

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拡張

中川 幸子¹ 成田 雅彦¹ 土屋 陽介¹ 加藤 由花^{1,a)}

概要:我々はこれまで、ロボットサービスのインターネット化を目指し、ロボットサービス向けのプロトコル仕様である RSNP (Robot Service Network Protocol) を利用したクラウドロボティクスのためのサービス基盤「Jeeves フレームワーク」の研究を進めてきた。本稿では、Jeeves フレームワークを拡張し、仮想空間上で提供されるユーザサービスと、実空間上で提供されるロボットサービスをマッピングするためのアーキテクチャを設計する。本フレームワークにより、制約の多い実世界サービスを隠蔽し、ユーザに提供するサービスを仮想空間上に構築することが可能になる。その結果、ロボット技術者は実空間やハードウェアに依存したサービスをフレームワーク上に構築し、Web 技術者はユーザに提供するサービスに適した仮想空間を構築することが可能になる。サービス例の提示により、提案するフレームワークの有効性を検証する。

1. はじめに

近年、RT ミドルウェア [1] や ROS[2] など、ロボットやロボットサービスのネットワーク化への関心が高まっている。特に、クラウドコンピューティング技術の進展により、クラウドロボティクスに関する研究が多く行われるようになってきた。例えば、RoboEarth[3] や Cloud Robotics[4][5] などが注目されている。

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 [6][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。

文献 [9][10][11] では、この 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にすることにより、複数ロボット間の処理分散を容易に実現する「Jeeves^{*1}フレームワーク」を提案してきた。Jeeves フレームワークで

は、Publisher/Subscriber モデルを採用することにより、ロボットの状態をフレームワークが管理しなくても、非同期にサービスを提供できるモデルを提案した。ここでは、ロボットが存在する実空間をクラウド上にデジタルデータまたはプロセスとしてマッピングし、これらをインターネット越しにユーザが利用、操作するモデルを構築している。しかし、ユーザが要求するサービスと、クラウド上で提供されるデータ、プロセスは一致しておらず、これらの機能間のマッピングはアプリケーションごとに個別に実装する必要があった。

本稿では、Jeeves フレームワークを拡張し、ユーザの要求と、ロボットサービスをマッピングさせるアーキテクチャを設計したのでその結果を報告する。本フレームワークにより、制約の多い実空間のサービスを隠蔽し、ユーザに提供するサービスを仮想空間上に構築することが可能になる。その結果、ロボット技術者は実空間やハードウェアに依存したサービスをフレームワーク上に構築し、Web 技術者はユーザに提供するサービスに適した仮想空間を構築することが可能になる。本稿では、サービス例の提示により、提案するフレームワークの有効性を検証する。

2. 関連研究

ネットワークロボット分野におけるクラウド基盤としては、前述の RoboEarth[3] や Cloud Robotics[4][5] などが提案されている。RoboEarth は、クラウド環境をデータストアとして利用し、クラウド上で Web サービスとの連携を実

¹ 産業技術大学院大学
1-10-40 Higashi-Ohi, Shinagawa-Ku, Tokyo 101-0062, Japan

^{a)} yuka@aiit.ac.jp

^{*1} 英国の小説に登場する有名な執事の名前。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

現するプラットフォームである。ただし、共通利用可能なロボットサービスをクラウド上に配備する仕組みは提供されておらず、ロボットサービス自身はロボットサイドに実装する仕組みになっている。一方、Cloud Robotics はこれまでロボットサイドに実装されてきたロボット機能をクラウド側に配備し、ロボット側の処理を軽量化することを目的としている。近年、軽量で簡易なロボットが数多く製品化されており、クラウド上のサービスと組み合わせることにより様々なサービスの提供が期待できる。ただし、様々な種類のロボットで共通利用可能なロボットサービスを開発する仕組みは用意されておらず、サービス開発者にとっては、ロボット上で実装していた機能を、クラウド上で同様に実装する必要がある。

