

Honda RaaS Platform

クラウドとロボティクスを融合させるプラットフォーム – Robotics as a Service Platform

Honda Robotics as a Service Platform Integrates Cloud and Robot Technology

東 治 臣¹

Haruomi HIGASHI

要 旨

Honda RaaS Platformとはロボティクスサービスを開発・運用するためのプラットフォームである。

ロボティクスサービス構築を行うための開発や運用の観点から、過去の ASIMO によるデリバリーサービスの実証実験より、ネットワーク、実験環境構築、開発環境、運用環境、サービス構築・拡張に関する検討すべき改善事項を得た。

Honda RaaS Platform は各改善事項に対して、SIM による稼働エリア拡大およびセキュアな通信環境、クラウドを利用したスケーラブルなインフラ環境、ネットワーク経由でアクセス可能な開発・デバッグ環境、リモート監視可能な運用環境、API 提供によるサービス構築・拡張環境を提供することで、ロボティクスサービスの開発や運用を改善できる。

また、Honda RaaS Platform のアーキテクチャやインタフェースを示すと共に、複数の異なるロボティクスデバイスを連携・協調させる一連のデリバリーサービスの事例により、コンセプトである「異能を繋ぐのりしろ」を実証した。

ABSTRACT

Honda Robotics as a Service Platform is a platform for the purpose of developing and operating robotics services.

In terms of development and operation for the purpose of establishing such services, past verification tests of delivery service using ASIMO identified issues for further work that required investigation relating to the network, the creation of testing environments, the development and operational environments, and the establishment and expansion of service.

Regarding each of the above issues, the Honda RaaS Platform can enhance the development and operation of robotics services by providing for a secure communication environment and expansion of the operating area by means of SIM, a scalable infrastructure environment using the Cloud, a development and debugging environment that is network accessible, an operational environment that enables remote monitoring, and an environment for establishment and expansion of service by API provision.

In addition to showing the architecture and interface of the Honda RaaS Platform, the concept of “collaboration of diverse talents all over the world” was also proven by case examples of a series of delivery services that create collaboration and coordination by multiple different robotics devices.

1. はじめに

Honda は創業 100 年を超える 2050 年を見据えて、2017 年に「2030 年ビジョン」を策定した⁽¹⁾。ここでは「すべての人に、『生活の可能性が広がる喜び』を提供するー世界中の一人ひとりの『移動』と『暮らし』の進化をリードするー」というステートメントが示されている。

¹ ライフクリエーションセンター

その具現化の手段のひとつとして、従来の Honda が得意とする多岐に渡るモノづくりに対し、ハードウェアだけではなく、ソフトウェアによる体験や経験といったコトづくりを含むソリューション創出力の強化を図るテーマとして、2017 年に Honda RaaS Platform が提案された。その担う役割は、Honda 社内の様々なアルゴリズムやアプリケーションに対する先進研究開発の成果を再利用可能な形

で集約し、すべての人に対して役立つようにまとめあげることで、Honda の中でソフトウェアの価値と重要性を高めることである。

2. Robotics Vision

Honda のロボット開発の歴史は 1986 年に 2 足歩行の研究開発からスタートした (Fig. 1)。ASIMO に代表されるヒューマノイドロボット研究から生まれるロボティクス技術、および応用製品の総称が Honda Robotics である。また、ASIMO の開発ではそれと並行して「人と共存、協調して社会の中で役に立つロボット」の実現に向けて、受付案内やデリバリーサービスの自動化のための統合制御システムを開発してきた^{(2), (3)}。

2018 年、Honda は新たなロボティクスのビジョンとして「人の素晴らしさが際立つロボティクス社会」の実現を目指すことを、3E コンセプト「Empower・Experience・Empathy」とともに発表した (Fig. 2)⁽⁴⁾。

3E コンセプトの Empower とは、人とロボティクスデバイスがそれぞれの得意領域を活かして協働することで人に時間的・精神的・肉体的な余裕が生まれ、その余裕を新たな方向に振り向けることで、人は自らの可能性を拡大してい



Fig. 1 Honda Robotics history



Fig. 2 3E concept

くことができる、という考えである。Experience とは、人がロボティクスを通して全く新しい体験を行い成長し、ロボティクスデバイスも人から学び、より人のためになるように自らを成長させていく、という考えである。Empathy とは、ロボティクスデバイスが人を思いやり共感して、人が行動するためのサポートを行う、という考え方である。

このような、3E コンセプトを実現するためには「人と協調する人工知能」が必要不可欠であると考えられる。

3. Robotics as a Service - RaaS

一般的に Robotics as a Service (RaaS) とは「ロボティクスの機能をクラウド経由で提供すること」とされる。これは、高負荷処理をクラウド環境に肩代わりさせることでロボット側の処理負荷を軽減する（オフロードできる）ことができる、という考え方である。それに加え、Honda では RaaS を用いると主に下記の三つのメリットがあると考えた。

