The 25th Annual Conference of
the Robotics Society of Japan

RSJ

第25回
日本ロボット学会
学術講演会

講演概要集

会期 ▶ 2007年9月13～15日
会場 ▶ 千葉工業大学津田沼キャンパス

主催 ▶ (社)日本ロボット学会

分散コンポーネント型ロボットシミュレータ OpenHRP3
–RT コンポーネントを用いた実機と可換な制御ソフトウェア開発機能–

○安藤慶昭　金広文男　中岡慎一郎　神徳徹雄　末廣尚士　比留川博久 (産業技術総合研究所)

OpenHRP3: Distributed Component Based Robot Simulator
–Concurrent Controller Module Development Environment Based on RT-Component–

*Noriaki ANDO, Fumio KANEHIRO, Shin'ichiro NAKAOKA,
Tetsuo KOTOKU, Takashi SUEHIRO, Hirohisa HIRUKAWA(AIST)

Abstract—A distributed component based dynamics simulator OpenHRP3 has been developed for efficient and systematic robot system development. OpenHRP3 provides seamless controller development environment between simulation and real system. Once a developer implements a controller as a RT-Component and tests it on the simulator, the identical controller module can be executed on the real system without being re-compiled. To realize this functional capability, ExecutionContext that is the execution entity object of the RT-Component is extended. In this paper, we will show the architecture of the OpenHRP3 dynamics simulator based on RT-Middleware and RT-Component with new ExecutionContext.

Key Words: OpenHRP, OpenRTM, RT ミドルウェア

1. はじめに

筆者らはソフトウエアのモジュール化と再利用を促進し，次世代ロボットの開発を効率化することを目的として、RT ミドルウェア [1] を基盤とした分散コンポーネント型ロボットシミュレータ OpenHRP3[2] の開発を行っている。

OpenHRP3 では RT コンポーネントとして実装されたロボットの制御モジュールをシミュレータと接続して検証し、検証が済んだモジュールを再コンパイルすることなしに実機に適用できる開発機能を提供している。

本稿ではこの実機と可換な制御ソフトウェア開発機能を実現するために RT ミドルウェアに対して行った拡張と、それを用いて実現した OpenHRP3 の制御ソフトウェア開発機能について述べる。

2. RT コンポーネント

RT ミドルウエアは RT コンポーネントと呼ばれるモジュールの組み合わせにより、ロボットシステムを効率よく構築するためのソフトウエアプラットフォームである。RT コンポーネントは、i) メタ情報取得のための共通インターフェース, ii) ロジックを実行するアクティビティ、iii) データ送受信を行うためのデータポート (InPort, OutPort)、iv) インターフェース経由の相互作用を行うためのサービスポート (Provider, Consumer) を備えたフレームワークに基づき作成されるソフトウエアコンポーネントである [1]。

2·1 実行コンテキスト

通常のプログラムは、起動されるとプロセスのメインスレッドや、プロセス内で生成されたスレッドにより処理が実行される。

これに対して RT コンポーネントでは、スレッドを抽象的に表現した実行コンテキスト (ExecutionContext) と呼ばれるオブジェクトがスレッドに相当する実行主体となる。実行コンテキストは実行周期や状態を持ち、状態に応じて RT コンポーネントのコアロジックに実装されたコールバック関数を呼ぶことで、主たる処理 (例えばロボットの制御等) が行われる。

2·2 実行コンテキストの動的関連付け

RT コンポーネントが生成されると、通常一つの実行コンテキストが RT コンポーネントに関連付けられ、コアロジックの実行が開始される。RT ミドルウエアでは、図 1 に示すように、動的リンク可能なモジュールとして作成された複数種類の実行コンテキストを、コンポーネント生成時に選択的に関連付ける (バインディングする) ことができる。

例えば、リアルタイム実行がサポートされない通常の Linux 上でコンパイルされた RT コンポーネントを、ARTLINUX のようなリアルタイム実行をサポートする環境で実行する際に、実行コンテキストモジュールの切り替えを行うだけで、リアルタイム実行を行うことが可能である。動的リンク機構を用いているため、RT ミドルウエアおよび RT コンポーネント共に、再コンパイルすることなくこの機能を利用することができる。

3. シミュレータ用実行コンテキスト

現実のシステムでは、各コンポーネントにおける時間の進み方は、現実の時間の進み方と同じであるが、シミュレータにおいては、シミュレータを構成する全てのコンポーネントはシミュレータが管理する時間 (= シミュレータ時間) で動作する必要がある。そこで、Figure 2 に示すように、同一のコンポーネントで現実時間での動作とシミュレータ時間での動作をシームレスに実現するための実行コンテキストの拡張を行った。

Figure 3 に示す CORBA IDL (Interface Definition Language) 定義のように、現実時間で動作させる際に使用する実行コンテキストの標準インターフェースを、

