

環境情報構造化プラットフォーム ROS-TMS における タスク管理機構

○橋口優香 (九州大学) 表允哲 (九州大学) 辻徳生 (九州大学)
諸岡健一 (九州大学) 倉爪亮 (九州大学)

1. 緒言

高齢化の進展に伴い、医療・介護現場で生活支援を行うサービスロボットの開発が期待されている。しかし、非産業分野におけるロボットの作業環境は多様で動的に変化し、期待される作業も複雑である。そこで我々は、環境側に固定センサを分散配置し、取得した情報を TMS(Town Management System) で管理する、環境情報構造化の枠組みを提唱している [1]。環境情報構造化とは、スタンドアロン型のロボットを発展させる考え方とは反対に、人間の生活空間をロボットが作業しやすい環境 (情報構造化環境) に整備することで柔軟なサービス提供を実現するものである。また、TMS とは、ネットワークを介して取得したセンサ情報を処理・分析し、意味のある環境情報として保持、ロボットに提供するシステムである。このシステムを導入することによって、搭載可能なセンサ数やその性能に限りのあるロボット単体でも、必要な情報をシステムから補充し、作業を遂行することができる。

情報構造化環境でロボットによる生活支援を行うためには、RT 要素技術を組み合わせたプラットフォームや環境センサ統合アーキテクチャが必要となる。これらの枠組みは、あらかじめ決められたロボットにより、特定サービスを実行するものでなく、指示内容や環境の状態から使用できるロボットを判断し、必要な情報や処理プログラムを柔軟に選択できる汎用性の高いものであることが求められる。これまで、多様なロボットを考慮したロボット作業の設計手法として、吉見らにより作業情報構造化という概念が提案されている [2]。ここでは、ロボット間の物理的な差異による制約を吸収するために、ロボットの形態依存、または非依存の観点から動作プログラムのモジュール化を行っている。前田らは、環境・作業情報構造化の概念を取り入れた空間知を実現する枠組みとして、「空間知ミドルウェア」を提案している [3]、[4]。

我々も、蓄積した環境情報を活用し、ロボットサービスを実行する枠組みとして、ミドルウェア ROS (Robot Operating System)[5] をベースにしたタウンマネジメントシステム (ROS-TMS) の構築を進めている [6]。本稿では、これまでの TMS において個別に設計されていたロボットサービスを見直し、サービスの構成要素としてタスクを再定義した。さらに、ROS-TMS に対し、作業情報構造化の概念を取り入れることで、ロボットやタスクの追加または変更に対して柔軟なアーキテクチャを構築した。提案するアーキテクチャでは、要求されるタスクに関する直感的な情報を、システムがロボットのコマンドレベルまでおとし、ロボットに依存し

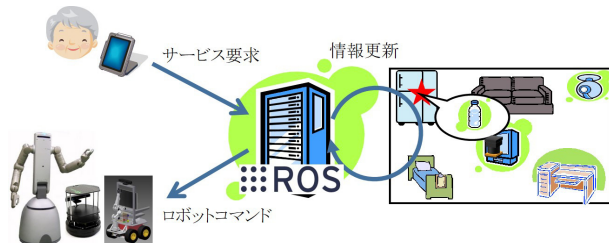


図 1 ROS-TMS 概念図

ない命令系統を用いてタスクを実行することができる。

2. システムの枠組み

我々は、環境情報構造化を具現化する、TMS の開発を行っている。本節では、新たな TMS の枠組みとして設計した ROS-TMS について述べ、新たに開発したタスク管理モジュールについて概説する。

2.1 ROS-TMS の概念

我々が 2011 年までに開発した TMS[1] は、センサ情報の管理と提供を行うことを主な用途とされており、物品の運搬・配膳といったロボットが行うサービスは、開発者が、担当するサービスとロボットを固定し、各々が TMS と通信しながらハードコーディング方式で開発を行ってきた。担当するロボットやサービスに特化した実装を行うことは、特定の条件において精度の高いサービスを実現できることもあるが、システムとして、ロボットやサービスの追加・変更に対応しづらく、汎用性が低いという問題があった。

そこで ROS-TMS[6] では、システムの機能拡張性や柔軟性を高めるために ROS(Robot Operating System) ベースで開発を行っている。ROS とは、ロボットソフトウェア開発を柔軟に行うために様々なツールやライブラリを提供しているオープンソースのミドルウェアである。ROS-TMS を構成する各モジュールを、ROS のノード (特定の処理を行うプロセス) として実装することにより、複数のモジュール間の通信やモジュールの追加・変更を容易に行うことができる。

図 1 で示すように、ROS-TMS は、システムがユーザからの要求を受けつけ、蓄積した環境情報を適宜ロボットへ提供しながら、ロボットコマンドを送信してタスクを実行させる包括的な枠組みと位置づけられる。

