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
ー第 2 報 異種ロボットによる作業情報構造化実験ー

Task management system for informationally structured architecture ROS-TMS
-Structured task information management experiments for a variety of robots-

橋口優香 (九州大)　○ピョユンソク (九州大)
正　辻徳生 (九州大)　諸岡健一 (九州大)　正　倉爪亮 (九州大)

Yuuka HASHIGUCHI, Kyushu University
Yoonseok PYO, Kyushu University, pyo@irvs.ait.kyushu-u.ac.jp
Tokuo TSUJI, Kyushu University, tsuji@ait.kyushu-u.ac.jp
Ken'ichi MOROOKA, Kyushu University, morooka@ait.kyushu-u.ac.jp
Ryo KURAZUME, Kyushu University, kurazume@ait.kyushu-u.ac.jp

This paper presents a framework of a task management system for the informationally structured architecture, ROS-TMS, and information structured task management experiments using different types of robots and service tasks. The proposed system interprets user's request, plans a proper robot service, issues a series of robot commands suitable for the structure of each robot, and executes the robot service by task executing machine, SMACH. We conducted a go-and-fetch task with two types of robots.

Key Words: Task management, Informationally structured environment, Service robot, ROS

1　緒言

超高齢社会において，医療・介護現場での労働力不足は大きな社会問題であり，その解決策の一つとして，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，サービスロボットの作業環境は多様で動的に変化するため，ロボットに搭載したセンサのみで環境を把握することは困難である．そこで我々は，2005 年から，環境側に固定センサを分散配置し，各センサから得られるデータをネットワークを介して取得・分析・保持するためのシステム (Town Management System，TMS) を構築してきた [1]．さらに 2012 年からは，TMS の機能拡張性やノード間通信の柔軟性を高めるために，オープンソースのミドルウェア ROS をベースにした TMS「ROS-TMS」の開発を開始している [2]．ROS-TMS の導入により，搭載可能なセンサ数やその性能に制限のあるロボット単体であっても，必要な情報をシステムから補充しながら，作業を遂行することができる．この概念を環境情報構造化，ロボットが作業しやすいよう整備された空間を情報構造化環境と呼ぶ (図 1)．

本稿では，ロボットが行う作業 (タスク) を実行・管理するタスクスケジューラモジュール [3] において，ROS-TMS と親和性の高いタスク記述法を新たに開発した．さらに，多様なロボットやサービスに対応できる新たな作業情報構造化アーキテクチャを提案し，その有効性を実機実験により検証した．

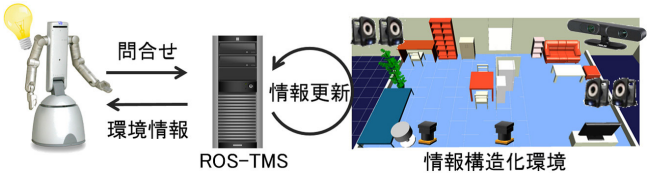

Fig.1 ROS-TMS

2　関連研究

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の概念が提案されている [4],[5]．ここではまず，システムを多様な環境に対応させるために，環境内のものに関するデータ記述を統一・定型化している．さらに，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，ロボットプログラムをシナリオ，スキル，スキルソリューション，サーボの 4 層に分け，従来各層に分散していたロボット固有の形態情報をシナリオ・スキル層に含めない実装手法を採用している．ロボット固有の形態情報を必要としないシナリオ・スキル層を共通レイヤ，それ以外を個別レイヤと呼ぶ．また，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 [6],[7]．空間知ミドルウェアは，ユーザインタフェース，センサ，ロボットを接続する機能と，ユーザからの要求を解釈し，要求に応えることのできる RT 要素の組み合わせ・実現方法を動的に生成する機能を持つことを目的に構築されている枠組みである．

3　ROS-TMS におけるタスク管理機構

3.1　ROS-TMS

ROS-TMS は，我々が開発している環境情報構造化プラットフォームである．上位モジュールとして，ユーザからの要求を受け付けてシステムに送信するユーザリクエストと，それを解釈してタスクを実行するタスクスケジューラが，センサ系のモジュールとして，センサドライバと，センサ情報からより高次のデータを抽出するためのセンサシステム及びステートアナライザが，ロボット系のモジュールとして，ロボットドライバとロボットの動作計画を行うロボットプランニングが実装されている (図 2)．これらは，ROS のパッケージとしてオープンソース化 [1] されている．