- (1) ロボットの共通管理による開発・運用の効率化
- (2) ロボットの稼働状況に応じた柔軟なコスト管理
- (3) 新機能の継続的なアップデート

(1) では、ロボットの管理や運用を行うための基本機能を共通化して再利用することで、ロボットサービスの開発や運用の効率を向上させる。

(2) では、例えばロボットを利用した分だけコスト計算を行う従量課金などを可能にする。

(3) では、ロボットやクラウド上のソフトウェアを継続的に更新することで、最新の状態にする。これにより、顧客のニーズや要望に対してロボットサービスを柔軟に対応させることができる。

4. Honda RaaS Platform Concept

Robotics Vision を実現するためには、人を中心とした複数の異なるロボットの協調と連携が必要不可欠であり、Honda だけではなく様々な開発パートナーのロボットやシステムをシームレスにつなぐことが重要である。Honda は CES 2019 にて、Robotics Vision である「人の素晴らしさが際立つロボティクス社会」と 3E コンセプトおよび Cooperative Intelligence (CI) を実現するためのベースシステムとして、Honda RaaS Platform Concept を発表した⁽⁵⁾。

Honda RaaS Platform とは、ロボティクスサービスを開発・運用を容易に行うためのプラットフォームである。これは、データ蓄積・共有、通信制御、状態遷移管理、ロボット間連携などの共通機能を、インタフェースである Application Programming Interface (API) やソフトウェア開発キットである Software Development Kit (SDK) として提供する (Fig. 3)。

さらに、Honda RaaS Platform は「異能を繋ぐのりしろ」というコンセプトを掲げている。これは様々な役割や機能の異なるロボットやシステムをつなぎ、相互に協調させることによって、単一のロボットでは不可能なサービスの実現を目指すコンセプトである。すなわち、様々なロボットメーカーやサービスインテグレータなどと共に、プラットフォームを育てながら、ロボティクスサービスを多くの人が量産できる仕組みを構築することが目指すべきゴールとなる⁽⁵⁾⁻⁽⁸⁾。

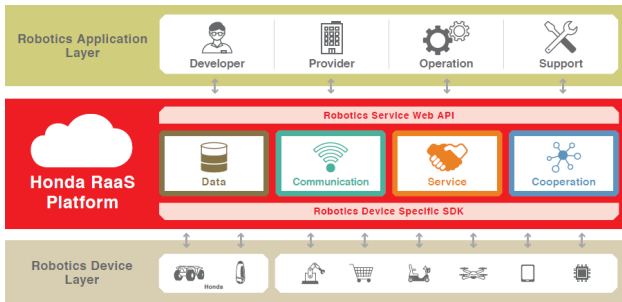


Fig. 3 Honda RaaS Platform conceptual diagram

5. ロボティクスサービス構築実験

Honda RaaS Platform を用いて実現するロボティクスサービスを検討する上で、様々な知見と気づきを得るために、過去に実施したデリバリーサービスの実証実験から改善事項の抽出を行った。具体的には、複数台の ASIMO が連携して、来客者からの注文によって依頼された飲み物を席まで運ぶサービスで、青山と和光市にある Honda の 2 拠点において行われた (Fig. 4)。

得られた改善事項を下記に示す (Fig. 5)。

ネットワークに関しては、実験ではロボットの稼働エリアが屋内空間に限られていたが、現実には屋外移動を含めた通信環境の確保が必要となる。また、ネットワークごとに認証方法が異なるため、統一的なセキュリティ確保にも不安があった。

実験環境構築に関しては、複数台のロボットを青山と和光市の各拠点で運用するため、拠点ごとに管理サーバを用意し、稼働させるロボットごとに設定を行う必要があった。つまり、サーバ設置に時間が掛かるうえ、さらに設定変更の手間も掛かっていた。

開発環境に関しては、ロボットのソフトウェア更新に、ネットワークが有効に活用できなかった。具体的には、あらかじめ物理的な USB メモリに入れたロボットの更新

ソフトウェアを、ロボット管理用の PC (デプロイ PC) に移した後、さらに PC からロボットに転送する必要があった。つまり、ロボットのソフトウェアを更新するプロセスに物理的な手作業が入るため自動化できず、煩雑となっていた。

運用環境に関しては、運用監視用の管理サーバを拠点ごとに設置して、運用監視のオペレータをアサインする必要があった。つまり、拠点が增多るごとにサーバ構築作業費や運用人件費のコストが増加する仕組みとなっていた。

サービスの構築や拡張に関しては、本実証システムは ASIMO のデリバリーサービス用に作りこまれており、もし他に新しいサービスを構築したい場合は、新たにシステムから作り直す必要があった。つまり、新規サービスを開発する場合には新たにシステムの開発コストが掛かり、またサービスを拡張する場合であっても、ASIMO や管理サーバのハードウェアリソースの制約により、拡張できる機能や処理に限界があり、その拡張性には懸念があった。

