.2.1.21. 走行環境グリッドマップ形式(2D)地図)

# 走行環境グリッドマップ形式(2D)地図	
float32 resolution	# 解像度
NavigationPixel pixel_list	# ピクセルリスト

○NavigationPixel.msg(3.2.1.22. 走行環境用ピクセル)

# 走行環境用ピクセル	
int16 x_position	# X 位置
int16 y_position	# Y 位置
int16 can_drive	# 走行可否

○NavigationGridMap3D.msg(3.2.1.23. 走行環境グリッドマップ形式(3D)地図)

# 走行環境グリッドマップ形式(3D)地図	
float32 xy_resolution	# XY 解像度
float32 z_resolution	# Z 解像度
NavigationVoxel voxel_list	# ボクセルリスト

○NavigationVoxel.msg(3.2.1.24. 走行環境用ボクセル)

# 走行環境用ボクセル	
int16 x_position	# X 位置
int16 y_position	# Y 位置
int16 z_position	# Z 位置
int16 can_drive	# 走行可否

4.1.2.2. データ型定義(常時通信データ)

○MoveState.msg(3.2.2.1.移動状態)

#移動状態	
string message_identifier	#メッセージ識別子
string move_command_message_identifier	#移動指示/中断指示メッセージ識別子
string robot_id	#ロボット ID
string timestamp	#日時
Pose current_pose	#現在位置
int16 operation_state	#動作状態

○LocomotionState.msg(3.2.2.3.機能状態)

#機能状態	
string robot_id	#ロボット ID
string timestamp	#日時
int16 current_state	#現在状態
ChargeState charge_state	#充電状態
Property[] properties	#プロパティ

○ErrorInformation.msg(3.2.2.4.エラー情報)

#エラー情報	
string timestamp	#発生日時
string error_information	#エラー情報

○ChargeState.msg(3.2.2.6.充電状態)

#充電状態	
float32 battery_level	#充電率
float32 current	#電流値
float32 voltage	#電圧値
float32 power_consumption	#消費電力量
uint16 charge_discharge_cycles	#充放電回数
float32 temp	#温度

○MoveInstruction.msg(3.2.2.7.移動指令)

#移動指令	
float32 linear_velocity	#並進速度
float32 angular_velocity	#回転速度

4.1.2.3. サービス定義(イベントドリブンデータ)

○SendMoveCommand.srv(3.2.3.2.移動を指示・中断する: 移動を指示する)

```
#移動を指示する
string message_id      # MessageId
string robot_id        # RobotId
MoveCommand move_command # Command
---
int16 status           #移動指示応答
```

○CancelMove.srv(3.2.3.2.移動を指示・中断する: 移動を中断する)

```
#移動を中断する
string message_id      # MessageId
string target_id       # TargetId
string robot_id        # RobotId
---
int16 status           #移動指示応答
```

○NotifyMoveCompletion.srv(3.2.3.4.移動完了を通知する)

```
#移動完了を通知する
string response_id     # ResponseId
string message_id      # MessageId
string robot_id        # RobotId
int16 state            #状態
string reason          #理由
Pose current_pose      #現在位置
LocomotionState current_state #現在状態
---
int16 return_code      # ReturnCode
```

○PauseMove.srv(3.2.3.6.移動を制御する: 移動を一時停止する)

```
#移動を一時停止する
string robot_id        # RobotId
---
int16 return_code      # ReturnCode
```

○ResumeMove.srv(3.2.3.6.移動を制御する: 移動を再開する)

```
#移動を再開する
string robot_id        # RobotId
---
int16 return_code      # ReturnCode
```

○SetMoveMode.srv(3.2.3.7.移動モードを切り替える)

```
#移動モードを切り替える
int16 target_mode      #切替先モード
---
int16 return_code      # ReturnCode
```

○SetCurrentPoseHint.srv(3.2.3.9.現在位置のヒントを与える)

```
#現在位置のヒントを与える
Pose current_pose      #現在位置
---
int16 return_code      # ReturnCode
```

○GetEnvironmentCoordinates.srv(3.2.3.10.走行環境座標を取得する)

```
#走行環境座標を取得する
string node_name          #ノード名
---
Pose environment_coordinates #走行環境座標
int16 return_code        # ReturnCode
```

○ClearEmergencyHalt.srv(3.2.3.11.異常停止を解除する)

```
#異常停止を解除する
std_msgs/Empty empty     #引数なし
---
int16 return_code        # ReturnCode
```

○StartCharging.srv(3.2.3.12.充電を制御する: 充電を指示する)

```
#充電を指示する
std_msgs/Empty empty     #引数なし
---
int16 return_code        # ReturnCode
```

○StopCharging.srv(3.2.3.12.充電を制御する: 充電を終了する)

```
#充電を終了する
std_msgs/Empty empty     #引数なし
---
int16 return_code        # ReturnCode
```

○SetMoveProperty.srv(3.2.3.13.移動機能プロパティを管理する: 移動機能プロパティを設定する)

```
#移動機能プロパティを設定する
Property move_property    #移動機能プロパティ
---
int16 return_code        # ReturnCode
```

○GetMoveProperty.srv(3.2.3.13.移動機能プロパティを管理する: 移動機能プロパティを取得する)

```
#移動機能プロパティを取得する
std_msgs/Empty empty     #引数なし
---
Property[] move_property_list #移動機能プロパティリスト
int16 return_code        # ReturnCode
```

○RemoveMoveProperty.srv(3.2.3.13.移動機能プロパティを管理する: 移動機能プロパティを削除する)

```
#移動機能プロパティを削除する
Property move_property    #移動機能プロパティ
---
int16 return_code        # ReturnCode
```

○SetRobotType.srv(3.2.3.14.ロボット種類 ID を設定する)

```
#ロボット種類 ID を設定する
string robot_id           #ロボット ID
string robot_type_id      #ロボット種類 ID
---
int16 return_code        # ReturnCode
```

○SetLocalizationMap.srv(3.2.3.15. 地図データを設定する: 自己位置推定用地図を設定する)

```
#自己位置推定用地図を設定する
LocalizationMap localization_map      #自己位置用地図
---
int16 return_code                    # ReturnCode
```

○SetNavigationMap.srv(3.2.3.15. 地図データを設定する: 走行環境地図を設定する)

```
#走行環境地図を設定する
NavigationMap navigation_map          #走行用地図
---
int16 return_code                    # ReturnCode
```

○GetLocalizationMap.srv(3.2.3.16. 地図データを取得する: 自己位置推定用地図を取得する)

```
#自己位置推定用地図を取得する
string robot_type_id                 #ロボット種類 ID
---
LocalizationMap localization_map     #自己位置用地図
int16 return_code                    # ReturnCode
```

○GetNavigationMap.srv(3.2.3.16. 地図データを取得する: 走行環境地図を取得する)

```
#走行環境地図を取得する
string robot_type_id                 #ロボット種類 ID
---
NavigationMap navigation_map         #走行用地図
int16 return_code                    # ReturnCode
```

○ClearNavigationMap.srv(3.2.3.17. 地図データを削除する: 自己位置推定用地図を削除する)

