

rtshell 3.0: RT ミドルウェア用コマンドラインツール

rtshell 3.0: Command-line tools for RT-Middleware

○ BIGGS Geoffrey (産総研) 松坂 要佐 (産総研) 安藤 慶昭 (産総研) 神徳 徹雄 (産総研)

Geoffrey BIGGS, National Institute of Advanced Industrial Science and Technology
Yosuke MATSUSAKA, National Institute of Advanced Industrial Science and Technology
Noriaki ANDO, National Institute of Advanced Industrial Science and Technology
Tetsuo KOTOKU, National Institute of Advanced Industrial Science and Technology

RT-Middleware is a framework for developing component-based software. A set of command-line tools, known previously as rtschell and rtsshell, are provided for managing RT-Components and RT-Systems from a command line, in order to allow low-resource use and scriptability. We report on new commands recently added to this toolkit that facilitate the debugging of RT-Components, visualisation and verification of RT-Systems, and managing documentation.

Key Words: RT ミドルウェア、RT コンポーネント

| | | | | |
|-----|-------------|----------|---------|------------|
| 3.0 | rtinject | rtexit | rtcheck | rtstodot |
| | rtprint | rtlog | rtdoc | rtcomp |
| 2.0 | rtdel | rtinject | rtprint | |
| | rtact | rtcat | rtcon | rtconf |
| | rtcryo | rtcwd | rtdeact | rtdis |
| 1.0 | rtfind | rtls | rtmgr | rtreset |
| | rtstart | | | rtstop |
| | rtresurrect | | | rtteardown |

Fig.1: rtshell の各バージョンに導入されたコマンド。バージョン 3.0 で rtinject と rtprint は全体作り治った。

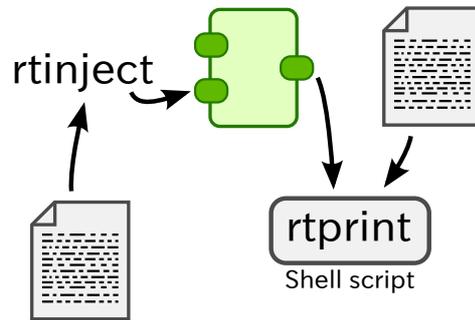


Fig.2: rtinject と rtprint によるコンポーネントテスト

1. はじめに

現在、高度なサービスロボットの開発をより効率的に行うため、OpenRTM-aist をはじめとするソフトウェアプラットフォームの研究開発が盛んに行われている。一般にこうしたソフトウェアプラットフォームには、モジュール化のためのフレームワーク、通信ミドルウェア、開発ツールやライブラリ等が含まれる。

著者らは、これまでに、コマンドラインからの RT コンポーネントを管理するためのツールセットとして、rtschell と rtsshell[1] の研究開発を行ってきた。これらのツールにより、省リソース、スクリプトによるタスクの自動化、および単純なツール群により機能を提供を実現することができた。

ツールセットは以下のようなコマンドセットを提供する。

- RT コンポーネントの activation および deactivation
- ポートの接続および切断
- マネージャを利用したコンポーネントのインスタンス化等
- 一つのコマンドによる全 RT システムを復元・開始

本稿では、rtschell の新たなバージョンとして、以下の重要な機能を付加し、コマンドとして追加する。

- コンポーネント入力および出力のデバッグ
- コンポーネント間で送受信されるデータの記録と再生
- RT システムの可視化
- RT コンポーネントのドキュメンテーションの管理

本稿では、これらの新しい機能について解説し、それらがロボット・ソフトウェア開発にもたらす利点について述べる。新しいツールは図 1 に表紙される。

2. RT コンポーネントのデバッグ

一般的なデータ駆動型 RT コンポーネントは、入力ポートと出力ポートを持ち、あるデータが入力ポートに入力された

場合、出力ポートには内部ロジックに応じた特定のデータを出力する。また、入力ポートのみ、あるいは出力ポートのみを持つ RT コンポーネントもありうる。こうしたコンポーネントのデバッグにおいては、(もし存在するならば) 入力ポートへあるデータを入力し、出力ポート (あるいは出力ポートを持たずに、実際のモータやハードウェア信号の出力) から出力されるデータを監視することになる。

こうしたコンポーネントをデバッグする際には、以下の困難さが生ずる:

- 入力ポートへテストデータの入力
- 出力ポートからデータの読み込みとチェック

一般的には、これらのデータを処理するためにファサード・コンポーネントを作成する必要があるが、一つのコンポーネントのデバッグのために新たな別のコンポーネントを作成するには多大な労力が必要となる。rtshell はこうした形式のデバッグを単純化する、rtinject および rtprint と呼ばれる 2 つのツールを提供する。

rtinject はコンポーネントの入力ポートへ固定値を送るために使用される。値はコマンドライン上で指定するか、コマンドの標準入力に与えることができる。こうしたツールは自動テスト時に有用である。出力値は Python 言語の表現として与えることができ、また、ユーザ定義型を持つデータポートに対しても柔軟に対応することが可能である。

rtprint は rtinject の逆で、出力ポートからデータを読み、標準出力 (一般にはターミナルに表示される) にそのデータを表示する。開発者はコンポーネントが予想通りにデータを生成していることを確認するために rtprint を使用することができる。さらに、複雑なデータ型を扱うためにカスタム・フォーマット機能を使用することができる。

rtshell は多数の小さなツール群により構成されているため、

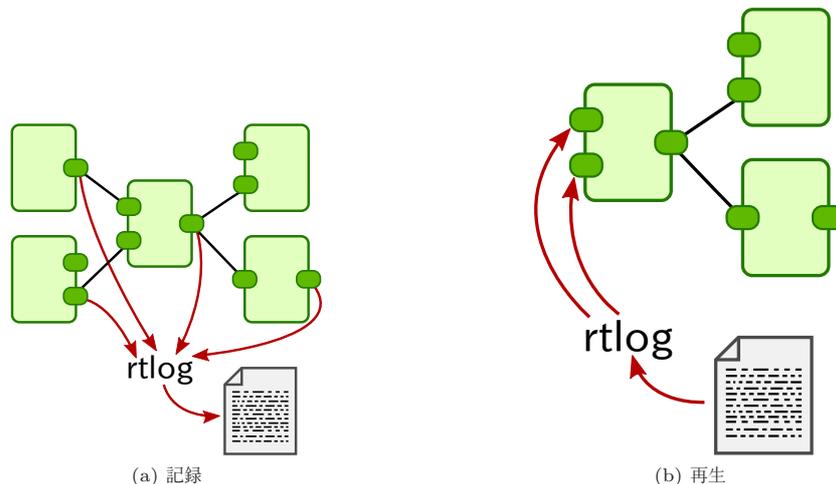


Fig.3: rtlog によるシステムのデータの記録、再生

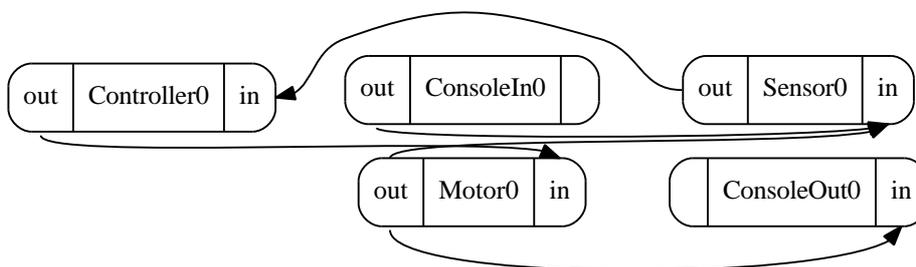


Fig.4: rtstodot の出力

例えば図 2 に示されるように、強力なリグレーションテストを行うために、2つのツールを組み合わせ使用することもできる。

3. RT コンポーネントのロギング

コンポーネントのデバッグとデータのロギングは表裏一体であり、後の分析およびシミュレーションのためにセンサー、アルゴリズムおよびドライバー・コンポーネントによって生産されたデータを記録することはロボット工学において一般的である。rtshell はこの機能性を提供する rtlog コマンドを提供する。rtlog は単一のファイルの中へ複数の任意の出力ポートからのデータを記録し、複数の任意の入力ポートに後でそれを再生することができる。図 3 は rtlog の使用イメージを示す。

データの記録時、rtlog はデータとともに出所や、データの各チャンネルのプロパティを記録する。一つのチャンネルは一つの出力ポートに相当する。再生時には、データの出力元と互換性をもつ入力ポートへデータが送られることを保証するために、記録時に格納された情報が利用される。また、記録されたすべてのチャンネルを再生するとは限らず、開発者は、任意のチャンネルを再生することができる。例えば、記録時には 3 つのレーザー・スキャナからのデータを保存し、再生時には、そのうち 1 つのレーザー・スキャナのデータを再生して利用することも可能である。

rtlog のもうひとつの重要な特徴は、再生時にデータの時間スケールを操作する機能である。開発者は時間インデックスを指定することで、例えば、記録開始から 30 秒から 60 秒の間のデータのみを再生するといった、ログデータのある期間のデータだけを再生することができる。さらに、記録されたデータを記録時よりもより速い速さで再生するといった、再生速度の変更も可能である。

