

ロボットサービスに必要なネットワーク知能の実現に向けた

アーキテクチャの提案

Proposal of architecture for realization of network intelligence required for robot service

村川 賀彦*1

Yoshihiko MURAKAWA

*1 株式会社富士通研究所

FUJITSU Laboratories LTD.

It is necessary to combine the use of the service of the intelligence processing on Cloud etc. and processing on the robot well to achieve the robot service. In this paper we propose the architecture to achieve this. Moreover, it is necessary to enable the development of various robot services in the spread of the robot service. It proposes the programming framework in this architecture that the web developer can easily work on the development of service for that.

1. はじめに

ロボットのソフトウェア技術は、新たな統合とビジネス化の時代へ入りつつある。従来、ロボットの共通プラットフォームの研究には、RT ミドルウェアプロジェクト[Ando 05], 次世代ロボット知能化技術開発プロジェクト[佐藤 10], ロボットサービスイニシアチブ (RSi: Robot Service initiative) [成田 08], ROS (Robot Operating System) [Queley 09]等の研究成果がある。また、要素技術としては、近年のブレイクスルー技術である特徴量を利用した画像や音声、SLAM (Simultaneous Localization and Mapping) 等の認識系計算技術の発展がある。環境地図作成、画像処理、アーム軌道計算など、ロボットへの適用が可能な要素技術のオープン化も進捗しつつあり、これらを統合することでロボットシステムを構築できるレベルとなった。一方、ロボットサービスとインターネットサービスの融合や、ロボットへのクラウド技術適用も模索されている。

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双方向通信可能な RSNP (Robot Service Network Protocol) の研究を進めてきた[成田 10]。RSNP は、2004 年に設立されたロボット開発関連の企業を中心とした業界団体である RSi (Robot Service initiative) により仕様化され[RSi], ライブラリ開発、高信頼通信、他のプラットフォームとの連携などが研究されている。RSNP 仕様に準拠したライブラリ実装としては、[FJLIB 10]があり、Java, JavaScript, HTML5 と組み合わせたロボットサービス開発を行うことができる。また、Objective-C で実装された iRSNP [成田 12b]もあり、相互運用を実現している。

ロボット技術は多くのブレイクスルー技術に基づいており、人工知能分野をはじめ、様々な分野の研究者やソフトウェア開発者が容易に利用できると、新たな研究や、広い産業領域への展開が期待できる。実際、人工知能分野では、オントロジーに基づくロボットを利用した先行研究例[小林 12]がある。しかしながら、ロボット工学の専門知識を持たないソフトウェアプログラマが、ロボットを用いたアプリケーションを開発することは、いまだ容易とはいえない。

そこで、我々は、RSNP を利用したロボットサービス開発(図 1 にそのプログラミングモデルを示す)におけるサーバ側およびロ

ボット側ソフトウェアを JavaScript 化することで Web 開発者でも容易にロボットサービスが開発できることを提案した[村川 14]。また、ロボットの非専門家である開発ユーザやプログラマのロボット技術の容易な利用を対象として、オープン化されつつあるロボット関連技術を RSi の技術上に構築することを目指し、クラウドベースのロボットサービス統合基盤が提案している[成田 13]。

本稿では、この2つの技術を融合することで、クラウド上の知能処理などのサービスの利用とロボット上での処理を上手く組み合わせることが可能となるアーキテクチャを提案する。次章以降では、まず、2 章と 3 章で、前提となるロボットサービスのためのプログラミングモデル RSNP を利用したロボットサービス開発の Javascript 化)とクラウドベースのロボットサービス統合基盤を構築するキーとなる技術であるマイクロサービスについて簡単に説明し、4 章で、それらの融合となるアーキテクチャを提案し、まとめる。

