

No. 04-4

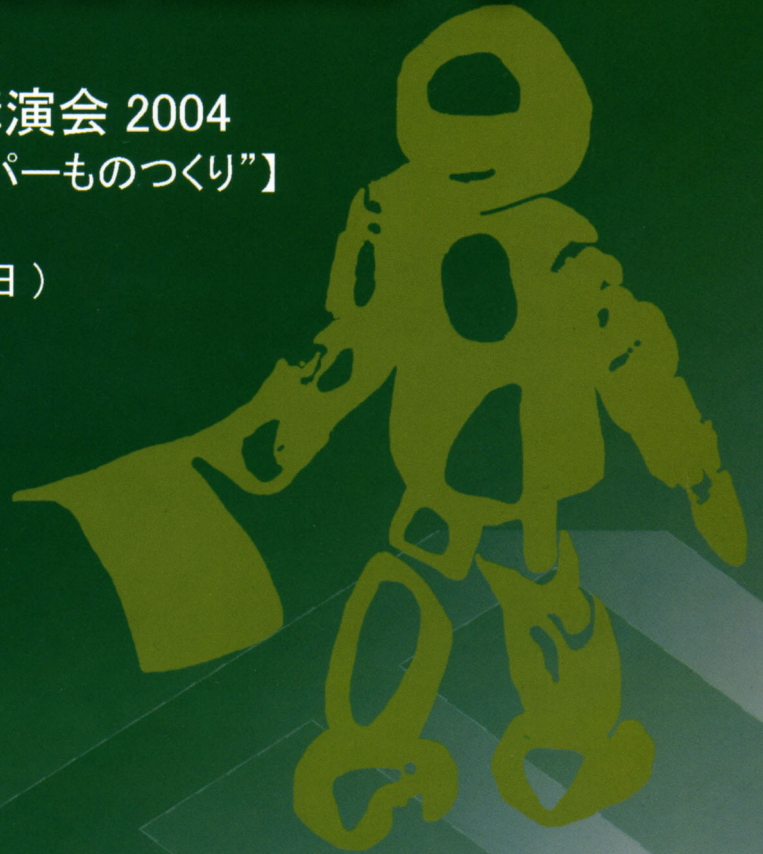
# ROBOMECH'04

2004 JSME Conference on Robotics and Mechatronics

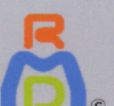
ロボティクス・メカトロニクス講演会 2004  
【安心・安全な環境のための“スーパーものづくり”】

2004年6月18日(金)～20日(日)  
名城大学(名古屋市天白区)

講演概要集



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





# RTコンポーネントによるマスタアーム制御システム構築

- RTミドルウェアの基本機能に関する研究開発（その6） -

Development of Master Arm Control System with RT Components

- R&D of RT Middleware Fundamental Functions (Part 6) -

○正 尹祐根 末廣尚士 北垣高成 神徳徹雄 安藤慶昭（産総研）

Woo-Keun YOON, Takashi SUEHIRO, Kosei KITAGAKI, Tetsuo KOTOKU, Noriaki ANDO

Intelligent Systems Institute,

National Institute of Advanced Industrial Science and Technology (AIST),

1-1-1 Umezono, Tsukuba 305-8568, Japan

**Abstract:** RT middleware project of the METI has been started since 2002. Middleware for the robot system is necessary to establish the modularization. In this project, we proposed a basic framework of the modularization of robotic functional elements as “RT component”. In this paper, we introduce an application for a master arm control system with RT components.

**Key Words:** RT (Robot Technology), Software component, Middleware, Master arm

## 1. はじめに

経済産業省のプロジェクト「ロボット機能実現のために必要な要素技術開発」が平成14年度から3年計画で実施されている(1)~(5)。

本プロジェクトの目的は、ロボットシステムを構成するロボット要素技術 (Robot Technology, RT) をソフトウェア的にモジュール化し、それを構成部品として自由に組み合わせることにより、新しい機能を持ったRTシステムを用意に構築可能とするための基礎技術を確立することにある。そこで、筆者らはロボット要素の部品化および再利用化を促進するRTミドルウェアを提案し、開発を進めている。

RTミドルウェアは、既存のロボット技術を部品化し、再利用を促進するための、

- コンポーネントフレームワーク、
- 標準的に再利用されるソフトウェア部品群、
- ライブラリ群、
- 標準サービス群、

などから構成される。

本論文では、これまで提案してきたコンポーネントフレームワークに基づき構築されたマスタアーム制御システムを紹介する。

## 2. RTソフトウェアコンポーネント

分散オブジェクトとは、IDLなどで記述された明示的なインタフェースを通じてのみクライアントにサービスを実施するサーバオブジェクトである。さらに、コマンドとして送信されてきたメッセージに対して応答するというパッシブな動作の側面のみをサポートし、そのオブジェクト内での処理については、基本的には立ち入らない。

