

## IXM: 移動ロボット上の機能統合および通信ミドルウェア

住谷拓馬 菅谷みどり

芝浦工業大学工学部情報工学科

## 1. 研究の背景

近年、お掃除ロボットなどの移動ロボットが家庭などで利用されている。本研究室でも、こうした背景に沿い、移動ロボットの活用方法を研究しており、移動ロボットの上にセンサやカメラを取り付けることにより、見守りや、落し物探索の研究を行っている。

しかし、移動ロボットの機能拡張や、ソフトウェアの改変には、デバイスごとに異なるソフトウェア開発環境を理解して利用しなくてはならず、手間がかかる問題がある。共通して利用できる統合ミドルウェアなどの環境があることが望ましいが、既存研究[1]の開発支援ツールでは、ソフトウェア改変までは行うことができない。

本研究では、移動ロボット上でのソフトウェア開発者に向けた、ソフトウェア開発環境の統合のための基盤として、IXM (Information eXchange Middleware) という統合ミドルウェアを提案する。IXM は、異なるソフトウェア開発環境を隠蔽し、ソフトウェア開発者が容易に拡張機能を利用できる API を提供する。また、本研究では、IXM を利用し、遠隔地から操作するために通信システムを実装した。評価では、IXM を使用した場合と使用しなかった場合のパラメータ変更プログラム行数を比較、および、移動ロボットを起動する際の工程数を比較し、その結果を示した。

## 2. 統合ミドルウェア (IXM) の設計

本研究では実装にあたり、まず、ソフトウェア開発環境に必要とされる API を明確にするために、ロボットへの要求を明確にし、関係を定義した。次に、定義に従って、IXM の API 設計および内部設計の検討、実装、評価を行った。そして、IXM を遠隔地から操作するための機能拡張を行った。次に詳細を述べていく。

## 2.1 移動ロボットへの要求

ソフトウェア開発環境で必要とされる API を明確にするために、まず、移動ロボットで必要となる動作を決めた。今回は、実際に開発を行っている見守りとキーボード操作の二つとした。次に、二つの動作に必要な別々に実装されているソフトウェアを整理した。見守りにおいては、追従、転倒検知、気温測定の三つとし、キーボ

ード操作においては、キーボード操作のみとした。また、今回ソフトウェアの改変する具体的なパラメータを、追従の追従距離、気温測定の測定間隔とした。

整理して木構造にまとめたものを図1に示した。

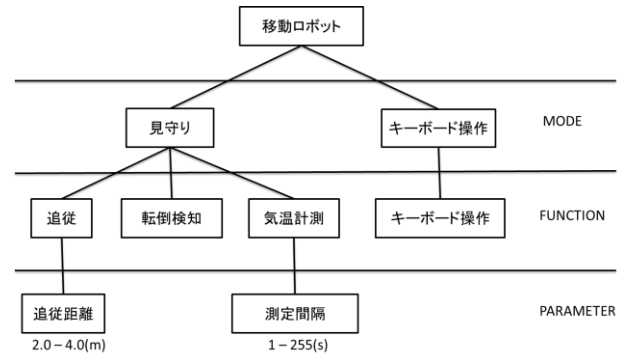


図1 機能関係図

## 2.2 API の実装

ソフトウェア開発者が、移動ロボットの拡張機能の利用、また、ソフトウェアの改変を容易に行えるように以下の API を設計し実装した。

- Ixm Set/Get Mode(struct ixm\_service \*ixm)
- Ixm Set/Get Func(struct ixm\_service \*ixm)
- Ixm Set/Get Param(struct ixm\_service \*ixm)
- Ixm Start/Stop(struct ixm\_service \*ixm)

Set/Get は、MODE、FUNCTION、PARAMETER を移動ロボットに設定、または、設定の取得をする。Start/Stop は、必要なソフトウェアを実行、停止する。複数のソフトウェアを実行する際には、それぞれのソフトウェアごとに子プロセスを生成し、非同期に実行できるものとした。

## 3. IXM の通信機能の拡張

移動ロボット上のセンサから取得した情報を、モニタリング及び、移動ロボットを遠隔地から操作できるようにするため、IXM のミドルウェア上での通信機能の拡張を行った。これにより、外部から移動ロボット上の PC との通信を可能とし、移動ロボット上の PC にログインせずともソフトウェアの改変、実行も可能となった。

また、ソフトウェア開発者が、移動ロボットで動作実験を行う際に、MODE、FUNCTION、PARAMETER の設定、ソフトウェアの実行、通信、そして、移動ロボット上のセンサからの取得情報のモニタリング、以上のすべてを行

**IXM: Functional integration and communication middleware of the mobile robot**

Takuma Sumiya, Midori Sugaya  
Information engineering, engineering department, SHIBAURA  
INSTITUTE OF TECHNOLOGY  
135-8548, koto, Japan