以上、この実証実験でロボティクスサービスの開発や運用に関する様々な改善事項を得ることができた。



Fig. 4 Tea delivery service by ASIMO

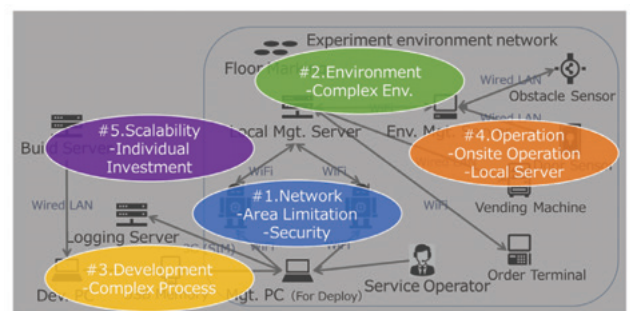


Fig. 5 Issues in building ASIMO delivery service

6. Honda RaaS Platform の特徴

改めて、Honda RaaS Platform とは「ロボティクスサービスを開発・運用するためのプラットフォーム」である。Honda RaaS Platform では、ASIMO のデリバリーサービスの実証実験から得られたロボティクスサービスの開発や運用に関する改善事項に対して、下記の五つのアプローチにより改善と解決を図った (Fig. 6)。

- (1) SIM による稼働エリア拡大とセキュアな通信環境
- (2) クラウドを利用したスケーラブルなインフラ環境
- (3) ネットワーク経由でアクセス可能な開発環境
- (4) リモート監視可能な運用環境
- (5) API 提供によるサービス構築・拡張環境

ネットワークの改善事項に対しては、(1)のアプローチより、ロボットに対して Honda RaaS Platform が提供する SIM を用いてセルラー通信を行うことで、WiFi 領域以外でも通信できるようにする。これにより、ロボットは屋内と屋外の様々な場所でネットワークに接続することが可能になり稼働エリアが広がる。また、各ロボットの SIM 情報を用いてデバイス認証を行い、インターネットを経由することなくセルラー回線経由で通信できることで、ロボットとクラウドの間でセキュアな通信を確立することができる。Honda RaaS Platform のセキュア通信は SIM によるセルラー通信と WiFi 通信の両方をカバーすることができる。WiFi の場合は、インターネットを経由する際に Honda RaaS Platform から発行される認証に必要なセキュリティトークンを用いることで、セキュリティを向上させる。

実験環境構築の改善事項に対しては、(2)のアプローチより、クラウド環境を用いることで処理能力に応じた動的な計算リソースの増加や負荷分散が可能 (スケーラブル) なインフラ環境を提供する。また、拠点ごとのサーバに個別にログインすることなく、Web ブラウザがあれば任意の端末から Honda RaaS Platform の Web GUI にアクセスして拠点ごとのロボットの登録や管理を容易に行うことができる。

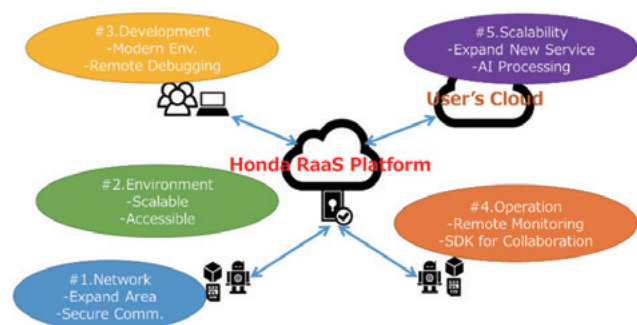


Fig. 6 Approach of Honda RaaS Platform

開発環境の改善事項に対しては、(3)のアプローチより、開発した新しいソフトウェアをネットワーク経由でロボットが利用可能な状態にできることで素早く継続的な開発や改修が可能になる。また、開発者は任意の端末から SIM 経由でロボットにリモートログインすることができ、ロボットの設定変更やログにアクセスしてデバッグを行うことができる。

運用環境の改善事項に対しては、(4)のアプローチより、オペレータは離れた場所からリモートでロボットの稼働状況を確認して監視することができることで、効率的な一括監視が可能になる。また、Web API や SDK を通して運用環境を様々な外部システムと連携させることができる。

サービス構築コストの改善事項に対しては、(5)のアプローチより、Honda RaaS Platform が提供する Web API を用いて、利用者のクラウド環境 (以下、User's Cloud) とシステム連携を行い、ロボティクスサービスを機能拡張することができる。これにより、Honda RaaS Platform が提供する基本機能に加えて、新しい機能を開発したり、既存のサービスを拡張したりすることができる。また、機械学習や AI 処理やシステムを横断するなどの高負荷な処理をクラウド環境にオフロード (負荷分散) できることで、例えば、特定の人を探して屋外に迎えに行くために他の外部システムと連携して特定の人々の位置情報を取得するなど、様々なユースケースを実現できる可能性が広がる。

