

# 第24回 日本ロボット学会 学術講演会

The 24th Annual Conference of  
the Robotics Society of Japan

# 講演概要集

会期 ▶ 2006年 9月14～16日

会場 ▶ 岡山大学津島キャンパス



# ネットワーク及びモジュール構成による 遠隔移動ロボットのシステム検討

## System design for Tele-operated mobile robot by a network and module construction

○原佑輔(中央大院) 高橋新吾(中央大院) 國井康晴(中央大) 安藤慶昭(産総研)

\*Yusuke Hara, Shingo Takahashi, 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、自然環境などでの調査、観測作業等においてロボットの利用が期待されている。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機能で構成されるため、開発の長期化が課題となる。このため、試作サイクルの短縮や並列開発による効率化が求められる。一方、多くの場合、システムは一度配置されると直接アクセスすることが難しく、パラメータの不適合やアルゴリズムの不具合、故障への対応が課題となる。殆どの場合、接続される何らかの通信回線及びネットワーク越しに対応することが要求されるため、何らかの仕組みが必要となる。

以上より、本研究ではハードウェア、ソフトウェアのモジュール化及びネットワークベースでの構成を検討する。特に、近年のインターネット普及によりネットワークソフトウェア技術の発展は目覚ましく、CORBA、DCOMなどの分散オブジェクト技術が注目される。これはオブジェクトをネットワーク上の複数の計算機に配置し、それらを連携動作させることによりシステムを構築する手法である。ロボット分野においてもCORBAをベースとしたRTミドルウェアの検討が産総研を中心に進んでいる[1]。RTミドルウェアは、ロボットシステム構築に特化したミドルウェアであり、CORBAの複雑さを隠蔽するフレームワークが用意されているため、実用性が高く、開発効率向上に寄与することが期待できる。

以上より本稿では開発フェーズ、運用フェーズの要求を洗い出し、システムアーキテクチャの検討を行う。また、検討結果より、遠隔移動探索システムを対象にRTミドルウェアを用いたシステムを構築し、動作の確認を行なった。

### 2. システム開発と運用の効率化

遠隔移動ロボットには開発フェーズと運用フェーズが存在する。本章では、それぞれのフェーズにおける課題を洗い出し、解決方法とそれを踏まえたアーキテク

チャの検討を行う。

#### 2.1 開発フェーズ

開発において開発効率を向上させるためには以下3つが要求される。

- 1) 継続性の確保
- 2) 並列性の確保
- 3) 柔軟性の確保

##### 1) 継続性の確保

一般にロボットは高度で複雑な機能を持つため、ハードウェアとソフトウェアが大規模化する傾向がある。その結果、検討、試作、実用に至るサイクルを繰り返すため研究開発期間が長期化する。新たな試作開発は、過去のシステムをベースに実施されるため、ハード・ソフト共に資産の再利用と変更、追加の容易さが効率化へ向けた課題となる。さらに、開発全体を通して一人の担当者が継続的に開発を実施することは難しく、新たな担当者は、ハード・ソフトの理解に時間を要し、当人の苦勞に加え、開発ポテンシャルの低下など開発効率に大きな影響を与える。このような問題を解決するためには一般に、ドキュメント等の管理整備に加え、システムの機能分割により機能を明確化し、ハードウェア・ソフトウェア共に一定の粒度以下でのモジュール化で対応可能である。これにより、担当者は全てのモジュールを理解せずとも、必要に応じ特定のモジュールのみ理解することで研究開発の継続が可能になり、各モジュールの入れ替えや接続関係の変更、新たなモジュールの追加などにより、システムが構築可能なため、効率性が向上する。特にハードウェアに関しては、インターフェースを共通化し、ハードの違いをソフトウェアで吸収する。

