

# 多重オントロジーに基づくセマンティックロボットサービスの設計と実現 Design and Implementations of Semantic Robot Service Based on Multi-Layered Ontologies

宮川智好  
Tomoyoshi Miyakawa

山本隆三  
Ryuzo Yamamoto

植田光  
Hikaru Ueda

山口高平  
Takahira Yamaguchi

慶応義塾大学  
Keio University

This paper presents the framework for Semantic Robot Services which enables cooperation and sharing of operation with several robots by making use of multiple ontologies; user request, robot service, robot function, behavior, cause, recovery, ontologies. The case studies with intelligent transportation show that the framework goes well with good robot action plans.

## 1. はじめに

近年、タイプの異なるロボットが利用可能になってきたことから、異種ロボットを連携させて、人に役立つサービスを生み出すための研究開発が開始されている。例えば、産業界が中心となって設立した組織である RSi(ロボットサービスイニシアチブ)[1]では、ロボットがネットワークに接続するための汎用通信機能と個々のロボットが所持する物理機能を管理するプロフィール上で、ロボットサービスを実現する方法を検討している。また、NEDO が提唱している「次世代ロボット知能技術開発プロジェクト」では、幅広い用途と人間共存環境での実用化に繋げるために、自立的な次世代ロボットの要素技術を確立しようとしている。しかしながら、こういった活動では、人とロボットのインタラクションを担うレイヤーが存在しない。これは人とロボットのインタラクションをとるには、人間が処理する漠然とした概念世界とロボットが処理する精緻な物理世界を融合する高次連携機構が必要であり、人工知能技術の導入が必須とあるが、現状のロボットサービス連携ではそこまで対応できていないといえる。

一方、オントロジーの利用方法という観点からはセマンティック Web、情報システム開発の上流工程支援、データベースの統合、Web サービスの連携、知識マネジメントなどがあるが、すべてソフトウェアへの応用であり、ハードウェアの連携にオントロジーを応用する試みは少なく、オントロジーを用いて意味を考慮しながらロボットの効果的な連携を考えるセマンティックロボットサービス(SRS: Semantic Robot Services)の研究は萌芽的であると考える。

本稿ではこのシステムのフレームワークの提案、構築したオントロジー群、オントロジー群の統合に基づいた SRS の設計と実現の可能性について述べる。

## 2. 提案システム

提案システムでは後述する複数のオントロジー群を用いることで、ユーザゴールを単一タスクに分解し、そのタスクを複数のサービスと関連付ける。そして、単一のサービスをロボットの可能機能動作に展開し、環境地図やロボットデータ、実行環境のオブジェクトデータを用いたデータ計算による実行ロボットの選定、競合解消を行う。実行ロボットとその実行順から構成される連携案を作成し、ロボットに実行命令を送ることでロボットの連携作業を実現する。また、連携作業中には、ロボットのタスクの成否をチェックし、エラーと判断された場合、エラーの原因を推論し、それに基づくリカバリーを実行する。このシステムフレームワークを SRS として提案する。概要を図1に示す。

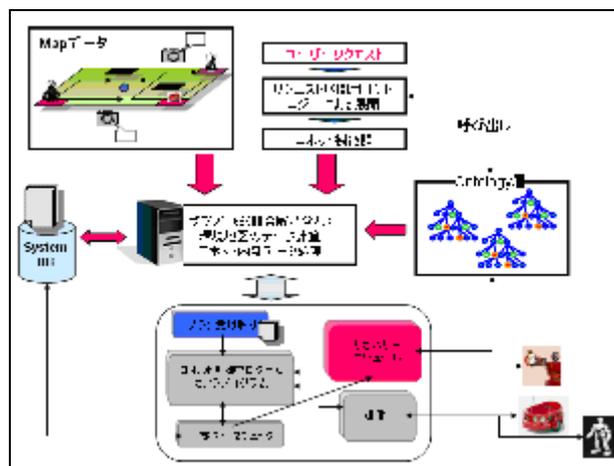


図1 SRS システム概要図

## 3. 作成したオントロジー群

本研究において作成し、システムに用いるオントロジー群の概要を説明する。ユーザーリクエストをロボット実機能へと展開する3つタスク展開用オントロジーと、エラー発生時にその原因を特定し、特定された原因に基づくリカバリー行動を決定するための異常特定のオントロジーの2群に分けられる。