6.1. Honda RaaS Platform の役割

Honda RaaS Platform の主な役割の一つは接続されているロボットの状態を把握して適切にタスクを振り分けて指示することである。例えば、お茶出しデリバリーサービスの場合は全体シナリオを実行する中で各ロボットへのタスク管理指示制御を役割として担う。各ロボットは Honda RaaS Platform のタスク管理指示制御からのタスク指示を受けて、お茶を積む、お茶を運ぶなどのタスクを実施する。すべてのタスクは Honda RaaS Platform で管理されており、開発者や運用者は Honda RaaS Platform の Web GUI 画面でタスクの実施状況を確認することができる (Fig. 7)。

Honda RaaS Platform の Web GUI 画面はロボットの現在地を示す地図 (Fig. 7(a)) とロボットの現在の状態を示すロボットリスト (Fig. 7(b)) によって構成されている。

ロボットリストにはロボットのタスクの状態などが管理されており、ロボットのタスクに更新があった場合は、対象のロボットのラインが青色でフラッシュ点滅すると同時にタスクが更新される。例えば、コミュニケーションロボットが注文を受けた後、Honda RaaS Platform からワゴンロボットのタスクが待機から配送に更新される例を示す (Fig. 8)。

6.2. 機能全体像

Honda RaaS Platform のデリバリーサービスで利用された機能全体像を整理すると下記ようになる (Fig. 9).

- (1) SIM による認証と TLS 保護されたセキュア通信
- (2) ロボットデータの User's Cloud 環境への転送
- (3) ロボットのタスク状態や位置情報の可視化
- (4) API を用いたサービスシナリオアプリの作成



Fig. 7 Web GUI configuration



Fig. 8 Honda RaaS Platform sends task

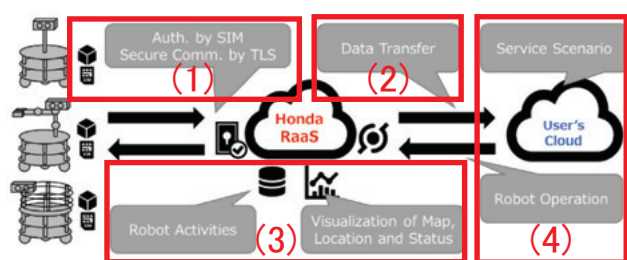


Fig. 9 Overall function

6.3. アーキテクチャとインターフェース

Honda RaaS Platform の全体像を理解するためのアーキテクチャを下記に示す (Fig. 10). Honda RaaS Platform はインフラストラクチャサービス層, アプリケーションサービス層, インタフェース層の三つのレイヤで構成される. インフラストラクチャサービス層では, 主にネットワークや認証に関わる機能が実装されている. アプリケーションサービス層では, Web API や Web GUI で必要な機能に関わる様々なアプリケーションが実装されている.

今回は主にインタフェース層に関して説明を行う. インタフェース層では, Honda RaaS Platform を利用するためのインタフェースとして, ロボットに対して, SIM と cirobot-sdk を提供する. ロボット開発者は開発するロボットに SIM を挿して, cirobot-sdk を組み込むことで Honda RaaS Platform に接続することができる. cirobot-sdk はプログラミング言語 python で実装された SDK である. cirobot-sdk は主に通信に関わる機能を提供しており, Honda RaaS Platform とロボット間の認証や通信プロトコル (HTTP や MQTT など) を隠蔽したソフトウェアになっている. また, ロボットの通信ミドルウェアである ROS⁽⁹⁾ 向けの ROS ノードも提供している. これにより ROS を用いたロボット開発者は, ROS Topic 通信を行うことで, クラウドを必要以上に意識することなく, クラウドに接続されている他のロボットとの連携・協調のためのプログラミングを行うことができる.

User's Cloud に対しては, Web API や ciraas-sdk (python クライアントアプリケーション) を提供する. サービスインテグレーションは Web API や ciraas-sdk を用いてロボティクスサービスを構築することができる. Honda RaaS Platform がロボットから受け取ったデータは User's Cloud のストリームサービスに転送される. User's Cloud ではユーザが Honda RaaS Platform の Web API を用いて自由にアプリケーションを構築することができる. Web API は, Web システムの HTTP インタフェースである REST API として設計されている. ciraas-sdk は, User's Cloud 上で対話的に

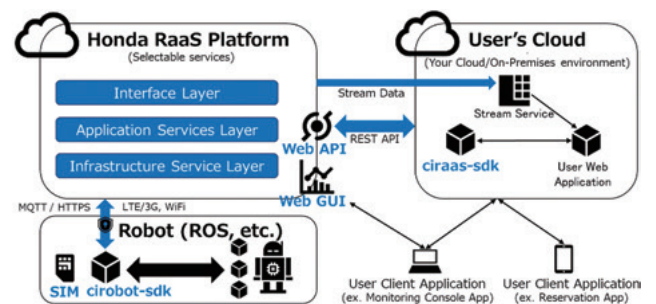


Fig. 10 System architecture diagram

プログラミングを行うための機能を提供する。

運用者の PC やタブレットなどの運用クライアントシステムに対しては、Web GUI をインタフェースとして提供する。運用環境を利用したいオペレータは、任意のデバイスから Web GUI にアクセスすることでロボットの登録や位置情報などを把握しながら運用監視を行うことができる。

