

クラウドロボティクスのための タスク分散フレームワーク

齊藤 由香利^{†1} 五十嵐 登^{†1} 大山 直人^{†1}
阪口 和明^{†1} 角田 龍太^{†1} 中川 幸子^{†1}
中山 央士^{†1} 清水 将吾^{†1}
成田 雅彦^{†1} 加藤 由花^{†1}

近年、インターネット分野とロボット分野の融合が加速しており、クラウド環境への適用を前提とした様々なプラットフォームが提案されている。我々はこれまで、ロボットサービス向けプロトコル仕様である RSNP を利用し、様々なデバイス、ロボット、サービス機能の連携を実現する分散処理フレームワークを構築してきた。本稿では、このフレームワークを拡張し、複数種類のタスク分散アルゴリズムを環境に応じて切り替え可能な仕組みを提案する。また、評価実験により、その有効性を確認する。

A Distributed Service Framework for Cloud Robotics

YUKARI SAITO,^{†1} NOBORU IGARACHI,^{†1}
NAOTO OHYAMA,^{†1} KAZUAKI SAKAGUCHI,^{†1}
RYUTA TSUNODA,^{†1} SACHIKO NAKAGAWA,^{†1}
HISASHI NAKAYAMA,^{†1} SHOGO SHIMIZU,^{†1}
MASAHIKO NARITA^{†1} and YUKA KATO^{†1}

Recently, many ICT companies as well as researchers are taking an increasing interest in robot services using cloud computing, and various service platforms for them have been proposed. From these background, we have developed a distributed service framework using Robot Service Network Protocol to integrate various devices and robots with service functions. In this paper, we extend the framework, and propose a scheme to apply suitable task distributed algorithms according to system environment. In addition, we conduct simulation experiments, and verify the effectiveness of the proposed mechanism.

1. はじめに

近年、クラウドロボティクスへの関心が高まっている。クラウドコンピューティングはインターネット分野の技術であるが、コンピュータのユビキタス化に伴い、Internet of Things¹⁾、Cyber-Physical Systems²⁾ など、実世界サービスとの融合が急速に進んでいる。ロボットやロボットサービスのネットワーク化については、これまでも RT ミドルウェア³⁾ やネットワークロボットフォーラム⁴⁾ などの活動があったが、上記分野との融合を視野に、ロボットサービスのクラウド化が進んでいる。例えば、RoboEarth⁵⁾ や Cloud Robotics⁶⁾ などの研究が行われている。

このような背景の下、我々はロボットサービスのインターネット化を目指して、RSNP (Robot Service Network Protocol) の研究を進めてきた⁷⁾。RSNP はロボットサービスをインターネット経由で利用・共有するためのロボットサービスプロトコルであり、業界団体である RSi (Robot Service Initiative)⁸⁾ により仕様化されている。Web サービス技術を基盤としていることから、インターネット分野の開発者を取り込み、ロボットサービスの普及に寄与してきた。しかし、RSNP はロボットとロボットサービス間の通信プロトコルを定義したものであり、ユーザの要求に対し複数のロボット間で処理を分散させるようなサービスの場合、スクラッチからアプリケーションを開発する必要があった。

そこで我々は、RSi のロボットサービスモデルを拡張し、サービス要求を行うユーザと、実際に処理を実行するロボットとの連携を疎結合とすることにより、複数ロボット間の処理分散を容易に実現するフレームワークとして Jeeves フレームワーク^{*1} を提案してきた⁹⁾¹⁰⁾¹¹⁾。ロボットを対象としたサービスでは、ロボットの動きやサービスの提供状態により動的で不安定なネットワーク構成が取られる場合が多く (ex. ロボットが移動することにより、無線 LAN の通信エリアから外れてしまうなど)、結合度の低いアーキテクチャが要求される。そのため、ユーザからの要求に応じ、複数のロボット間で処理分散が可能なアーキテクチャとして Publisher/Subscribr モデルを採用している。

Jeeves フレームワークは、クラウド側でロボットの状態管理を行わないところに大きな特徴があり、これまで、ユーザからの要求は先着順でロボットに割り当てられていた。しかし、具体的なロボットサービスを検討していくと、フレームワーク側に、ある条件にマッチした

