

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

阪口和明[†] 中川幸子[†] 大山直人[†] 中山央士[†]
五十嵐登[†] 角田龍太[†] 清水將吾[†] 成田雅彦[†] 加藤由花[†]

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。ただし、ロボットサービスを対象とした場合、デバイスの種類と数が増え、様々な分野の開発者が参入できる仕組みの構築が必要であり、既存のプラットフォームでは十分でない。本稿では、ロボットサービス向けプロトコル仕様である RSNP (Robot Service Network Protocol) 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提案する。さらに、提案するフレームワークを利用したプロトタイプシステムとしてペットシッターサービスを実装し、フレームワークの有効性を検証する。

A Distributed Service Framework for Integrating Robots and Internet Services

Kazuaki Sakaguchi[†], Sachiko Nakagawa[†], Naoto Ohyama[†],
Hisashi Nakayama[†], Noboru Igarashi[†], Ryota Tsunoda[†],
Shogo Shimizu[†], Masahiko Narita[†] and Yuka Kato[†]

With the rapid advance of the integration of internet and robot areas, various service platforms are proposed that assume cloud environments. For robot services, as the numbers of devices and their types increase, a mechanism that enables the entry of developers in various areas is required. However, existing platforms are insufficient to the problem. In this paper, we propose a distributed service framework that realizes the coordination of various devices, robots, and service functions based on RSNP (Robot Service Network Protocol), a protocol specification for robot services. Furthermore, we implement a pet-sitting service as a prototype system that uses a proposed framework and verify the effectiveness of the framework.

1. はじめに

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の提案が行われている。例えば、ネットワークロボット分野における RoboEarth[1]や Cloud-Robotics[2][3]、センサネットワーク分野における Sensor-Cloud[4]や X-Sensor[5]などである。ただし、ロボットサービスを考察対象とした場合には、プラットフォームにつながるデバイスの種類と数が増え、様々な分野の開発者が参入できる仕組みの構築が必要であり、既存のプラットフォームではこれらの点への考察が不足していた。

このような背景のもと、本稿では、ロボットサービスイニシアチブ (Robot Service initiative: RSi) [6]によって策定されたオープンなロボットサービス向けプロトコル仕様である RSNP (Robot Service Network Protocol) [7]を利用し、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を提案する。フレームワークの提案にあたり、解決すべき課題は下記の3点である。

- (1) 多種多様なロボットやロボット以外のデバイス、サービス機能を相互に接続する仕組みを実現すること (インターオペラビリティ)
- (2) クラウド環境への適用を前提とした統合プラットフォームとしての仕組みを実現すること (インターネットとの親和性)
- (3) ロボットらしいサービスを提供できること (ロボットらしさ)

本稿では、(1) および (2) の課題を解決するために、様々なデバイス、ロボット、サービス機能 (音声認識機能などのサービス部品のこと。以降、マイクロサービスと呼ぶ) の連携を実現し、タスク分散が可能なアーキテクチャを設計する。(3) の課題を解決するためには、ロボットとクラウド間の通信に RSNP を利用し、ロボット研究者の定義するロボットらしいサービスを実現する。ここでロボットらしいサービスとは、(i) 画像やセンサ情報をロボットサービスへ連続的に転送でき、種々のロボットプラットフォームとの連携ができること、(ii) ロボットの自律運転中に割り込み、簡単な組み合わせ動作を行わせる遠隔割込操作ができることと定義する。これは RSi のロボットサービスモデルにおける定義である[8]。

本稿ではさらに、提案するフレームワークを利用したプロトタイプシステムとして、ペットシッターサービスを実装した結果を報告する。ここでは、ペット見守りロボットを用いたペットホテルにおける統合的なサービスの実装結果から、フレームワークの有効性を検証する。

[†] 産業技術大学院大学 産業技術研究科
School of Industrial Technology, Advanced Institute of Industrial Technology

2. Jeeves フレームワークの提案

2.1 設計指針

フレームワークの設計にあたっては、1章で述べた課題を解決するために、以下の設計指針を策定した。

- RSiのロボットサービスモデルを適用する。これにより、クラウド環境での利用、多種多様なロボットやロボット以外のデバイス、サービス機能を相互に接続できるフレームワークを実現する。
- 通信プロトコルとして RSNP を利用する。これにより、ロボットらしいサービスの提供を実現する。さらに、RSNPはWebサービス基盤を利用していることから、ロボット分野の専門知識を持たないソフトウェア開発者の取り込みが期待できる。そのため、ロボットサービスを容易に開発できるフレームワークの実現を目指す。