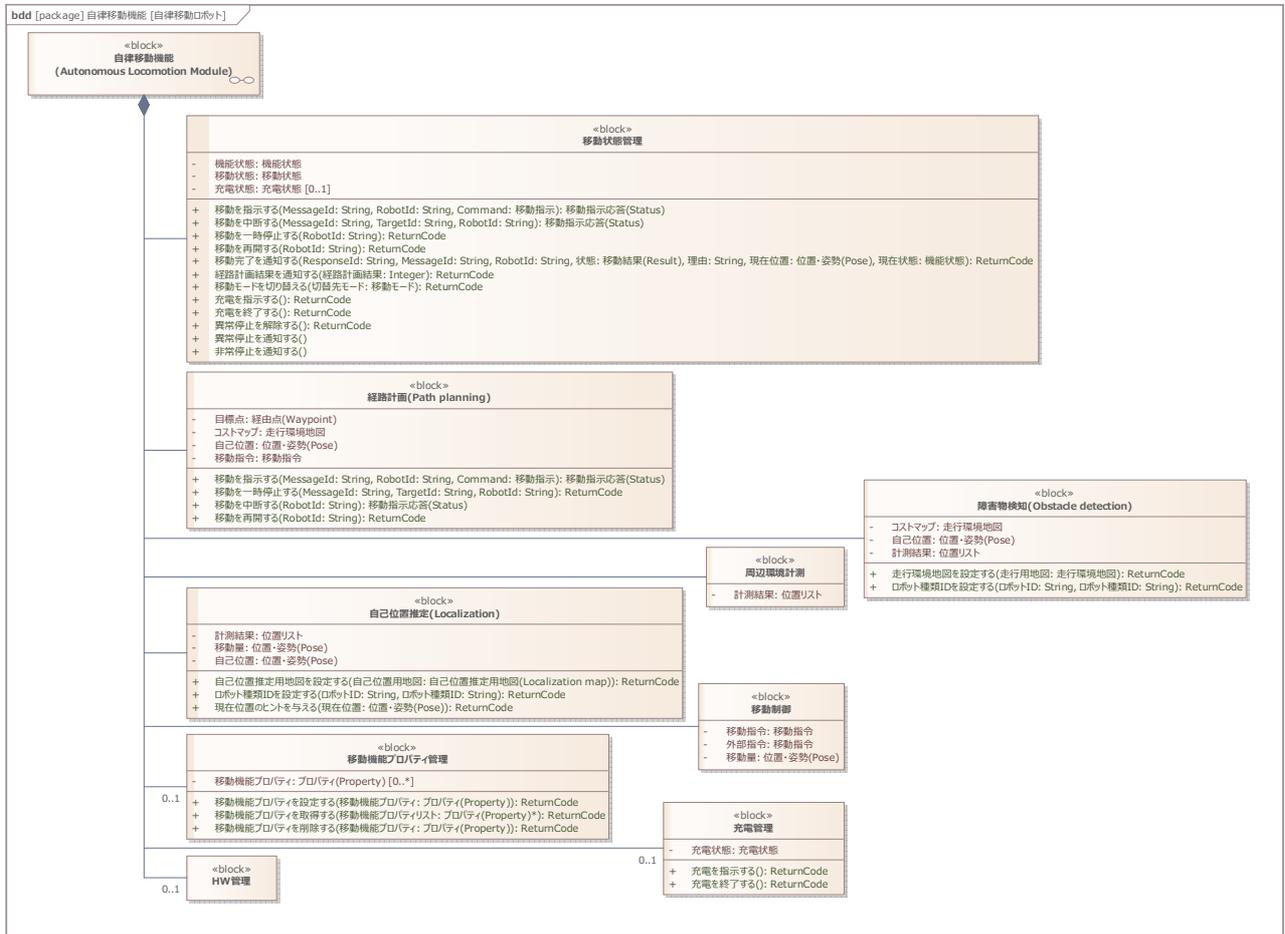
```
#自己位置推定用地図を削除する
std_msgs/Empty empty                #引数なし
---
int16 return_code                    # ReturnCode
```

○ClearNavigationMap.srv(3.2.3.17. 地図データを削除する: 走行環境地図を削除する)

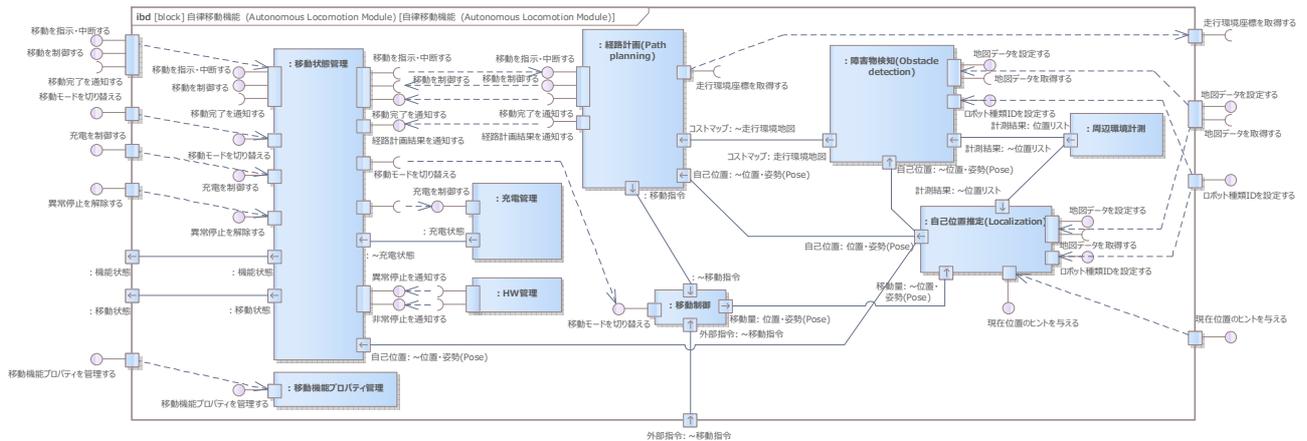
```
#走行環境地図を削除する
std_msgs/Empty empty                #引数なし
---
int16 return_code                    # ReturnCode
```