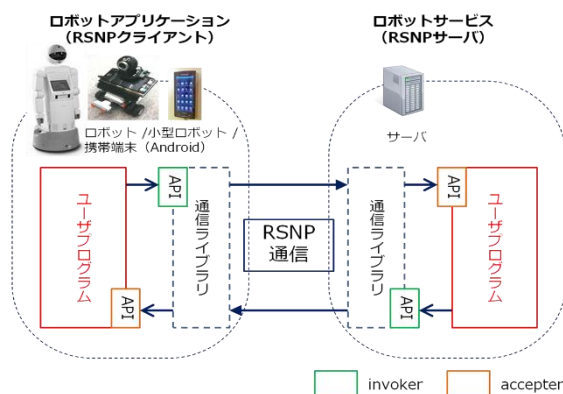


図 1 RSNP プログラミングモデル

2. ロボットサービスプログラミングモデル

RSNP のサーバ側実装を行ったものを RSNP サーバと呼ぶ。この RSNP サーバを利用してロボットへ指示を出す I/F を WebAPI 化することで、スマート端末上の HTML5 と JavaScript で書かれた Packaged web apps (通常サーバに配置されている Web アプリをパッケージという形で配布し、通常のアプリのように端末に常駐させる技術[PWA])から、ロボットへ指示が可能とな

る。この WebAPI 化が RSNP の Web サービス化(これを「RSNP サービス」と呼ぶ)となる。これにより、RSNP プログラミングモデルのサーバ側実装が JavaScript(JS)化されることになる。

また、サーバ側と対になるロボット側の実装についても統一されたフレームワークで作成できる環境を整えたいと考えた。そこで、このフレームワークについて、RSNP プログラミングモデルを基に、JS 向けの MVC (Model-View-Controller) フレームワーク (Backbone.js[Osmani 13]や Angular.js[Green 13]などのライブラリで提供されている)を参考に、実装を検討している。JS 向けの MVC フレームワークは、ブラウザでの表示と処理をモデル化したものであるため、ロボットにはそのまま当てはめることが難しいので、何を変更すれば良いかなど、プログラミング事例から抽出する必要がある。特に、MVC の View 部分がロボット側と端末側に分散されてあるため、それらを統一的に扱えるような Model と Controller も必要になる。その前提となる、ロボット側の実装についても JS 化が必要である。

サーバ側とロボット側をついで作成するプログラミングモデルの実現イメージを図 2 に示す。ロボットサービス作成者は、ロボットサービス向けのフレームワークを利用し、ロボットサービスをサーバ側とロボット側を対で作成する(①)。それを、まずは、シミュレーション環境で十分動作確認する。できれば、ロボット実機でも動作確認する。それを、ロボットサービスを配布する何らかの仕組み(例えば、アプリストアなど)の上に置く(②)。ロボットサービス利用者あるいは提供者(まとめて利用者とする)は、スマート端末にロボットサービスをダウンロード、あるいは、ロボットサービスが(例えば、場所に応じて)配布される(③)。利用者は、スマート端末でアプリを起動する(④)。すると、それに連動して、スマート端末から、利用するロボットにコードが送り込まれる(⑤)。提供されるロボットサービスにもよるが、複数台のロボットを利用するものなら、同時に複数台のロボットにコードが配布される。この時、利用ロボットの予約などの仕組みが必要になる。これにより、利用者は、スマート端末でロボットを使ったロボットサービスが簡単に利用できる。

