

移動ロボットの行動要素授受システムの構築 及び行動要素オブジェクトの設計

Constructing behavior-module transfer system and Designing behavior-module object

○忍頂寺毅*, 松川卓二**, 王志東*,
高橋隆行*, 中野栄二*, Majid N. Ahmadaabaadi*

○Takashi Ninjouji*, Takuji Matsukawa**, Wang Zhidong*,
Tkayuki Takahashi*, Eiji Nakano*, Majid N. Ahmadaabaadi*

*東北大学大学院情報科学研究科, **東北大学工学部

*Graduate School of Information Sciences in Tohoku University,

*Faculty of Engineering in Tohoku University,

キーワード: 行動要素の授受(transfer elemental-behavior),
分散オブジェクト(distributed object), ネットワークロボットシステム(networked-robot system)

連絡先: 〒980-77 仙台市青葉区荒巻字青葉 東北大学大学院 情報科学研究科 中野研究室(青葉山キャンパス機械系)
忍頂寺 毅, Tel.: (022)217-7025, Fax.: (022)217-7023, E-mail: ninjouji@robotics.is.tohoku.ac.jp

1. 緒言

移動ロボットの研究において複数ロボットによる協調作業といった分野がある。こうした動きはサブサンプリング・アーキテクチャの提唱¹⁾にはじまってビハイビア・ベーストのロボットの研究からも活気づけられているように見受けられる。

とくに通信を行えるロボットが複数いることで協調作業を実現するために、次のような研究がなされている。ある作業に複数台のロボットを投入することで、そのうちの何台かが機能不良の事態に陥ったとしても、残りのロボットが作業を遂行する²⁾ことを目指すもの。また、一台では作成に時間がかかる地図情報の獲得を、複数のロボットによって分担

して行う³⁾もの。そして学習成果を共有することで解の探索空間を分割する⁴⁾、といったような研究があげられる。

自律ロボットシステムの設計においては、下記の手順にしたがってなされるのが一般的である。

- 1) ロボットのタスクを想定する
- 2) タスクに基づいてハードウェアの仕様を決める
- 3) タスクとハードウェアに応じてソフトウェアを作成する

こうした工程のために、自律ロボットは設計時に想定されたタスクのためのアルゴリズム、すなわち行動やその手順しか持たないことになる。そして、そ

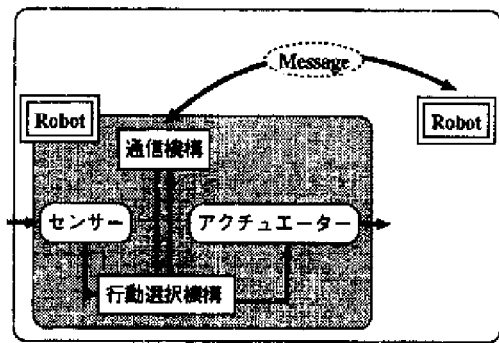


Fig. 1 ロボット間の通信と行動選択

の結果、ロボットたちは作業遂行における役割を、あるていど定められた枠の中から割り振られることになる。

しかし今後、自律ロボットの研究開発が進むにつれてロボットが自分のもつ動作を適宜組み合わせることで柔軟に作業に対応できることが望ましいと考えられる。そのとき、ロボットが活動中であってもある作業の手順や、そこで必要とされる行動要素といったものを受け取り、実行できるシステムがあれば、上記の要求の一部を満たせると考えられる。

複数ロボットにおいては、作業の手順を持っているロボットがその手順を持たないロボットにその手順を伝えることで、受け手のロボットは作業に参加することができる。また、行動要素についても、それを持っているロボットが持たないロボットに伝授することで、受け手のロボットは送り手のロボットの機能を得ることができる。そこで本研究はこうしたシステムの基本構成を提案し、本稿においてシステムの基礎部分について示すことにする。今後は、このシステムの有用性をシミュレーションと実機で検証する予定である。

このシステムは、図1に示すように、通信機構と行動選択機構の二つからなる。通信機構によって他のロボットと行動の手順や行動要素といった情報を通信し、行動選択機構によって受け取った行動要素が実行可能なものであるならば自身の行動の選択肢として活用する。

2. 分散オブジェクト環境によるシステム設計

ロボット間で作業の手順や行動要素を授受するためには、その情報が手続きを含んだデータ構造であることや、その行動要素が他の行動要素を含んでいることも考えられる。以下、こうした表現力をもったデータ構造を検討し、それをロボット間でやりとりするための枠組について見ていく。そうして、本研究で提案するシステムへのオブジェクト指向概念の導入、行動要素をオブジェクトとして表現すること、行動要素オブジェクト授受システムを分散オブジェクト環境によって構築することを明らかにする。