以下、RSiのロボットサービスモデルとRSNPについて詳述する。

2.2 RSNP

2.2.1 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、もしくは物理的サービスと定義している[9]。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る。また、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。ロボットをインターネットに接続する利点は、人と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である。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実現できる。

2.2.2 システムアーキテクチャ

RSNPの特徴を下記に示す。

- インターネットとの整合性が高く、標準化された豊富な機能の上にサービスを構築でき、多様なロボットと各種サービスに適用できる。さらに、複数実装間でも相互接続が可能である。
- 計算機資源の限られているロボットへ、情報提供や指示が迅速に行える。
- ロボットらしいサービスを提供できる。具体的には、画像やセンサ情報をロボットサービスへ連続的に転送でき、種々のロボットプラットフォームとの連携が実現できる。

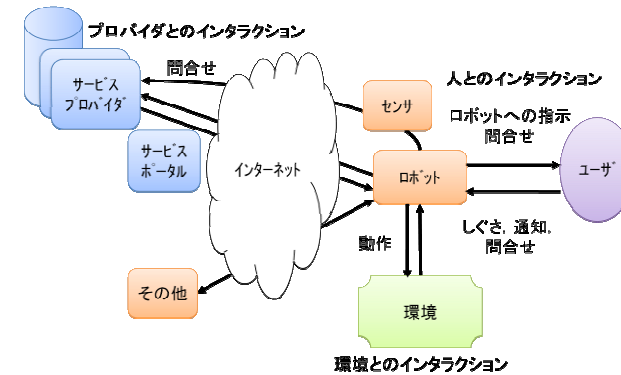


図1 RSiのロボットサービスモデル
Figure 1 The robot service model on RSi



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

RSNP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[10]。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、インターネットと整合性の高い標準化された機能を利用可能である。RSNPのシステムアーキテクチャを図2に示す。

プロトコル自身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、基本プロファイルとして、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提供し、応用プロファイルとして、情報サービス・天気サービス・防災情報サービス・見守りサービス・リモート制御などのサービスを提供する。このように、ロボットの機能とロボットサービスの機能をモジュール化して提供することで、機能の異なる多様なロボットや各種サービスへの適用を可能としている。

2.3 フレームワークの提案

2.1 節で述べた設計指針に従い、フレームワークの設計を行う。本稿では、1 番目の設計指針の実現形態として、Publisher/Subscriber モデルを採用することにした。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、結合度の低いアーキテクチャが要求されるためである。Subscriber からの接続には RSNP を利用することにより、2 番目の設計指針を実現する。我々はこのフレームワークを「Jeeves*フレームワーク」と名付けた。フレームワークの構成を図 3 に示す。

Jeeves フレームワークは、動的かつ柔軟な分散リソースの発見と参加を可能とし、ロボットをはじめとする多種多様なリソースの連携を実現する。このフレームワークを利用することで、インターネット分野とロボット分野、及びロボット以外のデバイス分野の技術者が個別にサービス部品を開発することが可能になり、その結果、複合的なロボットサービスの提供が期待できる。

2.4 フレームワークの機能

Jeeves フレームワークは、エンドユーザのロボット等（ロボット、各種デバイス、マイクロサービス等）への作業依頼を効率的に処理するために以下の機能を持つ。

- エンドユーザの作業依頼をクラウド上にキューイングする機能
- ロボットが自ら作業内容の可否を判断し、実行可能な場合は作業キューを取得する機能

なお、今回の実装では、ロボットからの状態通知の到着順に作業を割り当てている。サーバ側で、稼働率の低いロボットや作業実行に最適なロボットを選択し、作業を割り当てる処理は行っていない。Jeeves フレームワークの処理フローを次に示す。

