

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2009
2009 JSME Conference on Robotics and Mechatronics

ROBOMECH 2009 in FUKUOKA

豊かな暮らしを創生するロボティクス・メカトロニクス
Robotics and Mechatronics for Creating an Affluent Society



太宰府天満宮本殿

Sun. 24th ~ Tue. 26th May, 2009

Fukuoka International Congress Center
Fukuoka, Japan

主催 社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



RT コンポーネントの複合化とその分類および構築ツールについて

Composition of RT-Components its classification and building tool

正 安藤 慶昭 (産総研) 正 清水 昌幸 (静岡大学) 正 原 功 (産総研) 正 比留川 博久 (産総研)

Noriaki ANDO, National Institute of Advanced Industrial Science and Technology, n-ando@aist.go.jp

Masayuki SHIMIZU, Shizuoka University

Isao HARA, National Institute of Advanced Industrial Science and Technology

Hirohisa HIRUKAWA, National Institute of Advanced Industrial Science and Technology

The composite component, which synthesizes a single RT-Component from multiple RT-Components, can be utilized for some purpose such as grouping, hiding, synchronization of execution, synchronization of status and so on. In this paper, classification of composite components is discussed from the aspect of relation between parent RTC and children RTCs in logic execution and status transition. An implemented composite component framework and composite component authoring tools are also mentioned.

Key Words: RT-Middleware, RT-Component, composite component

1. はじめに

著者らは、RT ミドルウェア (RTM): OpenRTM-aist において、幾つかの RT コンポーネント (RTC) を合成してひとつの RTC にする手法として複合コンポーネントを提案してきた [1]。また、稲村ら [2]、清水ら [3] はこれとは異なる視点から別の複合コンポーネントの実現手法を提案している。

その後、国際標準化組織 OMG において RT コンポーネントモデルが標準化され、2008 年 4 月に OMG Robotic Technology Component Specification として一般公開された [4]。標準仕様で定義されている RTC モデルは従来のものとは異なるため、複合コンポーネントモデルも変更する必要がある。

複合コンポーネントは、子 RTC の隠蔽と公開、実行の同期あるいは状態の同期などの観点から、さまざまな種類のものが実現可能である。本稿では、これら複合コンポーネントを用途と特性から分類し直し、新たなコンポーネントモデルに適用する。このモデルに基づき新たな複合コンポーネントフレームワークを提案するとともに、実現されたフレームワークとこれを操作するためのツールを示す。

2. OMG RTC モデル

OMG で標準化された RTC モデルにおいては、コンポーネントのロジックの実行は実行コンテキスト (Execution Context: EC) と呼ばれる論理スレッドにより行われる [4]。EC と RTC は多対多の関連を設けることが可能であり、たとえば、ひとつの EC が複数の RTC を駆動したり、逆に複数の EC がひとつの RTC を駆動したりすることが可能である。

2.1 複数のコンポーネントの協調

複数の RTC が協調して動作する際に、最も単純な方法として、RTC {A, B, C} のロジックを一つのスレッドで直列に実行する方法が考えられる。これを実行コンテキストと RTC の関係で表現すると、ひとつの EC と 3 つの RTC {A, B, C} を関連付けることで実現することが可能である。

このように、OMG RTC モデルにおいては、EC と RTC の関連により複数の RTC が持つロジックの実行順序や同期メカニズムを実現する。

3. 複合コンポーネント

3.1 親 RTC との関連付け

RTC の複合化とは「2 つ以上の RTC について階層関係を設定すること」であるといえる。

オブジェクトの階層化の手法として Composite パターンがある。Composite パターンは図 1 (a) に示すように、あるクラスが自身を保持できるように、自身に対して集約関係を持たせる方法である。こうしたオブジェクトには一般に `add_component()`、`remove_component()` などの関数を

持たせ階層的な関連付けを行う。以前の RT ミドルウェア (OpenRTM-aist-0.2) では Composite パターンを利用した複合化が提案されていた [1]。

一方、OMG RTC 仕様においては RTC の本体である RTObjct 自体には、このような構造を規定おらず、参照仕様である SDO において規定されている Organization オブジェクトを使用する構造となっている (図 1 (b))。

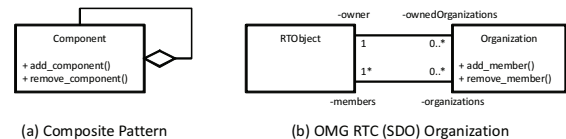


Fig.1 Composite pattern and OMG RTC (SDO) organization.

