

RT ミドルウェアを利用した遠隔移動ロボットシステムの構築

System design for Tele-operated mobile robot based on RT-Middleware

○原佑輔 (中央大院) 國井康晴 (中央大) 安藤慶昭 (産総研)

*Yusuke Hara, Yasuharu KUNII, Noriaki Ando

Abstract—Construction of robot systems complicated at various functions, such as recognition, movement control, and an action plan, is required of realization of a remote mobile robot. However, the time and the labor which the method of building a system and development take pose a big problem. In this paper, the remote mobile robot architecture based on RT middleware is proposed.

Key Words: Remote mobile robot, Architecture, RTMiddleware

1. はじめに

近年、災害現場や極限環境で探査を行うロボットや、家事を行うお手伝いロボットなどの遠隔移動ロボットが注目されている。このようなロボットはセンサ、カメラ、マニピュレータ、移動機構など様々な機器を搭載し、認識や運動制御、行動計画などの多くのサブシステムから構成されるため、開発プロセスに莫大な時間を要する。また、遠隔作業においては、故障やパラメータ、アルゴリズム不適合など運用面のリスクを回避することが重要になる。

以上より本稿では開発プロセス、運用プロセスの要求を洗い出し、その要求を満たしたシステムアーキテクチャの検討を行い、探査ローバを対象にシステムを構築する。

2. アーキテクチャの検討

2.1 開発プロセス

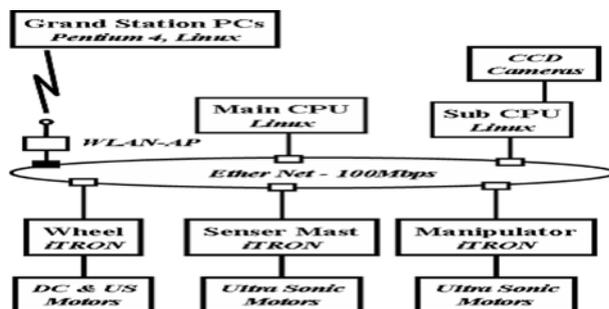
開発プロセスにおいては、複数の人間が世代を超えて開発を実施するため、ハードウェア、ソフトウェア共に研究開発資産の活用による労力、期間の効率化が求められる。このためシステムの

- インターフェースの共通化
- モジュール化

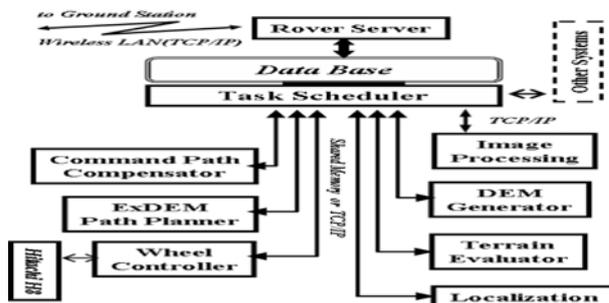
の導入を検討すべきである。

システムは粒度の違う様々なサブシステムにより構成され、有機的に接続されることで成立している。各サブシステムは複数の人間により開発され、各担当者が境界部分に関し共同して開発するため、各システム間の調整に時間を要し、試験などのために担当者間のスケジュール調整などが発生し効率が低下する。また、開発中様々な検討を実施するためのシステム変更が容易に行えることが効率化に繋がる。このため、モジュール化とインターフェースの共通化によるブラックボックス化による効率化を図る。また開発の効率化に加え、機能や負荷分散のためある程度の粒度のサブシステムに積極的にサブ CPU の配置を実施することとする。

一方、ソフトウェアにおいてもモジュール化により開発の規模が縮小され、複雑なシステムでもプログラ



(a) Hardware System Structure



(b) Software System Structure

Fig.1 System Architecture for Tele mobile-robot

ミングが行い易くなる。またインターフェースの共通化によるブラックボックス化により管理が容易になり、修正や拡張、再利用が容易になる。さらに複数の手法を実験で評価し、試行錯誤するプロセスが容易に実施可能になる。

2.2 運用プロセス

ロボットとオペレータが離れているというのが遠隔移動ロボットの最大の特徴である。遠隔移動ロボットを運用するには以下の点が重要になる。

- 再構築可能なシステム構成
- サブシステムの独立性
- サブシステムへの透過性

ロボットは遠隔地で作業をしているためネットワークを通して再構築可能なシステム構成にすることが重要である。また、障害を起こしても直接復旧にはいけな

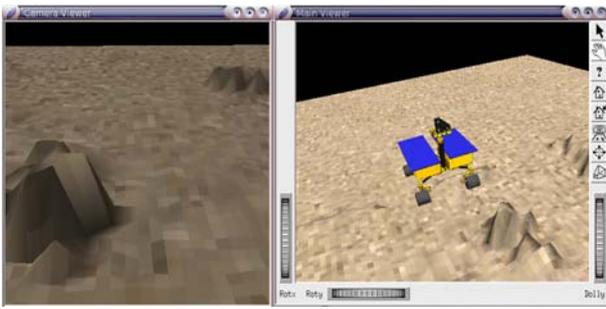


Fig.2 Simulator

いため、障害を最小限に抑え早急に原因を究明・復旧できるシステム体系にすることが必要である。1つのサブシステムが障害を起こしたときに障害がシステム全体に広がらないようにする独立性とシステム障害を起こしたときにオペレータが直接障害を起こしているモジュールにアクセスできる透過性が重要になる。

