

JAUS および RT ミドルウェアを規範とした分散アーキテクチャの提案

Proposal of Distributed Architecture Based on JAUS and RT Middleware

稲 村 渡	技術開発本部総合開発センター制御システム開発部
藤 井 正 和	技術開発本部総合開発センター制御システム開発部
村 上 弘 記	技術開発本部総合開発センター制御システム開発部 主幹
富 沢 正 明	技術開発本部総合開発センター制御システム開発部 部長

RT システムの構築において、標準技術でのモジュール化による開発手法は、機能の変更や向上が容易で、再利用性が高いなどの効果があり、産業の発展に寄与すると期待されている。本稿では、アプリケーションレベルの仕様として JAUS (Joint Architecture for Unmanned Systems), 実装技術として RT ミドルウェア (Robot Technology Middleware) を規範とした分散アーキテクチャを提案する。構成した分散アーキテクチャを移動ロボットに搭載し、実験によって効果を確認した。

For construction of a robot system, a modular approach by standardized technologies is expected to improve compatibility and reusability of robotic functions. It is expected to contribute to industry progress because robot systems will be developed more rapidly and flexibly by integrating various modules in the near future. This paper proposes a distributed architecture for robot systems using standardized technologies. We constructed the distributed architecture based on both of JAUS (Joint Architecture for Unmanned Systems) as a specification of an application layer and RTM (Robot Technology Middleware) as a development framework. The architecture was applied to mobile robots. The experimental results show the effectiveness and reliability of the proposed architecture.

1. 緒 言

RT (Robot Technology) とは、「ロボット技術を活用した、実世界に働きかける機能を持つ知能化システム」の総称である⁽¹⁾。近い将来、RT が生活を支えることで、豊かで安心・安全な社会を実現することが期待されている。RT が生活空間に浸透するためには、多様化するニーズにこたえる RT システムが必要であり、そのためにシステムの開発、保守、拡張のコストを低減する開発手法が求められる。これまでの RT システムの開発においては、CPU 能力の制約などから、開発するシステムごとにソフトウェアの設計が行われてきた。近年の CPU 能力や通信システムの著しい発展によって、モジュール化されたソフトウェア部品を組み上げてシステムを構築することが可能となってきた。多数のソフトウェア部品を組み合わせることで多様な機能を短時間で実現することや、必要なソフトウェア部品のみを開発して組み替えることで新しい機能の追加や機能の高性能化をする手法の確立が期待されている。こうした背景から、共通の規約を策定するために RT のソフトウェア基盤の標準化がさまざまな形で推進されている。

モジュール化による開発手法の有効性は誰もが認めるところであるが、実際にはなかなか浸透していないのが現状

である。研究開発グループごとに、これまでの RT システム開発でのソフトウェア資産の蓄積があるので、新しい手法を導入すると一時的にシステム開発の効率が下がってしまうことがその要因である。そのうえ、現状では使用可能なソフトウェア部品がほとんどないため、導入初期段階では標準化されたモジュール化手法に合わせて、ソフトウェア部品をひとつひとつ開発しなければならない。逆に、ソフトウェア部品が充実して容易に入手できるようになれば、開発効率が上がるので利用者も増えることになり、必要に応じて新しいソフトウェア部品が開発され、一層ソフトウェア部品が充実していく循環が起きると考えられる。そのためには、RT システム開発のソフトウェア基盤を国内外で標準化された仕様に準拠していくことが重要となる。標準技術の普及には、進みつつある標準技術を実際に使い、効果と性能評価、課題を示し、改良やノウハウの蓄積を進めていくことが有効である。

本稿では、ソフトウェア部品によって RT システムを構築するための分散アーキテクチャを提案する^{(2),(3)}。提案する分散アーキテクチャは、標準化として取り組まれている二つの仕様を相互補完することによって構築した。また、提案した分散アーキテクチャを用いて、移動ロボットのシステムを実際に開発するなかで、不足する機能の仕様