[1] https://github.com/irvs/ros_tms

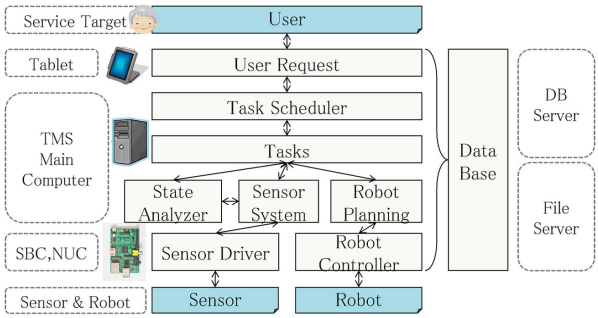

Fig.2 modular composition of ROS-TMS

3.2 作業情報構造化を取り入れたタスク設計

作業情報構造化の実現には，ロボットプログラムを，ロボットのハードウェア構成依存･非依存の観点から切り分けた実装が必要である．本節では，ロボットの種類や作業内容に対するプログラムの汎用性を向上させるために，ROS-TMS においてこの概念をどのように導入・実現したかについて述べる．

3.2.1 サブタスク

タスク設計にあたり，移動・把持・受け渡しなどの，タスク間で共有できるロボットの基本動作としてサブタスクを定義する．タスクを任意のサブタスクの集合として定義することで，タスク記述をシステムで統一でき，効率的なタスク管理が行える．また，タスク間で必要な機能を使いまわすことができ，プログラムの再利用性が向上する．

3.2.2 共通レイヤと個別レイヤ

ROS-TMS において，システムがユーザから要請を受け取った後，ロボットに命令コマンドを送信するまでに，ユーザリクエスト，タスクスケジューラ，ロボットプランニング，ロボットコントローラモジュールが使用される (センサ系のモジュールは常時稼働しており，必要な情報はデータベースに保存されている)．このうち，ロボットコントローラモジュールでは各ロボット固有のドライバが起動している．一方で，ロボットプランニングモジュールでは，移動経路やアームの軌道計算のために，ロボットの移動ベースの形状やマニピュレータの自由度，リンク間距離などの情報が必要になる．そこで，各サブタスクについて，ロボットプランニングモジュールで，ロボット ID ごとに分岐して種々の計算を行わせる実装を行う．これにより，上位モジュールでは，ロボットのハードウェア構成に依存しない情報だけを扱うことができる．すなわち，ROS-TMS においては，個別レイヤをロボット系のモジュールに，共通レイヤを上位モジュールに位置付けている．

なお，現在は，ユーザが使用したいロボットを選んでシステムに通知する機構を採用しているが，将来的にはユーザがしてほしいことをシステムに要請するだけで，システム側で使用できるロボットやその仕事に適していて効率的にサービスを提供できるロボットを選択することが望ましい．システムがロボット選択機構を備えることで，上位モジュールはロボットの情報が完全に不要になり，より高度な作業情報構造化が実現できる．

3.3 タスクスケジューラ

タスクスケジューラモジュール (TS) は，ユーザリクエスト (UR) から送信された要求を受け付け，必要な情報をデータベース (DB) と連携して補充しながら要求を解析，ロボットコマンドを発行して作業を実行させる機能を持つ．タスク実行の流れを図 3 に示す．まず，ユーザからタスクが要請されたとき (4.2①)，そのタスクがどのようなサブタスクの集合からなるかをデータベースに問い合わせる (4.2②，③)．次に，得られたサブタスクの集合を解釈して，後述のタスク実行マシンへの入力となるスクリプトを発行する (4.2④)．そして，スクリプトに記述された順序・並列度に従って，各サブタスクをスレッドで実行させる (4.2⑤)．タスクスケジューラから実行命令が出されたサブタスクは，ロボットプランニングモジュール (RP) において与えられた抽象的な情報をデータベースと連携しながら具体的な数値データに変換し，指定されたロボットを動かすためにロボットコントローラモジュール (RC) にコマンドを送信する (4.2⑥)．