2.2 ROS-TMS のモジュール構成

ROS-TMS のモジュール構成を、図 2 に示す。この枠組みにおいて、「ロボット A でタスク B をしてほしい」というような要求は、ユーザリクエストを介してタス

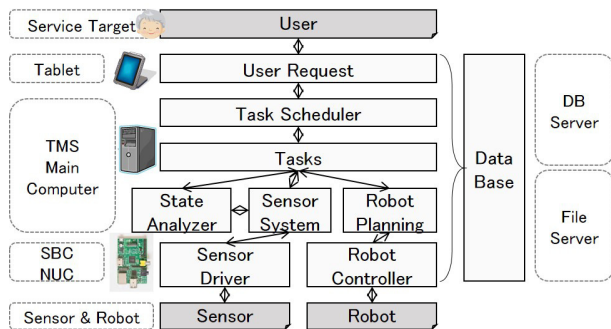


図 2 ROS-TMS モジュール構成

タスクスケジューラに送信される。タスクスケジューラでは、要求されたタスクがどのようなサブタスクから構成されるかをデータベースに問い合わせ、分解する(ブロック化)。その後、サブタスクごとに適切な引数を設定して、ロボットへの動作指令を自動生成し、タスク実行マシンを通して逐次実行する。経路計画や把持計画などの、ロボットの形態に依存する部分は、下位モジュールであるロボットプランニングモジュールで吸収する。

2.2.1 下位モジュール

まず、環境内には環境固定センサや複数台のロボットがあり、それらを制御するドライバやコントローラが SBC(Single Board Computer) や NUC (Next Unit of Computing) 上で実行されている。これらの上に、センサデータから高次の情報を生成するセンサシステム、センサデータを基に環境状態を推定するステートアナライザ、また、ロボットの移動・把持といった動作計画を行うロボットプランニングシステムがある。これらは ROS-TMS の下位モジュールと位置づけられ、対象としている環境を把握し、ロボットによるサービス実行時に必要な情報を提供する。

2.2.2 上位モジュール

ROS-TMS では、ロボットが行うサービスのことを「タスク」と呼び、独自に定義づけを行っている(3.1 節参照)。タスク実行の開始はシステム利用者(ユーザ)によって指示されるものや、システムが自動的にタスクを開始するものがある。ユーザからのタブレット端末を介したサービス要請は、ユーザリクエストを通してタスクスケジューラに送信され、3.2 節で説明する方法で実行される。

以下の節では、システムが管理するタスクの定義と、タスクスケジューラモジュールの開発について述べる。

3. タスク管理

本節では、ROS-TMS の枠組みの中でロボットが行うサービスとして「タスク」を定義し、それを管理するタスク管理モジュール「タスクスケジューラ」について説明する。

3.1 タスク定義

まず、これまで独立したロボットサービスとして設計されていたサービス動作を、タスク間で共有できる

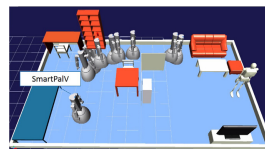


図 3 move の実行画面 (SmartPalV)

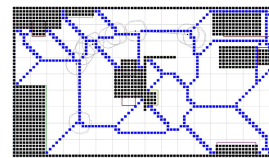


図 4 move 実行時の出力経路 (SmartPalV)

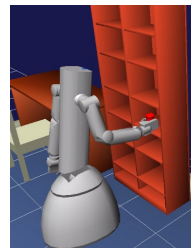


図 5 grasp の実行画面

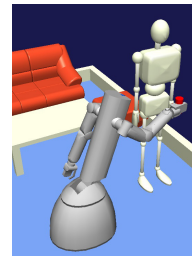


図 6 give の実行画面

ロボットの作業単位、すなわちサブタスクの集合に分解する(ブロック化)。ここで、サブタスクとは、開発者があらかじめ定義した、ロボットが提供できる特定のサービスのことで、例えば、move(目的地までの移動)、grasp(物品把持)、give(対象者への物品受け渡し)などが考えられる(図 3～図 6、ID については後述)。ブロック化されたサブタスクを複数組み合わせることで、意味のある作業が生成され、これをタスクと定義する。この定義によって、プログラムの再利用性が向上し、また、ROS-TMS の枠組みの中でロボットが行うタスクの定義が統一されることにより、タスクを実行する命令系統や実行機構も統一することができる。

具体例として、物品取り寄せタスクをブロック化して生成できる 3 つのサブタスクを示す。各サブタスクは、2.2 節で述べた、ROS-TMS におけるロボットプランニングモジュールとして実装されている。