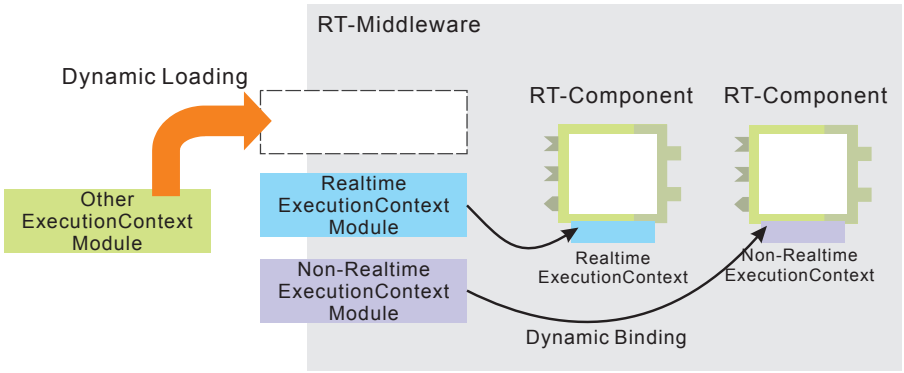

Fig.1 Dynamic Loading and Binding of ExecutionContext.

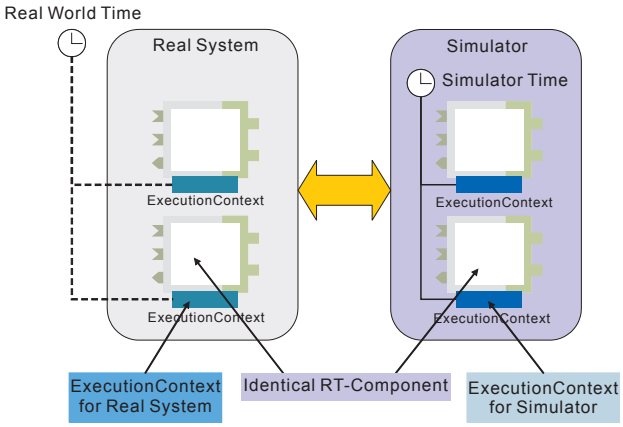

Fig.2 Extended ExecutionContext for Simulator.

```
interface ExtTrigExecutionContextService
 : ExecutionContextService
{
  void tick();
};
```

Fig.3 Extended Interface of ExecutionContext

シミュレータ用に拡張し、シミュレータが設定する刻み幅で時刻を進める為のオペレーション (`tick()`) を追加した実行コンテキストを作成した。

現実の時間刻みでロジックを駆動する代わりに、シミュレータが供給するクロック信号 (=`tick()` オペレーション呼び出し) により、関係する RT コンポーネントのロジックが駆動される。これにより、同一の RT コンポーネントを内部の変更や再コンパイルを行うことなく、実システムとシミュレータシステムの両方に利用することができる。

4. RT ミドルウエアを用いたコントローラの分散コンポーネント化

シミュレータ内部は大きく分けて、順動力学計算や視野画像生成を行うシミュレーションエンジン部とロボットの制御量を計算するコントローラ部、それらの動作タイミングをシミュレーション世界の時間に合わせて制御するスケジューラ部の 3 つ部分からなる。これら 3 つの部分の呼び出し関係と処理の流れを Figure 4 に示す。

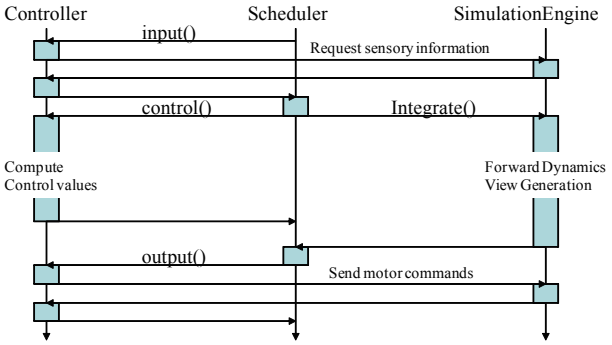

Fig.4 Inside of OpenHRP3

コントローラ部は Fig.5 に示す IDL によって定義されるインタフェースを持っており、これらのインタフェースはスケジューラ部によって制御周期に一度呼び出される。

input シミュレーションエンジン部から制御計算を行うのに必要なセンサ情報を取得する

control センサ情報と内部状態に基づいてロボットの制御量を計算する

output 計算した制御量をシミュレーションエンジン部に送る

```
interface Controller
{
  // 略
  void control();
  void input();
  void output();
  // 略
};
```

Fig.5 Controller Interface Definition.