- (1) 作業を行うロボットもしくはマイクロサービスをサーバに登録する。実装を予定しているサービスを提供する能力を、ひとつでも備えていると登録可能。
- (2) ロボットに作業を依頼したいユーザが、受けたサービスサーバに伝える。
- (3) ロボットもしくはマイクロサービスは、自らの状態が「空き」か「実行中」であるかを保持しており、状態が「空き」かつ作業内容が実行可能なものであれば、サーバに対して作業を要求する。
- (4) サーバがロボットに対して、作業を先着順に割り振る。
- (5) ユーザに対しては、割り当てられたロボットが通知される。遠隔操作により提供されるサービスの場合、ユーザは直接ロボットに指示を送ることができる状態となる。
- (6) キューに蓄積された作業がなくなるまで、(3) (4) (5) の処理を繰り返す。

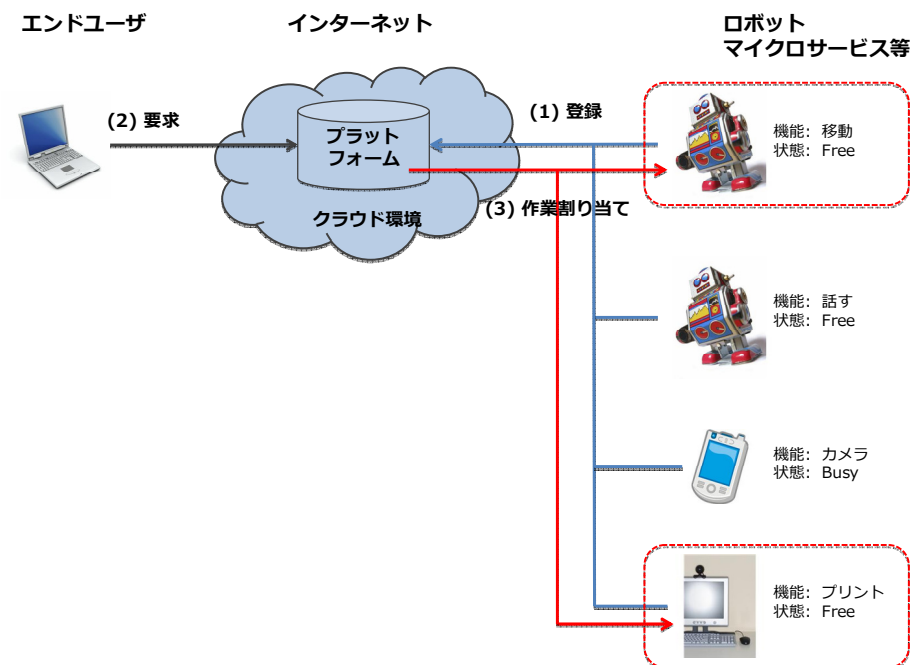


図 3 フレームワークの構成

Figure 3 System architecture of the framework

* フレームワーク名の由来

イギリスで有名な執事の名前である。執事とは、一般に事務を管轄する者を意味し、高位の人物の家や寺社で家政・事務を執りしめる者を指す。サービスロボットを「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。

3. プロトタイプシステムの実装

Jeeves フレームワークを利用したプロトタイプシステムとして、簡易版ペットシッターサービスを構築した。本章ではその結果を報告する。

3.1 実装したサービスの概要

プロトタイプシステムとしては、ロボットや各種デバイス、マイクロサービス等が連携したサービスを実装する必要がある。そのため本稿では、ペットホテル等でペットの世話をするロボットと監視サービスを組み合わせた、ペットシッターサービスを対象とすることとした。サービスのイメージを図4に示す。ここでは、動物病院やペットホテル等の施設での利用を想定し、カメラ撮影による監視をはじめ、ペットの世話を担当するロボット等が構成要素となる。さらに、ロボットが取得したペットに関する情報はクラウド上にアップロードされ、ペットを預っている施設の職員や飼い主は、Webブラウザで作業の要求と作業結果の確認を行う。

3.2 システム構成

構築したプロトタイプシステムの構成を図5に示す。ここでは、サービスアプリケーション（RSNPサーバ）とデータベースを学内に設置したサーバ上に構築した。サービスアプリケーションは、Linuxサーバ上に常駐するJavaアプリケーションであり、ユーザに対する認証、操作画面の提供、ユーザから要求された作業のデータベースへの保存を行う。また、ロボットに対しては、認証、作業の割り当て、画像・ライフログ情報のデータベースへの保存を行う。データベースはMySQLで実装し、ユーザ認証用、ロボット認証用、依頼可能な作業一覧、作業と作業中のロボットの関連付けの4種類のテーブルを持つ。なお、ロボットの認証は本来第三者機関を通じて行うことが推奨されているが、プロトタイプシステムではデータベース内にテーブルを作成して行っている。

