

「R-env:連舞™」クラウド対応型インタラクション制御技術 ～デバイスを連携させたサービスを簡易に作成するためのアーキテクチャ～

Cloud-based integrated development environment for creating human-machine interaction services

松元 崇裕*¹ 松村 成宗*¹ 細淵 貴司*¹ 望月 崇由*¹ 吉川 博*¹ 山田 智広*¹
Takahiro Matsumoto Narimune Matsumura Hosobuchi Takashi Takayoshi Mochiduki Hiroshi Yoshikawa Tomohiro Yamada

*¹ NTT サービスエボリューション研究所
NTT Service Evolution Laboratories

1. 序論

介護, 教育, ナビゲーションなど様々な分野においてインタラクションロボットによるサービス実現に向けた研究・開発が進められている。これらのサービス実現には, ロボット単体だけでなく, 周囲のセンサと連携しながら外部状況をより正確に認識することや, サービスに合わせて複数のロボットやデバイスを組み合わせることが必要となることが多い。例えば, 環境に設置された人の位置を取得するセンサと連携することで, ロボットは自身のセンサ検知範囲外でもユーザ位置を把握することができる。またナビゲーションでは, 地図アプリと連携することでディスプレイに道順を表示したり, 非移動型ロボットが他の移動ロボットに案内を引き継いだりすることでサービスの質を向上することができる。

一方, 複数のロボット・センサ・アプリが連携したサービスの作成には, 連携する機器が相互に情報を送受信するためサーバを用意したり, 各機器の IF 仕様を理解しそれに合わせた連携制御のプログラムを作成したりする必要がある。そのためロボット単体のサービス開発と比べて, 複数の機器が連携するサービスの開発には, より多くの手間や幅広い専門的な知識が必要となり開発コストが高かった。

本論文は複数のロボット・センサ・アプリが連携したサービスを簡単に作成可能なクラウド対応型インタラクション制御技術 R-env:連舞™(以下 R-env)について, その機能およびアーキテクチャを述べる。また二日間のハッカソンで, R-env を初めて利用した人が作成したサービスを通じて, R-env を利用したサービス開発に対する効果について調査結果を述べる。

2. 従来研究

ロボット単体のサービスを開発するフレームワークには Robot Operating System(ROS)が提案されている[1]。ROS はロボットに必要な各機能をライブラリによりノードとして提供し, 複数のノードを管理・実行することでロボットサービスを実現する。しかし, ROS は基本的に 1 つのロボット制御を主目的としたフレームワークであり, 複数のロボットやアプリを連携させたサービスを作成する仕組みまではサポートされていない。一方, 本論文で提案する R-env は個々のロボット機能追加は問わず, ROS を利用して作成されたロボットも含めて, 複数のロボットやアプリを連携させて利用するサービスの作成をサポートする技術である。

また, ロボットやスマートフォン, センサなどを連携させたサービスを提供する基盤としてはユビキタスネットワークロボットプラットフォーム(UNR-PF)の研究が進められてきた[2]。UNR-PF ではロボットの持ち得る機能をモジュールとして抽象化し, サービス

開発者が個々のロボット仕様を気にすることなく開発できることを特徴としている。一方で, 人型ロボットに限っても個々のロボットが持つ機能は様々な種類があり, ガジェットやセンサまで含めると, 機能の全てを予めプラットフォーム側で抽象化するのは難しく, 規定された抽象化の枠組みでは扱えないロボット・センサ・アプリが存在する問題があった。また UNR-PF は機器間のメッセージを規定するものの, サービス開発をサポートする環境が存在するわけではなく, 実行環境も1つのサービス毎に準備を行う必要があった。そこで R-env は接続するデバイスをプラットフォーム側で規定せず, 接続する側が自由に機能を宣言可能とすることで, 扱えるロボット・センサ・アプリの制約をなくす。更に, 機器間のメッセージの送受信をどのように制御するかを記述するグラフィカルプログラミング環境をクラウド上で提供することにより, サービス開発・実行を容易にできる環境を実現する。

3. R-env:連舞™クラウド型統合開発・実行環境

R-env はクラウドシステムとしてブラウザからプログラムを作成可能な開発環境機能と, 作成したプログラムをクラウド上で動作させる実行環境機能を持つことを特徴とする。