2.1 情報のデータ構造

ロボットが利用する情報のデータ構造は、いわばコンピュータシステムが使うデータ構造に還元できる。

データ型	構造
基本型	int型, double型, etc.
構造体	基本型の複合
オブジェクト	構造体 + メソッド

すると、パラメータとそれを利用する手続きを一まとめにして扱うためにはオブジェクトで表現することが妥当と考えられる。また、オブジェクトは他のオブジェクトを属性として持つことができる。これは行動要素が他の行動要素を含むことや、あるいは作業の手順を行動要素によって表現することを可能にし、ロボットの作業の手順や行動を授受するシステムの構築に有効である。

2.2 通信基盤

CPU パワーやコンピュータネットワークの技術の向上が著しい今、本研究においてはインターネット技術によってロボット間の通信を実装する。ここでロボットたちはTCP/IPによるネットワークを展開する。

こうしたネットワークにおいてコンピュータ同士

が通信する場合、ソケットかあるいは Remote Procedure Call (RPC) によるものが一般的である。いずれの手法を採用するにせよ、行動要素は他の行動要素を含む場合も考えられ、通信で扱う情報の種類が増えれば増えるほどプロトコルの設計は面倒なものになってしまう。したがって、オブジェクトをオブジェクトのまま扱える手法を採用することが望ましいと言える。

2.3 分散オブジェクト環境

異なるホストに存在するオブジェクトを参照し、また、オブジェクトをパラメータとして相手先に渡したり、戻り値ととしても受け取れるようにするために図2に示す分散オブジェクト環境を構築する。

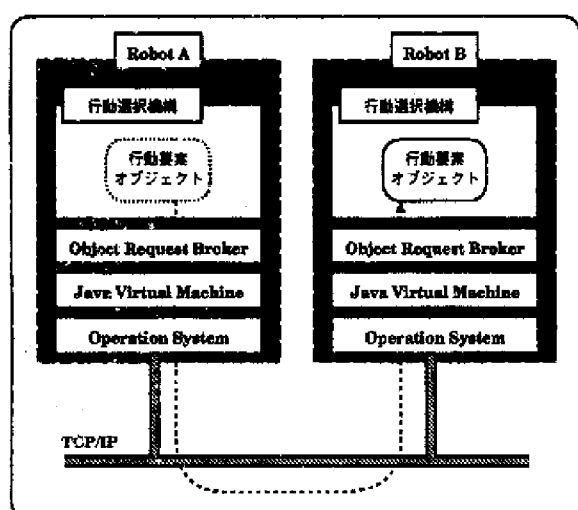


Fig. 2 自律移動ロボットのための分散オブジェクト環境

上記の環境を構築するために以下のものを用いた。

OS	Linux 2.0.31
ORB	JavaRMI
JDK	JDK1.1.3

Operating System に Linux を採用したのは、当研究室において研究および開発が進められているメッセンジャロボット⁵⁾などのように、自律移動ロボットシステムの基幹 OS として十分に機能していることによる。また、本研究におけるシステムの開発のため、JAVA 言語を利用した。この言語はオブ

ジェクト指向言語であるとともに、ネットワークプログラミングを重視した設計をとっている。そのため、分散オブジェクト環境を構築するために必要とされる Object Request Broker として、JavaRMI を Java Development Kit のデフォルトのパッケージに持っている。

さらに JAVA 言語の特徴として、その実行環境が豊富であるということがあげられる。Java 実行環境は実に数多くの OS で実装されており、本研究において用いている Linux を始め、Solaris, Windows95/NT や MacOS でも利用が可能である。このことは、本研究において構築を目指すシステムが多くの実行環境を有することを意味している。また、JAVA 言語はプログラム実行中であっても新しいクラスをロードし、そのインスタンスを取得することが可能である。これは本稿第4章の第2節で後述する機能を付化するのに必須のものである。

