

# ネットワークロボットにおける コストを考慮したサービス引き継ぎの検討

金澤 祥弘<sup>†</sup> 鈴木 亮平<sup>†</sup> 戸辺 義人<sup>‡</sup>

<sup>†</sup>東京電機大学大学院工学研究科情報メディア学専攻

<sup>‡</sup>東京電機大学工学部情報メディア学科

## 1 はじめに

ロボット技術の発展に伴い、生活の中で使われるようなロボットが増えてきた。身近に活動を行う中で、ロボットの停止は人や作業に大きな影響を与えることが考えられる。このようなロボットサービスの中断に対して、アプリケーション自身が周囲にいるロボットに移動し、行われていた仕事を引き継ぐことで、ロボットサービスの継続を目指すフレームワークを検討する。また、実際にモバイルエージェントと移動ロボットを用いてロボット上のアプリケーションが移動する実験を行った。本稿では、ロボット間のサービス引き継ぎについて検討した結果を述べる。

## 2 ネットワークロボット

ネットワークロボットではエージェントという形で各ロボット上で動作する自律的なプログラムを用いることが多い。また、ロボット以外のエージェントに関する研究としては、コンピュータ上を移動しながらプログラムが仕事を行う、モバイルエージェントに関する研究<sup>[1]</sup>がある。本研究ではこれらの研究を元に、ロボット上のアプリケーションを移動させることでロボットサービスの継続手法を確立することを目指す。

## 3 設計指針

周囲のロボットを利用してサービスを継続するシステムを設計するに辺り以下の二点に着目した。

- ・サービス中断期間の減少
  - ・ロボットサービスが停止した場合、すぐ近くに他のロボットがいるとは限らないため、他の場所にいるロボットを呼びサービスを継続させる。この時、周囲のロボットの中からできるだけ早く駆けつけられるロボットを選択する。
  - ・ロボットにあわせた動作の提供
- 例え身近にロボットがいたとしても、そのロボットが同じサービスを提供することができるとは限らない。しかし、道案内動作や警備動作といった点では、必ずしも元の動作をそのまま再現する必要はない。そのため、各ロボットが備えているデバイスや機能によってサービスを変化させることを目指す。

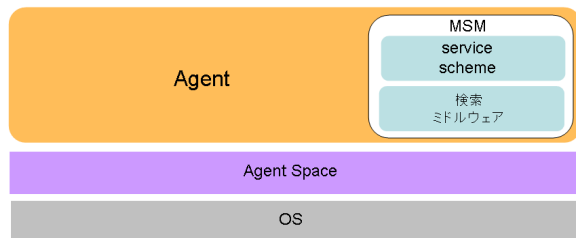


図1 MSMとアプリケーションの構造

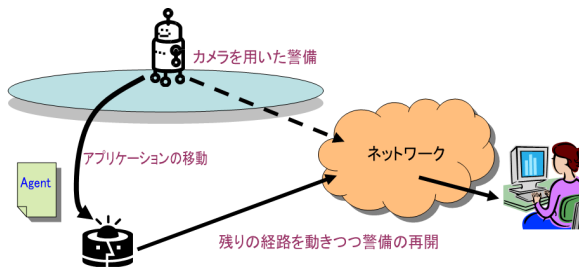


図2 サービス移動の概念

## 4 システム設計

### 4.1 MSM (Mobile Service Middleware)

各ロボット上のアプリケーションを移動させるミドルウェアとして MSM を設計した。MSM はロボットアプリケーションの移動先として適切なロボットを検索する検索ミドルウェアと、各ロボットが備えているデバイスをまとめアプリケーションに対して提示する Service Scheme からできている。図1に MSM とアプリケーションの構造を示す。また、図2にサービス移動の概念図を示す。

### 4.2 移動先の検索

アプリケーションが移動するために、周囲のロボットから移動可能なロボットを検索するための機構を以下に示す。まず、サービスを継続できなくなったロボット(Source)は、周囲に対してブロードキャストを行い、ロボット上で動作するアプリケーションを各ロボットに移動させる。移動したアプリケーションはロボットの位置と、備えているロボットデバイスを収集し、アプリケーションが実行するサービスに必要なデバイスが揃っているかを確認する。もし最低限必要なデバイスが揃っているならば、デバイスと Source までの距離をコストという値にまとめる。まとめたコストは CRPP(Cost RePly Packet)として Source に送信され、Source 上では周囲のロボットから集めたコストの中でもっとも低いロボットを選択しサービスを継続させる。この時、他のロボット上のアプリケーションを破棄させるために、ESRM(End of Search

Consideration of Service Migration based on Cost in Networked Robots.

