

研究終了報告書

「データ中心開発パラダイムを実現する包括的な IoT システム開発環境」

研究期間：2018年10月～2022年3月

研究者：高瀬 英希

1. 研究のねらい

本研究提案では、構成要素ならびに設計階層を包括した開発フレームワークを確立し、IoT 開発におけるパラダイムシフトの実現を目指す。IoT 技術の本質をなすビッグデータの処理と通信を柱に据えて、IoT システムアーキテクチャにおける関数型プログラミング言語 Elixir の新たな可能性を開拓する。

本研究提案の“包括的”が示す 1 つめの意味は、IoT システムの各構成要素における設計階層の縦断である。設計階層とは、SoC アーキテクチャ、コンパイラ、インタプリタ、実行環境(インタプリタ・OS)、通信ミドルウェアおよびライブラリを指す。それぞれの設計階層の機能および利点を協調的に活用することで、各構成要素における実行効率、リアルタイム性、ネットワーク応答性および省電力性を最大化できる実行基盤を確立する。特に、組込み向けデバイスの活用が前提となる IoT デバイスおよびエッジサーバについて、実行性能およびメモリ効率の観点で優れた革新的な実行環境を提供する。“包括的”の 2 つめの意味は、IoT システムの各構成要素を横断して一体的にアプリケーション開発を進めることを指す。ビッグデータの処理に重点を置いたシステムレベルの観点から、IoT システム全体の開発を一気通貫に進められる開発環境の構築を目指す。IoT アプリケーションの各構成要素への分散配置に加えて、要素間の効率的な通信機構を自動生成できるようにする。すなわち、データ中心型の統一的なプログラミング技法によって、各構成要素のアプリケーションを同時並行的に開発できるようにし、システム系全体の設計生産性の向上を実現する。

2. 研究成果

(1)概要

縦断方向の研究課題としては、Elixir を設計言語とする SW/HW 協調システム開発技術、ROS と FPGA の統合開発を実現する ZytelBot プラットフォーム、および、Elixir の IoT フレームワーク Nerves におけるリアルタイム性能の改善手法に関する研究成果を挙げた。SW/HW 協調システム開発技術については、Elixir コードから FPGA 向けのハードウェア回路を合成する技術およびプロセッサからハードウェア回路に効率的にアクセスできるインタフェース機構の生成技術を確立し、高品質な SW/HW 協調システムを開発できる環境を構築した。Zytelbot プラットフォームについては、ロボットシステム向けの開発プラットフォームである ROS (Robot Operating System) において、ROS の利点を活用しながら FPGA の技術統合の容易化を実現するための実行環境およびソフトウェア志向の協調設計手法を提案した。Nerves の改善手法については、IoT 技術向けの様々な指標を定量的に評価可能なベンチマーク環境を整備し、特にリアルタイム性に影響を与える主要な指標を明らかにし、これらの制御によって Nerves のリアルタイム性を改善できることを示した。

横断方向の研究課題としては、ROS 2 の主要な側面である通信ミドルウェア機能の IoT システム構築への応用に着目し、組込み IoT デバイス向け ROS ノード軽量実行環境 mROS、および、Elixirを開発言語とする ROS 2 クライアントライブラリ Rclex に関する研究成果を挙げた。mROS については、まず ROS 1 ノードを組込みデバイス上で実行可能とする mROS の研究開発を進め、任意型のメッセージを扱うための通信機能の拡張に取り組んだ。続けて ROS 2 への展開に取り組み、単一障害点の運用に関する課題を克服可能な軽量実行環境である mROS 2 を開発した。ロボットシステムにおける組込みデバイスの活用促進と開発生産性の向上を図ることができ、高いリアルタイム性と通信性能を発揮できることを確認している。Rclex については、関数型言語 Elixir による ROS 2 のクライアントライブラリであり、Elixir の利点のひとつである並列分散処理向けのプロセスモデルを活用している。ROS 2 の通信性能に関するスケーラビリティを向上させることができ、多数のデバイスが接続された IoT システムのような環境への貢献が期待される。