・リクエスト展開用オントロジー群

### ① User Request Ontology

User Request をどのようなサブタスク群に展開できるかを表現している。概要図を図2に示す

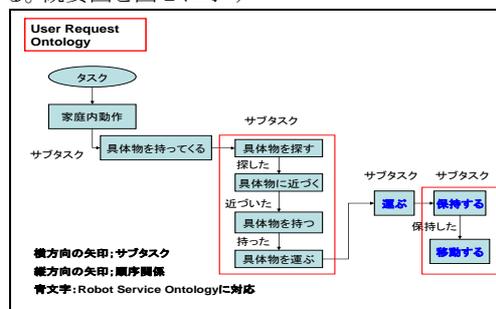


図2 User Request Ontology の概要図

図2に示すように、ユーザーの要求をドメインとし、そのユーザーリクエストを達成するために必要なサブタスク群→展開した

サブタスクを達成するために必要なサブタスク群→・・・といった様に段階的に展開していく形でリクエストを表記したものである。

### ②Robot Service Ontology

ロボットにとってのサービスを表現している。User Request Ontology によって展開されたユーザーの要求のサブタスク群がロボットにとってどのような行為(サービス)であるかを表記したものである。

### ③Robot Function Ontology

ロボットそれぞれが可能な実機能を表現している。ユーザーリクエストが各個体ロボットの機能につながるようなオントロジー群と展開パターンで、現実存在する各ロボットの機能(スキル、アクション)を体系化したものが Robot Function Ontology である。ここでは研究で用いる各ロボットの機能一つ一つが最下位概念となる。更にこのロボット実行可能機能概念と対応するロボット動作プログラムには、移動した後に目的地とのずれを補正すると言ったようなリカバリー行動が含まれているケースが存在する。その場合にはエラーが発生したとしてもロボットが単独でリカバリーを行うことが可能と言え、その部分をシステムでいちいち判断しなくてもよくなり、冗長な推論を行わなくてもよくなると言える。

以上の各オントロジーを用いた、ユーザーリクエスト展開の流れを以下に図3として示す。

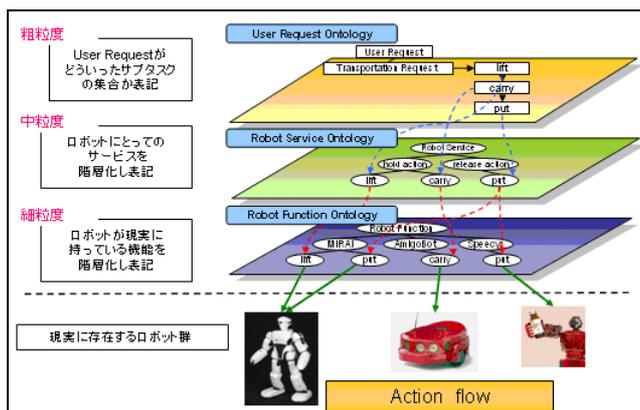


図3 リクエスト展開の流れ

・異常特定用オントロジー群

### ④Behavior Ontology

実際に観測されるロボットの異常な振る舞いを体系化したものである。異常な振る舞いを異常の原因と混同しないように、区別し、体系化している。この際にロボットの振る舞いをロボットのそれぞれの機能レベルまで細かく展開し、分類している。

### ⑤Cause Ontology

実際に観測された異常振る舞いを引き起こす原因を体系化したものである。原因とは異常の振る舞いに対してその原因ということであり、構造的な原因、機能的な原因という2つの分類で分けて体系化を行った。ここでは運搬作業という実験環境下において考え得る異常原因を列挙し、体系化している。

### ⑥Recovery Ontology

リカバリーアクションについて体系化したものである。ロボット連携作業における異常原因に対応して、リカバリーアクションを分類したもので、リカバリーを行うロボットやその作業対象によって体系化されている。

## 4. システム展開

### 4.1 ユーザーリクエスト展開

本節では、ユーザーリクエストをロボット実行可能機能フローに展開するシステムでの展開方法を述べる。

この部分は大きく分けて以下の3つに分けることができる。

#### ①ロボット実行可能機能フローの作成

User Request Ontology 部分でタスクを展開、そして User Request、Robot Service、Robot Function の3つのオントロジー群の対応関係の取得によるロボット実行可能機能フローを作成する。