3.2 ポートの委譲と隠蔽

幾つかの RTC を一つの複合コンポーネントにすることで、ひとまとまりのサブシステムを一つのビルディングブロックに見せることが可能である。

RTC は他の RTC と相互作用を行うための端点としてポートを持つ。複合コンポーネントは、まとめられたひとつのサブシステムをどのようなビルディングブロックとして表現するかに応じて、適切なポートを持ち、あるいは隠蔽すべきである。

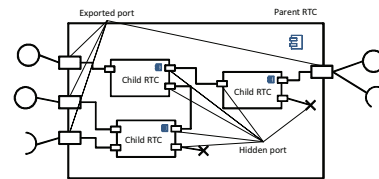


Fig.2 Exported ports and hidden ports in composite component.

子 RTC は親である複合コンポーネントに自身のポートを委譲することで、親がポートを持っているように振る舞うことができる。どのポートを委譲または隠蔽するかは、構成するシステムに依存するため、複合コンポーネントの作成時に、どのポートを親に委譲するかを選択できなければならない。

3.3 実行の同期・非同期

各 RTC が固有に持つロジックの実行は、通常 RTC 生成時に同時に生成される実行コンテキスト (Execution Context: 以下 EC) によって行われる。従って、通常各 RTC 同士は図 3(a) のように互いに独立かつ非同期に実行される。複合コンポーネント化が単に複数の RTC から構成される複雑な



Fig.3 Executed ports and hidden ports in composite component.

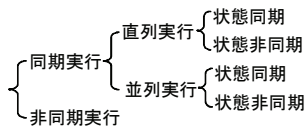


Fig.4 Executed ports and hidden ports in composite component.

システム内部構造の隠蔽が目的なら，RTC 間の同期は必要ない（非同期複合コンポーネント）。

これに対して，リアルタイムサーボ処理や，フィルタ構造のシステムにおいては，複数の RTC を同期的に動作させる必要がある．これを実現する方法として，親 RTC の EC を子 RTC に関連付けることにより実行を同期させる，同期複合コンポーネント [1] がある．同期の必要性の有無は複合コンポーネントの適用対象によって異なる．

3.4 並列実行・直列実行

実行が非同期である場合，各 RTC は独立した EC により駆動されるため，実行は並列的である．一方，実行が同期的である場合は，各 RTC の EC による駆動は並列実行と直列実行を考慮する必要がある．直列実行は，図 3(b) のように，複合コンポーネントに含まれる子 RTC の実行がある一定の順序に従って同期的かつ直列に行われる．並列実行は，図 3(c) のように，複合コンポーネントに含まれる子 RTC の実行が，同期的かつ並列に行われる．

3.5 状態の同期・非同期

EC は複数の RTC に関連付けが可能であり，各 RTC 毎に独立の状態（Inactive-Active-Error の三状態）を管理している．前項の同期複合コンポーネントを形成する場合を考える．

例えば，ある処理を行う RTC-B に対して，前処理を行う類似の RTC-A, RTC-A', RTC-A" の内一つを選択的に Active 状態にして使用する場合，これら 4 つの RTC の状態は独立でなければならない．一方，複合コンポーネントに含まれる 3 つの RTC, RTC-X, RTC-Y, RTC-Z がそれぞれ依存関係がある場合，すべて同時に Active, Inactive, Error に遷移する必要がある．

このように，複合コンポーネント内の RTC の実行の同期と状態の同期はそれぞれ独立に考慮する必要がある．複数 RTC 間の実行が非同期の場合には，各 RTC の状態遷移が同期的に行われるとは限らないため，厳密な意味での状態同期は不可能である．

3.6 分類

以上から，複合コンポーネントの種類を分類したものを図 4 に示す．上述したポートの委譲と隠蔽は，すべての複合コンポーネントに必須の機能であるため，この分類には含めない．非同期実行の場合，子 RTC はすべて独立であるので，直列・並列または状態同期・非同期の制御は不可能である．同期実行の場合は，直列・並列と状態同期・非同期の組み合わせが存在する．