包括的な開発環境としては、IoT/クラウドロボティクス時代の仮想環境である「箱庭」の研究開発に取り組んだ。仮想環境である箱庭上に様々なソフトウェアやサービスを持ち寄って、机上で実証実験できる場を提供する。IoT の各要素が連携される複雑なシステムの事象や状態を、「箱」の中ではシナリオに応じた同じ挙動が再現されるようにする。そして、箱庭のシステム構成要素である「アセット」を差し替えることで、検証の対象/抽象度/レベルを任意に変更することができるように設計している。

(2) 詳細

研究テーマ A「Elixir を設計言語とする SW/HW 協調システム開発技術」

Elixir には Flow ライブラリに代表される並列処理機構を標準で備える。また、Enum ライブラリおよびパイプライン演算子によって、データ統計処理の流れを直接的に表現できる。研究者は、Elixir の並列プログラミングモデルに着目し、これがハードウェア向けに実現されるデータフロー型処理と親和性が高いことを明らかにした。この性質を活かして、論理を書き換え可能なデバイスである FPGA 向けのハードウェア回路を合成する技術を確立した。加えて、Elixir アプリケーションの稼働するプロセッサより密結合された FPGA 上のハードウェア回路に効率的にアクセスできるインタフェース機構の生成技術を確立した。これらにより、Elixir コードのみから高品質な SW/HW 協調システムを開発できる環境を構築した。

研究テーマ B「ROS と FPGA の統合開発を実現する ZytileBot プラットフォーム」

ロボットシステム向けの開発プラットフォームである ROS (Robot Operating System) において、FPGA ハードウェア設計技術の技術統合の容易化を目指した。具体的には、ROS 2 の利点を活用した実行環境およびソフトウェア志向の協調設計手法を提案し、これをプラットフォームとして構築した。設計者は実行環境において、ROS 2 ツールを活用したハードウェア開発を実現できる。提案する協調設計手法では、設計支援ツールにハードウェアで実行したいソフトウェアアプリケーション記述および構成情報を入力とすることで、ROS 2 のコンポーネント指向開発に沿った協調アプリケーションを生成する。

提案するプラットフォームによって、ロボットシステム開発における、プログラマブル SoC の

導入と協調設計の加速化を促進できる。提案手法を自律移動ロボットの開発に適用し、その有効性を評価した。画像の推論による物体認識をハードウェアで実装し、プログラマブル SoC において実行されるロボットシステムに組み込んだ。組み込み CPU と比較したところ、アルゴリズムの動作単体で約 188 倍高速なハードウェアが、ドライバアプリケーションを含めても、約 30 倍の高速化および約 21 倍の性能対電力効率であることを確認した。

本プラットフォームは ZytileBot と名付け、GitHub 上に OSS として公開して研究開発を進めている。国際会議 FPT2019 での論文発表および関係する FPGA Design Competition での優勝を収め、電子情報通信学会リコンフィギュラブルシステム研究会での招待講演を実施した。

研究テーマ C「Elixir の IoT フレームワーク Nerves におけるリアルタイム性能の改善手法」

Nerves とは、関数型言語 Elixir における IoT デバイスのための実行環境であり、Linux の rootsfs およびブートローダ、最小セットの Erlang VM のみで Elixir アプリケーションを駆動することができる。研究者は本技術の可能性に着目し、コア開発者と積極的な情報交換および議論を交えながら、その動向調査および産業界展開に取り組んだ。研究成果としては、複数の IoT ボード上でベンチマークアプリケーションを実行し、IoT 技術向けの様々な指標を定量的に評価可能な環境を構築した。本環境による評価から、特にリアルタイム性に影響を与える主要な指標を明らかにした。これを踏まえて、ガーベッジコレクションの設定、プロセスの優先度、および、CPU 周波数のそれぞれの制御により、Nerves のリアルタイム性を改善できることを示した。本研究の成果は、“Nerves in Action”という Nerves 開発者向けの Webinar における招待講演として公開しており、Elixir で実装される IoT システムの品質向上に繋がるものである。