^{†1} 産業技術大学院大学

Advanced Institute of Industrial Technology

*1 英国の小説に登場する有能な執事の名前。ロボットをメイドととらえ、メイドを指揮管理する者として命名した。

ロボットを割り当てる機構が必要であることが明らかになった。そこで本稿では、Jeeves フレームワークにタスク割当機構を導入し、シミュレーション実験によりその有効性を検証する。割当ルールはフレームワーク上で提供されるサービスにより異なるが、Jeeves フレームワークではこのルールを入れ替え可能な仕組みを実現し、これに対応する。

2. Jeeves フレームワーク

本章ではまず、Jeeves フレームワークで利用している RSNP について述べた後、フレームワークの概要を説明する。

2.1 RSi のロボットサービスモデル

RSi では、ロボットサービスを、ネットワークを介してロボットが提供する情報サービス、もしくは物理的なサービスと定義している。図 1 にモデルの概要を示す。このモデルは、ロボットやサービスプロバイダ、サービスポータル、ユーザなどから構成され、同期・非同期の通信による動作や動作パターンの指示や結果の取り出し、ロボットからプロバイダへの問い合わせ・通知、サービスの提供、ユーザを含む外界とのやり取りを行うことができる。

このようにロボットをインターネットに接続する利点は、人とロボットが協調してサービスを行う「協調型」ロボットを実現できること、インターネット上のコンテンツを再利用できることなどにある。RSNP は、このモデルに従ってサービスのプロトコルを規程しており、異なるベンダで独立して開発したロボット／サービスの間での相互運用が可能である。RSNP のシステムアーキテクチャを図 2 に示す。

プラットフォームのベースは、インターネットやシステム構築向け通信基盤である Web サービス基盤を利用している。そのため、高信頼メッセージング機能、セキュリティ機能等、インターネットとの整合性の高い標準化された機能を利用可能である。プラットフォーム自体は共通サービスとロボットサービスから成り、共通サービスでは、各種サービスをサポートするために、PULL 型・PUSH 型、同期・非同期型の通信モデルを提供している。一方、ロボットサービスは基本プロファイルと応用プロファイルから成り、カメラ・音声入出力などのマルチメディア機能や、前後回転動作など単純な動作・パターン動作などのロボットの動きを基本プロファイルとして提供し、情報サービス・天気サービス・防災サービス・見守りサービス・リモート制御などのサービスを応用プロファイルとして提供する。

2.2 フレームワークの構成

Jeeves フレームワークのシステム構成を図 3 に示す。ユーザは、HTTP 経由で実行したいサービス要求をフレームワークに送信する (Publisher)。ロボットは、自身が実行可能な

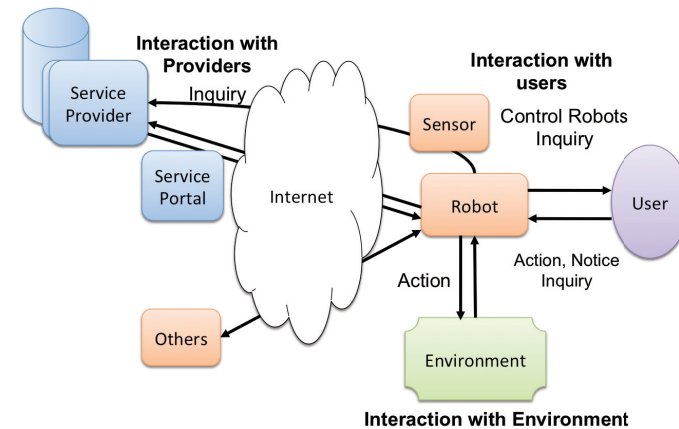


図 1 RSi のロボットサービスモデル
Fig.1 The robot service model on RSi.