Appendix.1.内部構成の設計例

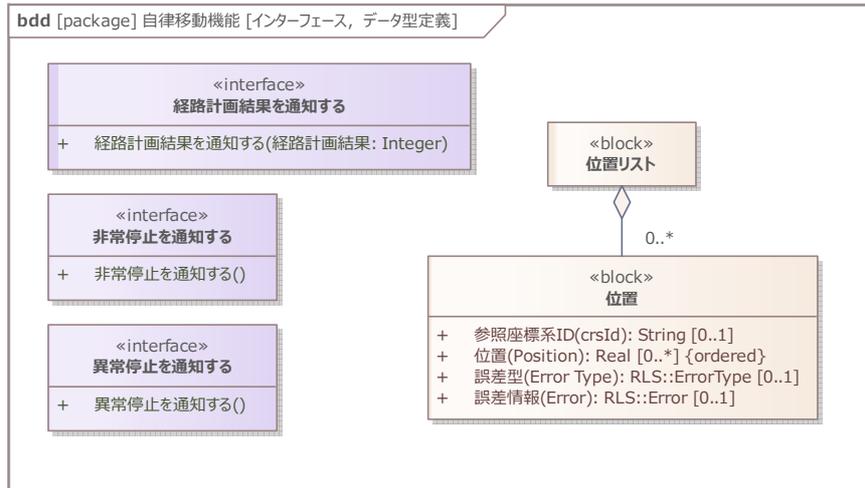
自律移動機能の内部構成およびそれらの接続関係の設計例を以下に示す。



図A-1-1 内部構成



図A-1-2 内部構成要素の接続関係



図A-1-3 内部構成要素で使用しているインターフェース，データ型定義

A.1.1. 移動状態管理

Description :「自律移動機能」の全体の振る舞いを管理する要素。「自律移動機能」の状態を管理し、「経由点管理機能」など外部からの指令の実行可否などを判断する。なお、指示された指令が実行可能な場合には、各指令の内容に応じて、適切に他の要素に動作指令を送信する。 また、適切な周期で「経由点管理機能」「外部アプリ」に対し、常時通信データの送信も行う。				
Attributes				
機能状態	機能状態	M	1	「外部アプリ」に通知する「自律移動機能」の内部状態
移動状態	移動状態	M	1	自律移動実行時の状態。「自律移動機能」の状態が「自律移動中」の場合に、定期的に「経由点管理機能」に対して送信される。
充電状態	充電状態	O	1	「充電管理」から通知されたバッテリーの充電状態。
Operations				
移動を指示する		M		「経由点管理機能」から移動目標の「目標点」を受信し、移動を開始するためのインターフェース。「移動指示」を受信した後、内部状態を確認し、すぐに「ステータス(Status)」を「経由点管理機能」側に返す(実際に移動が完了するのは待たない)。状態が「待機中」もしくは「自律制御中」の場合のみ受信可能。
in	MessageId	String	M	メッセージ識別子。移動指示を識別するために使用する。
in	RobotId	String	O	指示対象ロボット識別子。移動指示を送信するロボットを識別するために使用する。
in	Command	移動指示	M	「経由点管理機能」から指定された移動情報
ret		移動指示応答(Status)	M	移動指示の受信状態。自律移動ロボットの移動状態ではない。
移動を中断する		M		「経由点管理機能」から移動を中断するためのインターフェース。受信した後、内部状態を確認し、すぐに「ステータス(Status)」を「経由点管理機能」側に返す(実際に移動が中断されるのは待たない)。状態が「自律移動中」の場合のみ受信可能。
in	MessageId	String	M	メッセージ識別子。中断指示を識別するために使用する。
in	TargetId	String	M	移動指令識別子。「経路計画」でどの移動指示に対する中断指示なのかを識別するために使用する。「移動を指示する」で通知された MessageId が設定される。
in	RobotId	String	O	中断対象ロボット識別子。中断指示を送信するロボットを識別するために使用する。
ret		移動指示応答(Status)	M	中断指示の受信状態。自律移動ロボットの移動状態ではない。

移動を一時停止する		M	「外部アプリ」側から移動の一時停止を指示するためのインターフェース。「経路計画」が実際に動作を一時停止した後、「ReturnCode」を「外部アプリ」に返す。状態が「自律制御中」の場合のみ受信可能。
in	RobotId	String	O 指示対象ロボット識別子。一時停止指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M 一時停止の実行結果。
移動を再開する		M	「外部アプリ」側から移動の再開を指示するためのインターフェース。「経路計画」が実際に動作を再開した後、「ReturnCode」を「外部アプリ」に返す。状態が「ユーザ指示一時停止中」の場合のみ受信可能。
in	RobotId	String	O 指示対象ロボット識別子。再開指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M 移動再開の実行結果。
移動完了を通知する		M	「経路計画」から移動指示の実行結果を受信するためのインターフェース
in	ResponseId	String	M メッセージ識別子。「経路点管理機能」側で移動完了通知を識別するために使用する。
in	MessageId	String	M 移動指示/中断指示メッセージ識別子。「移動を指示する」もしくは「移動を中断する」で設定されていた MessageId を設定する。どの指示に対する応答なのかを識別するために使用する。
in	RobotId	String	O ロボット識別子。どのロボットからの応答なのかを識別するために使用する。
in	状態	移動結果(Result)	M 自律移動ロボットの動作結果
in	理由	String	O 移動が失敗した場合や中断した場合に、その理由を設定する。
in	現在位置	位置・姿勢(Pose)	M 移動が完了した時点での現在位置・姿勢。
in	現在状態	機能状態	O 移動が完了した時点の状態
ret		ReturnCode	M 「移動状態管理」の通知受信結果。
経路計画結果を通知する		M	「経路計画」から「移動状態管理」に対して、経路計画の実行結果を通知するためのインターフェース。周辺に対応不能障害物が存在するために経路計画に失敗し、「障害物停止中」に遷移する場合や、対応不能障害物が除去されたことで経路計画に成功し、「自律制御中」に遷移する場合など、「自律移動機能」の状態遷移に関連するイベントを通知するためのインターフェース。
in	経路計画結果	Integer	M 「経路計画」の経路計画の実行結果。「自律移動機能」の状態遷移に関連するイベントを設定する。
ret		ReturnCode	M 経路計画の実行結果の受信結果。
移動モードを切り替える		M	「外部アプリ」側から「自律移動機能」に対して、移動モードの変更を通知するためのインターフェース。
in	切替先モード	移動モード	M 「自律移動機能」の遷移先のモード。
ret		ReturnCode	M 指定された移動モードへの変更結果。
充電を指示する		O	充電を開始することを指示するためのインターフェース。実際に充電を開始した後、「ReturnCode」を「外部アプリ」に返す。本インターフェースはオプション。内部要素に「充電管理」が存在する場合に実装可能。
充電を終了する		O	充電を終了することを指示するためのインターフェース。実際に充電を終了した後、「ReturnCode」を「外部アプリ」に返す。本インターフェースはオプション。内部要素に「充電管理」が存在する場合に実装可能。
異常停止を解除する		O	「外部アプリ」側から異常停止を解除するためのインターフェース。
異常停止を通知する		O	「HW 管理」から異常停止の発生を受信するためのインターフェース。本インターフェースはオプション。内部要素に「HW 管理」が存在する場合に実装可能。
非常停止を通知する		O	「HW 管理」から非常停止の発生を受信するためのインターフェース。本インターフェースはオプション。内部要素に「HW 管理」が存在する場合に実装可能。

A.1.2.経路計画(Path planning)