6.4. ロボティクスサービスの構築方法

Honda RaaS Platform の利用者であるロボティクスサービスを構築したいエンジニアは、User's Cloud を用意してアプリケーションを実装する必要がある。例えば、User's Cloud として Amazon Web Services (AWS)⁽¹⁰⁾ のクラウド環境を用いてロボティクスサービスを構築する事例を以下に示す (Fig. 11)。

User's Cloud 側でロボットからアップロードされたデータストリームを受けるためには、AWS のデータ転送サービスである Amazon Kinesis Data Streams⁽¹¹⁾ を用いる。また、今回のようなお茶出しサービスのシナリオは、AWS 上にアプリを容易に構築できるサービスである AWS Lambda⁽¹²⁾ を用いてプログラミングを行う。

これにより AWS エンジニアは容易にロボティクスサービスのシナリオを実装して、お茶出しサービス以外にも、受付サービスや案内サービスなどのロボティクスサービスを構築することができる。

6.5. スケーラブルなアーキテクチャ

Honda RaaS Platform は AWS 環境に構築されている (Fig. 12)。Honda RaaS Platform の中で特にコアとなる

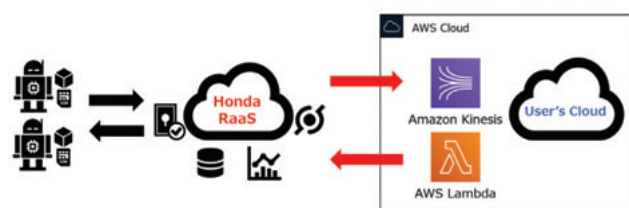


Fig. 11 How to build robotics service on AWS

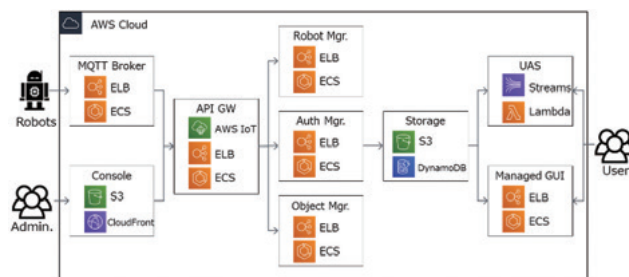


Fig. 12 Cloud architecture diagram

コンポーネントとしては、Robot Manager, Authentication Manager, Object Manager がある。

Robot Manager はロボット登録情報の管理を行う。Authentication Manager はロボットの認証管理を行う。Object Manager はデータの管理を行う。それらを通して、ロボット管理者は Honda RaaS Platform の Web Console (管理者用 GUI) にアクセスして、ロボットの追加や削除などの管理を行うことができる。また、ロボットの運用者は Managed GUI (Web GUI) にアクセスすることでロボティクスサービスの運用を行うことができる。

Honda RaaS Platform には様々な場所の様々なロボットからの同時並列的なアクセスが想定される。アクセスが集中する可能性が高いコンポーネントは、トラフィックを自動的に分散する Elastic Load Balancing (ELB)⁽¹³⁾ による負荷分散と仮想化サービスである Amazon Elastic Container Service (ECS)⁽¹⁴⁾ 上に配置されたコンテナ管理ツールの Docker 環境により、オートスケールするための対策が行われている。

6.6. API とユースケース

Honda RaaS Platform が提供する API の例として、State API と Object Storage API の二つの API とそれぞれを利用した際のユースケース例を下記に示す。

6.6.1. State API

State API を用いると、ロボットの状態を設定・取得することができる。ユースケースとしては、ロボットのバッテリー情報の取得やロボットにタスクを設定するケースを示す。

ロボットは State API を用いて、下記の Fig. 13 に示す通り、ロボットのバッテリー情報を Honda RaaS Platform 経由で User's Cloud まで送信することができる。ここでは“battery”というキーで“20”(=%)という値を送信している。データの転送は cirobot-sdk を用いて、軽量通信プロトコルである MQTT で通信される [Fig. 13 図中 (1)]。ここでロボットが送信した値 {“battery”:20} は Honda RaaS Platform 上の DB に格納される [Fig. 13 図中 (2)]。次に Honda RaaS Platform 経由で User's Cloud の Amazon

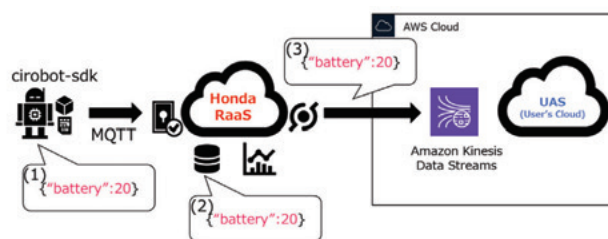


Fig. 13 Battery notification from robot with State API

Kinesis Data Streams に転送される [Fig. 13 図中 (3)].