タスクを RSNP 経由でフレームワークから取得する (Subscriber)。ここで、ロボット側で実行されるタスクは RSNP アプリケーション (RSNP クライアント) として実装されており、クラウド側で実行されるタスクは RSNP サービス (RSNP サーバ) として実装されている。サービス要求は複数の RSNP サービスの組み合わせとして定義されている場合もあり、この対応付けは Jeeves フレームワーク内で行われる。つまり、Jeeves フレームワークは、サービス要求と RSNP サービスの対応リスト、RSNP サービスと RSNP アプリケーションの対応リストを保持しており、このリストを基にサービス要求をロボットタスクに割り当てる (図中 (2))。

このとき、処理開始のトリガはロボット側のタスク取得のタイミングとなるため、サービス要求とタスク取得の処理は非同期に行われることになる。ただし、ロボットとクラウド間の通信は、RSNP サービスと RSNP アプリケーション間のやりとりになるため、ロボットにタスクが割り当てられた後は、同期・非同期いずれの処理も可能である (図中 (3))。

以上の構成から、Jeeves フレームワークの特徴は以下の 3 点にまとめられる。

- 疎結合：ユーザからのサービス要求は、ロボットの状態、実装を意識せずに行える。つまり、サービス要求とロボット処理の分離が可能である。
- 複数機能の協調：ユーザからのサービス要求は、複数ロボットで同時に処理が可能である。これは、サービス要求を複数の RSNP サービスと対応させること、または複数ロ

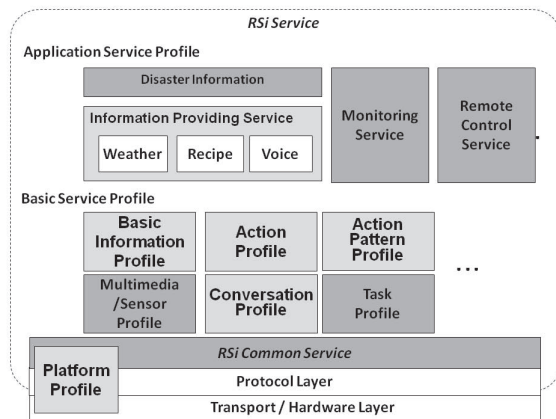


図2 RSNP のシステムアーキテクチャ
Fig.2 System architecture of RSNP.

ロボットを同時に利用する RSNP サービスを実装することにより可能である。

- 同一機能の負荷分散：ユーザからのサービス要求は、同一の機能を持つ複数台のロボットで分散処理させることが可能である。ここでは、負荷分散を行う RSNP サービスを実装することで対応する。RSNP を利用することによりロボットの実装が隠蔽されるため、負荷分散サービスの実装は比較的容易である。

3. タスク割当機構

ここまで述べてきたように、Jeeves フレームワークでは、ユーザからのサービス要求を複数の RSNP サービスに対応付け、ロボット自身が実行可能なタスクをフレームワークから取得することにより、タスクの割当を実現してきた。つまり、あるタスクを実行可能なロボットが複数存在する場合、負荷分散機構を導入していない限り、先にタスクを取得したロボットに処理が割り当てられる。しかし、フレームワークを利用したロボットサービスを考察すると、先着順ではなく、明示的なタスク割当機構が必要である。本章ではこのタスク割当機構の設計結果を述べる。

3.1 要求条件

まず、クラウド環境への適用を前提とした具体的なロボットサービスを想定し、それらが要求するタスク割当機構について考察する。ロボットサービスとしては、以下に示す 4 種類