以上より、モジュール化によるブラックボックス化によって、中身を理解せずとも利用、応用が可能になり、属性が確保された結果、時間的な効率化が図られる。

##### 2) 並列性の確保

遠隔移動ロボットシステムは運動制御技術や認識技術、行動計画技術など、粒度の違う様々なサブシステ

ムにより構成され、それらが有機的に接続されている。対象となる技術分野が多岐に渡るため、各サブシステムは専門知識を持つ複数の技術者により並列に開発が進められる。そのため境界部分が生じ、初期的な機能試験などに際しても共同作業が発生し、担当者間のスケジュール調整や開発遅延などによる効率低下が懸念される。さらに、実際の作業障害に加えて、各開発者に定常的な精神的なストレスを与え、士気の低下などに繋がり開発効率に影響を与える。よって、可能な限り各サブシステムの独立性を確保し、各サブシステムの開発遅延の影響を最小限に抑えることが重要となる。

以上より、モジュール化と入出力の規格化により独立性の確保に対応する。特に、機能、従属関係を整理し粒度を調整し、モジュール化することにより、開発規模を縮小し機能を局所化できるため、並列的な開発が容易になる。

### 3) 柔軟性の確保

システムが大規模になると一部の機能の修正や拡張が多数のプログラムに及ぶため、システムの変更が難しくなる。しかし、開発フェーズでは複数の手法を実験で評価し試行錯誤するプロセスが重要なため、機能の変更、修正、拡張が繰り返され、その作業に多くの時間を有するため、効率化が課題となる。

解決に向け、モジュール化によるブラックボックス化と汎用的なインターフェースの構築を検討する。モジュールを小さい粒度で構成し、インターフェースを汎用化することで、モジュール間の接続関係の変更が容易に実現される。

## 2.2 運用フェーズ

運用フェーズにおいて、ロボットとオペレータの間には物理的な遠距離が存在するため、以下の3つが重要となる。

- 1) 適応性の確保
- 2) 透過性の確保
- 3) 可用性の確保

### 1) 適応性の確保

遠隔移動ロボットの運用では非可視運用や状況によっては通信容量の制限や通信時間遅れなどが発生するため、オペレータがロボット自身の情報やその周辺環境情報をリアルタイムに理解することは難しい。そのため、多くの自律機能が付加されている。しかし、自律機能は想定された状況下において有効に機能するものの、想定外の状況に対してはリスクが発生する。自律機能が何らかの機能不全に陥った場合、通常、制御ゲインなど各種パラメータの変更での対応や、根本的なアルゴリズムの変更で対処し、障害を乗り切る必要がある。よって、プログラムやハードウェアの接続関係をソフトウェア的に柔軟に変更できる必要があり、モジュール化とネットワークベースのシステム構築で対応する。

### 2) 透過性の確保

遠隔地で活動するシステムを運用するためには、常にシステムの状態を把握する必要がある。特に障害が発

生した場合、その原因の特定と回避のため、ハード、ソフトを問わず、全てのシステムの状態の把握とアクセスが欠かせない。このため、システムの透過性の確保が重要な課題となる。しかし、通常、システムには接続関係が存在し、またハード的な接続方式によっては、情報の直接的取得が難しい場合がある。これに対し、接続をネットワークベースとし、接続関係をネットワーク上に維持することで、必要に応じ各機能に直接アクセスしたり、接続関係のソフト的に切り替えることが可能になり、透過性を維持することが可能となる。

### 3) 可用性の確保

障害やシステム変更時にも安定した動作を実現するため、可用性、すなわち以下の項目が重要となる。

- 障害を起こしにくいシステム
- システムを停止させずに変更可能

まず、障害を起こしにくいシステムの実現に向け、ハードウェアを冗長構成にすることが有効である。これをマルチCPU化とネットワーク化により実現する。ネットワーク上に複数のCPUを分散配置することで複数のタスクをハードウェアの接続関係に縛られることなく動作させることができる。これにより、1つのプロセスに障害が起きた際、同じプロセス（以下バックアッププロセス）を他のCPUで動作させ、対象をバックアッププロセスと接続することで障害を最小限に抑えることができる。