3. 行動要素オブジェクト

行動要素オブジェクトは図3に示す通り、抽象クラス Behavior を継承して作られる。

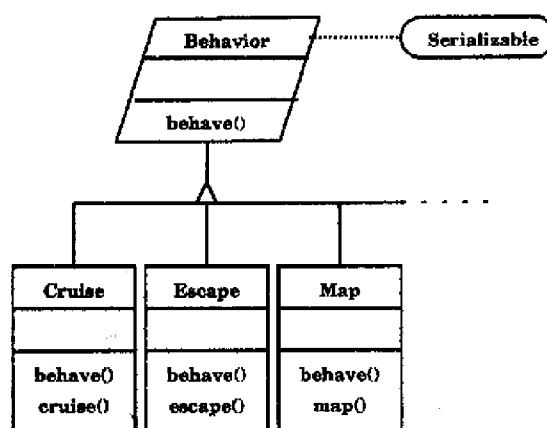


Fig. 3 行動要素オブジェクトのクラス図

Behavior クラスの behave() メソッドは抽象メソッドであり、各行動要素オブジェクトはこのメソッドの実装することで自身の機能を定義し、また、このメソッドの呼び出しによって機能する。新たに行動要素を作るときは、この Behavior クラスを継承し、behave() メソッドを実装すればよい。例えば移動に

関しての行動要素はそれぞれ, Cruise:移動, Escape:回避, そして Map:現在位置確認の3つを想定している。また, java.io.Serializable インターフェースを実装することで直列化が可能なオブジェクトとなり, ネットワーク上を移動することが可能である。

ロボットは図4にあるように, オブジェクトとして自らの行動を取り扱うこととなる。それぞれの行動要素オブジェクトはそれ自身が一つの行動となり, あるいは他の行動を含むことで作業の手順を表現することも可能である。そうしてロボットの行動選択機構において機能する。

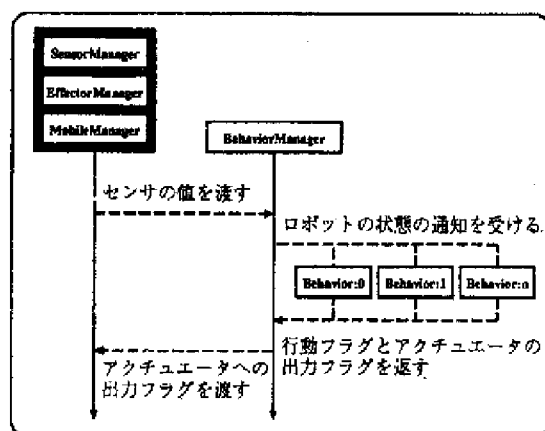


Fig. 4 行動要素オブジェクトによる行動選択

本研究で提案するシステムを大まかに描くと図5のようになる。

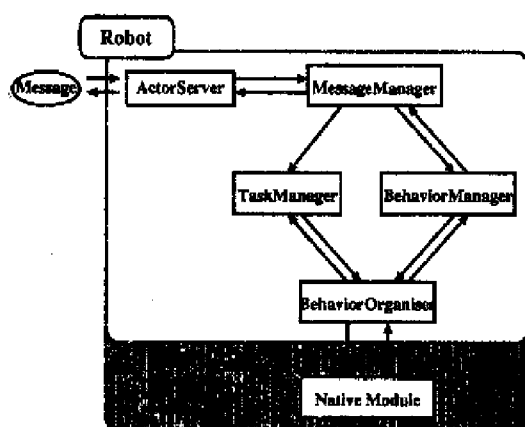


Fig. 5 行動要素オブジェクト授受システムおよび実行環境概観

行動要素オブジェクトを実際に運ぶのは Message オブジェクトの役目であり, 作業の手順かあるいは行動要素を示すものに応じて MessageManager が

振り分ける。作業の手順を示すものは TaskManager に送られることになる。行動要素オブジェクトを運んで来た Message は, BehaviorManager に送られる。それが行動要素インスタンスであるならば, インスタンスが取り出され BehaviorOrganiser に送られる。もし, 行動要素クラスであるならば, BehaviorManager のクラスローダによってクラスオブジェクトを取得することとなる。また, BehaviorManager は MessageManager から ActorServer を通じて他のロボットに行動要素オブジェクトを送る。BehaviorOrganiser は行動選択過程として機能し, NativeModule からロボットの状態を得たあと, 次を取る行動を決定し, そこに出力フラグを送る。NativeModule はセンサやアクチュエータを担うモジュールである。

4. 行動要素オブジェクト授受システムの利用形態

行動要素オブジェクトの授受には, 以下に述べるように大きく4つの利用形態が考えられる。「行動要素を受け取る」ことに注目した場合が前半の2つであり, 「行動要素を送る」ことに注目したのが後半の2つである。