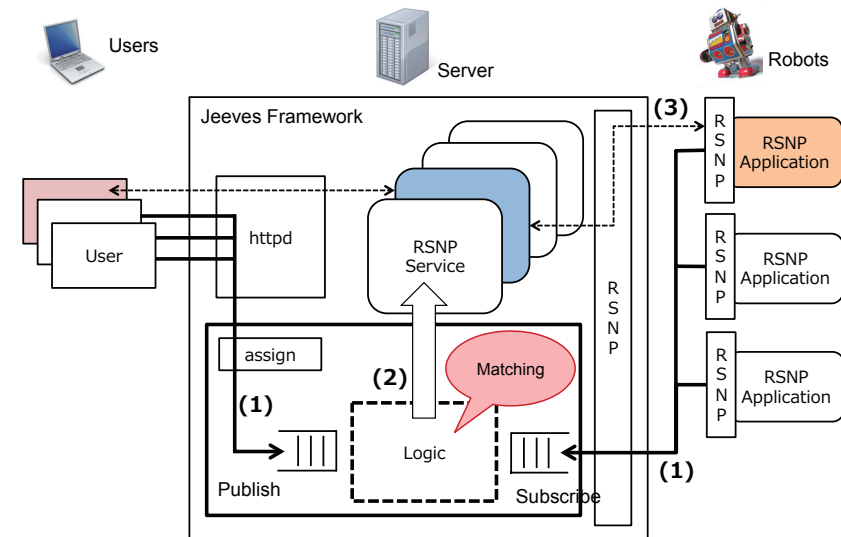


図3 フレームワークのアーキテクチャ
Fig.3 System architecture of Jeeves framework.

のサービスを取り上げる。これらのサービスは文献¹²⁾において設計されたものである。

- ペットシッターサービス：動物病院やペットホテルで、遠隔地の飼い主とペットのコミュニケーションを支援するペットの見守りサービスである。本サービスは遠隔地からペットの様子を確認する機能と、ペットライフログの取得により定期的な健康、環境チェックを行い、異常があれば飼い主及び施設職員に知らせる機能を有し、可能であればロボットが自律的に異常に働きかけて対応を行う。イメージを図4に示す。
- 観光ロボットレンタルサービス：観光地に配備されたロボットの遠隔レンタルサービスである。ユーザは、観光現地に赴かなくても、ロボットを遠隔で操作することができ、観光を楽しむことができる。
- 最適な場所を探索するサービス：設定した目的に対して最適な場所を複数ロボットで探索し、一つの地図を完成させるサービスである。例えば、複数の探索ロボットを走らせ桜木の開花状況を探知し、リアルタイムなお花見の最適ポイントの発見と最短経路を発見後、場所取り機能を持つロボットがポイントへ駆け付けて場所取りを行う等の複合的なサービス展開が期待できる。

- コンシェルジュサービス：博覧会や美術館において、個人の趣向や過去の経験等に合わせ、ブースや経路を案内し、展示物や作品の解説を提供するガイドツアーサービスである。展示物や作品の情報をクラウド環境に保管することで、地域巡回型の企画展や他の美術館でも、出展先に設置されたロボット端末を利用して、クラウド内のデータを容易に流用できる。

これらのサービスは、サービス利用者の視点から「遠隔操作型」「情報活用型」の2種類に分類できる。前者2つのサービスは遠隔操作型であり、ファイアウォール内に設置されたロボットの遠隔操作を人の手で行う。人の代わりに、ロボットがその場にはいない個人の行動を補うサービスの特徴としている。一方、後者2つのサービスは情報活用型であり、クラウド環境に大量に蓄積されたデータの分析を行い、情報の共有と配信を行う。その場での個人の行動を支援する情報サービスの特徴としている。

これらのサービスは、ユーザがロボットの状態や実装を意識せずに利用できることが望ましく、Jeeves フレームワークに適したサービスである。例えば、ペットシッターサービスであれば、ユーザはサービス要求として「ペットの見守り」をフレームワークに依頼する。この要求は、RSNP サービスとして実装された「Web カメラによる見守りサービス」に対応付けられ、このサービスはペットホテルに設置された複数台の「Web カメラ」がRSNP クライアントとして実行可能である状況等が考えられる。この例の場合、先着順で一番先にタスクを取得した Web カメラがサービスを提供することになるが、サービスの目的からは、対象物に最も近い、または対象物を最も適切に撮影できる Web カメラが選択されるべきである。つまり、タスクの取得要求をキューイングしておき、サービスの目的に最もマッチしたロボットを選択することが望ましい。

今回考察対象とした遠隔操作型サービスでは、遠隔地という場が重要な意味を持つため、位置情報の利用が有効である。一方、情報活用型サービスでは、適切なサービスを選択することが意味を持つため、ロボット種別の利用が効果的であると考えられる。ロボット種別は遠隔操作型サービスでも有効な情報であるため、本稿では、「位置情報」および「ロボット種別」をタスク割当機構におけるパラメータとして利用することとした。