その他、ロボット技術に関するソフトウェア基盤としては、RT ミドルウェア [1] や ROS[2] などが存在するが、これらはハードウェアとロボットアプリケーション間の通信を規定したものであり、クラウド環境は主にデータストアまたは膨大な計算資源として利用されている。

このように、これまで提案されてきたソフトウェア基盤は、いずれもクラウド上のリソースを使ってロボットを高機能化することが主な目的であり、実空間に存在するロボットの投影として、クラウド上のサービスが規定されている。しかし、ユーザにとってのサービスは、必ずしも実空間そのもののマッピングである必要はない。例えば、位置情報に基づく地図サービスであっても、カーナビとグルメマップでは提供の仕方は大きく異なるはずである。サービスごとにユーザに提供される仮想空間を適切に定義し、各種データをマッピングさせることができれば、実世界利用サービスとして様々な応用が可能となる。本稿ではこのような立場から、ロボットサービス基盤のアーキテクチャ提案を行う。

3. 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、これまで提案してきたフレームワークの概要を説明する。

3.1 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

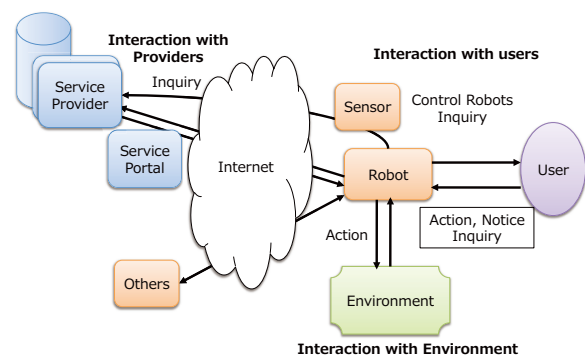


図 1 RSi のロボットサービスモデル
Fig. 1 The robot service model on RSi.

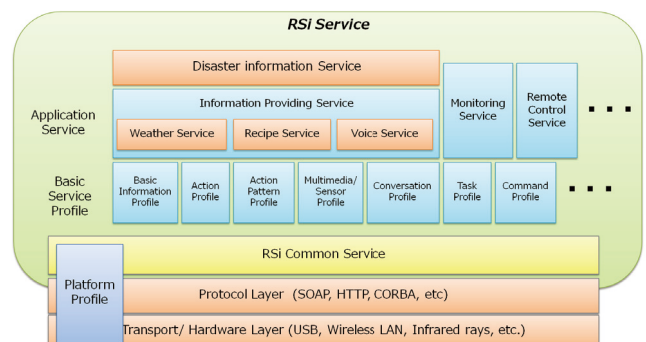


図 2 RSNP のシステムアーキテクチャ
Fig. 2 System architecture of RSNP.

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定しており、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。

3.2 フレームワークの構成

次に、これまで提案してきた 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

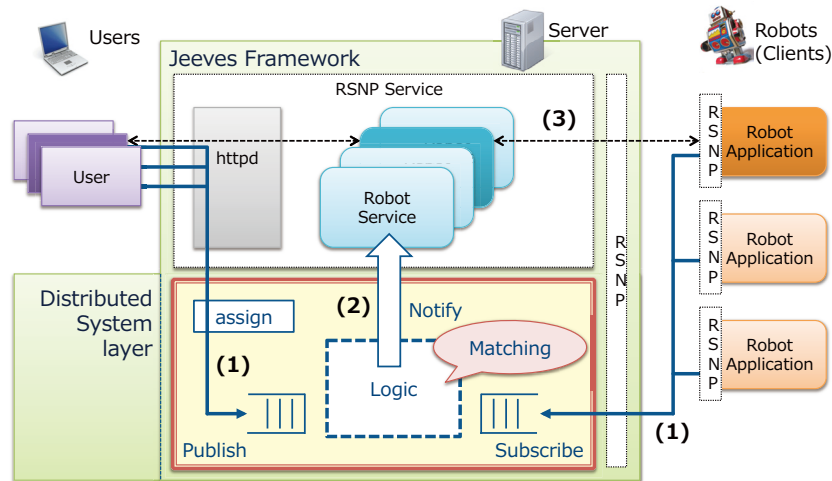


図 3 Jeeves フレームワークのアーキテクチャ
Fig. 3 System architecture of Jeeves framework.