研究テーマ D「組み込み IoT デバイス向け ROS ノード軽量実行環境 mROS」

先述した ROS は、出版購読型の通信ミドルウェアとしての側面があり、ロボットのみならず IoT システム構築への応用が期待される。研究者は、ROS 通信ミドルウェアの機能を省電力性かつリアルタイム性に優れた組み込みマイコンで活用可能とする技術の開発に取り組んだ。本研究項目は、デバイス間およびデバイスとサーバ間の通信機構の効率化に繋がる、IoT 構成要素の横断に関わるものである。

まず、本研究開始時の主流であった ROS 1 向けを対象として、組み込みデバイス上で ROS 1 ノードを実行可能とする mROS の研究開発を進めた。本研究では、mROS において任意型のメッセージを扱うための通信機能の拡張に取り組んだ。ROS のビルドシステムで定義されたメッセージの型に関するヘッダ情報を組み込みデバイス向けに変換し、この出版購読通信を軽量に実現するための内部構造を設計した。他の組み込みデバイス向け通信手法と比較評価し、提案手法はリアルタイム性およびメモリ効率の観点で優れていることを定量的に示した。実装成果は GitHub でオープンソースとして公開しており、また、本研究の成果は情報処理学会論文誌として出版された。これにより、ロボットソフトウェア開発の効率化、および、ROS ノード間の通信処理の最適化が達成できる。

続けて、ROS の新世代版である ROS 2 への展開に取り組んだ。ROS 2 では micro-ROS のプロジェクトが知られているが、欠点として agent node を実行する必要がある。これは ROS 1/mROS における通信経路の調停のための master node に該当し、ROS 2 でも依然として単一障害点の運用が避けられないことを意味する。これを解決するため、我々は組み込みデバイ

スのための ROS 2 ノード軽量実行環境である mROS 2 の研究開発に取り組んでいる。図 1 に示すように、RTPS 層においては組込み向け DDS 実装を採用している。この上位層に ROS 2 互換な API を提供することで、単一障害点の運用に関する課題を克服し、ロボットシステムにおける組込みデバイスの活用促進と開発生産性の向上を図っている。定量的評価では、micro-ROS と比較して高いリアルタイム性と通信性能を発揮できることを確認している。

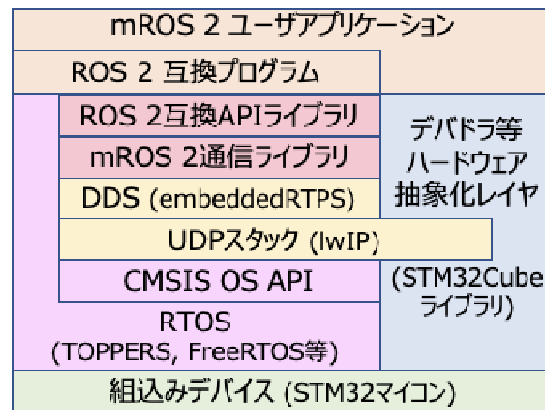


図 1 : mROS 2 のソフトウェア階層

研究テーマ E「Elixir を開発言語とする ROS 2 クライアントライブラリ Rclex」

大量のノードから構成される ROS 2 システムでは、トピックを介した出版購読型のノード感通信において通信遅延やメモリおよび CPU 使用率が增大することが予想される。本研究課題では、ROS 2 のスケーラビリティを向上させるため、関数型言語 Elixir による ROS 2 のクライアントライブラリ Rclex を提案した。Elixir ではプロセスモデルに基づく並列分散処理を容易に実現でき、プロセスごとにメモリ管理機能が備わっている。提案手法では、まず、ROS 2 の共通機能が実装された RCL のうち、データの出版購読機能のための必要な関数および構造体を NIF (Native Implemented Functions) 機能によって呼び出せるようにする。そしてトピックに対して、大量の ROS ノードを同時に生成できるように API を設計する。出版購読通信や付随するコールバック処理は、軽量な Erlang プロセスの実行として実現している。

提案手法によって、ROS 2 のノード数を増加させた場合でも、メモリ消費量や CPU 使用率が抑えられることが確認できた。つまり Rclex は、スケーラブルな ROS 2 アプリケーションの開発に貢献することができ、IoT システムのような多数のデバイスが接続された環境への貢献も期待される。