3.2 割当方式の設計

2章で述べたように、Jeeves フレームワークは、疎結合、複数機能の協調、同一機能の負荷分散という3つの特徴を持つ。本稿では、これらの特徴に適したタスク割当方式を設計する。

まず、1番目の要件に対しては、フレームワークがロボットの状態を常時監視するという

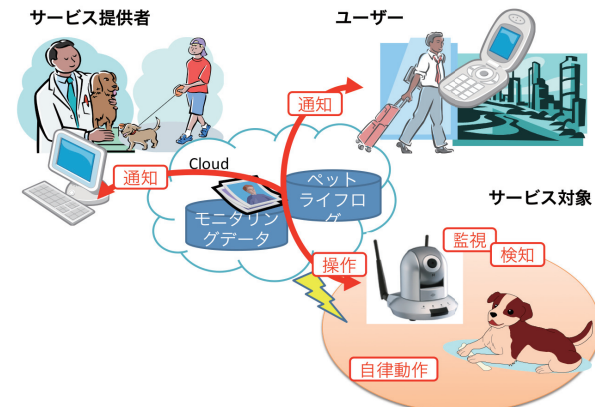


図4 ペットシッターサービスのイメージ
Fig.4 An image of a pet-sitting service.

ことはしないので、ロボットがタスクの取得をフレームワークに要求するときにその要求をキューイングし、前節で規程したパラメータを付加することとした。ただし、キューに滞在する時間はタイマーで管理し、指定した時間を経過した要求は破棄する。2番目の要件に対しては、異なるタスクを異なるRSNPサービスと対応付ける、または異なるタスクを受け付けるRSNPサーバを実装することで対応する。3番目の要件に対しては、1番目と同様、タスク取得の要求をキューイングすることにより対応する。

Jeeves フレームワークにおけるタスク割当のフローを以下に示す。今回の設計では、キューイングされた複数のタスク取得要求と、ユーザからのサービス要求をマッチングする必要があるため、処理開始のトリガーを、タスク取得のタイミングから、サービス要求のタイミングに変更している。ここで、ユーザからのサービス要求を $\{S_1, S_2, S_3, \dots\}$ 、クラウド側で提供可能なRSNPサービスを $\{s_1, s_2, s_3, \dots\}$ 、ロボット側で実行可能なRSNPアプリケーションを $\{c_1, c_2, c_3, \dots\}$ とする。2章で述べたように、フレームワークは、サービス要求とRSNPサービスの対応リスト、RSNPサービスとRSNPアプリケーションの対応リストを保持している。

Step 1 タスク取得要求の登録：

フレームワークは、ロボットからのタスク取得要求 c_i を受け取り、これをSubscriber側のキューに入れる。 c_i は、ロボットの位置情報 $loc_c(i)$ 、ロボット種別(機能) $fun_c(i)$ 、

時間の経過ごとに減算されるタイマー T_c を保持している。

Step 2 サービス要求の登録：

フレームワークは、ユーザからのサービス要求 S_j を受け取り、対応する RSNP サービスの一覧 s を抽出する。これを Publiiser 側のキューに入れる。 s_k は、ユーザの要求するサービスが実施される位置の情報 $loc_s(k)$ 、ユーザの要求するロボット種別（機能） $fun_s(k)$ 、時間の経過ごとに減算されるタイマー T_s を保持している。

Step 3 前処理：

- (1) $T_c < 0$ となった c_i をキューから取り除く。
- (2) $c = \{\emptyset\}$ なら、タスク取得要求が到着するまで待機する。
- (3) $T_s < 0$ となった s_k をキューから取り除く。

Step 4 割当処理：

事前に指定した割当ルールに従い、 s と c のマッチングを行う。 s_a に c_b が割り当てられたと仮定すると、フレームワークは要求元ユーザに s_a を、 s_a に c_b を通知し、サービスが開始される。