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.

4. フレームワークの拡張

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、タスクの割り当てを実現してきた. タスクの割り当ては、各ロボットが持つメタデータ (位置情報、所持機能、環境等) を利用し、Jeeves フレームワークの (2) の部分でマッチング処理を行っている. 本章では、このマッチング機構を拡張し、ロボットが実在する実空間のモデルと、ユーザがサービスを受ける仮想空間のモデルをフレームワーク上に構築し、それらモデル間のマッピングによりサービス提供を実現するアーキテクチャを提案する.

拡張したフレームワークの構成を図 4 に示す. ここに示すように、サービス基盤の構成要素は下記の 3 層になる.

- (1) ロボットが存在する実空間 (RSNP クライアント)
- (2) 実空間をデジタルデータとしてサーバ上に投影した世界 (RSNP サービス)
- (3) ユーザがサービスを受ける仮想空間 (サービス要求)

フレームワーク上では、RSNP サービス、サービス要求のレイヤが提供される. 実空間に対応するサービスドメインは RSNP クライアントとして実装され、RSNP サービスに対して自身の提供する機能をマッピングする. ユーザはサービス要求レイヤとして定義される仮想空間上でサービスを受け、サービス要求は RSNP サービスにより実空間

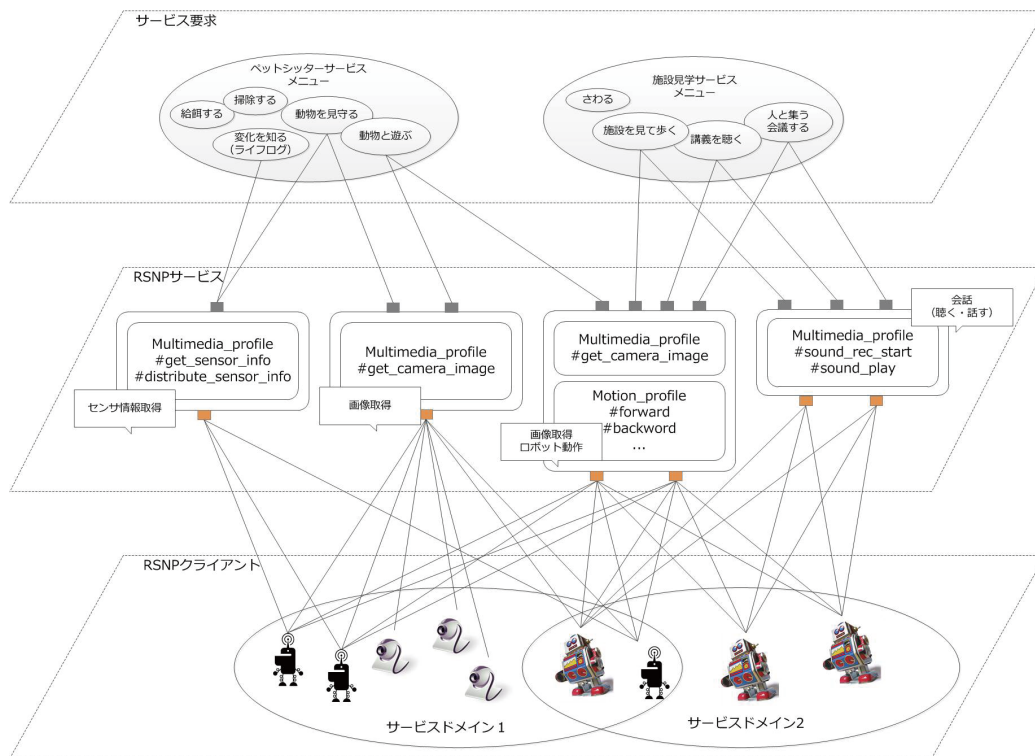


図 4 拡張したフレームワークの構成

Fig. 4 System architecture of enhanced Jeeves framework.