<sup>†</sup>Information Systems and Multimedia Design, Graduate School of Engineering, Tokyo Denki University

<sup>‡</sup>Department of Information Systems and Multimedia Design, Tokyo Denki University

Robots Message)をブロードキャストし、引き継ぎ対象でないロボット上のアプリケーションを破棄する。また、もしブロードキャストした時点で周囲に引き継ぎの可能なロボットがいなかった場合は、Source が NMRM(No Migrated Robot Message)をブロードキャストする。このメッセージを受け取った周囲のロボットは、それぞれがブロードキャストを行い、引き継ぎ可能なロボットを見つけるまで動作を繰り返し行う。上記のアルゴリズムを図 3 に示す。

```

1: rcv_pkt: receive packet
2: while rcv_pkt is ESRM do
3:   get_position(){get robot position}
4:   get_devices(){get robot equipped devices}
5:   diff(RequestedDevices, EquippedDevices): compare necessity
   device list with equipped devices{TRUE, FALSE}
6:   if diff(RequestedDevices, EquippedDevices)=TRUE then
7:     unicast CRPP to Source
8:   else
9:     unicast CRPP with NoRequestedDevicesFlag on
10:  end if
11:  if rcv_pkt is NMRM then
12:    broadcast AGENT to Neighbor Robots
13:  end if
14: end while

```

図 3 移動先ロボット検索アルゴリズム

### 4.3 Service Scheme

本研究では、アプリケーションが移動先のロボットにあわせたサービスを行えるよう、各ロボットが備えたデバイスをアプリケーションに対して提示するインタフェースとして、Service Scheme を設計した。Service Scheme はアプリケーションが各ロボット上に移動した後、そのロボットの有するデバイスをまとめ、リストとして保存する。そのリストを元にアプリケーションは使用するデバイスが備わっているかどうかを判断する。

## 5 実装

PC から USB を通して IO インタフェースを操作できる USBIO と、駆動系であるモータとタイヤを組み合わせ、PC 上から操作できる移動ロボット(HANIWA<sup>[2]</sup>)を作成した。本ロボットは JNI(Java Native Interface)を使用し、JAVA アプリケーション上から USB 機器に対して信号のオン・オフを切り替えることができる。また、各種モータと USBIO を繋ぐことで、前進、後進、右旋回、左旋回の動作を実装した。図 4 に本ロボットのソフトウェア構成を示す。

## 6 実験

実際にロボット上で動作するアプリケーションが障害物によって動作を停止し、他のロボットに移動して仕事を継続する動作を行った。今回は、5 章で述べたロボットにカメラを付け、巡回ロボットとして周囲の映像を取得しながら走行させた。また、周囲の障害物を検知するために PSD(Position Sensitive Detector)を装備させた。その他に、カメラデータを受信する PC を一台用意し常時受信データ

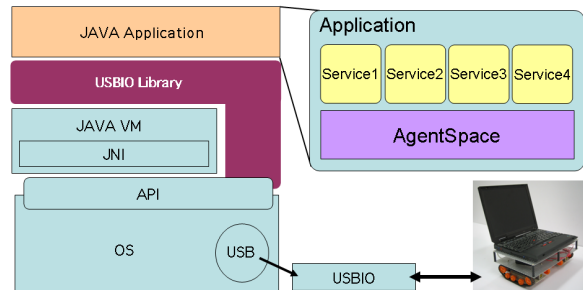


図 4 HANIWA のシステム

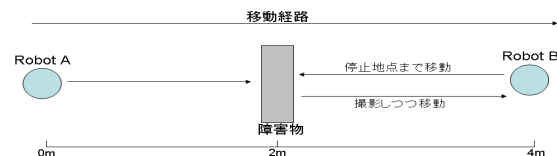


図 5 ロボットと障害物の配置と移動経路

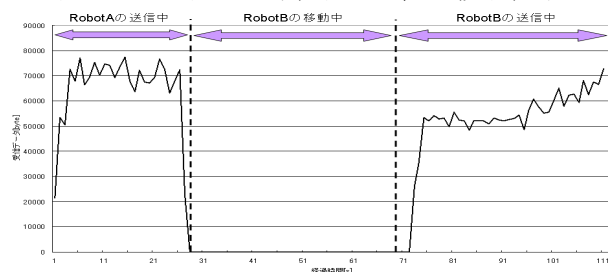


図 6 経過時間と受信データ

量モニタリングした。図 5 に二台のロボットと障害物の配置を示す。また、モニタリング結果を図 7 に示す。各ロボットの動作時間を測定し、図 6 上の受信データと照らし合わせたところ、RobotA が停止してから RobotB が移動し、RobotA の停止地点まで移動してからデータを送り始めるのが確認できた。しかし、RobotB が障害物まで移動するまでの間、RobotB はデータを送信していない。ロボットのサービス次第では必ずしも故障地点まで移動する必要はないため、今後は引き継ぎ先ロボットの動作を細かく設定する必要がある。

## 7 おわりに

本稿では、ネットワークロボットにおけるサービスの引き継ぎを行うために必要なアーキテクチャを検討し、実装を行った。また実際に移動ロボット上で動作させ、ロボット上のサービスが移動したあとも継続してサービスを行っていることを確認した。今後はデバイスの異なるロボットを複数用意し、Service Scheme の有効性について実証していく予定である。

## 参考文献

- [1] Ichiro Satoh, "MobileSpaces: A Framework for Building Adaptive Distributed Applications using a Hierarchical Mobile Agent System", Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS'2000), 2000.
- [2] HANIWA, <http://www.unl.im.dendai.ac.jp/haniwa/>