次にシステムを停止させずにシステム変更可能な構成にするためには、インターフェースの共通化とタスク管理が重要である。遠隔移動ロボットシステムは各サブシステムが協調して1つの動きを達成するが、常に全てのプロセスが動作しているわけではない。各サブシステムはシーケンス的に動作することが多く、ある期間において動作していないサブシステムが存在する。これを利用し、動作していないときにサブシステムを変更することでシステムを停止させることなくある一部分を変更することができる。このとき、インターフェースの共通化がなされているとスムーズにシステム変更を行うことができる。また、サブシステム間の従属関係が強いとシステムの変更が他のサブシステムに波及してしまうため、ソフトウェア間の従属関係は低い方が望ましい。

## 2.3 システム要件

以上をまとめるとハードウェア・ソフトウェア双方においてモジュール化、インターフェースの共通化、マルチCPU化、ネットワーク化によるシステム構築が求められることになる。次章以降で具体的なシステム設計を検討する。

## 3. システム検討

ハードウェアにおいては1つのサブシステムに1つのCPUを割り当てるマルチCPU化を行う。マルチCPU化の利点は処理の分散による性能の向上と複数のCPUを使用することによる可用性の向上である。また、このとき可用性を高めるためにシステムは冗長構成に

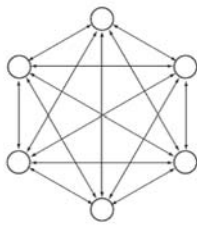


Fig.1 Custom Interface

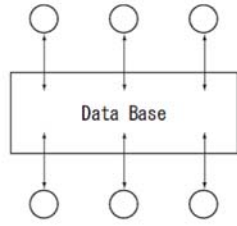


Fig.2 Common Interface

し、バックアップ用の CPU も搭載するものとする。  
 一般にマルチ CPU 化を行うと必要な透過性や独立性の確保が困難になるが、スター型のネットワークを適用することにより確保し、システムを構築する。スター型ネットワークにより、インターフェイスを共通化することができ、個々のハードウェアの違いを隠蔽することができる。また、各サブシステムがネットワークで接続されているため、オペレータやモジュール同士が直接アクセスすることができ、透過性の向上、障害が発生したモジュールだけを切り離すなど独立性の向上が期待できる。ハードウェア構成例を Fig.3(a) に示す。

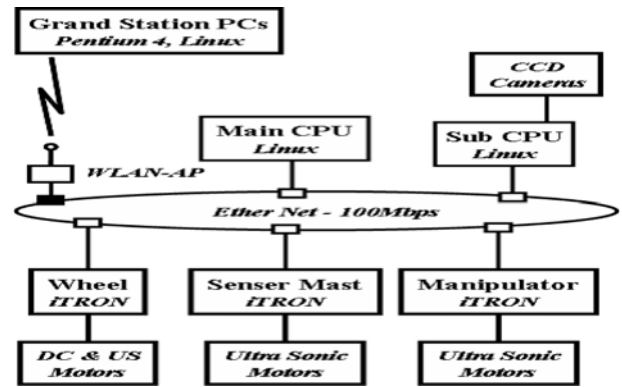
ソフトウェアにおいてはモジュール構成によるシステム構築を行う。モジュールの接続形態として、特定のモジュール同士を接続する方法 [2], データベースを用いる方法 [3] がある。前者の手法はシステムに高い即応性が期待できるが、ネットワークが複雑になる。また高い即応性を十分に引き出すには接続の数だけ独自のインターフェイスを設計する必要がある (Fig.1)。一方後者は即応性は劣るがデータベースとの共通のインターフェイスで全ての接続を表現できるため、モジュールの追加が容易である (Fig.2)。前章で議論した結果より、アーキテクチャでは柔軟性が高く機能の追加が容易である必要があるため、後者の接続形態を採用する。ソフトウェア構成例を Fig.3(b) に示す。

よってこのような分散オブジェクトシステムを構成するため、分散オブジェクト記述ツールの利用を検討する。一般に、CORBA や DCOM などのツールが知られているが、本研究では、それらをベースにロボットシステムに特化したミドルウェアである RT ミドルウェアを適用することとする [4]。