#### ②プラン決定

複数ロボット機能によって同一サービスを達成できるケースが多々起こる。そういった場合の競合解消やデータベース及び環境地図データとのアクセスを経て、最終プランを決定する。

#### ③プラン実行

実行プラットフォームにプランを流し、ロボットに実行させる。ロボットが作業実行中はアクションが行われている部位やそこにかかっているはずの負荷をデータとして取得し、比較することで、合わない場合にはエラー発生と判断し、リカバリー行程に移行する。

### 競合解消

ロボットの連携アクションプラン決定の際に使用可能ロボットの選定、競合解消が行われる。競合解消の流れを図4に示す。

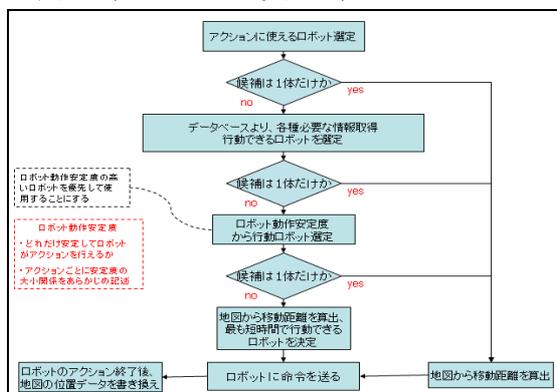


図4 競合解消の流れ

以下、図についての説明を行う。

#### (1)ロボットデータ及び搬送物データの比較

データベースにアクセスを行い、提案されたロボット群が、実際にアクションを実現することが可能か判断する。例えば、物体を「持つ」というアクションに対して、搬送物の重量とロボットの最大持ち上げ重量を比較することでアクションが可能なロボットの絞り込みを行う。

#### (2)ロボット動作安定度による比較

ロボットがどれだけ安定してアクションを行えるかを表す指標である。安定度は、あるロボットのアクションが他のロボットの同じアクションと比べて安定しているかないか、という相対的な大小関係のみを表す値を記述している。

#### (3)行動時間による比較

安定度の比較でも連携案が単一に絞り込めなかった場合、各ロボットの行動時間を比較する。例えば「移動する」というアクションならば、地図上のロボットと目標物との距離を求め、移動速度との比較によって移動に要する時間を算出し、行動時間の最も短いものを最終的な行動ロボットに決定する。

## 4.2 リカバリー工程

ここでは、ロボット連携の失敗をリカバリーするシステムについて説明する。ロボット連携プラン実行中にエラーが発生した場合、リカバリー工程に移行する。この際、それまでの全ての連携が一時的に停止される。実行アクションからリカバリーまでの流れを図 5 に示す。

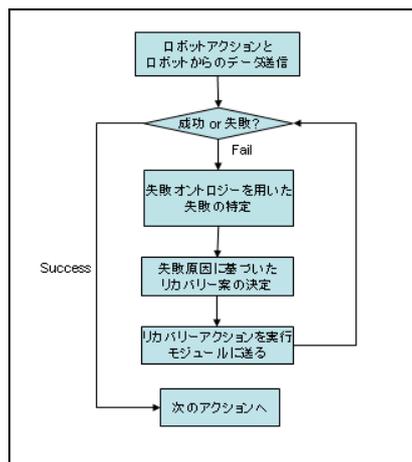


図 5 ロボットアクションからリカバリーまでの流れ

ロボットアクションの失敗時にロボットから送られてくるデータを基にまず異常状態を特定する。その際に以下の 2 つの工程に分かれる。

### ①異常振る舞い特定

異常発生の際にロボットから得られるデータを基に、Behavior Ontology を参照してどのような振る舞いが起きているのかを特定する。ここでは異常行動に対して、その異常の原因を特定するわけではなく、異常がどのような振る舞いに対して起きたのかを特定するのみである。

### ②異常原因特定

Behavior Ontology によって異常振る舞いを特定することができたら、特定された異常振る舞いとロボットからのデータを基に Cause Ontology を参照して、異常の原因を特定する。ここで特定された異常原因をもとにリカバリー案が特定される。また、ここでは異常原因を一つに特定できない場合や複数の原因が考えられる場合もある。

異常原因が特定されたら、それに対して、トラブル復旧のためのリカバリープランを提案する。このとき初期プランへの復旧を目的とした局所的なトラブル解決をその目標としている。