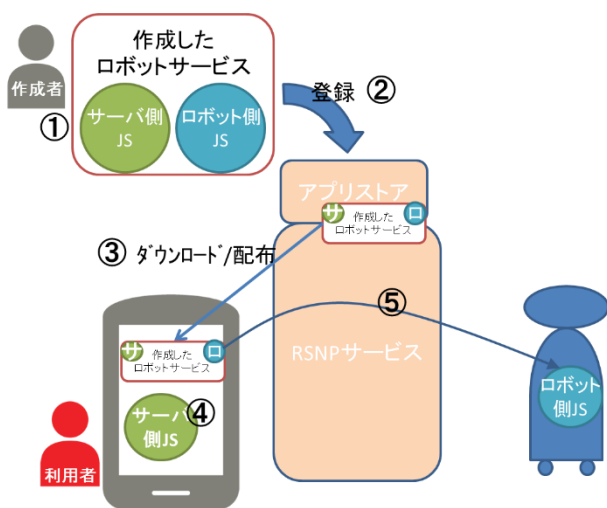


図 2 ロボットサービスプログラミングモデル

3. マイクロサービス

RSi では、2011 年 6 月に Data Push profile 機能の仕様を組み込んだ RSNP 2.3 を策定している。Web サービスでは、通常、

ロボット側(クライアント)からサービス側(サーバ)へ接続するが、Data Push profile は、サービス側からのトリガでデータをロボットに push するための仕組みである。RSNP は、Push 機能を利用することで構内 LAN やファイアウォール内に配備されたロボットにインターネット経由でデータを送付する仕組みを持つ。この仕組みをサービス部品に拡張すると、クラウド上に機能を配備することなく、インターネット経由で共通の機能を利用する仕組みが実現できる。我々はこの仕組みをマイクロサービスと呼んでいる[加藤 13]。これを用いることで、ロボットサービスのクラスタリング構成が可能になり、スケーラブルな性能向上と可用性が実現する。マイクロサービスの機構を実現するために、マイクロサービスを呼び出すロボット、プラットフォーム上で要求を受け付けるモジュール (MF: Micro-service Front)、実際にサービスを提供するマイクロサービス本体 (MB: Micro-service Body) との間の処理の流れを設計している。処理のイメージを図 3 に示す。この例では、サービスロボットとして配置された人を検知する機能を、監視ロボットから呼び出して利用する構成を想定している。これにより、負荷分散が可能となり、ロボットの接続数や負荷に応じて処理を動的に分散することができる。

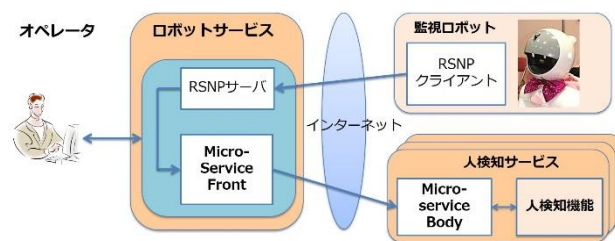


図 3 マイクロサービスの概要

4. アーキテクチャの提案

ロボットサービスプログラミングモデルとマイクロサービスを融合することで、以下に示す特徴を持つアーキテクチャが考えられる。

- ロボットサービス作成者は、知能化機能を新たに作成する必要がなく、クラウド上に配置された機能を利用する部分を作成するだけでよい。
- 知能化機能利用部分も含め、サーバ側とロボット側の実装と3対で作成するプログラミングモデルとなり、作成者は3対の作成用フレームワークを利用することで、それをあたたかも一体で作成できる。
- 3対のプログラムは、ロボットの利用者が一括してスマホにダウンロードし、利用するロボットと知能化機能を提供するマイクロサービスにそれぞれスマホからアプリを転送することで一体のロボットサービスとして機能させることができる。

提案するアーキテクチャを図 4 に示す例で説明する。ロボットサービス作成者は、ロボットサービス作成用フレームワークを利用し、サーバ側、ロボット側、マイクロサービス側のユーザ実装を3対一体で作成する(①)。作成したロボットサービス用のプログラムをシミュレーション環境で十分検証し、出来れば、実際にロボットでの動作確認を行う。そして、アプリストアなどのようなロボットサービスをスマート端末に配布する仕組みに登録する(②)。ロボットサービスの利用者、あるいは、ロボットを利用してサービスを提供する提供者(まとめて利用者とする)は、スマート端末にロボットサービスを3対まとめてダウンロード、あるいは、利用者のいる場所に応じて配布される(③)。利用者は、スマー

ト端末でアプリ(サーバ側実装となるもの)を起動する(④).すると、それに連動して、スマート端末から、利用するロボットにコードが送られる(⑤).同時に、ロボットが利用する知能化機能を使うためのコードが知能化機能を提供するサーバに送られる(⑥).提供されるロボットサービスにもよるが、複数台のロボットを利用するものなら、同時に複数台のロボットにコードが、複数の知能化機能を利用するなら、同時に複数の知能化機能サーバにコードが配布される.この時、利用ロボットの予約などの仕組みが必要になる(RSNP のタスクプロファイルで提供されている仕組みを利用することが可能).これにより、利用者は、スマート端末でロボットを使ったロボットサービスが簡単に利用できる.