うコマンドが必要になると考え、IXM コマンドを API を利用して実装した。コマンド実行時のシステム図を図 2 に示した。

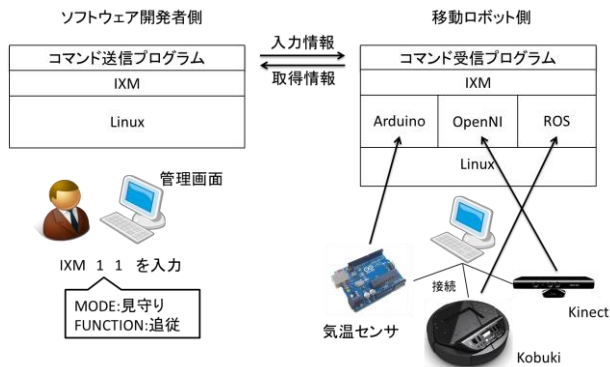


図 2 システム図

IXM コマンドは引数に MODE、FUNCTION をとる。例として、図 2 では MODE に見守り、FUNCTION に追従を選択した場合である。

3. 1 IXM コマンドの処理

図 3 に、IXM コマンド実行時の、ソフトウェア開発者側と移動ロボット側との通信のシーケンスを示した。

IXM コマンドを実行すると、まず、コマンド送信プログラムを実行する。その際、使用する FUNCTION の中に PARAMETER 変更可能なものがある場合、PARAMETER 入力を行い、コマンド受信プログラムへと情報を送信する。次に、コマンド受信プログラムは、入力情報を受信し、まず MODE 設定を行う。そして、FUNCTION 設定を行い、FUNCTION に応じて PARAMETER 設定を行う。最後に、各設定に基づき、ソフトウェアを実行する。

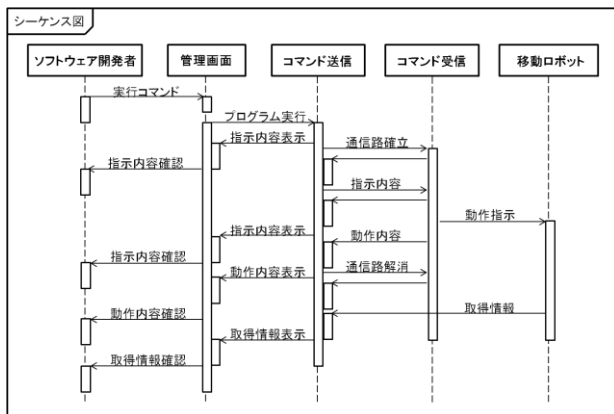


図 3 実行時のシーケンス図

4. 評価

ソフトウェア開発者が、IXM を利用した場合のソフトウェア開発効率を評価することを目的とし、ソフトウェア開発者が、IXM コマンドを用いて、PARAMETER の一つである追従距離の変更を行う際の評価を行った。評価では、IXM コマンドを使用しなかった場合とした場合の、ソフトウェア変更時に必要となるプログラム行数を比較した。

	プログラム行数
IXM コマンドを使用せずに、プログラムを書いた場合	25
IXM コマンドを使用し、プログラムを書いた場合	1

表 1 プログラム行数比較表

次に、ソフトウェア開発者が、IXM コマンドを用いて、移動ロボットに FUNCTION である追従をさせる際の評価を行った。評価では、IXM コマンドを使用しなかった場合と使用した場合の、移動ロボットを起動するまでの工程数を比較した。工程数は、コマンド実行数とした。

	工程数
IXM コマンドを使用せずに、移動ロボット起動に必要なコマンドを呼び出した場合	4
IXM コマンドを使用した場合	1

表 2 移動ロボット起動工程数比較表

3. 2 結果

表 1 のとおり、IXM コマンドを使用した場合、使用しなかった場合より 24 行プログラム行数を削減できた。また、IXM コマンドを使用すると、プログラム行数の削減だけではなく、複数のソフトウェアの並列実行や、異なる言語で書かれたソフトウェア環境のソフトウェア変更を行うこと、移動ロボット上のセンサからの取得情報をモニタリングすることが、プログラム 1 行で可能になった。これは、ソフトウェア開発効率上評価できる。

移動ロボット起動までの工程数は、表 2 のように IXM コマンドを使用することにより、3 工程削減できた。これにより、ソフトウェア開発者の手間を省くことが可能になると言える。しかし、数としてはわずかであり、さらなる API の追加が必要である。

5. 今後の課題

今回の研究により、ソフトウェアの変更を容易にすることができた。しかし、ソフトウェアを追加する際の API は実装されておらず、移動ロボットの機能拡張に関しては、まだ難しいものとなっている。今後、ソフトウェア追加の API を実装し、移動ロボットの機能拡張を容易なものとし、さまざまな動作ができる移動ロボットを実現したいと考えている。また、今後の API の追加に伴い、今回実装した API のさらなる改良も必要であると考えられる。

参考文献

[1] 村上 青児(筑波大学/産総研), 安藤 慶昭(産総研), 関山 守(産総研), 鍛冶 良作(産総研), 神徳 徹雄(産総研): 組込み用 Linux ボード, Beagle Black における RT ミドルウェアの動作と開発支援ツール