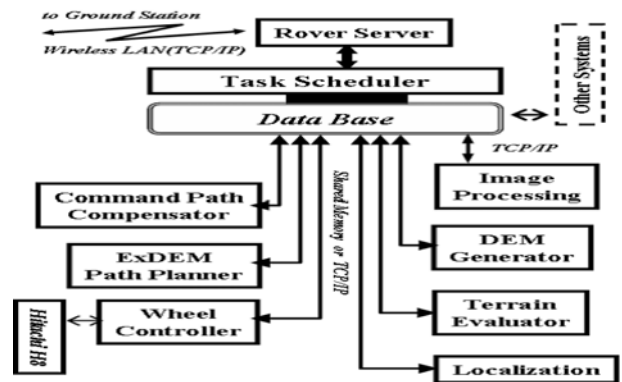
#### 4. RT ミドルウェアの利用

システム構築のため、RT ミドルウェアの導入を検討する。RT ミドルウェアは、経済産業省のもと、(社) 日本ロボット工業会、(独) 産業技術総合研究所などが中心となって開発を行っているミドルウェアである。多様なニーズに応えられるロボットシステムの実現を目指し、ロボット本体や機能部品をソフトウェアレベルでモジュール化し組み合わせることで、システム設計者の目標とするロボットシステムを比較的容易に構築することを可能にする技術である。RT ミドルウェアを導入する利点としては

- OS・ネットワーク・言語非依存
- オペレータによる遠隔制御
- コンポーネントの接続変更
- オブジェクトモデルやフレームワークの提供



(a) Hardware System Structure



(b) Software System Structure

Fig.3 System Architecture for Tele mobile-robot

#### ● RT コンポーネント

が挙げられる。RT ミドルウェアは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。このように RT ミドルウェアの特徴は他の分散オブジェクトミドルウェアと違い、コンポーネント自信がアクティブに動作すること、コンポーネント間の密な連携など、RT システムに固有な機能をサポートしていることである。また、システム構築にあたっては、必要に応じ RT ミドルウェアの改良、拡張を検討する。

#### 5. ネットワーク分散型システムの構築

2章、3章で議論したシステムを4章で紹介した RT ミドルウェアを用いて構築する。RT ミドルウェアのプロトタイプ実装である OpenRTM-aist-0.2.0 を使い、各サブシステムのモジュール化を行う [5]。本稿では以下のサブシステムをモジュールとして構成した。

- ユーザインタフェース
- 地形評価システム
- カメラシステム
- 走行システム

各モジュール同士の接続は、インターフェイスを共通化するためデータベースを介して接続する。また、移動観測システムではモジュール間の通信が膨大になることが予想される。このような状況ではオペレータがロ

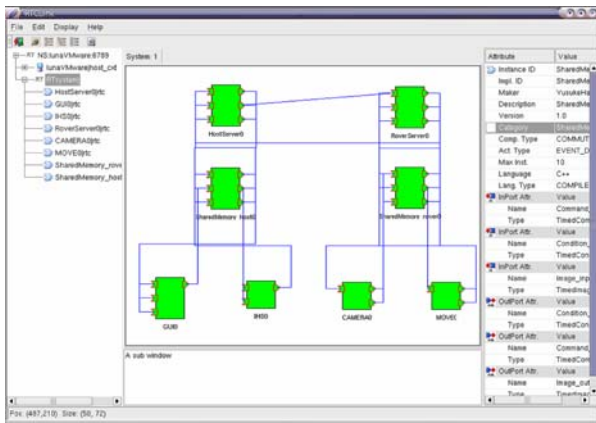


Fig.4 RTC-link

ボットの現状を把握しづらい上に通信量に起因する通信エラーが起こる可能性がある。よって、モジュール間のデータのやり取りを明確にし、無駄なモジュール間通信を減らすために内容別にデータのやり取りを行う。本システムでは Condition(姿勢情報など), Command(モジュールに対する指令など), Image(DEM や画像データなど) に分けて通信を行った。各モジュールの接続・制御には RTC-Link を用いた。シミュレーション環境として VineLinux, Pentium 4, 2.8GHz を用意し、各コンポーネントは Fig.4 のように接続し、Fig.5 に示したシミュレーションシステムで RT ミドルウェアを適用したシステムで一連の走行ミッション