各サブタスクはスレッドで起動されるので，タスクスケジューラでは，現在走っているスレッド数と終了したスレッド数の整合性をとる必要がある．そこで，各サブタスクは，サブタスクが正常に終了したかやロボットプランニング，ロボットコントローラレベルでエラーが発生し，タスクが異常終了したかといった通知を実行タイミング制御ノードに送信する (4.2⑦)．全てのサブタスクが正常終了すると，要求されたタスクの完了を通知する (4.2⑧)．

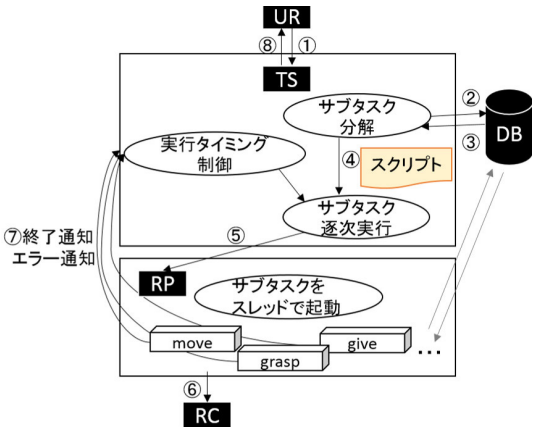

Fig.3 TMS_TS architecture

3.3.1 SMACH を用いたタスク実行

タスクスケジューラにおいて，タスクの実行には，タスク実行マシン SMACH[10] を用いる．SMACH は，複雑なロボットのふるまいを簡単に記述し，生成するためのタスクレベルのアーキテクチャである．SMACH は，ROS 依存の Python ライブラリとして提供されており，タスクレベルのロボット作業を記述・実行するための階層型状態機械を生成することができる．本システムでは，1 つの状態を 1 つのサブタスクに割り当て，サブタスクを直列または並列に接続した状態遷移図でタスクを表現する．

3.3.2 タスク記述

新たに構築したタスク記法について説明する．前提として，タスクは文字列型の情報として管理する．記法は，逆ポーランド記法を採用し，トークンの区切り文字を「 (スペース)」，サブタスクを直列に接続するか，並列に接続するかをそれぞれ演算子「+」,「|」で表す．終端文字を読み込むまで以下の手順を繰り返して字句解析を行い，対応する SMACH 実行スクリプトを出力する．

1. 特殊文字スペースで区切ったトークン (サブタスクまたは演算子) を，左端から読み込む．

2. 参照した文字がサブタスクブロックであればスタックにプッシュする．演算子ブロックであれば，既にスタックに積まれているブロックを 2 つポップして演算子で繋ぐ．

ここで，サブタスクは，サブタスク ID(9001〜9999) と 1 個以上の引数を $ で繋いで表記する．例えば，サブタスク ID9001 の move が，引数として目的地の棚 ID(6010) をとるとき，そのサブタスクは「9001 $ 6010」と表される．以上より，データベースにはタスク情報として図 4 のような文字列が格納されることになる．

type	id	name	etcdata
IDtask	8000	----------	
task	8001	get_object$rid	9001$oid 9002$oid + 9003$uid +
task	8002	patrol$rid	9001$rid 9006$oid 9007$oid | +

Fig.4 Task description in Database

4 実験

新たに構築したタスク設計・管理機構を用いて，多様なロボットに対して作業設計が可能な，柔軟で汎用性の高いアーキテクチャが構築できることを実験により検証する．具体的には，構築した命令系統を用いて，異なるハードウェア構成を持つ 2 台のロボットが，同じタスクを実現できることを実機で示す．

4.1 使用するロボット

4.1.1 SmartPalV(安川電機)

本稿では，棚の中の物品を把持するために，多自由度双腕ロボット SmartPalV(安川電機) を使用する (図 5)．仕様を表 1 に示す．

外寸	幅 555mm× 奥行 613mm× 高さ 1325mm
重量	119kg(バッテリ，センサ含む)
自由度	21 自由度（腕 7×2，手 1×2，腰 2，移動 3）