を補った。さらに、開発したシステムを用いて移動ロボットの実験を行い、その機能を確認した例を示す。

2. ソフトウェア標準技術

2.1 RT ミドルウェアの概要

RT ミドルウェア（以下、RTM と呼ぶ）は、「様々なロボット要素（RT コンポーネント）を通信ネットワークを介して自由に組み合わせることで、多様なネットワークロボットシステムの構築を可能にする、ネットワーク分散コンポーネント化技術による共通プラットフォームを確立すること」⁽⁴⁾を目的として開発された標準技術である。独立行政法人新エネルギー・産業技術統合開発機構の21世紀ロボットチャレンジプログラムにおいて、ロボット用分散ミドルウェアの研究開発プロジェクトによって研究が開始され、RTM のプロトタイプが開発された。現在はソフトウェアの標準化推進の業界団体である OMG (Object Management Group) の Robotics-DTF (Domain Task Force)⁽⁵⁾で、国際標準仕様として標準化が進められている。国内では、日本ロボット工業会内に委員会が設置され、筆者らも参加して、標準化活動に協力している。

RTM は、ロボットのソフトウェアを部品単位で開発する場合に、相互に接続しデータ交換可能であることを保証する標準仕様である。システムのソフトウェア部品として機能するために、RTM のソフトウェア部品である RT コンポーネントの基本動作を決定するための状態遷移が定義されている。また、RT コンポーネント間の連携を容易にするために、通信チャンネルが定義されている。

OpenRTM-aist は、産業技術総合研究所が開発した RTM のプロトタイプの実装技術である。OS に依存しない分散オブジェクト技術 CORBA (Common Object Request Broker Architecture) を利用することで RTM を構成している。RT コンポーネントや通信チャンネルとしてのポートは、CORBA オブジェクトとして実装されているため、容易に RT コンポーネント間の通信が実現できる。また、ポートで送受信するデータ型は、基本的な整数型や実数型、その配列などが定義されており、アプリケーションでの必要に応じて独自に定義することも可能となっている。

このように、RTM によって、アプリケーションに依存しない汎用的な RT システムの部品を構築することが可能である⁽⁶⁾、⁽⁷⁾。しかし、RT システム構築においてソフトウェア部品を組み合わせる有効に活用するためには、アプ

リケーションレベルにおいてもソフトウェア部品の仕様が統一されている必要がある。

2.2 JAUS の概要

JAUS (Joint Architecture for Unmanned Systems)⁽⁸⁾は、無人システム (UGV: Unmanned Ground Vehicle, UAV: Unmanned Air Vehicle, UUV: Unmanned Underwater Vehicle)などを指し、典型例は移動ロボットである)において、システムのライフサイクルコストの低減、開発期間の短縮、技術導入の容易さ、新機能の拡張性を実現することを目的にアメリカ国防総省が推進している標準仕様である。多くの研究機関やメーカーが参加する JAUS WG によって仕様の策定が行われている。JAUS は、ハードウェアやソフトウェアの実装技術、システムの実行する作業に依存しないことを目標に仕様が構築されている。

JAUS においては、モジュール化の単位をコンポーネントと呼ぶ。システムはコンポーネントの組み合わせによって構成され、各コンポーネントはメッセージ通信をすることによって連携して動作する。無人システムを構成するために必要な各機能をコンポーネントに割り当てる仕様が定義されている。

JAUS では、実装技術は規定されていないことから、実際のシステムを構築するためには、なんらかの実装技術を採用しなければならない。また、JAUS は基本的な移動ロボットを構成するために必要な内容は規定されているが、標準化途上の仕様であることから未定義の部分も残っている。

3. 分散アーキテクチャの提案

3.1 分散アーキテクチャの基本構成

これまで述べたように、RTM は、アプリケーションレベルの仕様に依存しない RT のモジュール化をサポートする仕様で、OpenRTM-aist が RTM の実装技術を含めたプロトタイプである。一方、JAUS はハードウェアや実装技術に依存しないアプリケーションレベルの仕様である。

したがって、RT システムのための分散アーキテクチャの基本構成として、これらの相互に補完な標準技術を採用して構築することを提案する。すなわち、OpenRTM-aist で実装するソフトウェア部品の機能を JAUS の仕様にに基づき規定する。また、RT コンポーネントの通信チャンネルであるポートを使用し、JAUS の仕様によって定義されたメッセージの通信を規定する。この結果、RTM におけるモジュールである RT コンポーネントが連携し、RT シ