研究テーマ F「IoT／クラウドロボティクス時代の仮想開発環境「箱庭」

多様かつ大量の情報機器がネットワーク通信を介して密接に絡み合う IoT システムでは、システム構築の問題発生経路の複雑化やサービス構築時の実証実験のコスト増が課題となる。このような技術的課題を解決し、IoT のシステム開発／サービス構築を加速化することができるプラットフォームとなること目指した「箱庭」の研究開発を進めた。複数の技術領域から由来する技術者が集合して IoT／自動運転システムを開発する際に、仮想環境である箱庭上に様々なソフトウェアやサービスを持ち寄って、机上で実証実験できる場を提供する。IoT の各要素が連携される複雑なシステムの事象や状態を、「箱」の中ではシナリオに応じた同じ挙

動が再現されるようにする。そして、箱庭のシステム構成要素である「アセット」を差し替えることで、検証の対象／抽象度／レベルを任意に変更することができるように設計している。

箱庭については、その狙いやコンセプトを立証するためのプロトタイプモデルを、単体ロボットシミュレータ向けに実装した。本実装では、制御プログラムを実行する組込みマイコンのシミュレータおよびロボットの振る舞いと外界環境を仮想化する物理エンジンが協調的に動作する。箱庭は複数のシミュレータが同時並行で動作するため、双方のシミュレータの実行を効率的に時間同期する機構を提案した。さらに、本プロトタイプを自律移動ロボットの開発に活用し、設計要件に見合う制御パラメータをベイズ最適化によって自動探索する手法を提案した。

3. 今後の展開

本研究課題は設計実装を重視して展開を進めてきており、研究成果の実用化ならびに社会実装への展開の達成に繋がるシーズ価値は高いものと考えている。

ROS 2における通信ミドルウェア機能を諸元とした mROS 2 および Rclex については、現在までの研究期間で挙げた成果としては、現行バージョンの定量的評価では競合技術と比べて有意な結果が得られているが、基本的なコア機能の設計および実装に留まっている。実用化に繋げるためには、任意型のメッセージ通信への対応などの機能拡充が不可欠となる。その後の社会実装への展開については、本成果の技術応用に興味のある企業を募り、機能拡充ならびにソースコードの品質向上をさらに推し進めていき、今後数年のうちでの研究成果の実用製品への採用・搭載を目指している。

仮想開発環境である箱庭については、本格的な研究開発の着手は 2019 年 4 月頃からであったが、その後の社会情勢変化によって注目度が増すこととなった。まずは、講義・演習のオンライン化という大きな影響を受けた教育現場において、演習用の環境としていくつかの大学にて採用されることとなった。また、ET ロボコンという組込み技術者向けのロボット・プログラミングコンテストにおいて、シミュレータによって実施された競技システムの一部に箱庭の技術成果が採用された。今後は企業での開発現場での採用に定めるために、デジタルツインやクラウドネイティブ技術を取り入れてさらなる技術進展を図っていく予定である。

4. 自己評価

本研究課題は、研究成果のオープン化を強く意識して推進する方針を取った。研究進行の早い段階から基本概念の設計実装を実施し、GitHub 上にその成果物のソフトウェアを一般公開していくこととした。基本実装したソースコードのリファクタリングや機能検証・テストについては、組込みソフトウェアの開発経験が豊富な企業への業務委託を実施し、その品質向上に努めた。また、取り纏まった研究成果はもちろんのこと、研究開発の途上の成果でも積極的に開発者会議や技術者イベントなどで発表し、技術コミュニティへの情報発信や知見の共有を進めた。国際的にも、Elixir に関する開発者・技術者会議である ElixirConf や Code BEAM で技術発表を行った。この結果、個人型の研究開発プロジェクトの成果としては十分といえる知名度を獲得することができたと考えている。

本研究課題の柱に据えている Elixir は知名度が必ずしも高いとはいえないため、これが研究成果の社会展開の障壁になりうる懸念があった。このため、研究・技術コミュニティの形成ならびに促進にも注力した。具体的には、まず、ROS の国内向け開発者会議である ROSCon JP の実