これに対して、ロボット要素の多くは（サーボなど）、それ自身のタスクを保持するだけでなく、必要なデータを自ら収集

したり、イベントの発生などの通知を実施したりなど、固有の処理（アクティビティ）を持っている。その固有の処理の中では、自身のインタフェースで要求されるデータを随時書き換えたり、他のRTモジュールヘクライアントとして要求を出したり、などが実行されることになる。

ロボットの要素を部品として再利用できる形でモジュール化するためには、上記の機能をひとまとまりとした単位で考える必要がある。そこで、筆者らは、これをオブジェクトと区別して、RTソフトウェアコンポーネント（以下RTコンポーネントと略す）と呼び、ロボット要素のモジュール化の単位とすることを提案している。

## 3. CORBAオブジェクトで構成されるRTコンポーネント

ここでは、コンポーネント自身がアクティブに動作し、相互に通信しながらシステムの機能を実現するタイプのモジュール化を考える。ひとつのRTコンポーネントはFig. 1に示す3種類のCORBAオブジェクトから構成される。

- RtComponent オブジェクト

RTコンポーネントの本体。コンポーネントのコマンドインタフェースはここで定義される。RtComponentでは、様々なコンポーネントに共通なものを定義する。個々のカテゴリーのコンポーネントはRtComponentを継承して作られる。ただし、複雑な多重継承をさけるために実装の方はRtComponentの実装を継承せずに個別にする方が良い。実装部はアクティビティ部を別threadとして持つ。入力ポート、出力ポートは、各々InPort, OutPortの“実装”を内部で生成することで実現する。それらもまた別threadで実行される。

- InPort オブジェクト

RTコンポーネントの入力ポート。IDLで公開されたput操作により外部からデータを受け取る。受け取った後の

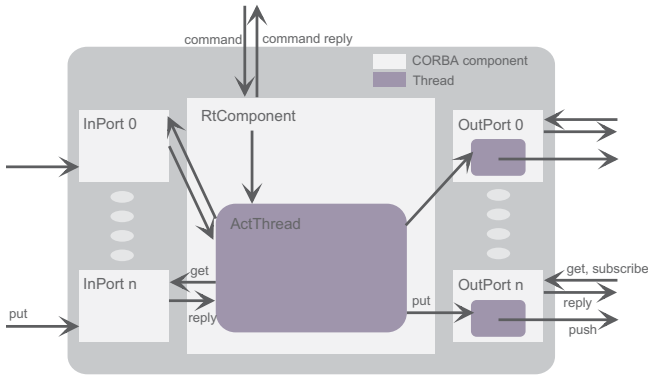


Fig. 1. RT component.

処理はコンポーネントごとに異なるが数種類のパターンに別けられると考えている。これらの違いは、それぞれの“実装”を用意することで吸収する。

- OutPort オブジェクト

RT コンポーネントの出力ポート。IDLで公開された get 操作により外部からの pull 型アクセスを受け付ける。また、InPort の object reference を引数にして subscribe することにより、指定された InPort に対してアクティビティ部でのデータ生成に応じて push 型のデータ出力を行う。

これらの IDL 定義を以下に示す。入出力ポートは、any 型のデータを用いることで IDL 定義を各々ひとつずつすむようにした。通信や処理の効率を考えると any 型でない方が良いが、その場合、ポートのデータ型に応じて異なる IDL 定義が必要となる。

```
enum ReqType {once,repeated};
//
interface InPort {
    void put(in any value);
    readonly attribute string name;
    readonly attribute CORBA::TypeCode port_type;
};
//
interface OutPort{
    any get();
    boolean subscribe(in ReqType r_type, in InPort i_port);
    boolean unsubscribe(in ReqType r_type, in InPort i_port);
    readonly attribute string name;
    readonly attribute CORBA::TypeCode port_type;
};
//
typedef sequence<InPort> InPortList;
typedef sequence<OutPort> OutPortList;
//
interface RtComponent {
    void on();
    void off();
    InPortList in_ports();
    OutPortList out_ports();
};
```

#### 4. マスタアーム制御システム

RT コンポーネントを利用したマスタアーム制御システムのひとつとして、シンプルなバイラテラルシステムを構築した。基本要素コンポーネントは、

- マスタアーム RT コンポーネント (マスタアーム RTC)

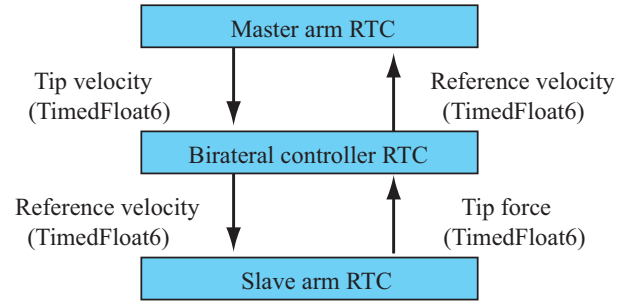


Fig. 2. Data flow of a birateral system.