1. カメラで DEM を取得
2. オペレータに送信
3. データを基に地形評価
4. ゴールを指定
5. ロボにゴールを送信
6. ゴールまで自律走行

が正常に稼動することを確認する。地形は 4[M] × 4[M] の DEM データを用い、Fig.5 の Operation System 右下のように地形評価・ゴールを指示し、Dynamics Simulator でシミュレーションを行った。その結果を Fig.6 に示す。これによりロボシステムの一連の走行ミッションが正常に稼動することを確認できた。

## 6. まとめと今後の課題

本稿では遠隔移動ロボットシステムの構築を目指し、開発フェーズと運用フェーズの要求を洗い出すことにより、アーキテクチャを検討し、ネットワークベースの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。

今後は GUI やマニピュ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と同時に運用フェーズにおける運用シナリオと必要機能の検討、RT ミドルウェアの改良、拡張を検討する。

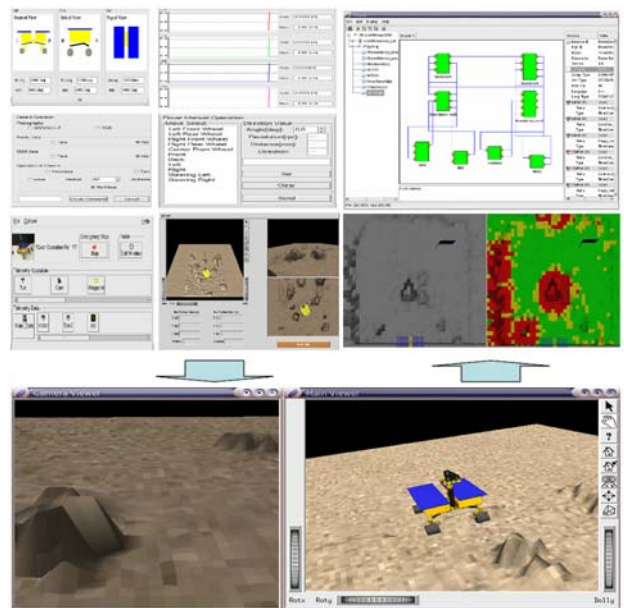


Fig.5 Operation System

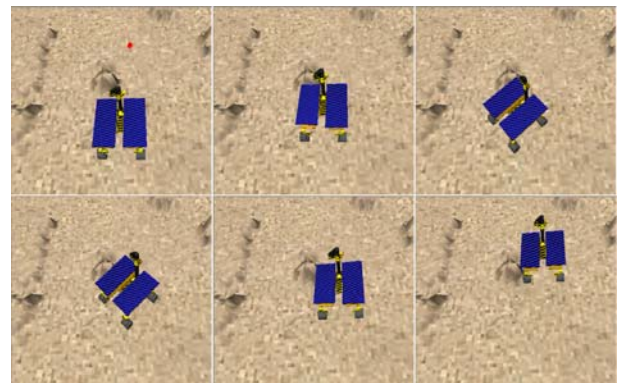


Fig.6 Simulation Result

## 参考文献

- 1) <http://www.is.aist.go.jp/rt/>
- 2) 岡哲資, 武田幸司, 稲葉雅幸, 井上博允”実時間並列計算モデルに基づく自律ロボットの運動システムの構造化記述と設計”, 日本ロボット学会誌, Vol.16, No.3, pp.337-344, 1998
- 3) 岡哲資, 田代純也, 植坂岳治, 高瀬國克 : ” BeNet と Java 言語に基づく統一的な記述手法によるロボットの総合ソフトウェアの開発 ”, 日本ロボット学会誌, Vol.19, No.2, pp.207-216, 2001
- 4) 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ー RT ミドルウェアの基本機能に関する研究開発 (その 14) ー ”, 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
- 5) OpenRTM: <http://www.is.aist.go.jp/rt/OpenRTM-aist-Tutorial/>