Table 1 SmartPalV specification

4.1.2 KXP(九州大学)

KXP は，Katana(Neuronics)，Xtion(ASUS)，Pioneer3-AT(MobileRobots) を組み合わせて開発したロボットである (図 6)．Pioneer3-AT は本来，4 輪スキッドステアの移動台車型ロボットであるが，後輪をキャスタに変更し，室内でその場回転ができるなど，小回りが利くように改良してある．仕様を表 2 に示す．

外寸	幅 500mm× 奥行 600mm× 高さ 960mm
重量	17kg
自由度	8 自由度（腕 5×1，手 1，移動 2）

Table 2 KXP specification

Fig.5 SmartPalV

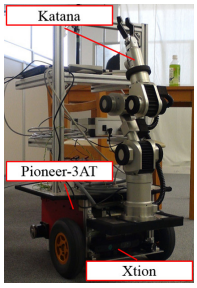

Fig.6 KXP

4.2 物品取り寄せタスク

SmartPalV は，初期値 (5500mm，1500mm，180degree) から，環境左奥にある茶色い棚に置いてあるチップスターを取り，人 (6500mm，1000mm，180degree) に渡すというシナリオで，また，KXP は，初期値 (4700mm，800mm，0degree) から，環境右奥にある茶色い机の上に置いてあるチップスターを取り，人 (1250mm，3000mm，-45degree) に渡すというシナリオでそれぞれ実施した．

ここで，実環境における移動体の自己位置推定は，高精度のモーションキャプチャシステム VICON を使用し，本体に取り付けられたマーカを追跡して決定している．追跡される対象は，ロボットとワゴン (移動家具)，人である．これらの位置姿勢はリアルタイムで ROS メッセージとして発行され，シミュレータに反映させることができる．人の位置は，人が装着するスマートグラス MOVERIO(EPSON) にマーカを貼り付け，その位置を反映させている．実験では，MOVERIO を三脚に固定したものを人の位置姿勢として取得し，システムに反映させている．図 7，8 では，人間のシルエットをオーバーレイ表示している．

4.3 実験結果

図 7，8 に，実機実験の様子をサブタスクごとにキャプチャした図を示す．「SmartPalV，私にチップスターを持ってきて」というユーザの要求は，図 4 の get_object を，rid=2002(SmartPalV)，oid=7001(チップスター），uid=1001（ユーザ）として呼び出し，TS で

1. SmartPalV がチップスターの前に移動 (9001)

2. SmartPalV がチップスターを把持 (9002)

3. SmartPalV がユーザへ物品受け渡し (9003)

という 3 つのサブタスクに分解され，以下のスクリプトが発行される (一部抜粋)．

```
smach.StateMachine.add('move',
  ServiceState('rp_cmd',
    rp_cmd,
    request = rp_cmdRequest(9001, 2002, [7001])),
    transitions={'succeeded':'grasp'})

smach.StateMachine.add('grasp',
  ServiceState('rp_cmd',
    rp_cmd,
    request = rp_cmdRequest(9002, 2002, [7001])),
    transitions={'succeeded':'give'})

smach.StateMachine.add('give',
  ServiceState('rp_cmd',
```

```
    rp_cmd,
    request = rp_cmdRequest(9003, 2002, [1001])),
    transitions={'succeeded':'succeeded',
                             aborted':'aborted'})
```

その後，各サブタスクが実装された RP で，必要な情報をデータベースと連携して補完しながらタスクを遂行する．

また，同じタスクを他のロボットで行いたいときも，ロボット ID を変更するだけで，システム内部で必要な計算に分岐してタスクが遂行できることを確認した．

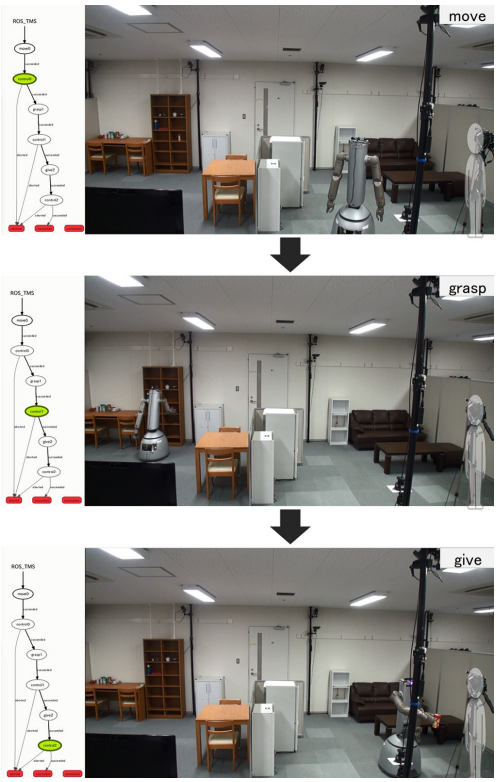

Fig.7 物品取り寄せタスク (SmartPalV)