- バイラテラルコントローラ RT コンポーネント (バイラテラルコントローラ RTC)
- スレーブアーム RT コンポーネント (スレーブアーム RTC)

である。

Fig. 2 に基本要素コンポーネント間のデータフローを示す。マスタアームコンポーネントは、手先速度指令制御により動作し、入力の手先速度指令値であり、出力は手先速度指令値である。また、オペレータがマスタアームに力を加えた場合、その力に応じてマスタアームは動作するため、入力される手先速度指令値が零の場合、出力される手先速度指令値はオペレータがマスタアームに加えた力に応じて生成される手先速度指令値となる。

バイラテラルコンポーネントは、マスタアームから入力される手先速度指令値にゲインをかけスレーブアームの手先速度指令値を出力し、スレーブアームから入力される手先力にゲインをかけマスタアームの手先速度指令値を出力する。

スレーブアームコンポーネントは、手先速度指令制御により動作し、入力の手先速度指令値であり、出力は手先力である。

本システムでの入出力データ型は TimedFloat6 と呼び、int 型のタイムスタンプおよび 6 個の float 型から構成される。

なお、コンポーネントの実装については、北垣らの発表<sup>(6)</sup>と内容が重複するため、本稿では省略する。

#### 5. RT コンポーネントの実行

本システムでは、まずマスタアーム RTC、バイラテラルコントローラ RTC、スレーブアーム RTC、コンポーネントを起動する。次にコンポーネントエディタなどで、各コンポーネント間を接続する。最後に、バイラテラルコントローラ RTC、マスタアーム RTC、スレーブアーム RTC の順にオペレーション on を発行することにより、システムが起動する。

なお、全てのコンポーネントは 1 [ms] 周期で動作しており、リアルタイムの実現には ARTLinux を採用した。

#### 6. 結言

RT コンポーネントを利用したマスタアーム制御システムの構築例を、簡単なバイラテラルシステムにより紹介した。

各システム要素をコンポーネント化し、さらに必要に応じて新しいコンポーネントを開発、接続することにより、様々な機能を有するロボットシステムを簡単に構築することが可能となる。例えば、本稿では、手先速度指令値によるバイラテラルシステムを構築したが、手先位置指令値によるバイラテラルシステムも容易に構築可能であり、さらに、通信時間遅れコンポー

ネットを開発，接続することにより，通信時間遅れ環境下でのバイラテラルシステムも構築可能となる．

今後の課題としては，北垣らの発表<sup>(6)</sup>と重複するが，以下に述べる．

- 各コンポーネントの入出力データ仕様各コンポーネント間のデータ仕様が統一されていれば，コンポーネント間の接続が容易となるが，汎用性は低下する．そのため，マスタアームやスレーブアームの入出力データ仕様を複数用意する必要がある．
- 複数コンポーネントに対するリアルタイム性バイラテラル制御などを対象とする制御システムの場合，システムのリアルタイム性が非常に重要となる．個々のコンポーネントをリアルタイム化するだけでなく，今後ネットワークで接続された複数コンポーネントを同期させ，リアルタイム化を実現する必要がある．

なお，本研究は新エネルギー・産業技術総合開発機構（NEDO）の「ロボットの開発基盤となるソフトウェア上の基盤整備」の委託を受けて実施されたものである．

## 参考文献

- (1) 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，第8回ロボティクスシンポジウム，pp. 487-492, 2003.
- (2) 末廣，北垣，神徳，尹，安藤，RT要素のモジュール化に関する検討-ロボット機能実現のために必要な要素技術開発（その1）-，第15回ロボット学会学術講演会，1F27, 2003．
- (3) 末廣，北垣，神徳，尹，安藤，RTコンポーネントの実装例-RTミドルウェアの基本機能に関する研究開発（その2）-，第15回ロボット学会学術講演会，1F28, 2003．
- (4) 神徳，北垣，安藤，尹，末廣，RTミドルウェアのソフトウェア開発支援機能の検討，第9回ロボティクスシンポジウム，pp. 282-287, 2004．
- (5) 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，第9回ロボティクスシンポジウム，pp. 288-293, 2004．
- (6) 北垣，末廣，神徳，尹，安藤，RTコンポーネントによるマニピュレータ制御システム構築-RTミドルウェアの基本機能に関する研究開発（その5）-，日本機械学会 ロボティクス・メカトロニクス講演会'04，2004（講演予定）．