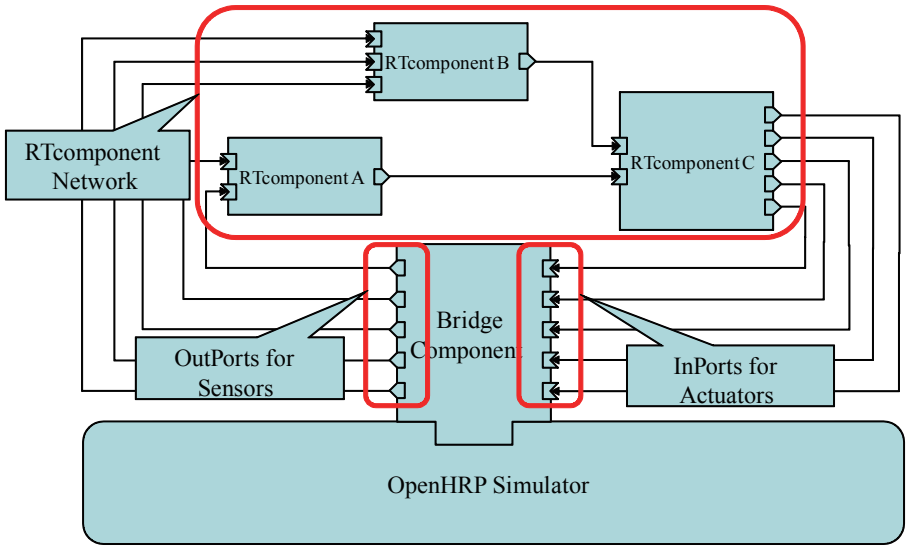

Fig.6 Configuration of RT-Component based OpenHRP3.

RT コンポーネントとして実装された制御モジュールをシミュレーション世界で動作させるためにはこのコントローラ部に RT コンポーネントを接続する必要がある。そこでコントローラ部のインタフェースの実装を持ち、ロボットに搭載されたセンサに対応する OutPort とアクチュエータに対応する InPort を備えた RT コンポーネントを開発した。これをブリッジコンポーネントと呼ぶ。ブリッジコンポーネントではコントローラ部のインタフェースをそれぞれ以下のように実装している。

input シミュレーションエンジン部からセンサ情報を取得し、それらを OutPort に設定する。

control 制御計算を行う RT コンポーネントでは外部トリガ付きの実行コンテキストを用い，このコンテキストの `tick()` オペレーションを呼び出して制御計算を 1 度だけ実行する。

output InPort から制御計算の結果を読み出し、動力学計算サーバに設定する。

ブリッジコンポーネントを用いて制御用の RT コンポーネントを OpenHRP3 に接続した際の構成を Figure 6 に示す。

この構成を用いて検証を行った RT コンポーネントは以下の 2 つの手順を行うことでそのまま実機に搭載可能である。

実行コンテキストの切り替え 実機での RT コンポーネントの実行制御は実時間 OS の周期実行機能を用いて行われるため、実行コンテキストを実時間 OS 用のものに切り替え、その周期をシミュレーション時の制御周期と合わせる。前述の通り、この切り替えは動的に可能であり、コンポーネントの再コンパイル等は不要である。

I/O コンポーネントの開発 ハードウェアからのセンサ情報を読み出しとハードウェアへの指令値の設定を行う I/O コンポーネントを開発し、これをブリッジコンポーネントと置き換える。

5. まとめ

本稿では、実機とシミュレータ間で可換な制御ソフトウェア開発環境を提供する RT ミドルウェアを基盤とした分散コンポーネント型ロボットシミュレータ OpenHRP3 のアーキテクチャを示した。RT コンポーネントのロジックの実行を行う実行コンテキストに対して拡張を行い、それを用いて実現した OpenHRP3 の制御ソフトウェア開発機能について述べた。

ロボットコントローラの開発者は、一旦コントローラを RT コンポーネントとして開発し、シミュレータ上で検証し動作することを確認すれば、同一のコントローラ RT コンポーネントを実機上でも再コンパイルすることなく動作させることが可能となる。これにより、コントローラコンポーネントの作成、シミュレータによる検証および実機での実行をシームレスに行うことができ、ロボット開発の効率化が期待できる。

謝辞

本研究は、文部科学省の平成１７年度科学技術振興調整費による「科学技術連携施策群の効果的・効率的な推進」の一環として実施したものである。ここに感謝の意を表する。

参考文献

[1] Noriaki ANDO et al., “RT-Middleware: Distributed Component Middleware for RT (Robot Technology)”, 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005), pp.3555-3560, 2005

[2] 比留川博久, 金広文男, 中岡慎一郎, 末廣尚士, 神徳徹雄, 安藤慶昭、中村仁彦, 山根克、齋藤元, 川角祐一郎: “分散コンポーネント型ロボットシミュレータ OpenHRP3”, 第 25 回日本ロボット学会学術講演会, 2007.