5 まとめ

情報構造化環境において，多様なロボットに対しその作業設計ができる，柔軟で汎用性の高い作業情報構造化アーキテクチャを提案した．タスク設計では，プログラムの再利用性を高め，システムの命令系統を統一するために，ロボットのタスクを，タスク間で共有できるサブタスクの集合として定義した．さらに，ROS-TMS に作業情報構造化の概念を導入するため，ロボットプランニングモジュール内でサブタスクを実装してロボットのハードウェア構成の差異を吸収，これより上位のモジュールではロボットに依存しない情報を扱える機構を構築した．タスク管理では，ROS-TMS の上位モジュールとしてタスクスケジューラモジュールを構築した．ここでは，設計したタスクを一意に管理するために，新たなタスク記述法を定義し，データベースと連携しながらタスク実行マシン SMACH を介してそれを遂行する一連の機構を構築した．開発したシステムに対し，実機実験により，提案したシステムの有効性を示した．

本研究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．

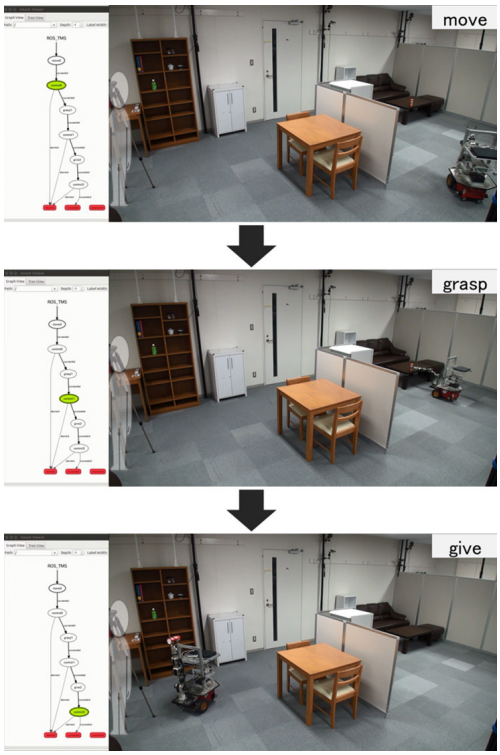

Fig.8 物品取り寄せタスク (KXP)

References

[1] 長谷川勉:“環境プラットフォーム「ロボットタウン」”，日本ロボット学会誌，Vol.26 No.05，pp.411-414，2008.

[2] ピョユンソク, 辻徳生, 橋口優香, 永田晃洋, 中島洸平, 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ア講演予稿集, 6D2, pp.624-630, 2014.

[3] 橋口優香, ピョユンソク, 辻徳生, 諸岡健一, 倉爪亮:“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”, 第 32 回日本ロボット学会学術講演会, 3C2-03, 2014.

[4] 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.

[5] 香月理絵, 西山学, 大賀淳一郎, 吉見卓, 菅原淳, 中本秀一: “異機種ロボット間のプログラム共通化を目指したソフトウェア構造-実装方法と東芝の双腕自律移動ロボットでの検証- ”, 第 9 回システムインテグレーション部門講演会, 1L2-4, pp.471-472, 2008.

[6] 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.

[7] 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.

[8] 中岡慎一郎:“拡張可能なロボット用統合 GUI 環境 Choreonoid”, 日本ロボット学会誌, vol.31, no.3, pp.12-17, 2013.

[9] 辻徳生, 原田研介:“grasppluginfor choreonoid”, 日本ロボット学会誌, Vol. 31, No. 3, pp. 232-235, 2013.

[10] Jonathan Boren and Steve Cousins: “ The SMACH HighLevel Executive ”, IEEE Robotics and Automation Magazine, pp.18-20, 2010.