

第9回（社）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



SICE[®]



2008.12.5-7
長良川国際会議場／未来会館

OpenRTM-aist-1.0の新機能

産業技術総合研究所 安藤慶昭

New Features of OpenRTM-aist-1.0

*Noriaki ANDO

National Institute of Advanced Industrial Science and Technology

Abstract— The RTC (Robotic Technology Component) specification has been released as official OMG (Object Management Group) standard in April 2008. Since our middleware OpenRTM-aist was based on the specification under work of FTF, it would be updated. This paper shows new features and changes from old version of OpenRTM-aist-0.4.

Key Words: RT-Component, real-time, execution context

1. はじめに

2008年4月に国際標準化団体 OMG (Object Management Group) から RTC (Robotic Technology) の標準仕様 Version 1.0 が正式に公開された [1]. RTC 標準仕様は産総研が中心となり策定された、ロボット用ソフトウェアコンポーネントの国際標準である。

これは、2006年9月に承認された RTC 標準仕様 (Alpha specification) が1年間の最終文書化委員会 FTF (Finalization Task Force) による、公式公開に向けた文書の校正と、仕様の実現可能性検証を経て公開されたものである。2007年5月に公開された OpenRTM-aist-0.4.0 [2] は、この Alpha specification の仕様に準拠したミドルウェアであり、その後の仕様変更と公式公開を受けて今回新たなバージョンとして OpenRTM-aist-1.0 をリリースする。

本稿では、公式公開された OMG RTC 標準仕様に準拠した RT ミドルウェア OpenRTM-aist-1.0 の新機能および以前のバージョンからの変更点を紹介する。

2. OMG における標準化

OMG (Object Management Group) は1989年に設立されたソフトウェア標準化団体である。OMG が主導して策定した代表的標準仕様としては、分散オブジェクトミドルウェア CORBA (Common Object Request Broker Architecture)、ソフトウェア分析・設計のためのモデリング言語 UML (Unified Modeling Language) 等がある。

OMG 標準は、MDA (Model Driven Architecture) に基づくモデルベースの仕様として記述される。MDA では特定の通信ミドルウェアプラットフォームに依存しない PIM (Platform Independent Model) として仕様を策定し、PIM から特定のルールに基づきマッピングされたインターフェース仕様 PSM (Platform Specific Model) に基づき実装を作成する。

RT コンポーネントモデルを標準化するため、著者らは2004年から OMG における標準化活動を行ってきた [3]。著者らおよび米国ミドルウェアベンダの二者が提案した “Robotic Technology Component Specification (RTC Specification)” の草案が2006年9月に、OMG の標準化作業部会 AB (Architecture Board) により承認された。この仕様は “Alpha specification” と呼ばれ、その後一年間かけて一二人の委員からなる最終文書化

委員会 FTF (Finalization Task Force) により、文書の校正が行われた。同時に、標準仕様に基づいた実装を作成することで、仕様の整合性・完全性の検証することが求められる。

委員らにより提示される全ての変更点は “issue” として管理され、委員による投票で仕様書への反映が決定される。OMG RTC FTF においては、合計28個の “issue” が挙げられ、うち24個が標準文書に反映された (Beta specification)。変更反映後、再度 AB および TC (Technical Committee) にて承認されたのち Version 1.0 公式文書として公開される。

3. インターフェース仕様の変更

FTF の過程では、Alpha specification の RTC インターフェースの仕様がいくつか変更された。仕様の変更は大きなものではないが、主な変更点は以下のとおりである。

3-1 attach_context オペレーションの変更

RTC に対し実行コンテキストを関連付けるオペレーションとして、ComponentAction インターフェースにあった attach_context が、LightweightRTObject へ移動された。実行コンテキストを関連付ける操作は、コンポーネントのコールバックインターフェースである ComponentAction に付属するよりも、コンポーネントの本体の LightweightRTObject に付属する方が、意味論 (セマンティクス) の観点から、より適切であるとの判断からである。

3-2 2つの DataFlow 型 RTC の統合

DataFlow 型 RTC とは、入力データを処理し出力する、いわゆるフィルタタイプのコンポーネントのことである。Alpha specification においては、DataFlow 型の RTC は、内部に入れ子状にコンポーネントを含むことができる DataFlowComposite 型と、DataFlowComposite 型の子コンポーネントになることができる DataFlowParticipant 型のコンポーネントが定義されていた。しかし、Composite パターンを導入することで DataFlowComposite と DataFlowParticipant 型のコンポーネントを区別なく扱うことができるため、これら2つの型は、DataFlowComponent 型として統合された。

3.3 実行コンテキスト取得操作の変更

RTC から見た実行コンテキストには、自らが所有する “owned” な実行コンテキストと、他の RTC が所有する実行コンテキストに参加している “participate” の 2 つの種類が存在する。Alpha specification においては、これら RTC が関連付けられている実行コンテキストを取得する操作は `get_contexts()` のみで、2 種類の実行コンテキストを明示的に区別して取得する操作が存在しなかった。FTF においてこの問題が指摘され、`get_contexts()` オペレーションは `get_owned_contexts()` と `get_participating_contexts()` に分割された。