RSNPクライアント端末としては、LEGO MINDSTORMS[11]にロボットを制御する役割を担うWindows PCを接続したものを使用した。LEGO MINDSTORMSとWindows PCはBluetoothで接続し、カメラ撮影を行うためのWebカメラも接続した。さらに、プロトタイプシステムでは、ペットの状態を監視するロボットサービスと、メール機能との連携を実現した。マイクロサービスとしてメール送信機能を実装し、異常を検知した際にユーザにメールが送信されるようになっている。ユーザによるサービスの利用イメージを画面遷移図として図6に示す。

なお、ペットシッターサービスの機能と、利用したRSNPプロファイルの一覧を表1にまとめた。ここでプロトタイプシステムの構築には、RSNPの一実装であるFJLIB（富士通製のJavaライブラリ）を利用した。

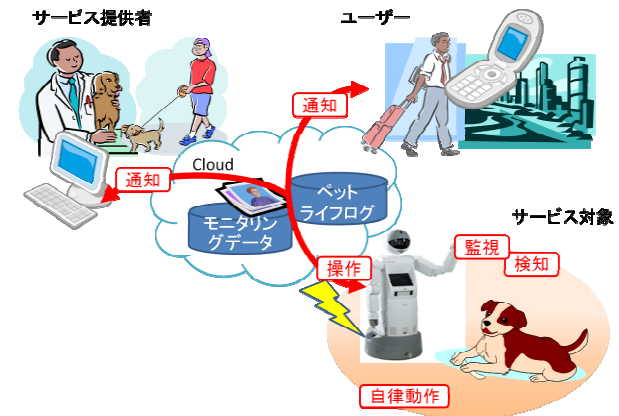


図4 ペットシッターサービスのイメージ
Figure 4 An image of the pet-sitting service

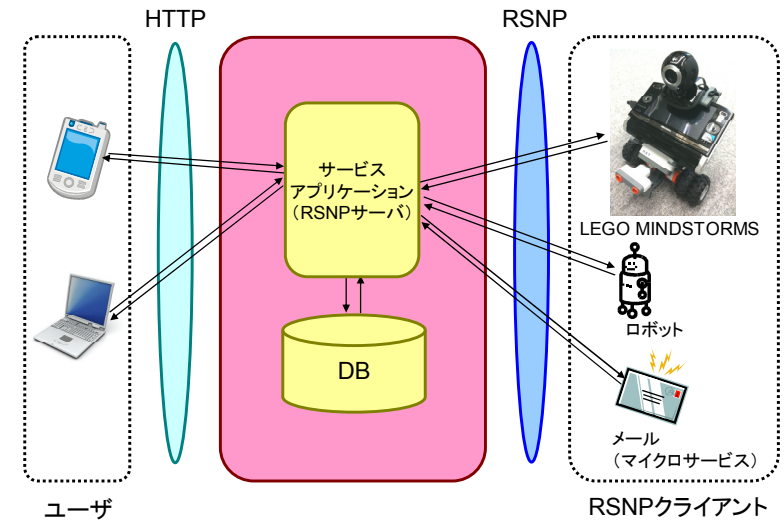


図5 プロトタイプのシステム構成
Figure 5 System architecture of the prototype system

表 1 ペットシッターサービスの機能
Table 1 System function of the pet-sitting service

機能(プロトタイプへの実装)	機能説明	RSNP プロファイル	RSNP プロファイルの説明
カメラ撮影 (実装済み)	ほぼリアルタイムの映像の配信, 再生	Multimedia_profile	ロボットのカメラやマイクなどのマルチメディア機器を操作し, データを扱う
ライフログ&異常検知 (実装済み)	<ul style="list-style-type: none"> センサを利用したロボットからの連続した情報の配信 収集した情報からの異常検知 マイクロサービスを使ったメール送信 	Contents_profile	サーバ・ロボット間で相手の提供するコンテンツを取得する
		Data_push_profile	サーバ・ロボット間で自身の持つデータを相手先に送信する
ペットと遊ぶ (実装済み)	<ul style="list-style-type: none"> ほぼリアルタイムの映像の配信, 再生 ロボット操作 (ペットへの接近, 動作によるペットの注意喚起) 	Motion_profile	ロボットに動作の処理を行わせる
		Multimedia_profile	ロボットのカメラやマイクなどのマルチメディア機器を操作し, データを扱う
ペット視点のカメラ (未実装)	ペットに搭載したカメラによるほぼリアルタイムの映像の配信・再生	Multimedia_profile	ロボットのカメラやマイクなどのマルチメディア機器を操作し, マルチメディアデータを扱う
給餌 (未実装)	給餌の実施	Command_profile	簡単な組み合わせの動作をロボットに行わせる