4.1 パラメータを変更した行動要素の授受

ロボット A が新しく設定したいパラメータによって, 行動要素の新しいインスタンスを作成する。そうしてロボット B に転送する。

4.2 未知の行動要素を授受

ロボット A がある行動要素インスタンスをロボット B に送ったが, ロボット B がその行動要素クラスを持っていないため, 受け取ることができない事態となる。そのとき, ロボット B はロボット A からその行動要素クラスのバイトコードを転送してもらい, それを受け取ったのちレゾルブし, クラスオブジェクトを取得し, 今後はそのインスタンスを利用

できるようになる。

4.3 行動要素の分配

ロボット A は現在利用中の行動要素インスタンスのクローンを作成し、それをロボット B に転送する。

4.4 行動要素の委譲

ロボット A は現在利用中の行動要素インスタンスのクローンを作成し、それをロボット B に転送する。ここまでならば分配とおなじだが、委譲の場合は元の行動要素インスタンスを消してしまう

5. 行動要素オブジェクト授受システムの動作シミュレーション

本稿第4章の第1節および第2節で提示した利用形態について、図6にあるようなシミュレーションを行った。

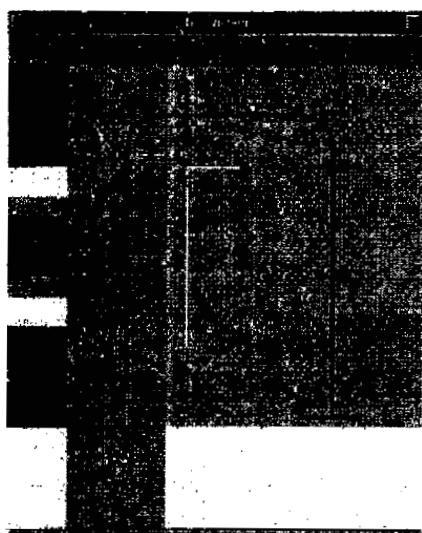


Fig. 6 行動要素オブジェクト授受機構シミュレータ

ここで登場する移動ロボットは対向2車輪型の移動モジュールをもち、プリミティブな制御機構、すなわちモーターへ出力することとエンコーダのバルスカウンタを得るといったことができる。ここで移動ロボット A は行動として Cruise:移動 オブジェクトと Escape:回避 オブジェクトをもっているとする。このロボットは前後に接触センサがあり、障害

物に当たったことを感知できる。普段はCruise によって移動し、センサが反応すれば Escape によって回避行動を取る。Cruise は移動モジュールに直進のための出力命令をだし、Escape は回避行動をとるための一連の動作の命令を移動モジュールにだす。Escape はロボットのたとえば前部の接触センサが反応したとき、一定時間の後退後、旋回して向きを変更し、その挙動を終える。この Escape オブジェクトは旋回方向をフラグとして持っており、その値が right ならば右旋回、left ならば左旋回を行う。初期値は left である。

5.1 パラメータを変更した行動要素の授受

ここで、移動ロボット B がおり、このロボットも回避行動として Escape という行動要素オブジェクトを持っている。移動ロボット B がその活動において、Escape の旋回方向が right であるとする。ここで、移動ロボット B が自身が利用している回避行動をロボット A に送るとする。このとき、以下に示す手順を経ることになる(図7, 図8)。

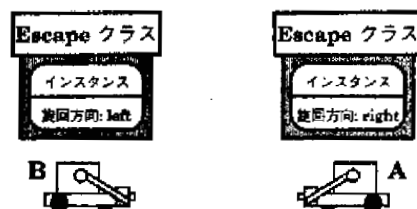


Fig. 7 A と B の行動要素オブジェクトの初期状態

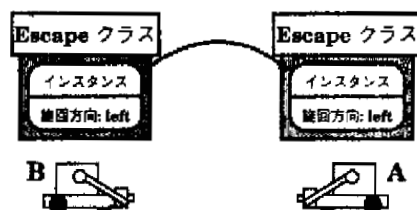


Fig. 8 B から A へ行動要素インスタンスを転送

5.2 未知の行動要素を授受

回避行動として EscapeRandom をもった移動ロボット C がいるとする。この行動要素オブジェクト



Fig. 9 A と C の行動要素オブジェクトの初期状態

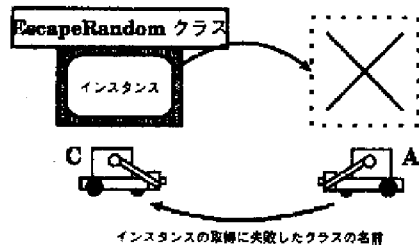


Fig. 10 C から A へ行動要素インスタンスの転送

と先程の Escape との違いはその旋回方向をランダムに決定するという点である。このオブジェクトは Escape とは異なる設計である。移動ロボット C がこの行動要素 EscapeRandom オブジェクトをロボット A に伝授する必要があるとする。このとき、まずは、ロボット C で利用している EscapeRandom インスタンスをロボット A に渡そうとする(9)。