2.3 ネットワークを用いたアーキテクチャ

以上を踏まえて考えるとスター型のネットワークを用いたアーキテクチャが有効である。次のような利点がある。

- インターフェースの共通化を実現でき、再構築が容易に行えるシステムを構成できる（適応性）
- オペレータ又はモジュール同士が直接アクセス可能（柔軟性）
- 障害発生時の故障箇所の特特定が容易にでき、障害が発生したモジュールだけを切り離すことが可能（独立性）

実際にローバに適応した例を Fig.1 に示す。

3. RTM の適用

前節で提案したようなネットワークシステムでは様々な計算機・OS・ネットワークを利用する。よって、このシステムに実装される各ソフトウェアモジュールには特定の計算機や OS 及び特定のネットワークに限定されないことが要求される。また、システムの運用を考えるとオペレータが各ソフトウェアモジュールを一括管理でき、なおかつ外部から直接制御できることが望ましい。そこで今回は RT ミドルウェア [1][2] を導入することにした。RT ミドルウェアは上記の内容を全て満たすだけでなく、オブジェクトモデルやフレームワークが提供されており、既存のソフトウェアの移行が容易というメリットがある。また、RT コンポーネントとしてモジュール化しておけば資産管理・継承が容易になり、開発効率の向上が期待できる。

4. システムの構築

OpenRTM-aist-0.2.0 付属の rtc-template を使用し、Camera, Pantilt, ExDEM Path Planner, Terrain Evaluator, Wheel Controller のソフトウェアモジュールに RTM を適用した。各ソフトウェアモジュールの制御には RTC-Link を使い、シミュレーション環境として VineLinux, Pentium 4, 2.8GHz を用意し、Fig.2 に示したダイナミクスシミュレータで RTM を適用したシステムの動作確認を行った。コンポーネントを起動

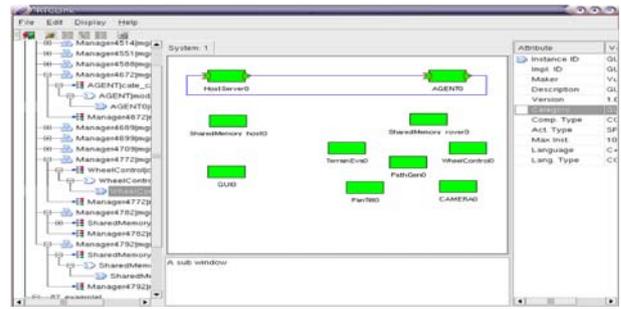


Fig.3 RTsystem

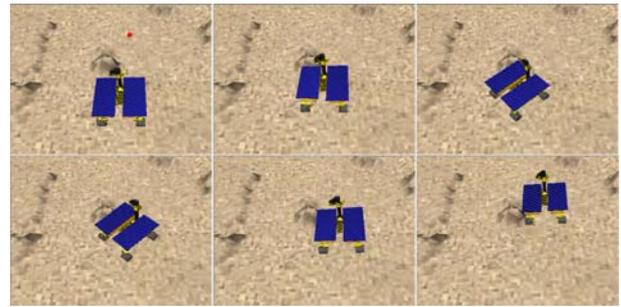


Fig.4 Simulation Result

し、ネーミングサービスに登録されているかを確認し、RTCLink のドローウインドウにコンポーネントブロックを表示・接続する。登録したコンポーネントを全て起動し、各コンポーネントがコマンド・データをやり取りしながら三次元カメラでの地形データ取得、地形評価、パス生成、走行など一連の自律動作ができるかを確認する。RTCLink の操作画面を Fig.3 に示す。ネーミングサービスの登録、RT コンポーネントの表示、RT コンポーネントの起動を確認することができた。次にシミュレーション結果を Fig.4 に示す。ローバの初期位置は図の左上であり、ゴールは赤丸のポイントを指定した。このシミュレーション実験によりシステムが正常に稼動することが確認できた。

5. まとめ

本稿ではネットワークアーキテクチャの提案、RT ミドルウェアを用いたシステム構築、シミュレーション実験によるシステムの動作確認を行った。RT ミドルウェアを適応することでネットワークシステムを比較的容易に構築することができた。今後は GUI やオペレータシステムなどの RT コンポーネント化などを行うと同時に RTC-L の導入も検討し、実機で実験を行うことを目指す。

参考文献

- 1) 安藤慶昭, 末廣尚士, 北垣高成, 神徳徹雄, 尹祐根, "RT 複合コンポーネントおよびコンポーネントマネージャの実装— RT ミドルウェアの基本機能に関する研究開発 (その 8) —", 日本機械学会ロボティクス・メカトロニクス講演会 2004, p.1C26, 2004.09
- 2) 安藤慶昭, 末廣尚士, 北垣高成, 神徳徹雄, 尹祐根, "RT コンポーネントによるシステム構築法— RT ミドルウェアの基本機能に関する研究開発 (その 14) —", 日本機械学会ロボティクス・メカトロニクス講演会 2005, 2005.06