3.1 開発環境機能

R-env はローカル PC 内で動作する開発環境とは異なりクラウド上で動作することを特徴とする。サービス製作者はブラウザより自身のアカウントで R-env にアクセスすることで, クラウド上でサービスの開発・保存・実行をすることができる。そのため PC への開発環境インストールの必要はなく, 異なる PC でも環境の違いを意識せずに開発をすることが可能となる。

R-env におけるプログラムでは, 各ロボット・デバイス・アプリへ命令を記述する「ノード」と, ノード間の移動条件を記述した「遷移」からなる遷移図(スクリプト)を GUI で記述することでプログラムを作成する。図 1 に R-env によるスクリプト作成時の画面を示す。R-env では上記のようなスクリプトを複数作成し, それらをシーンという単位でまとめて並列・逐次実行することで 1 つのサービスを実現する。各シーンにおけるスクリプトは同時に並列実行

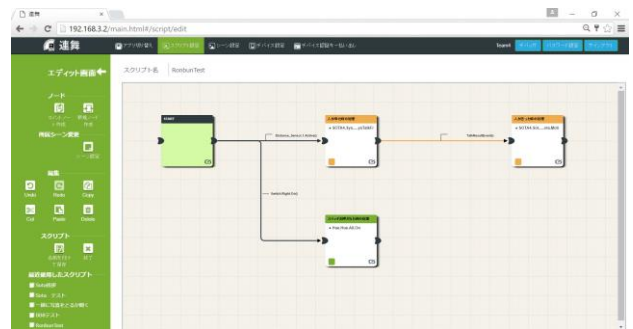


図 1 R-env において機器の連携を記述するスクリプト作成画面

連絡先: 松元崇裕, NTT サービスエボリューション研究所,
神奈川県光の横須賀市光の丘 1-1, (046) 859 - 3566
matsumoto.takahiro@lab.ntt.co.jp

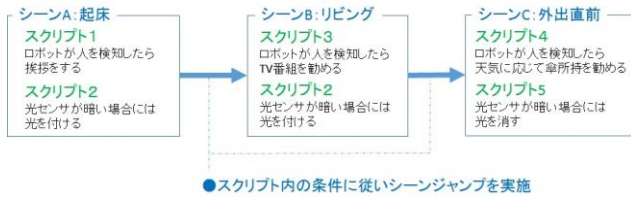


図 2 簡単なサービス例によるスクリプトとシーンの関係

され、何れかのスクリプトでシーンジャンプが記載されたノードに到達すると、現在のシーンで実施していた全のスクリプトを中断して次のシーンへ移動する。また 1 つのスクリプトは複数のシーンに所属させることが可能である。図 2 にスクリプトとシーンの関係を簡単なサービス例として示す。

図 2 のようにスクリプトとシーンに分割することで、一つの遷移図で全プログラムを作成するよりも遷移図をシンプルに記述することが可能であり、1 つのスクリプトが複数のシーンで使えることで、作成プログラムの再利用性を高めることができる。

3.2 実行環境機能

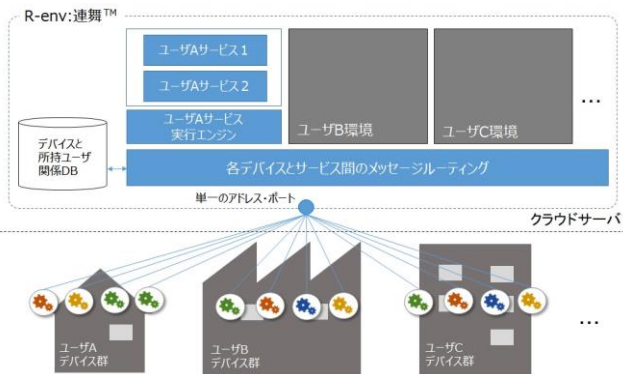


図 3 R-env における全体のアーキテクチャ構成

作成されたサービスはユーザ毎に紐づけられ、ブラウザから実行できる。図3に R-env のアーキテクチャ全体を示す。R-env から操作するデバイスは全て R-env サーバ指定の単一のアドレス・ポートへ接続し R-env とメッセージの送受信が可能である必要がある。サービスが実行されると、サービス実行エンジンは GUI で記述されたプログラムを解析し、デバイス間のメッセージのルーティングを実施する。そして各デバイスが R-env から受け取ったメッセージに応じて動作を行い、センシングや動作の結果を R-env へ返すことでサービスが実現される。