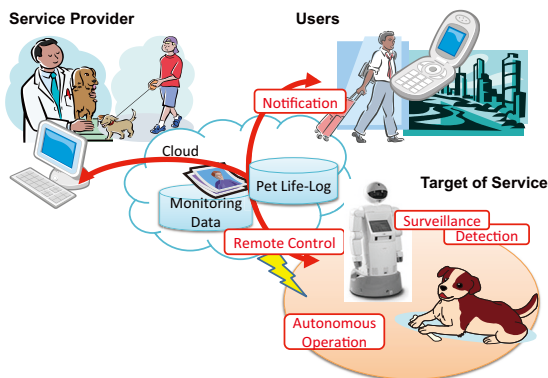


図 5 ペットシッターサービスのイメージ

Fig. 5 An image of a pet-sitting service.

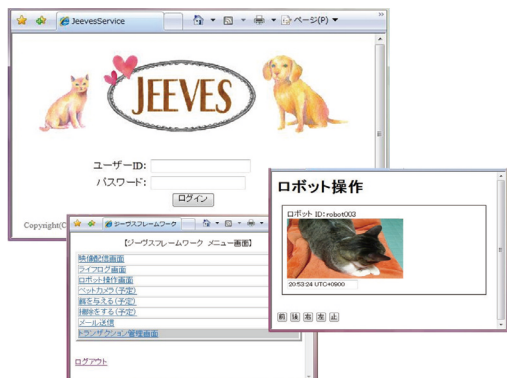


図 6 サービスイメージ

Fig. 6 An image of a provided service.

のサービスにマッピングされ、実際のサービスが実行される。つまり、ユーザは実空間のサービスを仮想空間越しに実行することになる。

Jeeves フレームワークやその他の既存手法では、これまで (2) の空間をユーザに提供するモデルが構築されてきたが、拡張モデルでは (2) と (3) のマッピング方法、実空間の投影として仮想空間はどのようにデザインされるべきか等が考察の対象になる。仮想空間についてはサービス提供者が設計するので、フレームワークとしてはマッピング機能を提供することになる。

具体的には、以下のパラメータを (2) の属性としてロボットからフレームワークに提供する。

位置：サービスの提供対象となる空間の情報。位置データで範囲を指定する場合や、地図データを提供する場合等が考えられる。

状態：利用の可否。例えば、バッテリー切れで動作不可、ネットワークが切断し、通信不可等の情報を提供する。

機能：対象となるロボットの持つ機能。カメラ、マイク、音声合成機能、センサー、把持、歩行・走行などがある。単機能ロボット、複数の機能を持つロボットの両方の存在を許容する。