### ③リカバリープラン提案

異常特定のフェーズで特定された異常振る舞い、異常原因、ロボットからのデータを基にリカバリーオントロジーを参照してリカバリープランを提案する。この際にプランが複数考えられる場合はロボットの動作安定度を基にしてプランの優先順位を決定する。

### ④リカバリープラン実行

安定度を基に順位付けられたプランを優先度の高い順に実行していき、リカバリーが成功した段階で元にシステムを再度起動させ、初期プラン達成を目指す。また、リカバリープランを実行しても復旧しなかった場合は最終的な受け皿としれ人間が直接介入し、リカバリーを行うこととする。

以上によって、リクエストを展開し、連携プランを実行し、また、リカバリープランの実行をする。

## 5. 実証実験

研究室の室内を実験環境として整備し、動的な障害(風やペットの存在等)の存在しない屋内という仮定の下、ロボット連携による「物質運搬シナリオ」とその際に何らかの異常が起きた場合のリカバリー行動の実験を行った。

### 5.1 実験環境及び使用ロボット

実験に使用するロボットは Speecys (大型人型ロボット)、MIRAI-RT (小型人型ロボット)、AmigoBot (運搬ロボット)、P3DX (アーム付運搬ロボット)それぞれ 1 台ずつである。環境地図作成には Mapper3 という MobileRobots 社が開発した環境地図作成用のソフトウェアを使用した。各オブジェクトの座標の値はメジャーで測定した現実世界のミリメートルに対応している。作成した環境地図および、実験場の様子を図 6 および図 7 に示す。

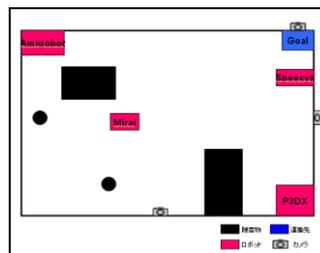


図 6 環境地図



図 7 実際の実験場

本実験においてはロボットの初期位置、障害物の位置、ゴールとなる目的地の位置は所与のものとしてあらかじめ配置される。

### 5.2 運搬シナリオ連携実験

本実験において、ユーザーリクエストがロボット実行可能機能フローへと展開される部分のシナリオを説明する。

運搬シナリオを構成するサブタスクは「持つ」、「運ぶ」、「置く」である。これらのタスクを実行可能なロボットが存在するかをファイルに問い合わせることで取得する。「具体物の運搬」に対して、「具体物を持つ」→「持つ」Speecys, P3DX, MIRAI  
「具体物を運ぶ」→「運ぶ」Speecys, AmigoBot, P3DX  
「具体物を置く」→「置く」Speecys, P3DX, MIRAI  
という関係が取得される。今回の実験はこの 3 つのアクションに分けられ実行されるが、それぞれのアクションを担当するロボットが異なる場合はロボット間での物体の受け渡しが発生する。この受け渡しタスクをシステムで自動的に挿入するのだが、その際のルールは以下の通りである。

- ①人型ロボットから搬送型ロボットへの受け渡しの場合  
搬送型ロボットが人型ロボットに近づき、人型ロボットが搬送型ロボットの上に運搬物を置くというフローの挿入。
- ②搬送型ロボットから人型ロボットへの受け渡しの場合

搬送型ロボットが人型ロボットに近づき、人型ロボットが搬送型ロボットから物を拾うというフローを挿入。

- ③搬送型ロボットから搬送型ロボットへの受け渡しの場合  
受け渡し場所に 2 台のロボットがともに向かうというフローの挿入。

続いて、具体的な例について本実験の説明を行う。

### (1) シナリオ展開パターン 1

状況:搬送物の体積と重量が大きく、搬送物の初期位置が床に置いてある場合

#### ①「持つ」ロボットの決定→MIRAI

重量の大きい搬送物を「持つ」ことが可能なロボットとして P3DX は除外され、Speecys と MIRAI の 2 体が結果として取得される。次にロボット動作安定度を参照すると、MIRAI の安定度のほうが高い為、MIRAI が選択される。