4. 実装

同期実行・直列実行・状態非同期の複合コンポーネントを実装した．複合化は，1) 子 RTC リストのセット，2) ポートの委譲処理，3) 子 RTC のデフォルトコンテキストの停止，4) 親 RTC の実行コンテキストを子 RTC へアタッチ，の順序で行われる．

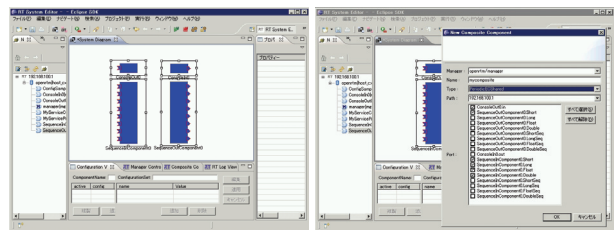


Fig.5 Selecting RTCs

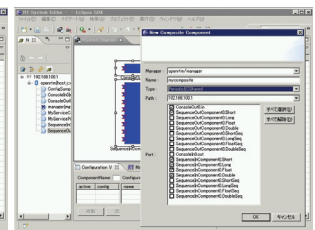


Fig.6 Selecting exported ports

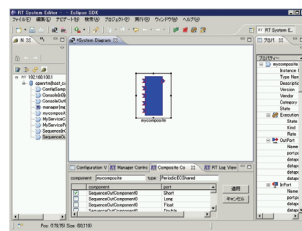


Fig.7 Created composite RTC

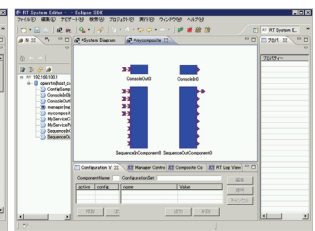


Fig.8 Inside composite RTC

コンポーネントの複合化は，複合コンポーネントの Organization に対して既存の RTC の参照をリストとして渡すことで行うことができる．しかしながら，これらの操作は CORBA (Common Object Request Broker Architecture) を熟知した上でプログラミングを行う必要があるため煩雑である．そこで，RTC 操作ツールである RTSystemEditor に，複合コンポーネント操作機能を追加し操作性向上を図った．

図 5~8 に RTSystemEditor の複合コンポーネント操作画面を示す．図 5 では複合化したい RTC を選択，図 6 では委譲するポートを選択，図 7 では複合化後の複合コンポーネントのアイコン，図 8 では複合コンポーネントの内部を表示している．なお，複合コンポーネントは，マネージャを拡張することにより，RTSystemEditor から動的に生成している．

このようなツールを利用することで，複合コンポーネントを利用した複雑なシステム構築が，より直感的に行うことが可能となる．

5. おわりに

本稿では，これら複合コンポーネントを用途と特性から分類し直し，新たなコンポーネントモデルに適用した．このモデルに基づき新たな複合コンポーネントフレームワークおよび，直感的な操作でコンポーネントの複合化を行うツールを実装し，その有用性を示した．今後は，すべての種類の複合コンポーネントを実現し，種々のシステムに適用するとともに，性能の評価を行う．

謝辞

本研究の一部は平成 20 年度，NEDO 次世代ロボット知能化技術開発プロジェクト（平成 19~23 年度）の補助を受けた．

- [1] Noriaki ANDO, et al., "Composite Component Framework for RT-Middleware (Robot Technology Middleware)", 2005 IEEE/ASME AIM2005, pp.1330-1335, 2005.07
- [2] 稲村 他, "RT ミドルウェアの性能評価 複合コンポーネントによるリアルタイム性能の改善", 計測自動制御学会 SI2006, p.3B2-7, 2006.12
- [3] 清水 他, "OpenRTM-aist-0.4.2 に準拠した複合コンポーネントフレームワークの開発", 第 9 回計測自動制御学会 SI2008, pp.863-864 (2L1-2), 2008.12
- [4] Object Management Group, Robotic Technology Component Specification Version 1.0, formal/2008-04-04 (2008)