清掃 (未実装)	清掃の実施	Command_profile	簡単な組み合わせの動作をロボットに行わせる

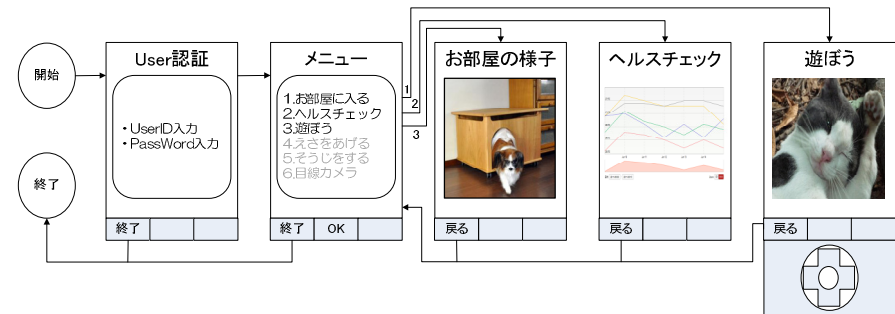


図 6 画面遷移のイメージ
Figure 6 Service images on a user client terminal

4. 考察

本章では, 前章までに述べたプロトタイプシステムの実装から, 1章で取り上げた課題(1)~(3)に対する検証をおこなう. また, Jeeves フレームワークと, 他のプラットフォームとを比較し, その有用性を示す.

(1)の課題には, ライフログ及び異常検知機能を実装することで, ロボットとマイクロサービスの連携を実現した. Jeeves フレームワークでは, RSNP を利用しているため, ロボット機能とロボットサービス機能がモジュール化されて提供されている. したがって, 多様なロボットから共通に利用可能なロボットサービスを開発することができた. (2)に対しては, ペットと遊ぶ機能を実装することで, ロボットの遠隔操作を実現した. 遠隔操作されるロボットは, セキュリティを考慮し, Firewall 環境や NAT 環境下に設置されることが想定される. インターネット側からの要求を常時受け付けるためには, Firewall 上でのポート解放などの作業や, 外部からのアタックに対して, 常時ネットワーク監視が必要になる等, ロボットサービス開発者にとっては困難である. Jeeves フレームワークでは, RSNP の擬似 Push 通信[10]を利用するため, インターネット側からのロボット制御が容易に実現可能である. また, ライフログ及び異常検知機能の実装により, クラウド上へ大量の情報の蓄積とデータの分析を実施することが可能になった. ロボット機能とロボットサービス機能がモジュール化されていることと, オープンな仕様である RSNP を利用することで, サーバへの大量なデータの蓄積と分析, マイクロサービスへの処理の分散を行うことができた. (3)に対しては, ペットと遊ぶ機能とカメラ撮影機能を実装し, ロボットらしさを実現した. RSNP では, ロボットらしいサービスを提供するための各種プロファイル群が用意されているため, 容易な実装が可能になった. 以上のように, 今回のプロトタイプシス

表2 各プラットフォームの比較結果
Table 2 A comparison result among three types of platforms

課題	Cloud-Robotics	Sensor-Cloud	Jeeves
多種多様なロボットやロボット以外のデバイス、サービス機能を相互に接続する仕組み	△ (ロボットやデバイスの専門家以外は難しい)	△ (ロボットサービス開発者以外は難しい)	○ (共通利用可能な開発環境の提供)
クラウド環境への適用を前提とした統合プラットフォームとしての仕組み	○ (独自実装)	△ (独自実装)	○ (標準仕様)
ロボットらしいサービスの提供	△ (ロボットやデバイスの開発者以外は難しい)	△ (ロボットサービス開発者以外は難しい)	○ (プロファイルの利用により、容易)