- move(ID:9001)
用途：安全な経路を通る、目標地点までの移動
入力：ロボット ID, 家具 ID(または物品 ID)
出力：現在地から、目的地となる家具付近の移動候補点までの、ロボットベースの移動コマンド(x,y,θの配列)
- grasp(ID:9002)
用途：把持対象物の把持
入力：ロボット ID, 物品 ID
出力：初期姿勢、把持姿勢、終了姿勢への関節遷移コマンド(各関節の値)
- give(ID:9003)
用途：対象者への物品受け渡し
入力：ロボット ID, 人 ID
出力：対象者とロボットの可操作度に基づく、最も良い受け渡し位置と姿勢への移動・関節遷移コマンド(x,y,θ,各関節の値の配列)[8]

ロボットプランニングモジュールで吸収するロボット固有の情報は、TMS-DB により提供される。ここで、DB

設計について簡単に述べる。DB では、システムが把握すべき対象物全てに一意的 ID を振って管理している。その振り分けは、人 (1001~1999), ロボット (2001~2999), センサ (3001~3999), 建物 (4001~4999), 空間 (5001~5999), 家具 (6001~6999), 物体 (7001~7999), タスク (8001~8999), サブタスク (9001~9999) となっている。各対象物に対して、位置姿勢やモデルの中心点からのオフセット、現在位置や状態等が更新・格納されている。家具に関しては、その家具に、あるロボットでアプローチしたいときに移動すべき候補点が格納されており、タスクに関してはそれがどのサブタスクからなっているか、どの形式の引数をとるかという情報があらかじめ格納されている。

3.2 タスクスケジューラ

タスクスケジューラとは、ユーザから要求されたタスクをいつ、どのように実行するか、時間拘束やリソース拘束を見てスケジューリングするタスク管理モジュールである。これは、タスク計画部とタスク実行部から構成される。本節で述べる形式でタスクを管理し、実行することで、新しいタスクに対して柔軟性の高いシステムを実現することができる。以下、各機能について説明する。

3.2.1 タスク計画部

タスク計画部は、実行順制御、並列制御の機能を持つ。

- 実行順制御

ユーザリクエストを通してシステムに要求されるタスクは1つではない。そこで、複数要求されたタスクをどの順序で実行させるかや、並列に実行できるかなど、タスクの実行順や実行形態に関して計画を行う必要がある。現在は、ユーザ定義の優先度の高い順としている。また、同じ優先度のタスクが要求された場合は要求を受け付けた時間の早い順に実行する。

- 並列制御

本システムを導入する環境内で、複数台のロボットが同時に作業することが求められたり、またそれを実現することで提供できるサービスの効率や質が向上すると考えられる。そこで現在、ROS の `nodelet` パッケージを利用し、タスク実行ノードが同時に 10 個まで起動できる仕組みを開発した。

3.2.2 タスク実行部

要求されたタスクは、それをブロック化し、生成された各サブタスクを逐次実行していくことで達成する。例えば、ユーザがあるタスク 1 を要求する。この要求は、ユーザリクエストを介してタスクスケジューラに送信される。タスクスケジューラでは、要求されたタスクのブロック化を行うため、TMS-DB にタスク ID で問い合わせる。タスクの実行に必要な情報は、あらかじめ開発者が定義し、DB に格納しておく。タスクを構成するサブタスクの集合がタスクスケジューラに返されると、各サブタスクの可変の引数 (移動の目的地や把持する物品など) を、ユーザリクエストで与えられた ID 列を解釈することで適宜当てはめる。そして、タスク実行マシン SMACH(State MACHine)[7] を通して実行する。

タスク実行マシン SMACH とは、タスクレベルで記述されたロボットによるサービスを、状態遷移の形で実行できるタスクレベルのアーキテクチャである。また、ROS との親和性も高いため、ROS-TMS に導入しやすい。ここでは、各状態をサブタスクとして置き換えた状態遷移図を、python スクリプトのファイル出力により自動生成する (図 7)。

4. シミュレーション

4.1 物品取り寄せタスク

ここまでに説明した ROS-TMS の枠組みを使い、物品取り寄せタスクを行うシミュレーションを実施した (図 8)。ユーザリクエストから送信される ID 列をタスクスケジューラで解釈し、指定されたロボットのロボットコントローラへ動作命令を送信するデータの流れを、各モジュールの挙動を示しながら説明する (図 7)。

- ユーザリクエスト

「ロボット A で物品 B を人 C に持ってくるタスク D をしてほしい」という形式で送られる (A~D: TMS-DB で管理される任意の ID)。

- タスクスケジューラ

タスク ID で DB を引くと、物品取り寄せタスクが、3つのサブタスク (9001;9002;9003) からなり、それぞれ引数に物体 ID や人 ID をとるということが分かる。一つ一つのサブタスクを SMACH における状態と定義し、DB で型が定義されている引数にユーザリクエストから受け取った具体的な ID を格納してサブタスクを逐次実行する SMACH の python スクリプトを発行する。