また、User's Cloud からロボットにタスクを設定する流れを Fig. 14 に示す。ここでは、Fig. 13 よりバッテリー情報の残量が 20% と少なくなってきたため、充電ステーションにロボットが移動するようにタスク指示を設定する。データの転送は HTTPS プロトコルで実行される (Fig. 14 図中 (1))。User's Cloud の中でプログラミングされた AWS Lambda から送信を行った値 {"battery":20,"task":"GoToStation"} は、Honda RaaS Platform 上の DB に格納される (Fig. 14 図中 (2))。そして、Honda RaaS Platform 経由でロボットに MQTT プロトコルでタスク指示の値が転送される (Fig. 14 図中 (3))。

Fig. 13, Fig. 14 の State API を用いることで、ロボットの状態を設定・取得することができる。ロボットの状態は常に Honda RaaS Platform 上で管理されており、ロボットとクラウドの間で状態を同期するロボットシャドウと呼ばれる仕組みを実装することで実現している。これによりロボットとクラウドが適切にロボットの状態を同期して管理することができる (Fig. 15)。

6.6.2. Object Storage API

Object Storage API を用いると、ロボットと Honda RaaS Platform の間で双方向に大きなデータの送受信を行うことができる。ユースケースとしては、User's Cloud から地図データをロボットに配信するケースを示す (Fig. 16)。

ロボットが実空間で移動するためにはロボット用の地図データが必要であるが、一般的に地図データはデータサイズが大きいため一括管理を行う。この地図データを共通データとしてロボットに配信できるようにするために、予

め User's Cloud 環境にアップロードされているものとする (Fig. 16 図中 (1))。

地図データは、User's Cloud 環境の AWS Lambda から Object Storage API を用いて Honda RaaS Platform に HTTPS プロトコルにて転送され (Fig. 16 図中 (2))、Honda RaaS Platform は転送された地図ファイルを DB に格納する (Fig. 16 図中 (3))。

ここで、Honda RaaS Platform は指定されたロボットに対して、MQTT プロトコルで地図データのダウンロード先の URL を発行する (Fig. 16 図中 (4))。

ロボットは任意の安全なタイミング、例えば、充電ステーションに帰還したタイミングで、HTTPS プロトコル経由で地図データをダウンロードすることができる (Fig. 16 図中 (5))。

ロボットは障害物を避けて移動するなど常に様々な処理を行っているため、Honda RaaS Platform が割り込みで大きな地図データを配信してしまうと既存の処理に影響を及ぼしてしまう可能性があるが、このように Object Storage API を用いると大きなデータも安全にロボットに配信することができる。

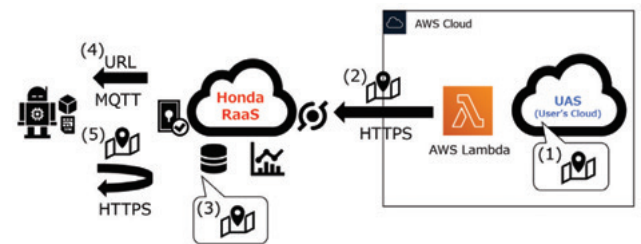


Fig. 16 Map distribution using Object Storage API

7. 実証事例

以下に Honda RaaS Platform による実証事例を示す。

7.1. サービスの分解と再構築

今回は Honda RaaS Platform による「異能を繋ぐのりしろ」というコンセプトを実証するために、複数の異なるロボティクスデバイスを連携・協調させることにより、一連のシームレスなサービス構築事例として、お茶出しのデリバリーサービスの実装例を示す。

まず、過去に行った ASIMO のお茶出しのデリバリーサービスを、単一のサービスという観点から三つのサービスに分解を行った (Fig. 17)。

(1) 注文を受けるサービス (コミュニケーションロボットが担当)

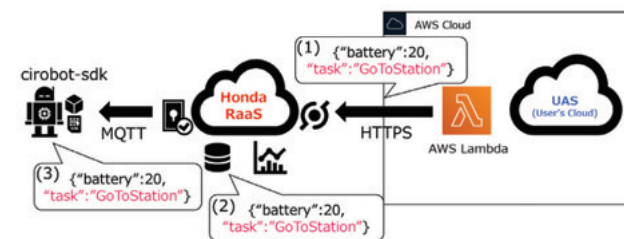


Fig. 14 Set task with State API from User's Cloud

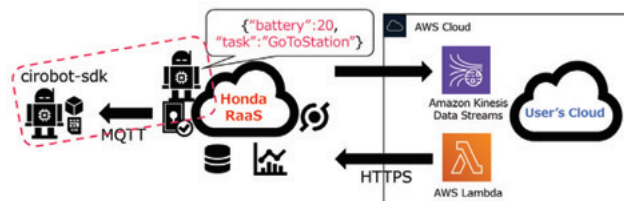


Fig. 15 State synchronization with robot shadows

(2) お茶を積むサービス (アームロボットが担当)

