

(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 System Integration Division
Annual Conference

SICE
SI 2005

講演概要集

SICE[®]

2005. 12. 16-18

熊本電波工業高等専門学校

RT コンポーネント・サービスフレームワーク

—RT ミドルウェアの基本機能に関する研究開発 (その 15)—

安藤慶昭 (産総研), ルメア オリビエ (産総研), 北垣高成 (産総研)
 神徳徹雄 (産総研), 末廣尚士 (産総研)

RT-Component Service Framework

— R & D of RT Middleware Fundamental Functions (Part 10) —

*Noriaki ANDO, Olivier Lemaire, Kosei KITAGAKI, Tetsuo KOTOKU and Takashi SUEHIRO
 National Institute of Advanced Industrial Science and Technology (AIST)

Abstract— In former RT-Middleware object model, there was no standard interface and framework to change internal status of RT-Component except InPort/OutPort and basic RT-Component interfaces. In this paper, we propose new concept “RT-Component Service”, which make adding user defined objects and interfaces to RT-Component possible. New framework for RT-Component service is also introduced.

Key Words: RT(Robot Technology), software component, middleware, robot system

1. はじめに

RT ミドルウェアでは、ソフトウェア開発手法としてコンポーネント指向ソフトウェア開発 (Component Based Software Development: CBSD) の考え方を取り入れ、基本単位となるソフトウェア部品: 「RT コンポーネント」の組み合わせによりシステムを構築する。著者らは、これまで RT ミドルウェアの中核部分である RT コンポーネントのアーキテクチャを提案してきた [1]。本稿では、RT コンポーネントにユーザ定義のオペレーションを追加する方法としての、RT コンポーネントサービスを提案する。

2. RT コンポーネント

著者らはこれまで、RT コンポーネントのオブジェクトモデルとして、図 1 に示す、InPort/OutPort、アクティビティ及び RT コンポーネント基本コマンドインターフェースからなるモデルを提案してきた。

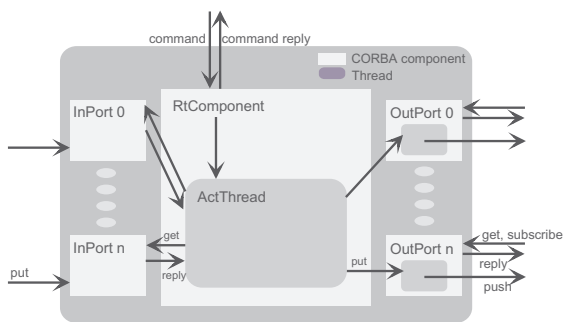


Fig.1 The architecture of RT-Component

コンポーネントは再利用性向上のために、再コンパイルすることなく、内部の詳細なパラメータなどを外部から変更するための十分なインターフェースを持たせる必要がある。

3. RT コンポーネントサービス

ここで、ロボットマニピュレータをコンポーネント化することを考える。手先の位置指令、速度指令などは

InPort によって与えることができるが、マニピュレータの制御モードの変更、あるいは制御ゲインの変更等は、オペレーションによって操作するのが自然である。CORBA IDL でこうしたインターフェースを記述すると、図 2 のように記述することができる。

```
interface Manipulator
{
    void setMode(MMMode);
    MMMode getMode();
    void setPosCtrlGain(in int axis,
                       in PIDGain gain);
    PIDGain getPosCtrlGain(in int axis);
    void setVelCtrlGain(in double axis,
                       in PIDGain gain);
    PIDGain getVelCtrlGain(in int axis);
};
```

Fig.2 IDL definition of “MyManipulator” service.

3.1 問題点

こうしたインターフェースをコンポーネントに付加する場合、継承またはコンポジション (または集約) によりこのインターフェースを持たせる方法が考えられる。しかしながら、RT コンポーネントにおいて、追加インターフェースを継承により持たせることには、

1. 継承関係が複雑になり全体の把握が困難になる。
2. 多重継承を許さない処理系がある。
3. シグニチャが衝突する可能性がある。

等いくつかの問題点がある。

継承本来の意味に鑑みて、“MyManipulator Component” は RtComponent の一種 (is-a の関係) であるが、“A Manipulator” や “A Gripper” の一種、すなわち意味的に内包しているのではなく、単に機能を継承しているだけに過ぎない。したがって、図 5 に示すようにコンポジション (has-a) 関係であらわす方が自然である。

こうした、コンポーネントに付随する「機能」を提供する実体を RT ミドルウェアにおいては、RT コンポーネント・サービス (RTCS: RT-Component Service) と

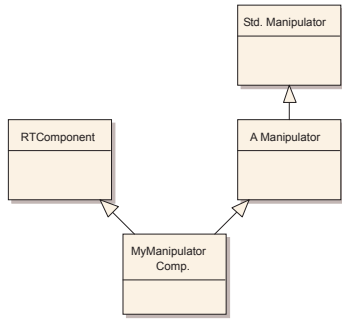


Fig.3 A Service realized by inheritance.