<p>Description : コストマップを用いて、現在位置から指定された目標点に移動するための「移動指令」を生成する要素。</p> <p>「経由点管理機能」から「移動状態管理」経由で指定された「目標点」、「自己位置推定」が推定した「自己位置」および「障害物検知」が生成した「コストマップ」を用いて、自律移動ロボットの「並進速度」「回転速度」を算出し、「移動制御」に対して「移動指示」を伝達する。「目標点」に「自己位置」が到達した場合には、「移動完了を通知する」を用いて「移動状態管理」に対して目標点到着を通知する。</p>				
Attributes				
目標点	経由点(Waypoint)	M	1	「移動状態管理」から指定された目標地点
コストマップ	走行環境地図	M	1	「障害物検知」が生成した現状の周辺環境の地図。移動禁止領域や周辺に存在する障害物の情報が反映された地図
自己位置	位置・姿勢(Pose)	M	1	「自己位置推定」が算出した現在の自律移動ロボットの位置・姿勢。
移動指令	移動指令	M	1	「移動制御」に対して通知する自律移動ロボットの並進速度、回転速度指令。
Operations				
移動を指示する		M	「移動状態管理」から移動目標点を受信し、移動を開始するためのインターフェース。「移動指示」を受信した後、内部状態を確認し、すぐに「ステータス(Status)」を「移動状態管理」側に返す(実際に移動が完了するのは待たない)。	
in	MessageId	String	M	メッセージ識別子。移動指示を識別するために使用する。
in	RobotId	String	O	指示対象ロボット識別子。移動指示を送信するロボットを識別するために使用する。
in	Command	移動指示	M	「移動状態管理」から指定された移動情報
ret		移動指示応答(Status)	M	移動指示の受信状態。自律移動ロボットの移動状態ではない。
移動を中断する		M	「移動状態管理」から移動を中断するためのインターフェース。受信した後、内部状態を確認し、すぐに「ステータス(Status)」を「移動状態管理」側に返す(実際に移動が中断されるのは待たない)。	
in	MessageId	String	M	メッセージ識別子。中断指示を識別するために使用する。
in	TargetId	String	M	移動指令識別子。どの移動指示に対する中断指示なのかを識別するために使用する。「移動を指示する」で通知された MessageId が設定される。
in	RobotId	String	O	中断対象ロボット識別子。中断指示を送信するロボットを識別するために使用する。
ret		移動指示応答(Status)	M	中断指示の受信状態。自律移動ロボットの移動状態ではない。
移動を一時停止する		M	「移動状態管理」側から移動の一時停止を指示するためのインターフェース。実際に動作を一時停止した後、「ReturnCode」を「移動状態管理」に返す。	
in	RobotId	String	O	指示対象ロボット識別子。一時停止指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M	一時停止の実行結果。
移動を再開する		M	「移動状態管理」側から移動の再開を指示するためのインターフェース。実際に動作を再開した後、「ReturnCode」を「移動状態管理」に返す。	
in	RobotId	String	O	指示対象ロボット識別子。再開指示を送信するロボットを識別するために使用する。
ret		ReturnCode	M	移動再開の実行結果。

A.1.3. 位置リスト

Description: 自律移動ロボットの周辺に存在する物体群の位置を示す型				
Attributes:				
計測リスト	位置	O	O..N	個々の計測位置を示すリスト

A.1.4. 位置

Description: 指定された座標系内での位置を示す型. 座標系の原点については, 事前に Sler が決定する				
Attributes				
参照座標系 ID (crsId)	String	O	1	位置の表現方法を定義している参照座標系(Coordinate Reference System)の Id. 位置の表現方法としてはデカルト座標系, 極座標系, 地理座標系および相対座標系などがあるが, どの表現方法で値を保持しているのかを示す Id. 位置で使用する単位も定義する. 関係者間で, 事前に使用する表現方法が規定されている場合には省略可能.
位置 (Position)	Real	M	0..N Ord	位置情報を表現する順序付き数値列. 各値が何を表現しているかはフレーム Id で定義される
誤差型 (Error Type)	RLS::ErrorType	O	1	位置の誤差の表現形式を示す. OMG Robotic Localization Service で定義されている以下の形式を指定可能. Reliability, CovarianceMatrix, Gaussian, UniformGaussian, MixtureOfGaussian, ParticleSet, LinearMixtureModel
誤差情報 (Error)	RLS::Error	O	1	位置の誤差情報. 誤差型で指定した形式で表現

A.1.5. 周辺環境計測

Description: 外界センサを用いて自律移動ロボット周辺の環境を計測する要素. 本仕様書では計測に使用する外界センサのハードウェアは特に規定していない. 計測した結果は, 誤差情報を含む位置データのリストとして, 「障害物検知」「自己位置推定」に伝達される.				
Attributes				
計測結果	位置リスト	M	1	外界センサを用いて計測した周辺に存在する物体の位置情報のリスト
Operations なし				

A.1.6. 障害物検知(Obstacle detection)

Description: 設定された周辺環境地図の情報と周辺環境の計測結果から「コストマップ」を生成する要素. 「自己位置推定用地図/走行環境地図管理」から設定された走行環境地図と「周辺環境計測」から伝達された「計測結果」および「自己位置推定」から伝達された「自己位置」を用いて, 現在の自律移動ロボットの周辺環境を表す「コストマップ」を作成する. 作成した「コストマップ」は「経路計画」に伝達される.				
Attributes				
コストマップ	走行環境地図	M	1	走行環境地図に設定された移動禁止領域や周辺に存在する障害物の情報が反映された地図
自己位置	位置・姿勢(Pose)	M	1	「自己位置推定」が算出した現在の自律移動ロボットの位置・姿勢
計測結果	位置リスト	M	1	「周辺環境計測」が計測した自律移動ロボットの周辺に存在する物体の位置情報のリスト.
Operations				
走行環境地図を設定する		O	「走行環境地図」を設定するためのインターフェース. 「自律移動機能」側から地図データを取得する場合には, 本インターフェースは不要.	
in	走行用地図	走行環境地図	M	「自律移動機能」が走行可能なエリアを設定したグリッドマップ形式の地図.
ret		ReturnCode	M	走行環境地図の設定結果.

ロボット種類 ID を設定する		O	「ロボット種類 ID」を設定するためのインターフェース。「自律移動機能」側から地図データを取得する場合には、本インターフェースは不要。	
in	ロボット ID	String	M	各ロボットを一意に識別するための識別子
in	ロボット種類 ID	String	M	ロボットの状況を基に、使用する地図を指定するための ID
ret		ReturnCode	M	ロボット種類 ID の設定結果。

A.1.7.自己位置推定(Localization)

Description ：自律移動ロボットの現在位置を推定する要素。 「自己位置推定用地図/走行環境地図管理」から設定された自己位置推定用地図と「周辺環境計測」から伝達された「計測結果」、 「移動制御」から伝達された「移動量」を用いて、自律移動ロボットの現在位置の推定を行う。推定した「自己位置」は「経路計画」および「障害物検知」「移動状態管理」に伝達される。 また、起動時に自律移動ロボットの初期位置の設定が必要な場合や使用している地図が切り替えられた場合には、「外部アプリ」から現在位置のヒントが通知される。				
Attributes				
計測結果	位置リスト	M	1	「周辺環境計測」が計測した自律移動ロボットの周辺に存在する物体の位置情報のリスト。
移動量	位置・姿勢(Pose)	M	1	「移動制御」から一定周期で伝達される自律移動ロボットの移動量。前回の位置からの差分データとして伝達される。
自己位置	位置・姿勢(Pose)	M	1	伝達された各種情報を用いて推定した自律移動ロボットの現在位置・姿勢。
Operations				
自己位置推定用地図を設定する		O	「自己位置推定用地図」を設定するためのインターフェース。「自律移動機能」側から地図データを取得する場合には、本インターフェースは不要。	
in	自己位置用地図	自己位置推定用地図	M	自律移動機能が走行時に自己位置を推定するために使用する地図
ret		ReturnCode	M	自己位置推定用地図の設定結果。
ロボット種類 ID を設定する		O	「ロボット種類 ID」を設定するためのインターフェース。「自律移動機能」側から地図データを取得する場合には、本インターフェースは不要。	
in	ロボット ID	String	M	各ロボットを一意に識別するための識別子
in	ロボット種類 ID	String	M	ロボットの状況を基に、使用する地図を指定するための ID
ret		ReturnCode	M	ロボット種類 ID の設定結果。