行委員および運営法人の理事を務めており、本イベントの開催運営を担ってきた。Elixir/Nerves については、この国内コミュニティである NervesJP を立ち上げ、多数の研究者・技術者と継続的に活動を進めてきている。また、組込み分野では高い知名度を持つ月刊誌である Interface において、Elixir および Nerves の技術解説に関する記事を寄稿した。商業雑誌において Nerves が取り上げられたことは日本初のことである。加えて、Erlang Ecosystem Foundation と呼ばれる国際的な法人における分科会である Embedded Systems Working Group の立ち上げに携わり、Erlang/Elixir の組込み技術の展開について積極的な議論と情報交換を行っている。これらの活動は当然ながらさきかけ研究期間の終了後も継続していく予定であり、Elixir/Nerves および本研究成果の普及促進に寄与するものであると考えている。

いっぽうで、研究活動としての成果である論文誌や学会発表で多くの業績を挙げることはできなかった。設計開発ツールの研究課題であったため妥当な定量的評価が難しいものであったことが要因のひとつではあるが、上述の活動等に重心を置きすぎた自覚がある。研究論文として未発表の成果も多いため、早急に原稿投稿を進めていく予定である。

5. 主な研究成果リスト

(1) 代表的な論文(原著論文)発表

研究期間累積件数: 2件

1. Hideki Takase, Tomoya Mori, Kazuyoshi Takagi and Naofumi Takagi, "mROS: A Lightweight Runtime Environment of ROS 1 nodes for Embedded Devices," Journal of Information Processing, Vol. 28, No. 2, pp. 150-160, 2020.

組込みデバイス向けの ROS ノード軽量実行環境である mROS を提案する。mROS は、リアルタイム OS および組込み向け TCP/IP スタックで構成され、汎用計算機上で動作する ROS ノードとの接続確立および出版購読通信機能を提供する。mROS の API は汎用の ROS と互換性があるように設計しており、ノードの移植性が高いという利点がある。評価実験および適用事例によって、提案手法がロボットシステムのエッジ処理におけるリアルタイム性の向上および省電力化に寄与することを示した。

2. 祐源英俊, 高瀬英希: ROS ノード軽量実行環境 mROS のユーザ定義メッセージ型の対応のための機能拡張, 情報処理学会論文誌, 2021 年, Vol. 62, No. 3, pp. 917-930.

ROS ノード軽量実行環境である mROS において、ROS のユーザ定義メッセージ型のヘッダファイルを自動生成する手法を提案する。ROS で規定されたメッセージの型ならびにユーザが独自定義する型を mROS 上で処理可能とするため、通信ライブラリの新たな動作フローを設計した。さらに本研究では、提案手法に基づく機能拡張を mROS に施したうえで、その性能を競合技術と比較評価した。これにより、本研究成果は組込み機器を活用する ROS ベースのロボットシステム開発の容易化に貢献することを示した。

(2) 特許出願

研究期間全出願件数: 0 件(特許公開前のももの含む)



(3) その他の成果(主要な学会発表、受賞、著作物、プレスリリース等)

1. Hideki Takase, “mROS: How to Integrate ROS Components into Embedded Devices,” the 19th International Forum on Embedded MPSoC and Multicore (MPSoC’ 19), Hakone, Japan, July 2019. [invited talk]
2. Hideki Takase, “Cockatrice: A Hardware Design Environment with Elixir,” ElixirConf US 2019, Colorado, USA, Aug 2019.
3. 高瀬英希:関数型言語 Elixir の IoT システム開発への展開, 第 23 回プログラミングおよびプログラミング言語ワークショップ(PPL 2021), 2021 年 3 月. [招待講演]
4. 高瀬英希, 武田大輝, 今西洋偉, 祐源英俊:ROS 2 Client Library for E², ROSCon JP 2021, 2021 年 9 月.
5. Hideki Takase, “Rclex: A Library for Robotics to meet Elixir,” Code BEAM America 2021, Nov 2021.