4. 評価実験

本章では、提案した割当手法の有効性を確認するためにシミュレーション実験を行う。本稿の目的は、タスク割当機構の導入により、フレームワーク上で適切なサービスが提供できることを検証することにある。最適な割当方式はサービスの種別により異なるが、ここでは、3章で抽出したパラメータの設定により、どの程度の効果があるかを評価する。なお、今回の実験では、複数機能の協調、同一機能の負荷分散に関する機構は導入せず、優先度に従ったタスク分散機構の検証のみを行っている。

4.1 実験のシナリオ

本実験では、前述した遠隔操作型サービスである「ペットシッターサービス」を想定し、実験シナリオを策定した。具体的には、ロボットカメラを用いて遠隔地の映像を確認するサービスとして、下記のシナリオを用いる。

- 飼い主はペットホテルに預けたペットの状況をインターネット経由で遠隔地から見守る。
- ペットは $10m$ 四方の四角い部屋の中におり、その部屋の中が見守り対象エリアとなる。
- 見守り対象エリアには、エリア内を自由に動きまわることができる複数台のロボットが配備されており、ロボットに設置されたカメラを利用し画像を取得する。ロボットの台数は $3m$ 四方に 1 台程度の密度の配備を想定し、9 台とする。

- ロボットの種別は 3 種類あり、状況に応じて配備されるロボットの種類は異なるものとする。ここでは、ノート PC に接続された Web カメラもロボットの一種である考え、この Web カメラ、高機能なサービスロボット、安価な小型ロボットが、ペットホテル側の都合により異なる組み合わせで配備されるものとする（9 台全てが同種の場合、3 種類全てが配備される場合等、全ての組み合わせを考慮する）。
- 飼い主が利用したいロボットの種別は、状況により異なるとする。カメラによる監視のみに着目すればどのロボットでも可能であるが、餌やり等を行いたい場合は高機能ロボットが選択される。
- ペットおよびロボットはエリア内を自由に移動可能であるが（Web カメラを除く）、モビリティはそれほど高くなく、今回の実験における 1 サイクル内では、どちらの位置も固定されているものとする。
- 配備されているロボットは、実験における 1 サイクル中は全てタスク取得を要求している。

4.2 実験の方法

上記のシナリオに従い、実験で利用するパラメータ値を以下のように設定した。

- 監視対象エリア： $10m \times 10m$
 - ロボットの配置 ($loc_c(i)$)：監視対象エリア内にランダムに 9 台を配置
 - ロボットの種別 ($fun_c(i)$)： A, B, C のうちいずれか 1 つをランダムに選択
 - ペットの位置 ($loc_s(k)$)：監視対象エリア内にランダムに配置
 - 飼い主の指定するロボット種別 ($fun_s(k)$)：以下からいずれか 1 つをランダムに選択 $\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}$
 - ロボットのタスク取得要求の順番：ランダム
 - 実験回数：100,000 回
- ロボットへのタスク割当方式としては、3章での考察に従い、以下の 4 種類の方式を比較することとした。
- ロボット位置指定方式： c の中から、 $loc_s(k)$ との距離が最も小さいロボットにタスクを割り当てる。
 - ロボット種別指定方式： c の中から、 $fun_s(k)$ に合致するロボットに先着順にタスクを割り当てる。
 - ロボット位置+種別指定方式： c の中から、 $fun_s(k)$ に合致するロボットを抽出し、その中で $loc_s(k)$ との距離が最も小さいロボットにタスクを割り当てる。

表 1 実験の結果
Table 1 Experimental results

割当方式	ペットとの距離の平均	指定した種別の適合率
ロボット位置指定方式	1.8m	57.1%
ロボット種別指定方式	5.2m	98.8%
ロボット位置+種別指定方式	2.7m	98.8%
先着順方式	5.2m	57.2%

- 先着順方式：先着順にタスクを割り当てる。比較対象として利用する。

実験シナリオにおいては、飼い主が指定した種別のロボットにタスクが割り当てられること、およびペットの近くのカメラで見守りができることが要求条件となる。そのため、評価尺度としては、飼い主の指定した種別のロボットが選択された割合（適合率）、およびペットとロボットとの間の距離を利用することとした。