A.1.8.移動制御

Description ：移動機構に対して制御指令を伝達し、自律移動ロボットの実際の移動を制御する要素。 「経路計画」もしくは「外部システム」から指定された「移動指令」を基に、自律移動ロボットの並進・回転移動速度を判断し、移動機構の制御を行う。				
Attributes				
移動指令	移動指令	M	1	「経路計画」で算出された移動指令。
外部指令	移動指令	O	0..1	「手動速度入力」から伝達された移動指令。ユーザが自律移動ロボットを手動で操作する際に設定される。
移動量	位置・姿勢(Pose)	M	1	自律移動ロボットの実際の移動量。「自己位置推定」に伝達される。本情報は、前回の位置からの差分量が一定周期で伝達される。
Operations なし				

A.1.9.移動機能プロパティ管理

Description : 「自律移動機能」が内部で使用する各種移動機能プロパティを管理する要素. 自律移動ロボットが使用するセンサのキャリブレーション値などを格納する.				
Attributes				
移動機能プロパティ	プロパティ (Property)	O	0..N	「自律移動機能」が使用する各種設定値.
Operations				
移動機能プロパティを設定する		O	「外部アプリ」側から「自律移動機能」に対して、移動機能プロパティを設定するためのインターフェース.	
in	移動機能プロパティ	プロパティ (Property)	M	設定対象の移動機能プロパティ.
ret		ReturnCode	M	移動機能プロパティの設定結果.
移動機能プロパティを取得する		O	「外部アプリ」が「自律移動機能」に設定されている移動機能プロパティを取得するためのインターフェース.	
out	移動機能プロパティ リスト	プロパティ (Property)	0..N	取得した移動機能プロパティのリスト.
ret		ReturnCode	M	移動機能プロパティの取得結果.
移動機能プロパティを削除する		O	「外部アプリ」側から「自律移動機能」に設定されている移動機能プロパティを削除するためのインターフェース.	
in	移動機能プロパティ	プロパティ (Property)	M	削除対象の移動機能プロパティ
ret		ReturnCode	M	移動機能プロパティの削除結果.

A.1.10.充電管理

Description : 自律移動ロボットが使用するバッテリーの充電を制御する要素. バッテリーの現在状態のモニタリングも行う.				
Attributes				
充電状態	充電状態	M	1	バッテリーの現在の充電状態.
Operations				
充電を指示する		O	「外部アプリ」側から「移動状態管理」経由でバッテリーの充電開始を指示するためのインターフェース.	
ret		ReturnCode	M	バッテリー充電開始の実行結果.
充電を終了する		O	「外部アプリ」側から「移動状態管理」経由でバッテリーの充電停止を指示するためのインターフェース.	
ret		ReturnCode	M	バッテリー充電停止の実行結果.

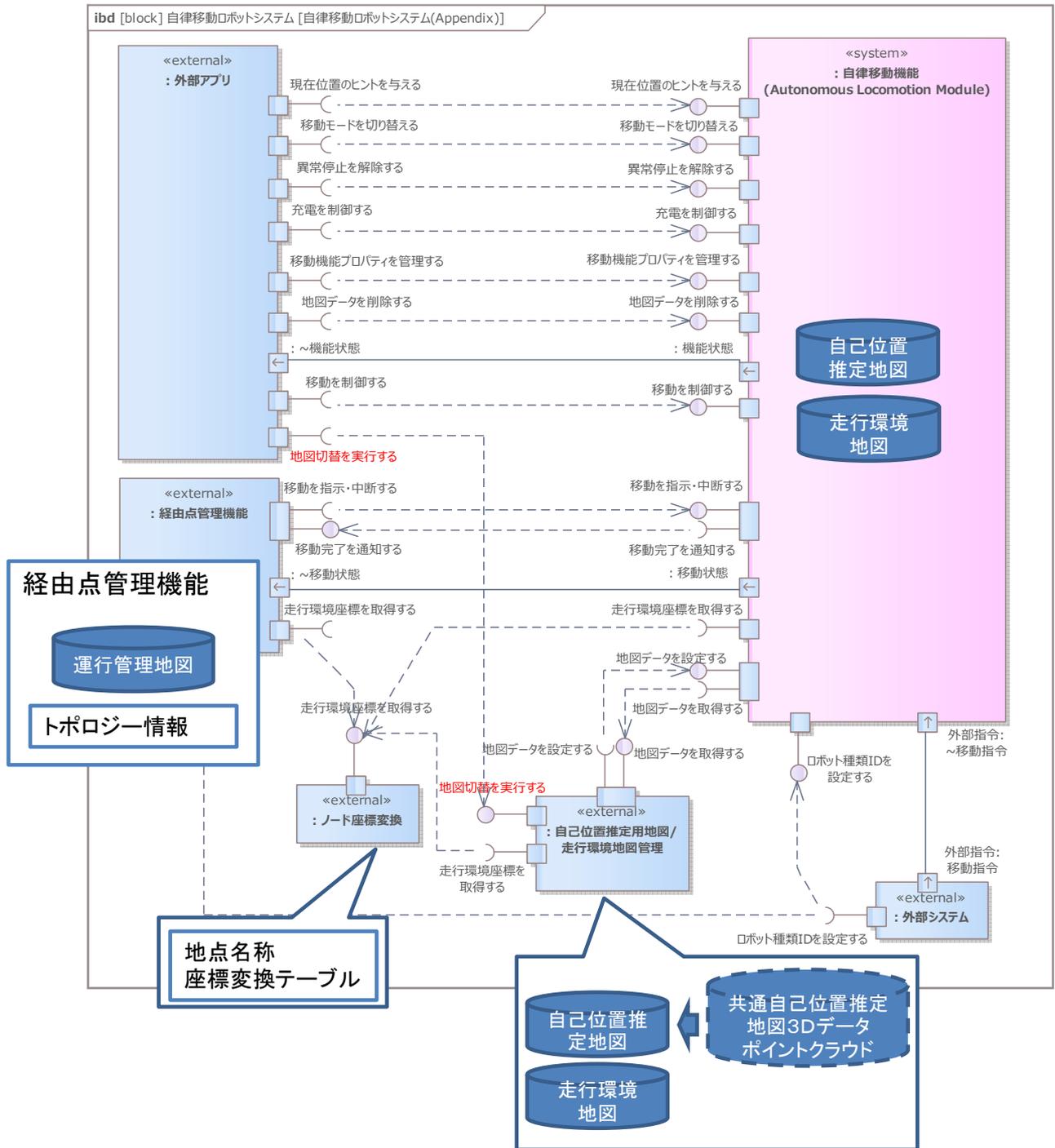
A.1.11.HW 管理

Description : 非常停止スイッチやバンパなど、自律移動ロボットのハードウェアにより停止処理が行われた事を「移動状態管理」に通知するための要素.				
Attributes なし				
Operations なし				