性能：CPU、メモリサイズ、ネットワーク規格等。

属性：割り当てアルゴリズムの必要に応じて、各種属性を持つことができる。ロボット自身の位置情報、ベンダ等がある。

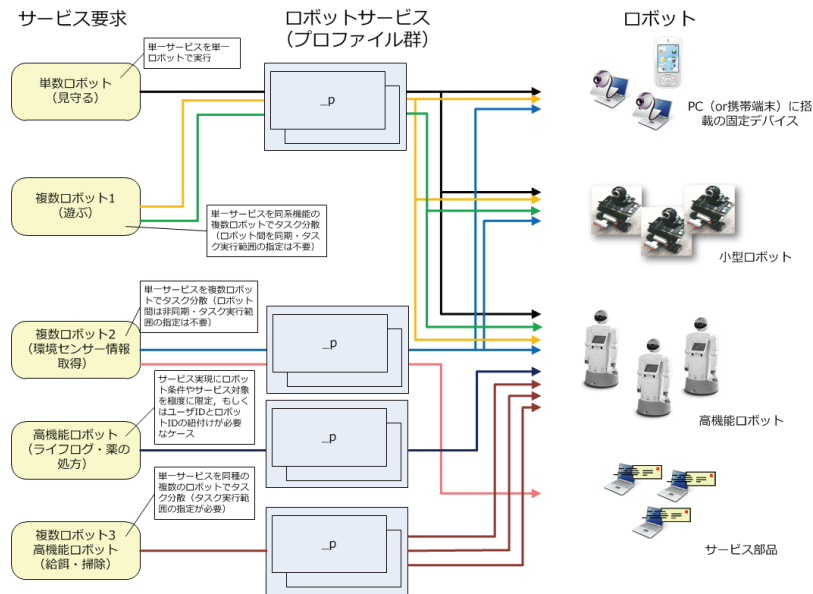


図 7 マッピングのイメージ

Fig. 7 An image of mapping process.

5. 適用例

5.1 サービスの概要

フレームワークの適用例として、ペットシッターサービスを取り上げる。これは文献 [12] において設計し、文献 [11] においてプロトタイプシステムを構築したサービスである。サービスのイメージを図 5 に示す。

具体的には、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。

5.2 サービスのモデル化

ペットシッターサービスの場合、実空間、仮想空間はそれぞれ以下のようにモデル化できる。

実空間：見守り対象となる部屋が実空間になる。その空間内に、カメラ、GPS 等のセンサデバイス、高機能ロボット、小型ロボット等が状況に応じて配置される。

仮想空間：提供サービスの一覧表示、Web 上での監視画面等が仮想空間になる。

本サービスでは、厳密な地図情報等は不要なので、対象エリアの位置情報をサーバ側に提供し、ロボットを含む各種デバイスは RSNP を介して RSNP サーバに接続する。その結果、監視対象の部屋とそこに配置されたデバイスが、実空間のモデルとしてサーバ側にマッピングされる。実空間から提供される機能は下記のとおりである。

- カメラ画像の送信 (ロボットからサーバへ)
- センサ情報の送信 (ロボットからサーバへ)
- ロボットの遠隔操作 (サーバからロボットへ)
- ライフログの送信 (ロボットからサーバへ)

一方、仮想空間側は、ペットシッターサービスをどのようにユーザに提供すべきかを考察し、Web サービスとして設計する。画面イメージを図 6 に示す。仮想空間は Web 技術者により構築されるが、本フレームワークを利用すると、ロボットの実体を意識せずに開発が可能になる。仮想空間によりユーザに提供される機能は下記のとおりである。

- 動物を見守る
- 動物と遊ぶ
- イベント通知
- 状態を知る (ライフログ)

これら仮想空間の機能を上記の実空間の機能 (ロボット側の機能) とマッピングすることにより、ユーザにサービスを提供する。マッピングのイメージを図 7 に示す。例えば、動物を見守る機能は、カメラ画像の送信、センサ情報の送信にマッピングされ、これらはそれぞれの機能を提供するロボットにマッピングされることになる。マッピングルールはフレームワークが提供し、サービス提供者はその詳細を知る必要はない。

以上をまとめると、ロボット技術者は RSNP を利用してロボットを開発し、ロボット依存のコードと RSNP の対応を記述する。Web 技術者は、仮想空間としてのインタフェースと提供サービスを設計し、それとフレームワークが提供する機能の対応を記述する。フレームワーク設計者は、機能と RSNP プロファイルのマッピングを記述することになる。

5.3 考察

インターネットを介してロボットを遠隔操作するシステムとしては、実用化されたものも含め、様々な提案が行われている。テレオペレーションと呼ばれる技術では、複数視点のカメラやセンサ情報を利用し、遠隔地のロボットを制御する様々な技術が開発されている。鹿島では、遠隔操作により建設機械 10 台を運用できる無人化施工システムを開発している [13]。重機 1 台に 5~6 台設置したカメラ映像を高精細かつ同時送信するための光ファイバケーブルを敷設し、映像と作動音を確認しながら遠隔操作を行う。リモート側に操縦席パネルを設置し、運転席に乗り込んでいるのと同様の操作性も実現している。また、テレグジステンスと呼ばれる一連の研究では、操作者にロボットの視覚や触覚などの情報を提示し、あたかも自身の体のように感じさせる遠隔臨場感を提供することで円滑な操作を実現している [14]。