4.3 実験の結果

実験の結果を表 1 に示す。種別を指定することにより適合率が向上し、位置を指定することによりペットとロボット間の平均距離が大幅に短縮していることがわかる。以上より、タスク割当機構が有効に働き、フレームワーク上で適切なサービスが提供可能であることがわかった。

タスク割当機構において利用するパラメータや割当ルールは、提供されるサービスによって異なるため、今回の提案方式が全てのサービスに適用できるわけではない。適切な方式をサービスごとに検討していく必要がある。ただし、Jeeves フレームワークにおいては、図 3 に示すマッチング機構の部分を入れ替えることにより、様々な方式の利用が可能な仕組みを実現している。また、3章で分析したように、遠隔操作型、情報活用型サービスでは、位置情報、ロボット種別情報の利用が有効であり、これらを利用した基本となる割当方式を拡張することにより、各サービスに適した手法を容易に開発することが可能であると考えられる。

5. まとめ

本稿では、我々がこれまで提案してきたクラウドロボティクスのための分散処理フレームワーク（Jeeves フレームワーク）を拡張し、ユーザの要求に対して適切なロボットにタスクを割り当てる機構を提案した。また、シミュレーション実験により、その有効性を確認した。

提案システムは、疎結合、自律性を実現するプラットフォームであるが、プロトコル処理、処理分散等によるオーバーヘッドの増加がみこまれ、性能劣化は避けられない。今後、実装性

能評価を進めながら、処理性能の向上に関する検討を進めていく予定である。

謝辞 本研究を進めるにあたり、(株) 富士通研究所の神田真司様、村川賀彦様、岡林桂樹様から貴重なご意見を頂きました。ここに感謝の意を表します。

参 考 文 献

- 1) Guinard, D., Mueller, M. and Pasquier, J.: Giving RFID a REST: Building a Web-Enabled EPCIS, *Internet of Things 2010 International Conference*, (2010).
- 2) Cyber-Physical Systems (CPS) : <http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm>
- 3) Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W.: RT-middleware: distributed component middleware for RT, *IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS 2005)*, pp.3933-3938 (2005).
- 4) 土井美和子, 山本大介, 荻田紀博: 新たなサービスを興すネットワークロボット技術, 東芝レビュー, Vol.64, No.1, pp.32-35 (2009).
- 5) RoboEarth: <http://www.roboearth.org/>
- 6) Kuffner, J.: What's Next: Cloud Enabled Humanoids?, *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010) Workshop*, (2010).
- 7) 成田雅彦, 村川賀彦, 植木美和, 岡林桂樹, 秋口忠三, 日浦亮太, 蔵田英之, 加藤由花: インターネットを活用したロボットサービスの実現と開発を支援する RSi (Robot Service Initiative) の取り組み, 日本ロボット学会誌, Vol.28, No.7, pp.829-840 (2010).
- 8) RSi - Robot Services initiative : <http://robotsservices.org/>.
- 9) 阪口和明, 中川幸子, 大山直人, 中山央士, 五十嵐登, 角田龍太, 清水将吾, 成田雅彦, 加藤由花: ロボットとインターネットサービスの融合を実現する分散処理フレームワーク, 情報処理学会マルチメディア通信と分散処理ワークショップ 2011 (DPSWS2011), pp. 57-62 (2011).
- 10) 中川幸子, 五十嵐登, 中山央士, 齊藤由香利, 大山直人, 角田龍太, 阪口和明, 清水将吾, 成田雅彦, 加藤由花: ロボットとインターネットサービスの融合を実現する分散処理フレームワークの評価, 電子情報通信学会技術研究報告, CNR2011-24, pp.1-6 (2011).
- 11) Nakagawa, S., Ohshima, N., Sakaguchi, K., Nakayama, H., Igarashi, N., Tsunoda, R., Shimizu, S., Narita, M., Kato, Y.: A Distributed Service Framework for Integrating Networking and Applications (AINA2012), (2012). (to appear)
- 12) 中川幸子, 大山直人, 阪口和明, 中山央士, 五十嵐登, 角田龍太, 清水将吾, 成田雅彦, 加藤由花: クラウド環境への適用を前提としたロボットサービスに関する一検討, 日本ロボット学会学術講演会, 3D2-3 (2011).