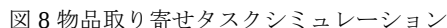
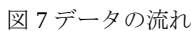
- ロボットプランニング

各サブタスクの引数を、コントローラに送るコマンドの形で発行できるように ID から値への変換を行う。例えば、`move` の引数 `pid(=place_id)` として送られてくる 6011 という ID で DB を引くと、`big_shelf` であり、この家具にアプローチするための候補点として $(x,y,\theta)=(2000,3700,90)$ という点があることが分かる。これを目標点として経路計画ノードを呼び出し、現在地 (ロボット ID で DB を引くと、現在地 x,y,θ が取得できる) から目標点までの安全な経路点を得る。この計算結果を、指定されたロボットのコントローラへ送ることで、ロボットが出力経路上の移動が実現される。同様に、把持や物品受け渡しのサブタスクなどについても、上位モジュールから送られる ID の情報を、最終的にロボットコントローラへ投げる命令の型に合わせて下位モジュールで必要な計算を行い、これを実現する。

把持計画には、産総研が中心となって開発しているロボット用統合 GUI ソフトウェア `choreonoid` の `graspPlugin` (アームの先端にハンドが取り付けられた構成のロボットシステムに対して、把持計画、軌道計画、作業計画など、種々の計画問題を解くことが出来るプラグイン) を使用している。

- ロボットコントローラ

各ロボットに対し、絶対座標へのベース移動や関節遷移を行う。



これまでに述べた命令系統で、ハードウェア構造の異なるロボットも制御できることを示す。図 3, 4 で示した安川電機の **SmartPalV** が行っているサブタスク **move** を例にとり、我々の研究室で製作された、アーム搭載移動ロボット **KXP** がこれ実行できることを確認した(図 9,10)。ベースの直径から、異なる出力経路が得られている。

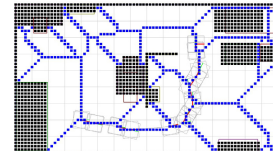


図 9 move の実行画面
(KXP)

図 10 move 実行時の出力経路 (KXP)

情報構造化環境における，ロボットや環境に依存しないタスク管理手法の開発を行った．プログラムの再利用性を高め，同じ命令系統で多様なロボットを統括できるよう，タスク定義を見直すとともに，各サブタスクにおいて，ロボットのハードウェアによる物理的な差異を下位モジュールであるロボットプランニングモジュールで吸収する設計を提案した．さらに，設計したタスクを管理する上位モジュールとしてタスクスケジューラを開発し，シミュレーションによりこれらの有効性を示した．

本研究の一部は文部科学省科学研究費補助金挑戦的萌芽（課題番号 26630099）の支援を受けた。

- [1] 村上剛司, 長谷川勉, 木室義彦, 千田陽介, 家永貴史, 有田大作, 倉爪亮: "情報構造化環境における情報管理の一手法", 日本ロボット学会誌, Vol.26, No. 2, pp.192-199, 2008.
- [2] 吉見卓, 香月理絵, 大賀淳一郎: "環境・作業情報構造化を利用したロボット作業技術", 東芝レビュー, Vol.64, No.1, pp.14-18, 2009.
- [3] 前田佳男, 石黒佑樹, 坂本武志, 水川真, 吉見卓, 安藤吉伸: "日常空間サービスプラットフォームとしての空間知ミドルウェア", 日本機械学会ロボティクス・メカトロニクス講演会, 講演論文集, 1A2-R01, 2012.
- [4] 前田佳男, 水川真, 安藤吉伸, 吉見卓: "空間知を活用した物品運搬サービスの構築", 日本機械学会ロボティクス・メカトロニクス講演会, 講演論文集, 1A1-M04, 2013.
- [5] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng: "ROS: an open-source Robot Operating System", ICRA workshop on open source software, Vol.3, No.3.2, 2009.
- [6] 表允哲, 辻徳生, 橋口優香, 永田晃洋, 中島洗平, 倉爪亮, 長谷川勉, 諸岡健一: "情報構造化アーキテクチャの提案とサービスロボットのオンライン動作計画の実現", 第19回ロボティクスシンポジウム講演予稿集, 6D2, pp.624-630, 2014.
- [7] Jonathan Boren and Steve Cousins: "The SMACH High-Level Executive", IEEE Robotics & Automation Magazine, pp.18-20, 2010.
- [8] 中島洗平, 長谷川勉, 辻徳生, 諸岡健一, 倉爪亮: "ワゴンを利用した生活環境中の日用物品運搬システムの開発", 第14回システムインテグレーション部門講演会, 2M2-3, pp.1819-1824, 2013.