(3) お茶を運ぶサービス (ワゴンロボットが担当)

過去の実証実験では ASIMO が 1 台ですべてを実施していたが、三つに分解したサービスを、三台の異なる役割を持ったロボットによって連携・協調させた一連のサービスとして、お茶出しのデリバリーサービスを再構築した。次にこれらの三台のロボットに SIM と SDK をインストールして Honda RaaS Platform に接続を行った。また、開発者と運用者が任意の端末から Honda RaaS Platform に Web 経由でアクセスできるようにした。

このようにロボットも人も Honda RaaS Platform にアクセスできる環境を構築した。これにより、すべてのロボットを Honda RaaS Platform 上で管理できることと、ロボティクスサービスを実現するためのシナリオをクラウド上で実装して制御することで、クラウドとロボティクスデバイスが一体となってサービスを実現できる形での融合を実現した (Fig. 18)。

7.2. サービスごとの作業分担

Honda RaaS Platform を用いて複数の異なるロボットを協調させたお茶出しのデリバリーサービスの実証実験を行った。

下記に、注文を受けるサービス (Fig. 19 - Fig. 20)、お茶を積むサービス (Fig. 21 - Fig. 22)、お茶を運ぶサービス (Fig. 23 - Fig. 25) の三つのサービスに分けて実施している様子を示す。

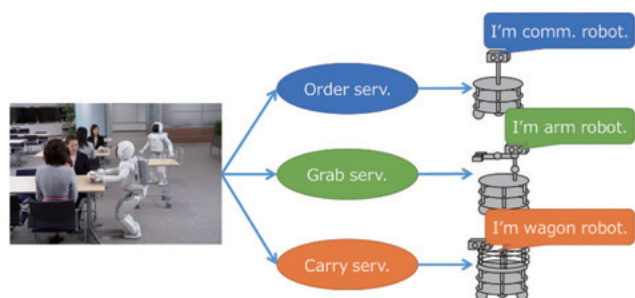


Fig. 17 Assigning role for each service to robot

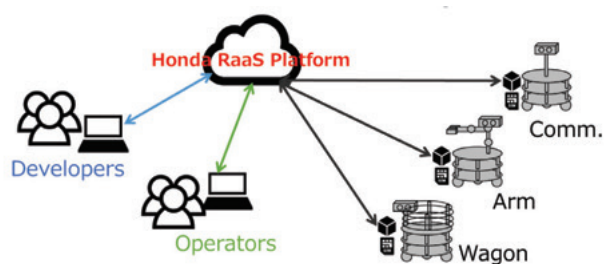


Fig. 18 Fusion of cloud and robotics device



Fig. 19 Request tea delivery service



Fig. 20 Acceptance of service request



Fig. 21 Start the task of grabbing tea



Fig. 22 Grab tea up and give to wagon robot



Fig. 23 Deliver tea to staff seat



Fig. 24 Tea delivery notification by communication robot



Fig. 25 Staff member receives tea

8. 今後の検討課題

同一環境内で複数のロボットが連携してサービスを提供するネットワークロボットシステムの実現に関して、各ロボットのタスク、移動経路、連携シナリオの調整が必要である⁽¹⁵⁾。

また、Honda RaaS Platform によって実現されるロボティクスサービスはSIMやクラウド環境を用いることで、屋内外の広範囲において様々な環境下でのサービスを実現するためのアーキテクチャ設計やリソース配分の調整が必

要になってくることが想定される。

例えば、屋外で人を探す場合は多くのロボットやIoTカメラの画像を取得して、クラウド上で人を認識・特定して、環境データや地図データなどの様々な情報からその場所までの最適な経路を算出する高負荷処理が必要になってくる。一方で、人を探す役割のロボットはその位置情報を受信して経路通りに移動するだけの低負荷処理を行えばよい、などの処理負荷の調整が考えられる。

クラウド環境ではロボット内部の限られたハードウェアリソースと比較して計算資源を豊富に利用することができるが、ロボットとクラウドの間で通信が発生するため、リアルタイム性が損なわれる。一方で、ロボット内部のハードウェアリソースを用いて処理を行うとリアルタイム性や即応性は高いが、計算資源に制約が発生する、というトレードオフがある。

そのため、ユースケースに応じて、どのように計算資源の配置を行うかのバランスを考えてアーキテクチャ設計を行う必要がある。

大枠の指針としては、例えば、複数のロボットをまたいだシナリオ制御や大局的なロボットの移動先の計算は複雑な計算を必要とするためクラウド上の豊富な計算資源を活用する。また、よりリアルタイム性が必要な処理、例えば、障害物の検出と回避や、人とのインタラクションなどはロボット側で処理を行うことが望ましいと考えられる (Fig. 26)。