Appendix.2. 地図について

Appendix.2.1. 地図の種類について

本仕様では、自律移動機能を用いたシステムを構築する際には、システム全体で以下の3種類の地図が存在すると想定している。各地図とモジュールの関係を示した概略図を図A-2-1に示す。また、図A-2-1では、Appendixで追加しているインターフェースを赤字で示している。



図A-2-1 地図情報の概略

- 自己位置推定用地図

自律移動機能が走行時に自己位置を推定するために使用する地図。

SLAMなどの技術を用いて事前に作成を行う。複数種類のロボット/センサを使用して、3Dのポイントクラウドのデータで表現した「共通自己位置推定用地図」を作成する場合もある。

本地図は、各自律移動ロボットの位置を表示するために使用する場合もある。また、新たに自律移動ロボットを追加した際に、既存の地図情報を利用するために配信する場合もある。更に、自律移動機能は利用せずに、別の手法で構築される場合もある。これらの理由から、本地図は「自己位置推定用地図/走行環境地図管理」から「自律移動機能」に対して配信する形となる。

- 走行環境地図

自律移動機能が走行可能なエリアを判断するために使用する地図。

自己位置推定用地図の情報を基に、移動対象エリア全体に対して走行禁止エリアをユーザが設定して作成する。

本地図は「経路計画(Path Planning)」で使用する。「経路計画(Path Planning)」では、各種センサで計測した周辺環境の情報も付加した上で、移動方向の判断を行う。なお、自律移動ロボットが移動した際に、「経路計画(Path Planning)」が使用する地図の範囲を切り替えるタイミングについては、「外部アプリ」側で判断を行う(フロア切替を含む)。

本地図は「自律移動機能」側のみが保持する。ただし、自己位置推定用地図と同様に、新たに自律移動ロボットを追加した際に、既存の地図情報を利用する場合もあるため「自己位置推定用地図/走行環境地図管理」から配信する形となる。

- 運行管理地図

自律移動機能に指令する経路点間の関係を記述したトポロジカル・グラフ形式の地図。

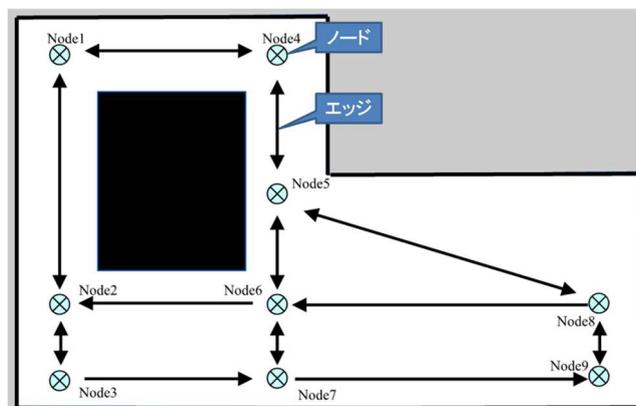
走行環境地図の情報を基に、ユーザが経路点の位置を定義するとともに、経路点間の接続関係を設定して作成する。また、各エッジに対して、進行可能方向や進入可能台数、制限速度などの制約条件を付加する場合や、各ノードに対して「荷積み場所」「荷下ろし場所」などの情報を付加する場合もある。

実運用時には、それぞれの自律移動ロボットから送信された「ロボット位置」を基に、各ノード、エッジの状態(各ノード、エッジに現在何台の自律移動ロボットが存在するか?など)を更新する。更に、単体ロボットでは取得できない情報を、他の外部システムから取得して地図の更新を行う(天候情報を取得し、雨だから通行禁止に設定、洪水予報が出ているため通行禁止に設定。など)。

そして、システム全体として適切な動作を実現できるように、各自律移動ロボットに対する「移動指示」を作成する。この際、各ノード、エッジに付加された制限事項を基に、「経路点管理機能」がエッジへの進入可否判断や制限速度の設定を行う。(例えば、あるエッジに「進入可能台数は1台」と設定されており、そのエッジに複数の自律移動ロボットが進入する必要がある場合に、どの自律移動ロボットに進入を許可するのは「経路点管理機能」側で判断を行う。そして、進入を許可しない自律移動ロボットに対しては、許可された自律移動ロボットが、その区間を抜けるまで「移動指示」を送信せず、その場で待機させる)

本地図は、システム全体として複数ロボットの動作を管理するために「経路点管理機能」側のみが保持する。(本仕様書の範囲外となるため、詳細なクラス定義などは記載していない。)