テムの実装により、課題(1)～(3)をみたくロボットサービスの開発が可能になることを実証した。なお、これらの開発は、ロボットを専門としない情報システム学専攻の学生が、RSNP2.3の調査からフレームワークの考案、プロトタイプの実装までを3ヶ月程度で実施しており、開発の容易性を検証できたと考えられる。

次に、他のプラットフォームとの比較を行う。ネットワークロボット分野におけるクラウドプラットフォームとしてCloud-Roboticsを、センサネットワーク分野におけるクラウドプラットフォームとしてSensor-Cloudを取り上げ、(1)～(3)の課題に対してJeevesフレームワークと比較した結果を表2に示す。Cloud-Roboticsは、これまでロボットサイドに実装されてい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。しかし、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た機能を、クラウド上で同様に実装する必要がある。Sensor-Cloudは、センサから収集される大量のデータをクラウド上で処理するモデルとなっており、収集されたデータを利用したアプリケーションは、すべてクラウド上に実装する必要があるため、ロボットサービスの開発者以外の参入は困難である。これらのプラットフォームと比較すると、Jeevesフレームワークは、RSNPがロボットサービスの標準仕様を定めているので、Webシステム開発者であってもロボットサービスを容易に開発することができる。

5. おわりに

本稿では、RSNPを利用した分散処理機構として「Jeevesフレームワーク」を提案し、そのプロトタイプシステムとして簡易型ペットシッターサービスを実装した結果

を報告した。提案したフレームワークの特徴は、多種多様なロボットの制御、ロボットらしさ、インターネットサービスの有効活用を実現している点にあり、プロトタイプシステムの構築を通してその有効性を検証した。

今後の課題としては、Jeevesフレームワークの機能追加、ペットシッターサービスの機能拡張、Jeevesフレームワークを利用した新たなサービスの開発等がある。また、RSNPでは開発したコンポーネントの再利用が可能であるため、今回利用したLEGO MINDSTORMS以外の他のロボット(Roomba[12], enon[13]等)への実装も行っていく予定である。

謝辞 本稿の執筆にあたり、貴重なご意見を頂きました(株)富士通研究所の神田真司様、村川賀彦様、岡林桂樹様に感謝いたします。

参考文献

- 1) RoboEarth. : <http://www.robearth.org/>.
- 2) Kuffner James. Cloud-Enabled Humanoid Robots. : Humanoids2010 Workshop “What's Next”, 2010年.
- 3) Rajesh Arumugam, Vikas Reddy Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, Foong Foo Kong, A.Senthil Kumar, Kang Dee Meng, and Goh Wai Kit. DAVINCI: A Cloud Computing Framework for Service Robots. :2010 IEEE International Conference on Robotics and Automation Anchorage Convention District. 2010.
- 4) Madoka Yuriyama, Takayuki Kushida. Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing. : 13th International Conference on Network-Based information Systems (NBIS2010),pp.1-8,, 2010.
- 5) 原隆浩, 神崎映光, 中山浩太郎, 義久智樹, 寺西裕一, 若宮直紀, 下條真司. X-Sensor : 大規模実証実験を可能とするセンサネットワークテストベッド. : 情報処理, 49(8), 950-955, 2008.
- 6) RSi - Robot Service initiative. 2011. <http://robotsservices.org/>.
- 7) ロボットサービスイニシアチブ. Robot Service Network Protocol 2.3 仕様書. 2010.
- 8) 成田雅彦, 村川賀彦. ロボット技術の標準化とRSi (Robot Service Initiative) の取り組み. 日本ロボット学会誌 Vol.29 No.4 pp353～356, 2011.
- 9) 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
- 10) 成田雅彦, 村川賀彦, 植木美和, 中本啓之, 日浦亮太, 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.
- 11) LEGO MIND STORMS. : <http://www.legoeducation.jp/mindstorms/>.
- 12) iRobot 自動掃除機ルンバ. : <http://www.irobot-jp.com/>.
- 13) 富士通サービスロボット「enon (エノン)」. : <http://www.frontech.fujitsu.com/services/products/robot/servicerobot/>.