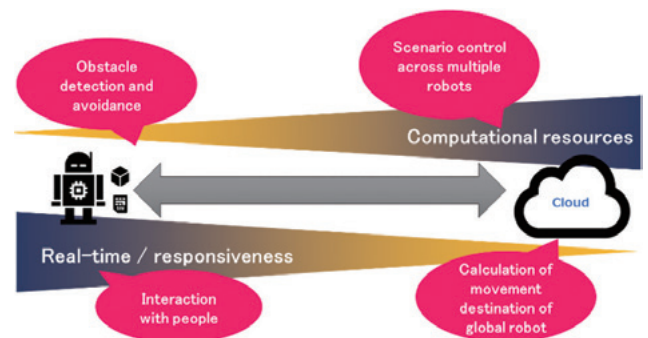


Fig. 26 Role sharing between robot and cloud

9. まとめ

Honda RaaS Platform とは「ロボティクスサービスを開発・運用するためのプラットフォーム」である。

本稿では、Honda RaaS Platform のアーキテクチャやインタフェースを示すと共に、コンセプトである「異能を繋ぐのりしろ」を複数の異なるロボティクスデバイスを連携・

Table 1 Comparison of enhancements

No.	Issues	Previous (ASIMO)	Present (Honda RaaS Platform)	Effect
1	Network security	Per-network authentication	Device authentication by SIM	Secure communication to cloud
2	Experimental environment	Complex individual construction	Build on cloud environment	Provide scalable environment
3	Development environment	Physical update	Update via remote	Enable agile development
4	Operation environment	Human operation at each site	Remote central operation	More efficient remote monitoring
5	Service scalability	Individual implementation	Utilizing common API	More efficient development

協調させることで、一連のデリバリーサービスの実証事例として示した。

Honda RaaS Platform の導入により、SIM による稼働エリア拡大およびセキュアな通信環境、クラウドを利用したスケーラブルなインフラ環境、ネットワーク経由でアクセス可能な開発・デバッグ環境、リモート監視可能な運用環境、API 提供によるサービス構築・拡張環境を提供することでロボティクスサービスの開発や運用を改善できることを示した (Table 1)。

今後は様々なユースケースの実証実験を通して、ロボティクスサービスの開発と運用に必要な機能やツールを拡充していく必要がある。また、屋内屋外の様々なエリアのユースケースに対応できるように、5G ネットワークの活用やロボットとクラウド間で柔軟にリソース配分が可能なアーキテクチャ設計の改修を継続していく必要がある。

10. おわりに

Honda としては初めての“ソフトウェアプラットフォーム”というコンテンツをグローバルに発信した。

Honda が掲げる 2030 年ビジョンや Robotics Vision を実現するためには、Honda 一社だけではなく、様々なパートナーとのエコシステムの構築が必須である。そのためにロボティクス領域はもちろんのこと、クラウドや IoT に関わる様々なエンジニアやプロフェッショナルに Honda RaaS Platform のことを知って頂き、共感と協力を頂きながら共に成長させていくことが必要である。

本稿が Honda RaaS Platform の「異能を繋ぐのりしろ」としての具体的な働きかけの一助となれば幸いである。

参考文献

- (1) <https://www.honda.co.jp/sustainability/report/pdf/2019/Honda-SR-2019-jp-016.pdf>
- (2) 重見聡史, 河口裕一郎, 吉池孝英, 川邊浩司, 小川直秀: 新型 ASIMO の開発, Honda R&D Technical Review, Vol.18, No.1, p.38-44
- (3) 中臺一博, 長谷川雄二, 関口達彦, 辻野広司: 人と共

存するロボットシステムの展望, 日本ロボット学会誌, Vol.27, No.1, p.6-9, (2009)

- (4) <https://global.honda/innovation/CES/2018.html>
- (5) https://www.honda.co.jp/CES/2019/detail/raas_platform/
<https://pages.awscloud.com/rs/112-TZM-766/images/J1-06.pdf>
<https://youtu.be/YgMkb9Fyojo>
- (6) <https://iotnews.jp/archives/128847>
<https://iotnews.jp/archives/127980>
https://pages.soracom.jp/Discovery2019_download.html
- (7) <http://roscon.jp/>
http://roscon.jp/presentations/ROSCon_JP_2019_presentation_5.pdf
<https://vimeopro.com/osrfoundation/roscon-jp-2019/video/370246969>
- (8) <https://iotloft5.splashthat.com/>
- (9) <https://robots.ros.org/>
- (10) <https://aws.amazon.com/>
- (11) <https://aws.amazon.com/kinesis/data-streams/>
- (12) <https://aws.amazon.com/lambda/>
- (13) <https://aws.amazon.com/elasticloadbalancing/>
- (14) <https://aws.amazon.com/ecs/>
- (15) 塩見昌裕, 神田崇行, Dylan F. Glas, 佐竹聡, 石黒浩, 萩田紀博: 複数の案内ロボットが連携してサービス提供するネットワークロボットシステムの実現, 日本ロボット学会誌, Vol.29, No.6, p.544-553, (2011)

■著 者■



東 治 臣