4. OpenRTM-aist-1.0

新たな OMG RTC Specification に準拠したミドルウェア OpenRTM-aist-1.0 を実装した。上述した Alpha specification から Version 1.0 への変更が実装に及ぼす影響は軽微であり、比較的容易に新たなインターフェースに追従することができる。これらの変更に加えて、OpenRTM-aist-1.0 では RT ミドルウェアの移植性の向上、機能の拡張を中心に変更を行った。

4.1 移植性向上のための OS 機能抽象化層

OpenRTM-aist-0.4 までは、多様な OS へ対応するために ACE (Adaptive Common Environment) と呼ばれる OS 抽象化ライブラリを使用していた。ACE は Washington University の D.Schmidt らにより開発されている OS 抽象化、多様なソフトウェアパターンの提供およびネットワークプログラミングフレームワークを提供する C++言語のためのライブラリである。しかしながら ACE は、サイズが大きい、 μ ITRON をはじめとする系組み込み OS には対応していないなど、実用的ロボットシステムに適用するには様々な制約がある。そこで、OpenRTM-aist と ACE を分離し、ACE の代替となる新たな OS 抽象化層 coil (Common Operating-system Infrastructure Layer) を実装した (Fig. 1)。coil は OpenRTM-aist に必要な ACE 代替機能のみを実装したライブラリであり、対応 OS は FreeBSD, Linux, Windows である。また、ACE へのアダプタも提供されているので、coil を ACE と共に利用することで上記以外の OS で ACE が対応している OS 上でも動作する。

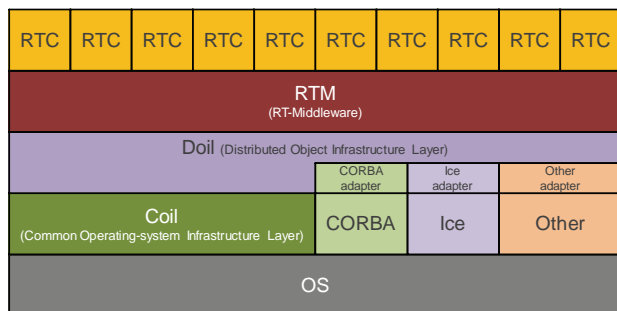


Fig.1 New architecture of OpenRTM-aist.

4.2 通信ミドルウェア抽象化層

OpenRTM-aist は通信ミドルウェアとして、CORBA (Common Object Request Broker Architecture) を利用している。しかしながら、ロボットシステムにおいては、Ice (Internet Communication Engine), Java

RMI (Java Remote Method Invocation), SOAP (Simple Object Access Protocol) 等も通信ミドルウェアとして用いられている。こうした他のロボット用ミドルウェアと連携する方法としては、[4] のようにブリッジを作成する方法などがある。しかし、ブリッジによる連携可能な機能は通常限定的であり、完全に透過的な相互運用は一般に困難である。

一方 OMG RTC 仕様は、ミドルウェアに依存しない PIM と呼ばれる抽象モデルで仕様が規定されているため、CORBA 以外へのマッピングさえ定まれば、他のミドルウェア上でも同様の機能を提供するコンポーネントフレームワークを実現することが可能である。

そこで、RTC を CORBA と分離すると共に、通常のオブジェクトと多様な通信ミドルウェアとの仲介を行う抽象化層 doil (Distributed Object Infrastructure Layer) を新たに導入した (Fig. 1)。CORBA 以外の通信ミドルウェアへの RTC インターフェースマッピングを行うと共に、doil フレームワークに従って、対象通信ミドルウェア用のいくつかのアダプタオブジェクトを実装するだけで、同一の RT コンポーネントを容易に他のミドルウェアへ対応させることが可能となる。

4.3 新たなデータポート

[5] で提案した、新たな機能拡張を行ったデータポートを実装した。これにより、一つの出力ポートから多数の入力ポートへ、個別にバッファリングとフロー制御を行うことができるようになる。また、コンポーネント間のデータフロー制御、オーバーフロー、アンダーフローの検知が行えるようになる。

5. 終わりに

本稿では、2008 年 4 月に国際標準化団体 OMG から公式にリリースされた RTC 標準仕様 Version 1.0 について述べた。また、OMG RTC 標準仕様 Version 1.0 に基づき実装された、新しい RT ミドルウェア OpenRTM-aist-1.0 の新たな機能を示した。今後は、他の coil の拡張により μ ITRON 等組み込み OS への対応を行うと共に、doil フレームワークに基づき、他の通信ミドルウェアベースの RTC を実現し、相互運用性の評価を行う予定である。

参考文献

- [1] Object Management Group, “Robotic Technology Component Specification Version 1.0”, formal/2008-04-04
- [2] 安藤 他 “OMG RTC 標準仕様に準拠した RT ミドルウェアの実装”, ロボティクス・メカトロニクス講演会 2007, 1P1-A02, 2007.05
- [3] 神徳 他, “RT ミドルウェア標準化活動への誘い”, 日本機械学会 ロボティクス・メカトロニクス講演会 2005, p.2P1-N-066, 2005.6
- [4] 山野辺他, “天気情報取得コンポーネントの開発 RT ミドルウェアグループとロボットサービスエンジニアチプとの連携”, 第 25 回日本ロボット学会学術講演会, 2I13, 2007.09
- [5] 安藤 他, “RT コンポーネント間のデータ送受信方法に関する考察”, 日本機械学会 ロボティクス・メカトロニクス講演会 2008, 1P1-E08, 2008.06