しかし、ロボット A は EscapeRandom クラスを持っていないため、そのインスタンスを受け取り、利用することができない(10)。

そのため、ロボット A は EscapeRandom クラスをロボット C に転送してもらうよう要求を出す。ロボット C はその要求に応じて EscapeRandom クラスのバイトコードをロボット A に渡す。ロボット A は受け取ったバイトコードから EscapeRandom クラスを取得し、自前のクラスローダーにあるデータベースに追加し、今後は、EscapeRandom オブジェクトを利用できるようになる(11)。

そうして新たにインスタンスを獲得し、行動選択機構を変更する(12)。

もし、あたらしく受け取ったクラスが既に持っているクラスの機能と重複する場合、古いクラスを捨ててあたらしいクラスを利用するようにすることも可能である。このシミュレーションでは、EscapeRan-

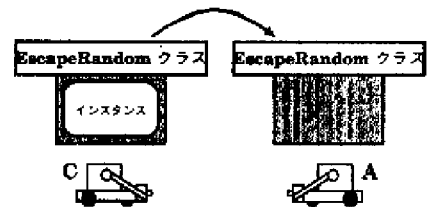


Fig. 11 C から A へ行動要素クラスの転送

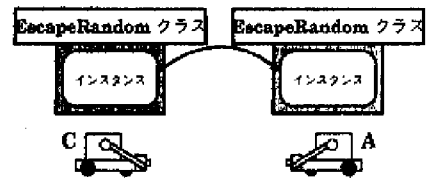


Fig. 12 C から A へ行動要素インスタンスの転送

dom クラスを得たことで、Escape クラスを利用可能な行動要素オブジェクトのクラスデータベースから消去し、行動選択機構において、入れ換えをしている。

5.3 考察

上述のように現段階では、既存のクラスあるいはインスタンスとぶつかる行動要素オブジェクトを受け取ったとき、上書きすることで対応している。これでは、既存のオブジェクトを利用した方が有効なときに、それが無駄になってしまう。今後は利用可能な行動要素オブジェクトを格納するデータベースにより、状況によって使い分けたり取捨選択するような評価機構を持つべきであろう。より柔軟にタスクに対応する自律ロボットのための行動要素オブジェクト授受システムの発展には不可欠な機能であると思われる。

6. 結言

本研究では、複数ロボットのために行動に関する情報を授受できるシステムの構築を目指し、分散オブジェクト環境を利用した行動要素オブジェクト授受システムを提案し、その基礎部分を示した。オブジェクト指向概念の導入により、ロボットの行動、その選択過程および授受機構を無理なく接続するこ

とができ、またオブジェクトで表現していることからモジュール性が高いため、今後の進展における拡張にも十分対応できると考えられる。また、JAVA言語を利用したことにより、ロボットが稼働中であっても未知の型の行動要素を受け取り、実行することが可能となった。

今後の課題としては、行動要素オブジェクトによってタスクを表現できるように拡張することがあげられる。この作業を行ったのち、最終的には実ロボットにおいて本システムを運用し、その有効性を検証することが必要である。

参考文献

- 1) R. A. Brooks: A Robust Layered Control System for a Mobile Robot, IEEE Robotics and Automation, 2-1, 14/23 (1986)
- 2) Lynne E. Parker: Task-Oriented Multi-Robot Learning in Behavior-Based Systems, IROS, 1478/1487 (1996)
- 3) 石岡宏治, 関一夫, 安西祐一郎: 複数の自律移動ロボットの個体差を考慮した地図獲得システムの設計と実装, 日本ロボット学会誌, 12-6, 68/78 (1994)
- 4) Ming Tan: Multi-agent reinforcement learning: independent vs. cooperative agents, Machine Learning, 330/337 (1993)
- 5) Keitaro Hanada, Takayuki Takahashi, Eiji Nakano: Software Architecture for Constructing a Mobile Robot with Highly Independent Modules, The Fourth Int. Conf on Control, Automation, Robotics and Vision (ICARCV'96), Singapore, Dec. 3-6, 1996. 2, 858/862 (1996)