いずれの研究も、遠隔地の状態（実空間の情報）をいかに忠実にネットワーク上に再現するかに注力しており、ペットシッターサービスに適用した場合は、前者であれば見守り対象の部屋に数多くのカメラやセンサを設置し、高精細な画像を監視しながら、見守りを実施する形態になるだろう。後者であれば、あたかも見守り対象の部屋にいるような状況を遠隔地に構築することになる。

用途によっては遠隔地の環境を高精度に再現することが必須の場合もあるが、本稿でのサービス例では、必ずしも忠実な再現である必要はない。むしろ、サービスを受けるユーザに適したインタフェース（仮想空間）でサービスが提供されることが望まれる。その結果、ロボット操作の専門家でなくても、また大がかりな設備がなくても、サービスの利用が可能になる。一方、ロボットサービスの開発者にとっては、ロボットが存在する実空間をそのままネットワーク上のサービスとして投影するのが自然であろう。フレームワーク上でこれら 2 つのモデルをマッピングすることにより、それぞれの技術者が容易にシステムを開発することが可能になると考えられる。

6. おわりに

本稿では、Jeeves フレームワークを拡張し、実空間と仮想空間をマッピングするアーキテクチャを提案した。また、サービスへの適用例を示すことにより、ロボットをユーザが直感的に利用できるサービスの構築が可能になることを示した。本稿ではモデル間のマッピングを静的に行なったが、現実空間ではバッテリー切れで使えなくなるロボットが存在するなど、マッピング自体を動的に行うことが要求される。今後、時間変動するパラメータを考慮した動的マッピング方式を検討していく。また、実際のサービスを構築し、フレームワークの検証を行っていく予定である。

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] Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W.: RT-Middleware: Distributed Component Middleware for RT (Robot Technology), *IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS 2005)*, pp. 3933-3938 (2005).
- [2] Quigley, M., Conley, K. and Gerkey, B.: ROS: an open-source Robot Operating System, *IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics 2009* (2009).
- [3] RoboEarth: <http://www.robearth.org/>.
- [4] Kuffner, J.: What's Next: Cloud Enabled Humans?, *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010) Workshop* (2010).
- [5] Arumugam, R., Enti, R., Bingbing, L., Xiaojun, W., Baskaran, K., king, F., Kumar, A., Meng, K. and KitVikas, G.: DAVinCi: A Cloud Computing Framework for Service Robots, *IEEE International Conference on Robotics 2010* (2010).
- [6] 成田雅彦, 村川賀彦, 植木美和, 中本啓之, 平野線治, 蔵田英之, 加藤由花: 普及期のロボットサービス基盤を目指す RSNP (Robot Service Network Protocol) 2.0 の開発, *日本ロボット学会誌*, Vol. 27, No. 8, pp. 857-867 (2009).
- [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 Service initiative: <http://robotsservices.org/>.
- [9] 齊藤由香利, 五十嵐登, 大山直人, 阪口和明, 角田龍太, 中川幸子, 中山央士, 清水将吾, 成田雅彦, 加藤由花: 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, *情報処理学会マルチメディア通信と分散処理研究会, DPS150*, pp. 1-6 (2012).
- [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, *信学技報, CNR2011-24*, pp. 1-6 (2011).
- [11] Nakagawa, S., Ohtama, N., Sakaguchi, K., Nakayama, H., Igarashi, N., Tsunoda, R., Shimizu, S., Narita, M. and Kato, Y.: A Distributed Service Framework for Integrating Robots with Internet Services, *the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA2012)*, pp. 31-37 (2012).
- [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).
- [13] 鹿島建設: 無人化施工システム, <http://www.kajima.co.jp/news/press/201204/18a1-j.htm>.
- [14] Tachi, S.: *Telexistence*, World Scientific Publishing Company (2009).