3.3 デバイスの接続方式

R-env では様々なデバイス(ロボット・センサ・アプリ)の開発者が自由に機能を定義しても、そのデバイスを簡単に R-env へ接続し GUI から扱えることを目的としている。そこで R-env に対する各デバイスの接続方式は、多くのデバイスへ簡単に実装可能で、メッセージを簡潔に記述可能な WebSocket による接続と JSON によるメッセージの送受信の方式を採用した。

新しいデバイスを R-env へ接続するには、デバイスの機能および機能の説明を WebSocket による接続時に指定の JSON フォーマットで R-env へ送信することが要求される。各機能は例えば「センサが人を検知したときにどのようなメッセージを R-env へ送信するか」、「R-env より指定のメッセージを受け取った時にロボットはどのような動作をするか」などを記述したものである。このとき R-env とデバイス間で規定されるのは機能の宣言方法のみであり、宣言の内容はデバイス開発者が自由に決定できる。

R-env はデバイスが接続されると、デバイスの機能と説明一覧をセットで開発環境の GUI へ反映する。それにより開発環境からは、デバイスの機能とその説明を見ながら GUI で作成中のプログラムへ追加デバイスの機能を組み込むことが可能である。

4. ハッカソン

R-env によるロボット・センサ・アプリを連携させたサービス開発について調査を行うため、ハッカソン形式で短時間の開発イベントを実施し、そこで作成されたサービスについて分析を行った。

4.1 開催内容と条件

ハッカソンは 2 日間で、5 人～6 人で構成される 4 つのチームが参加した。参加者は一部ロボットの開発経験を持つ人が含まれるが、全員がハッカソンで R-env を初めて利用する。参加者に対しては初日の朝に 1 時間ほどの演習を行う形で R-env の使用方法は教示された。各チームはサービスアイデア検討や演習を除き、約 12 時間の中でサービス作成を行った。

R-env に接続済みのデバイスとしては、開催側より事前に 4 種類のロボット、照明、表示 URL を R-env の指示で遷移するブラウザアプリ、4 種類のセンサ(近接・照度・加速度・モーション)が準備された。各チームには 1 つのユーザアカウントが付与され、1 つのチームが 1 つのサービスを作成する形で実施された。

4.2 作成されたサービス内容と分析

各チームが作成したサービスに利用されたロボット・センサ・アプリ数は最大 14 個～最少 6 個であり平均が 10.5 個であった。またデバイスを制御するサービスプログラムの全チーム平均はノード合計が 47.5、スクリプト数は平均 7、シーン数が 4.75 であった。事前の準備無しで 2 日間という時間の制約の中、何れのチームも多数の機器を連携させたサービスが作成されていることが分かる。以下が実際に作成されたサービス内容である。

- 占いと相性診断を行う接客ロボットサービス
- 温泉で受付と入浴アドバイスをする接客ロボットサービス
- ゴミ捨て場でカラス撃退と挨拶を行う監視ロボットサービス
- アミューズメント施設で案内兼接客を行うロボットサービス

また開催者側が事前に準備したデバイス以外にも全チームが 1 つ以上の自作デバイスをサービスに追加しており、デバイス接続方式が簡潔である点の効果が確認された。追加デバイスには、画像処理による入浴姿勢の点数判定アプリ、V-SidoOS を搭載した小型の人型ロボット、IFTTT、HTML5 による写真撮影アプリなどであった。接続されたデバイスが多岐に渡ることから、デバイス接続者が自由に機能を設定して R-env へ接続可能とする設計が、効果的に機能していることが確認できた。

5. まとめ

本論文は複数のロボット・センサ・アプリを簡単に制作するためのクラウド対応型インタラクション制御技術 R-env について、機能と全体のアーキテクチャについて述べた。また、ハッカソンにおける短期間のサービス開発の結果から、R-env のサービス作成の簡易性について調査を行い、その分析結果を述べた。

参考文献

- [Eitan 2010] Eitan, M. E., Eric, B., Tully, F., Brian, G. and Kurt, K., : The Office Marthon : Rubust Navigation in an Indoor Office Environment, Robotics and Automation, 2010.
- [Nishio 2013] Nishio, S., Koji, K. and Norihiro, H.: Ubiquitous Network Robot Platform for Realizing Integrated Robotic Applications, Intelligent Autonomous Systems 12, 2013.