システムとして統合することが可能となる。

JAUS では、コンポーネントやメッセージに ID コードが割り当てられており、ユーザ用のコード範囲も用意されている。JAUS で定義済みのものに関してはそれを使い、JAUS で未定義のものに関しては、ユーザに開放された仕様に基づき、JAUS の設計思想に準拠するように新たに定義した。

3.2 コンポーネントの基本設計

JAUS の機能を RTM で実装するには、基本となるコンポーネントが RT コンポーネントで実現できなければならない。JAUS では、コンポーネントの共通の基本動作として状態遷移を規定している。また、RTM でも RT コンポーネントの状態遷移を規定している。以下に、本分散アーキテクチャで採用した状態遷移とメッセージ通信の設計を述べる。

(1) 状態遷移

RT コンポーネントでは、実装を考慮し JAUS よりも詳細な状態遷移が定義されている。この状態遷移を、JAUS の状態遷移に対して矛盾のないようにマッピングを行った結果を第 1 図に示す。この状態のマッピングによって、RT コンポーネントを利用して、JAUS のコンポーネントの状態遷移機能を満たすことができた。

さらに、この状態遷移によって、各コンポーネントの基本動作が統一され、個別に開発された RT コンポ

ネントを統合してシステム化することができる。

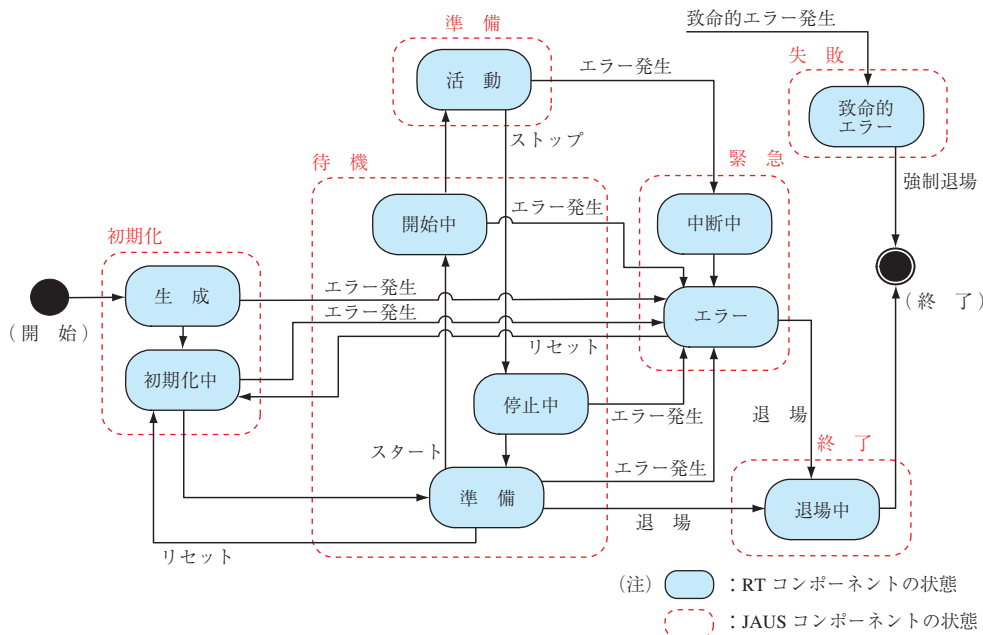
(2) 入出力ポート

OpenRTM-aist では、RT コンポーネント間の通信チャンネルであるポートを CORBA オブジェクトとして実装しているため、CORBA の仕様に従って定義された構造体を通信データ型とすることができる。ポートは、データ型ごとに用意され、同じデータ型の入力ポートと出力ポート間のみで通信することができる。

そこで、用途に応じて定義されている JAUS のさまざまなメッセージをポートで通信できるように構造体として定義した。ただし、JAUS では、すべてのコンポーネントが共通して扱わなければならない中核メッセージ群（コンポーネントを操作するコマンド系のメッセージ群）と各コンポーネントの機能に応じた固有のメッセージが定義されているため、中核メッセージ群については、構造体の集まりである一つの共用体として定義した。この結果、本分散アーキテクチャにおける各コンポーネントの RTM のポートは、JAUS としての中核メッセージ用とコンポーネント固有のメッセージに対応したものを用意すればよいことになる。