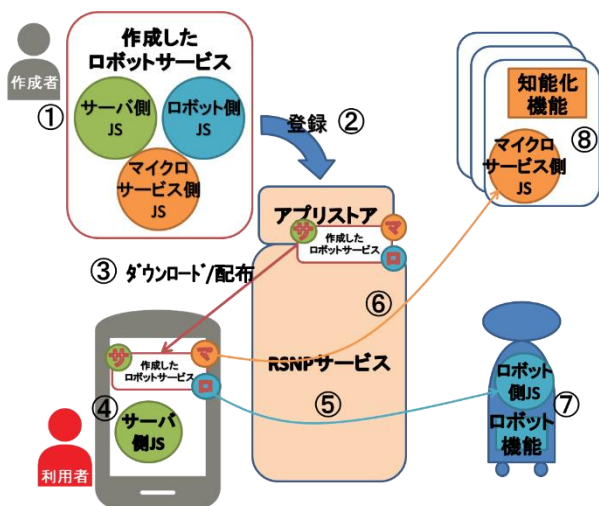


図 4 アーキテクチャの概要

5. まとめ

本稿では、ロボットサービスの作成を容易にするために、RSNP を利用したロボットサービス開発のプログラミングモデルとマイクロサービスを融合したアーキテクチャを提案し、その特徴を述べた. 今後は、提案したアーキテクチャに従い、クラウド上の知能化機能を容易に利用することが可能となる、新たなロボットサービスプログラミングモデルの構築を行っていく予定である.

参考文献

- [Ando 05] Noriaki Ando, Takashi Suehiro, Kosei Kitagaki, Tetsuo Kotoku and Woo-Keun Yoon: "RT-Middleware: Distributed Component Middleware for RT (Robot Technology)", Proc. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005), pp. 3555-3560, 2005.
- [佐藤 10] 佐藤知正, 岡野克弥: "RT ミドルウェアと知能モジュール構築プロジェクト", 日本ロボット学会誌, Vol.28, No.5, pp.546-549, 2010.
- [成田 08] 成田雅彦, 島村真巳子, 日浦亮太, 山口亨: "ロボットサービスイニシアチブ (RSi) の活動を通して実現したロボットサービス共通プラットフォーム仕様", 日本ロボット学会誌, Vol.26, No.7, pp.785-793, 2008.
- [Queley 09] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng: "ROS: an open-source Robot Operating System"

- Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA2009) Workshop on Open Source Robotics, 2009.
- [成田 10] 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.
- [RSi] RSi-Robot Service initiative: <http://robotsservices.org/> [Online]
- [FJLIB 10] ロボットサービスイニシアチブ, Robot Service Network Protocol 2.3 仕様書 1.0 版, 2010.
- [成田 12b] 成田雅彦, 加藤由花, 土屋陽介: "小型ロボットを利用した簡易型 RSNP ロボットサービス開発環境の構築", 産業技術大学院大学紀要, Vol.6, pp.53-59, 2012.
- [小林 12] 小林昭太郎, 樋川暁, 山口高平: "対話と動作の連携を目指したオントロジーに基づく知能ロボット", 第 26 回人工知能学会全国大会, 3K1-R-11-7, 2012.
- [成田 12a] 成田雅彦, 加藤由花, 中川幸子, 小川紘一: "ロボット技術のオープンイノベーション - Robot-OS(ROS)の戦略と我が国ロボット技術開発の方向性 -", IAM DPS #28 東京大学知的資産経営総括寄付講座, 2012.
- [村川 14] 村川賀彦: "新たな RSNP プログラミングモデルの提案", 第 32 回日本ロボット学会学術講演会, 1G1-06, 2014
- [成田 13] 成田雅彦, 他: "クラウドベースのロボットサービスの統合基盤", 第 27 回人工知能学会全国大会, 3A1-NFC-03-1, 2013.
- [PWA] "Packaged Web Apps (Widgets)", <http://www.w3.org/TR/widgets/>
- [Osmani 13] Addy Osmani: "Developing Backbone.js Applications", O'REILLY, 2013.
- [Green 13] Brad Green, et al.: "AngularJS", O'REILLY, 2013.
- [加藤 13] 加藤由花, 岡部泉, 村川賀彦, 岡林桂樹, 植木美和, 土屋陽介, 成田雅彦: "インターネットサービスプラットフォームにおけるロボットサービス提供手法", 情報処理学会論文誌, Vol.54, No.2, pp.1234-1246, 2013.