運行管理地図のサンプルを以下に示す。



図A-2-2. 運行管理地図の例

Appendix.2.2.地図の切り替えについて

階層や走行エリアが変わる場合に、地図の切り替えを行う際の振る舞いを以下に示す。

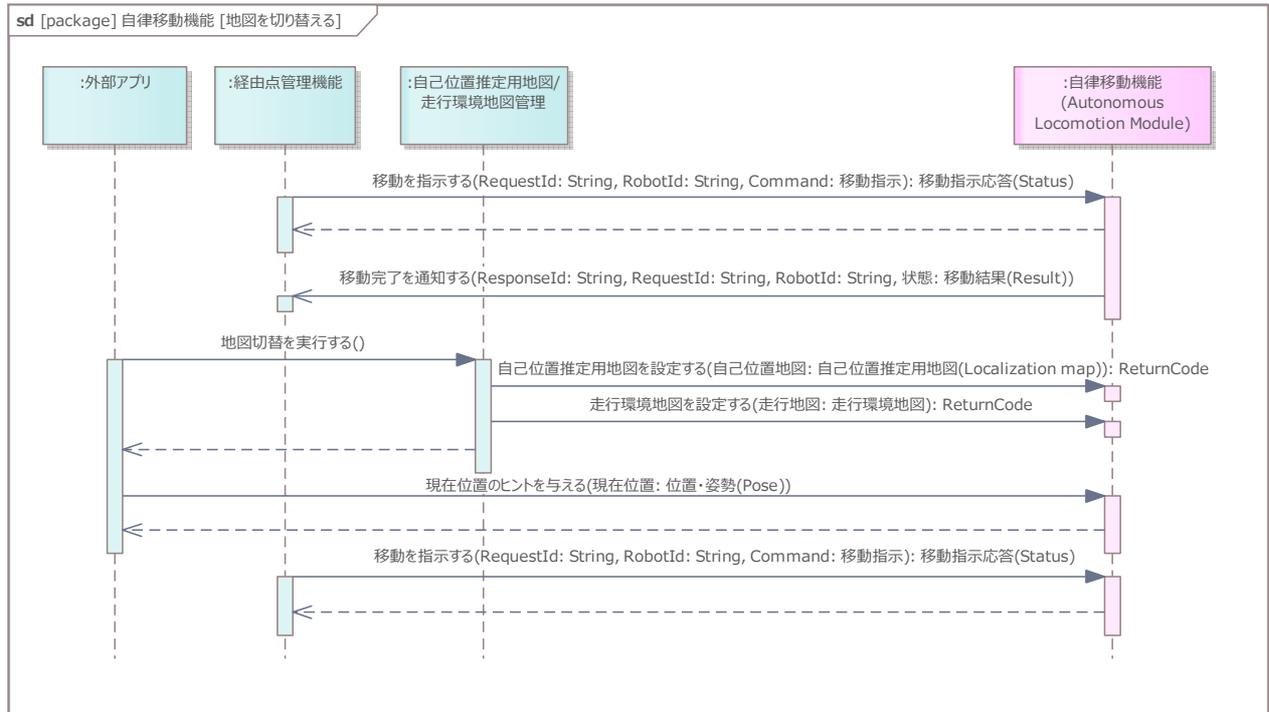


図 A-2-3. 地図を切り替える

「自律移動機能」が使用する地図の切り替えについては「外部アプリ」側で管理を行っている。

まず「経路点管理機能」が「移動を指示する」というメッセージで、地図切り替えを行う地点を「目標点」として通知する。「自律移動機能」が指定された目標点まで到達すると、「外部アプリ」は次の移動先から必要となる地図を判断し、「自己位置推定用地図/走行環境地図管理」に対して「地図切替を実行する」というメッセージで、地図の切り替えを依頼する。「自己位置推定用地図/走行環境地図管理」は必要となる地図を判断し、「自己位置推定用地図を設定する」「走行環境地図を設定する」というメッセージで「自律移動機能」へ地図の配信を行う。

「自己位置推定用地図/走行環境地図管理」から地図の更新が完了されると、「外部アプリ」が更新された地図内での「自律移動機能」の現在位置・姿勢を「現在位置のヒントを与える」メッセージを使用して通知する。

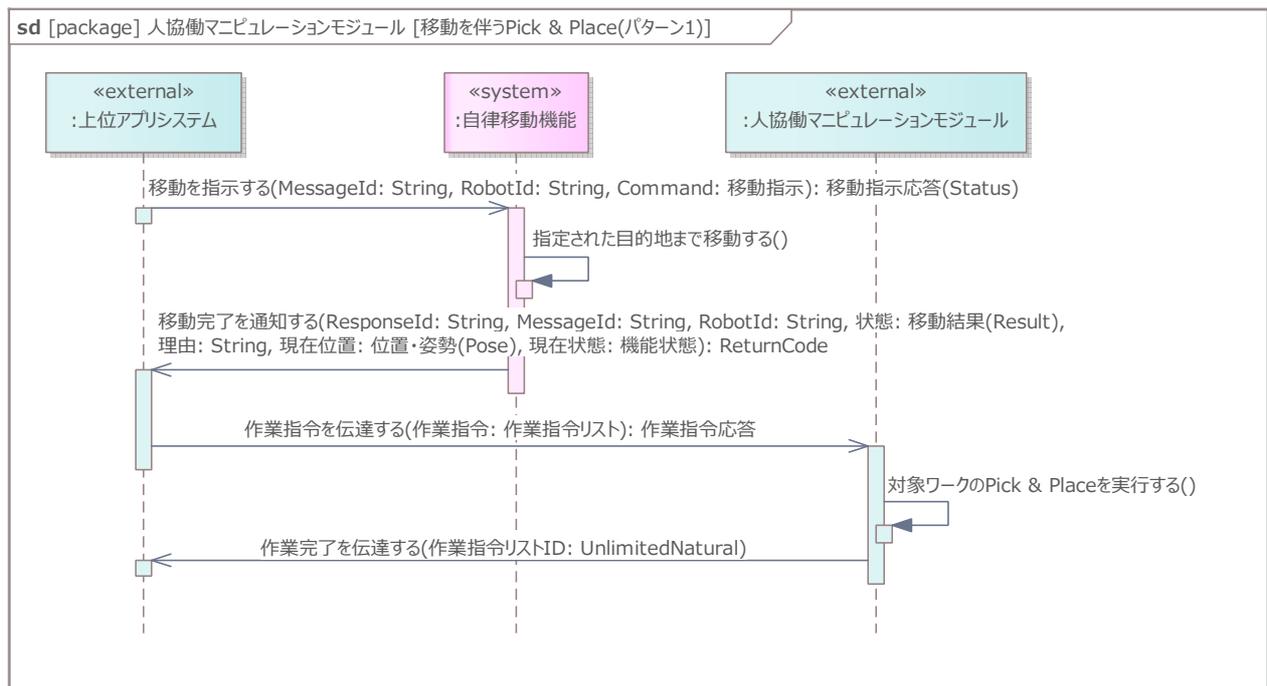
その後、「経路点管理機能」から「自律移動機能」に新しい目標点が通知され移動を開始する。

Appendix.3. 「人協働マニピュレーションモジュール」と連携する際の動作について

ロボット革命・産業 IoT イニシアティブ協議会が公開している「人協働マニピュレーション機能インターフェース仕様書」では、本仕様書で規定されている機能、インターフェースを有する「自律移動機能」を利用して、移動を伴う搬送(Pick & Place)作業を行う事も想定している。

「自律移動機能」および「人協働マニピュレーションモジュール」の振る舞いや詳細については、それぞれの仕様書で説明を行っているが、これらを組み合わせて使用する場合の振る舞いがわかりづらい部分もある。このため、「人協働マニピュレーションモジュール」と「自律移動機能」を組み合わせて使用する場合の全体の振る舞いを以下で説明する。

○パターン 1 (Pick & Place が完了してから移動する場合)



図A.3.1 移動を伴うPick&Place パターン1

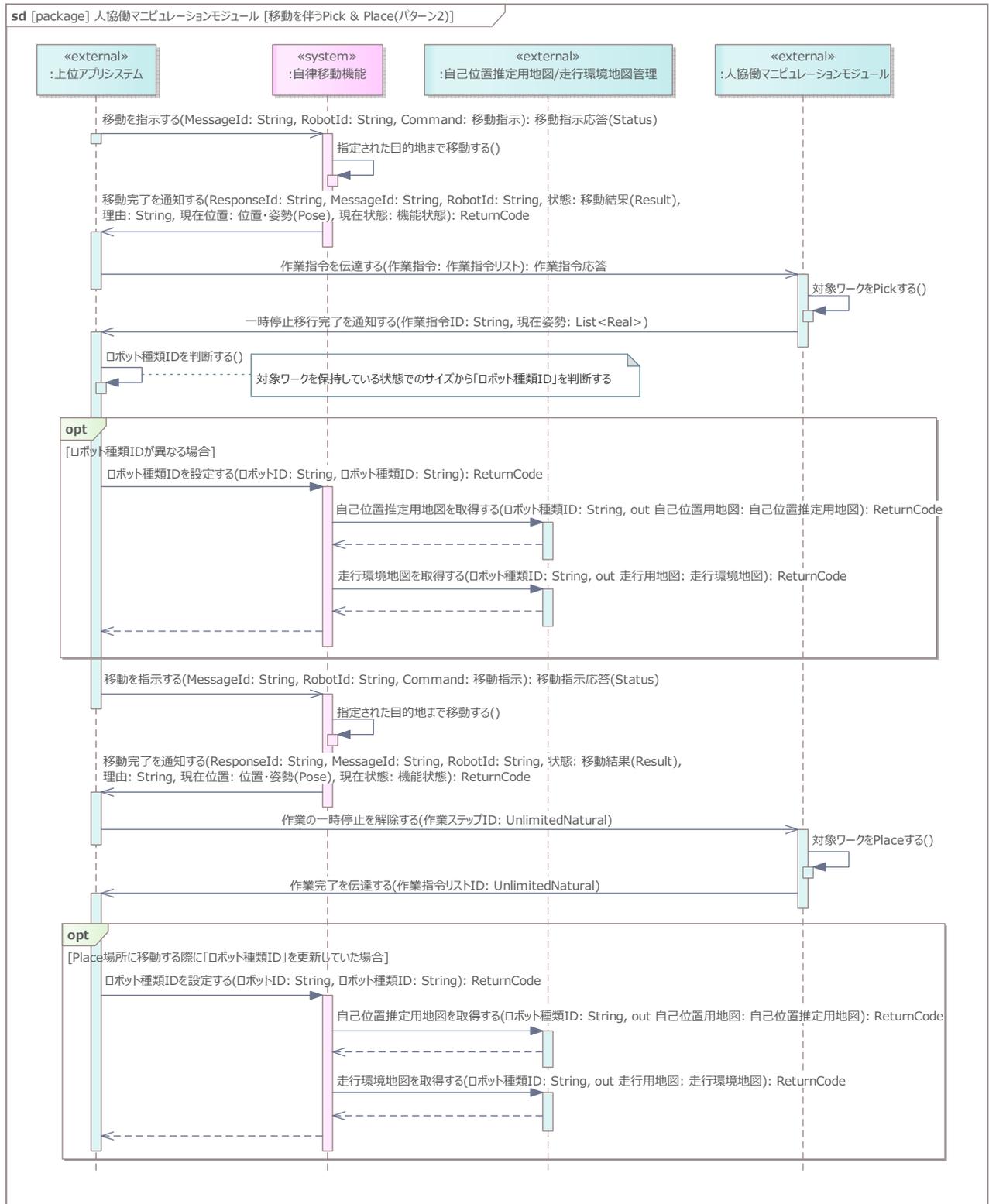
このパターンの場合、「自律移動機能」による移動と、「人協働マニピュレーションモジュール」による動作が完全に別々となるため、「上位アプリ」から順次指令が送出される形になる。