3.3 移動ロボット用 RT システムの設計

本稿で提案している分散アーキテクチャで、力制御可能なマニピュレータ搭載型移動ロボットの RT システムを設計した。第 2 図に、システム構成を示す。JAUS で定義されているさまざまなコンポーネントのうち、移動ロボット



第 1 図 コンポーネントの状態遷移
Fig. 1 State chart of components



第3図 協調搬送実験
Fig. 3 Experiments of cooperative transportation

ポートをもつように設計し、さらにマニピュレータの手先偏差情報を受け取る入力ポートを追加した。また、マニピュレータ手先制御ドライバには、手先の偏差情報を出力するポートを追加した。

従来のローカルベクタドライバは、単にベクタ情報で与えられた目標走行軌道に追従する機能を持つコンポーネントであった。後方の移動ロボットにおいて、これと同様のポートを持つ前述の搬送ドライバに入れ替えることで協調搬送作業を行う機能を保有させることができる。今回、協調搬送実験によってその機能拡張、変更を確認した。

このように、本分散アーキテクチャにおいて、対象となったコンポーネントのみの改造または入れ替えで、ほかの部分の変更を行うことなくシステムに新たな機能を付加することが可能であることを確認した。今後、新たな制御アルゴリズムの検証や実験を行う場合にも一部のコンポーネントのみを更新することで対応可能である。

5. 結 言

本稿では、標準化された仕様に従って部品を開発し、その部品を組み合わせるシステムを構築する手法を示した。二つの標準技術、JAUS のアプリケーション仕様と RTM のプロトタイプ実装技術である OpenRTM-aist を規範として構築した分散アーキテクチャを提案した。提案したアーキテクチャを 2 台の移動ロボットに適用し、モジュール化による機能変更・拡張の容易性、柔軟性を示した。

現在は、RT が社会を支える技術として普及していく黎明期である。標準化が進みつつある現在であるからこそ、

それを実際に使い先例を示す意義は大きいと考える。本稿が RT 産業の発展の一助なれば幸いである。

参 考 文 献

- (1) 社団法人日本機械工業連合会, 社団法人日本ロボット工業会: 21 世紀におけるロボット社会創造のための技術戦略調査報告書 日機連 12 先端 - 8 2001 年 5 月 p. 1
- (2) 稲村 渡, 藤井正和, 坂野 肇, 田中康仁: JAUS および RT ミドルウェアを規範とした分散アーキテクチャの提案 第 6 回計測自動制御学会システムインテグレーション部門講演会論文集 2005 年 12 月 pp. 967 - 968
- (3) 稲村 渡, 藤井正和, 坂野 肇, 田中康仁: JAUS および RT ミドルウェアを規範とした分散アーキテクチャの提案—複数移動ロボットによる協調制御システムへの適用— 第 11 回ロボティクスシンポジウム論文集 2006 年 3 月 pp. 538 - 543
- (4) RT ミドルウェアプロジェクトホームページ: <http://www.is.aist.go.jp/rt/> (参照 2007 - 04 - 10)
- (5) OMG Robotics DTF ホームページ: <http://robotics.omg.org/> (参照 2007 - 05 - 10)
- (6) 安藤慶昭, 末廣尚士, 北垣高成, 神徳徹雄, 尹祐根: RT コンポーネントによるシステム構築法—RT ミドルウェアの基本機能に関する研究開発(その 14)— ロボティクス・メカトロニクス講演会 2005 講演論文集 2005 年 6 月 2A1 - N - 072 (1) - (4)
- (7) 北野 斉, 桑田 亨, 中原智治, 荒木秀和, 石川武志, 小宮山優: RT ミドルウェアを用いた生活支援ロボットシステム ロボティクス・メカトロニクス講演会 2005 講演論文集 2005 年 6 月 2P1-N-068 (1) - (4)
- (8) JAUS ワーキンググループのホームページ: <http://www.jauswg.org/> (参照 2007 - 05 - 10)
- (9) 藤井正和, 稲村 渡, 田中康仁, 小菅一弘: 複数移動ロボットによるフックを用いた協調搬送制御 第 11 回ロボティクスシンポジウム論文集 2005 年 3 月 pp. 326 - 331