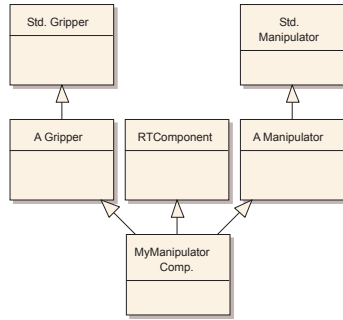


Fig.4 Services realized by multiple inheritance.

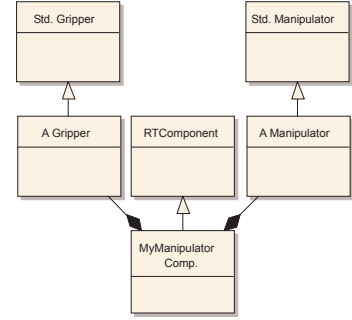


Fig.5 Services realized by object composition.

```
interface RTComponent {
    :
    ServiceProfileList get_service_profiles();
    ServiceProfile get_service_profile(in string id);
    RTCService get_service(in string id);
};
```

Fig.6 New Interface of RT-Component for RTCS.

呼ぶ。サービスをコンポジションにより RTComponent に持たせるため、サービスあるいはそのプロフィールを外部から取得するインターフェースを図 6 のように定めた。

4. OpenRTM-aist におけるサービスのためのフレームワーク

上記の RTCS のためのインターフェースに従い、実装 OpenRTM-aist において、以下のようなフレームワークを提供する。コンポーネントデベロッパは、既存のソフトウェア資産の関数・クラス等を RT コンポーネントサービスとしてコンポーネントに持たせることで、分散オブジェクトとして外部から自由にその機能にアクセスすることができるようになる。

4.1 サービスの登録

OpenRTM-aist では、ユーザ定義のインターフェースは図 2 に示すように CORBA IDL で定義される。図 2 のインターフェース定義に基づいて、サービスを実装し、コンポーネントのコンストラクタにおいてメソッド registerService() により登録することができる。

5. コードジェネレータ: rtc-template

上述したサービスの実装コードのメソッド定義および、コンポーネントにおけるサービスの登録は、ほぼ定型な処理であり生成を自動化できる。OpenRTM-aist において提供されている rtc-template を拡張しサービスをコンポーネントに対して付加するコードの生成を自動化した。コンポーネント開発者は、サービスの内部を実装だけで、コンポーネントに必要なサービスを付加できる。

図 7 に示すように、RT コンポーネントはコンポーネントの基本仕様 (specification) 及び、サービス定義 (service definition) からコンポーネント全体の仕様が一意に決まる。rtc-template は、specification 及び service definition からコードの雛形を生成する。これらをビル

ドすることにより、単一の実行ファイルおよびロードブルモジュールとしてのコンポーネントが生成される。

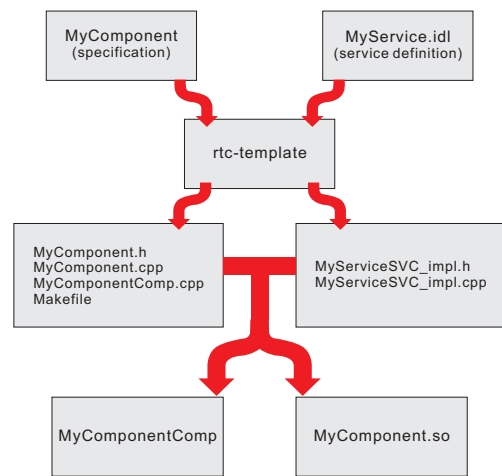


Fig.7 RT-Component service development flow by using rtc-template.

6. おわりに

本稿では、RT コンポーネントにユーザ定義のオペレーションを追加する方法としての、RT コンポーネントサービスを提供することにより、コンポーネントを単なる入出力の関係だけでなく、より柔軟な構成によりシステムを構築することが可能になった。また、著者らが提供する実装 “OpenRTM-aist” における RT コンポーネントサービスのための新たなフレームワークを示した。雛形コードを生成する新たな rtc-template により、コンポーネントに対するサービスの追加の定型な処理に関してほぼ自動化されることを示した。今後は、サービスを持つコンポーネントを柔軟に組み合わせる標準的な方法を提供する予定である。

参考文献

- [1] 安藤, 末廣, 北垣, 神徳, 尹, 「RT 要素のモジュール化および RT コンポーネントの実装」, ロボティックシンポジウム予稿集, pp.288-293, 2004.
- [2] 北垣, 末廣, 神徳, 平井, 谷江: “RT ミドルウェア技術基盤の研究開発について -ロボット機能発現のために必要な要素技術開発-”, ロボティックシンポジウム予稿集, pp.487-492, 2003.