4. RT システムの可視化

以前の rtshell ツール群では、基本的にテキストベースのインターフェースで使用するため、一度に 1 つのコンポーネントしか表示できず、多くの RT コンポーネントと接続からなるシステム構成を把握するのが困難であった。rtshell の新バージョンではこれを解決する rtstodot という新しいコマンドが追加された。

rtstodot は RT システム構成を記述する RTSPProfile を、Graphviz[2] で使用するためのドット・ファイル・フォーマットに変換する。rtstodot は他のツールからの主力を受けることができるように、標準入力から RTSPProfile を読み込む。例えば、一般的な使用方法として現在のシステム構成を視覚化するには以下のようにコマンドを実行する。

```
rtcryo | rtstodot | dot -T xlib
```

図 4 に可視化例を示す。このツールを使用すれば、RTSystemEditor のようにメモリを大量に消費することなく、ディスプレイ上で現在の RT システムを視覚的に把握することができる。

5. RT システムの検証

本稿で開発した rtshell は、以前のバージョン同様、完全な RT システムを管理するためのコマンドが含まれる。rtresurrect、rtstart といったコマンドは、保存された RTSPProfile から、完全な RT システムの接続および配置を復元させるために使用され、また、システムを開始、停止、さらにその状態をディスクに保存することができる。大規模なシステムを運用する場合、このようにして復元された動作中のシステムが、そのシステム構成を記述した RTSPProfile と同等であるかどうかを確認したい場合がある。すなわち、動作中のシステムのコンポーネントの構成、コンポーネントのエラーの有無、あるいはすべての接続が RTSPProfile に記述されたとおりであるかを確認する方法が必要となる。

rtcheck コマンドは格納された RTSPProfile に対して、実行中の RT システムの整合性を確認するために使用される。接続がすべて正確なプロパティとともに確立され、余分な接続がないか、あるいはすべてのコンポーネントが正常な状態であることを確認する。このコマンドは一般的には、RT システムを起動にする前の最終確認として使用される。

6. RT コンポーネントのドキュメント

RT コンポーネントの有用性を高めるためには、インターフェースやコンフィギュレーション・パラメータあるいは想定される振る舞いに関するドキュメントが十分に提供されることが重要である。

新たな rtshell のコマンド (rtdoc) は、ドキュメントをコンポーネント内に埋め込むことを可能にする。埋め込まれた情報は、rtdoc を用いてフォーマット済みのドキュメントとして取り出すことができる。開発者はコンポーネントを起動して、rtdoc を用いてドキュメントを取得し、使用方法を学習することができる。rtdoc を使用することで、常にドキュメントを RT コンポーネントとともに開発者に対して提供することができる。

7. おわりに

本稿では、RT ミドルウェア用のコマンドライン・ツール rtshell の新たなバージョンに含まれるいくつかのコマンドについて、新しい特徴のうちのいくつかを示した。RT コンポーネントおよび RT システムを作成する場合、これらの機能により開発効率を大きく向上させることができる。特に、コン

ポーネントのデバッグ、データのロギング機能を提供するツールにより、開発者が実際のデータをテストに使用することができるようになったため、コンポーネントの作成効率と信頼性が大きく向上させることができた。

データのロギングとデバッグツールはユニークではない。例えば、ROS[3] に似てるツールはある。RT システムの可視化ツールも、他のシステムに似てるツールはある。しかしながら、RT システムの検証ツール他のフレームワークにはない。システムの復元時、RTSPProfile と同等であるかどうかを確認することはロボスタシステムにとって重要なことである。OpenRTM-aist を利用するロボットはよりロボスタになることを支援する。ドキュメントツールは他の似てるツールと比べて情報をコンポーネントから得るからよりロボスタである。コンポーネント意外のファイルは必要ではないのでドキュメントはいつも得られると保証する。

謝辞

本研究の一部は平成 21 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。

参考文献

- [1] Geoffrey BIGGS, "rtshell: Command-line tools for OpenRTM-aist", 日本機械学会 ロボティクス・メカトロニクス講演会 2010, p.2P1-A23, 2010
- [2] Graphviz, <http://www.graphviz.org/>, 2011.
- [3] ROS Wiki, <http://www.ros.org>, 2011.