#### ②「運ぶ」ロボットの決定→AmigoBot

「運ぶ」タスクでは AmigoBot と P3DX が提案される。移動にかかる時間の比較によって最終的に AmigoBot 選択される。

#### ③「置く」ロボットの決定→Speecys

「置く」ことが可能なロボットとして、「持つ」タスクの場合と同様の理由で P3DX は除外され、Speecys と MIRAI だという結果が取得される。ここではロボットの運搬可能高度による比較が行われる。ゴールとなる棚の高さに対してタスク実行が可能なロボットとして Speecys が選択される。

#### ④受け渡しフローの挿入

①→②、②→③においてタスクの実行するロボットが異なり受け渡しが行われる。①→②は MIRAI(人型)から AmigoBot(搬送型)なので受け渡しフロー①が挿入される。一方②→③では AmigoBot(搬送型)からSpeecys(人型)なので受け渡しフロー②が挿入される。

以上のタスク群によってリクエストは達成された。

続いて、展開パターン 1 において異常が発生した場合について説明する。タスク①→②の際に AmigoBot が MIRAI まで移動して接近するが、この際に経路に障害物があり、進行できない場合を想定した。

#### ①AmigoBot が停止し異常発生が認識される。

②ロボットデータ、環境データを基に Behavior Ontology を参照して AmigoBot が進行不能であるという振る舞いが特定される。

③特定された振る舞いと原因特定ためのデータから、Cause Ontology を参照して AmigoBot に対して障害物があるという異常原因が特定される。

④特定された「障害物あり」という異常原因から、Recovery Ontology を参照して「進路変更」というリカバリーが提案される

以上の工程を経てリカバリーは成功し、初期プランが続行された。

### (2) シナリオ展開パターン 2

状況:運搬物の体積や重量が小さく、初期状態として運搬物が AmigoBot の上に乗っている場合

①シナリオ展開パターン 1 と同様に「持つ」、「運ぶ」、「置く」タスクを実行するロボットを決定する。ここでは 3 つのタスクに対して、いずれも P3DX が選択される。

②それぞれのタスクはすべて同一ロボットによって行われるため、各タスク間においては受け渡しタスクは行われない。しかし、初期状態として運搬物が AmigoBot 上に乗っているのでここで、受け渡しフロー②が実行される。

以上のタスク群によってリクエストは達成された。

ここで、同様に異常が発生した場合についても考える。P3DX が AmigoBot 運搬物をつかむ際に失敗するという状況を想定する。展開パターン 1 と同様の処理が行われる。

#### ①P3DX の異常認識

②P3DX がつかむ動作に失敗したという振る舞い特定

③位置ずれという異常原因の特定

④グリップ位置修正というリカバリープランの提案

以上の工程を経てリカバリーは成功し、初期プランが続行された。

## 6. 結論

本研究では、ロボット連携システムの構築において、オントロジー技術をもちいることで高次元なロボット連携を達成することを目的とし、“具体物を持ってくる”というリクエストが、多数のオントロジーを用いたタスク展開によってロボットへ指令を送って実行させ、またエラーが起こった際にロボットが取得できたデータおよび、オントロジーを参照することで、それを回復させることが可能であるということを示した。

これによってロボットの意味的な高次連携によって単体では行うことの出来ないサービスを行うことができる可能性を示すことが出来たとと言える。

今後の展望としては、研究に用いるロボットの機能を更に充実し、展開可能なタスクオントロジーや各オントロジーの語彙数を増やしていくことなどが必要であると考えられる。また、サーモやソナーのような内部情報だけではなく外部情報を取得にも力を入れることで、特定可能な失敗状況の幅も広がり、正確性も増すと考えられる。さらに、機能やインスタンスが増加に伴い、オントロジーを根本から改定する必要があったり、より効率的なりカバリーのプランニング等、課題は多くあると考えられる。

### 参考文献

- [1] Robot Services initiative  
<http://www.robotservices.org/>
- [2] Protégé: <http://protege.stanford.edu/>
- [3] 森田武史, 山口高平: 日本語概念を対象にした領域オントロジー構築支援環境 DODDLE の機能拡張, 電子情報通信学会, 信学技報(知能ソフトウェア工学研究会), KBSE2006-1 (2006.5)
- [4] 神崎正英, セマンティックウェブのための RDF/OWL 入門, 2005
- [5] 山口新平, 宮川智好, 山口高平, 谷崎浩一, 岩永将資: 多重オントロジーに基づくセマンティックロボットサービスの設計と実現, 第 22 回人工知能学会全国大会, 3J2-2