まず「上位アプリ」は「自律移動機能」に対して、「移動を指示する」というメッセージで移動対象のロボット、移動時の目標点を通知する。そして、自律移動ロボットが目標点に到着すると、「自律移動機能」から「上位アプリ」に対して「移動完了を通知する」というメッセージを送信し、目標位置に到着した事を通知する。

次に「上位アプリ」は「人協働マニピュレーションモジュール」に対して、「作業指令を伝達する」というメッセージで、その場所で搬送(Pick&Place)すべきワークの情報(種類、数など)を通知する。そして、「人協働マニピュレーションモジュール」は指示された作業が完了したタイミングで「上位アプリ」に対して「作業完了を伝達する」というメッセージを送信し、その場所での作業が完了した事を通知する。

そして、更に別の場所に移動して作業を行う必要がある場合には、同様な処理が繰り返される。

○パターン2 (Pick後にワークを把持したまま移動し、その後でPlaceする場合)



図A.3.2 移動を伴うPick&Place パターン2

このパターンの場合、「自律移動機能」による移動と、「人協働マニピュレーションモジュール」による動作が混在する形になるとともに、移動中の積載物が変化することで、ロボットの全体のサイズも変化する可能性があるため、より複雑な振る舞いとなる。

まずパターン1の場合と同様に、作業場所までの移動を行うため「上位アプリ」は「自律移動機能」に対して、「移動を指示する」というメッセージで移動対象のロボット、移動時の目標点を通知する。そして、自律

移動ロボットが目標点に到着すると、「自律移動機能」から「上位アプリ」に対して「移動完了を通知する」というメッセージを送信し、目標位置に到着した事を通知する。

次に「上位アプリ」は「人協働マニピュレーションモジュール」に対して、「作業指令を伝達する」というメッセージで、対象ワークの情報(種類、数など)を通知する。この際、「作業指令」の「一時中断あり」を **True** に設定するとともに、「待機姿勢」も指定し、対象ワークを **Pick** した後に移動を行うことも通知する。そして「人協働マニピュレーションモジュール」は、指定されたワークを **Pick** した後、「待機姿勢」まで移動する。そして「上位アプリ」に対して「一時停止移行完了を通知する」というメッセージを送信し、「一時停止中」状態となった事を通知する。この際、ワークの把持状態などによっては、指定された「待機姿勢」に到達できない可能性もあるため、実際の「待機姿勢」における「関節角度リスト」も併せて通知する。

そして「上位アプリ」は、通知された「関節角度リスト」から、(ワークも含めた)ロボット全体のサイズを算出し、対応する「ロボット種類 ID」を判断する。ここで「ロボット種類 ID」が異なる場合には、「上位アプリ」から「自律移動機能」に対して「ロボット種類 ID を設定する」というメッセージを送信して、「自律移動機能」が使用する地図(自己位置推定用地図、走行環境地図)を変更する(対象ワークの大きさや把持姿勢によっては、ロボット本体からはみ出してしまう場合があり、走行可能な場所が変わってしまう事があるため)。

その後、**Place** を行う作業場所まで移動するため、「上位アプリ」から「自律移動機能」に対して「移動を指示する」というメッセージで目標点(**Place** を行う作業場所)を通知する。そして、自律移動ロボットが目標点に到着したら、「自律移動機能」から「上位アプリ」に対して「移動完了を通知する」というメッセージを送信する。

そして、「上位アプリ」から「人協働マニピュレーションモジュール」に対して「作業の一時停止を解除する」というメッセージを送信して、対象ワークの **Place** を行う。「人協働マニピュレーションモジュール」は **Place** 作業が完了したタイミングで「上位アプリ」に対して「作業完了を伝達する」というメッセージを送信し、作業が完了した事を通知する。

最後に、**Place** 場所に移動する際に「ロボット種類 ID」を更新し、「自律移動機能」が使用する地図を更新していた場合には、これを元に戻す必要があるため、「上位アプリ」から「自律移動機能」に対して「ロボット種類 ID を設定する」というメッセージを送信して、「自律移動機能」が使用する地図(自己位置推定用地図、走行環境地図)を元に戻す。

搬送対象のワークが残っている場合には、同様の処理を繰り返す。

改訂履歴

版番号	公開日	備考
1.0	2020/06/01	初版
2.0	2021/06/01	移動機能の運用を想定した状態遷移とその追加に伴うインターフェース仕様の追記・修正
2.1	2022/06/01	全体の章構成を見直し、座標系を指定するIDの名称を変更
2.2	2022/11/01	データモデルの可視性をprivateからpublicに修正 3.2.1.19.走行環境用ピクセル(pixel), 3.2.1.21 走行環境用ボクセルの説明を修正
2.3	2023/06/01	「運行管理機能インターフェース仕様書」の内容を基に、インターフェース定義、構成要素を見直し
2.4	2024/06/01	「運行管理機能インターフェース仕様書」の内容と一致していなかった部分を修正
2.5	2024/12/01	「運行管理機能インターフェース仕様書 Ver.1.1」の内容に合わせる形で、3.2.1.2 進入可否制御方式の内容を修正
2.6	2025/10/01	Platform Specific Model (PSM)として、ROS PSMの章を追加.



ロボット革命・産業IoTイニシアティブ協議会
Robot Revolution & Industrial IoT Initiative