

# 分散処理システムを用いたロボットシステムの構築

## Construction of the robot system using a distributed processing system

○学 藤田 和友 (弓削商船) , 正 前田 弘文 (弓削商船)

Kazutomo FUJITA, Yuge National College of Maritime Technology, 1000 Yuge Shimoyuge, Kamijima-cho Ochi-gun, Ehime, 794-2506, Japan

Hirofumi MAEDA, Yuge National College of Maritime Technology, 1000 Yuge Shimoyuge, Kamijima-cho Ochi-gun, Ehime, 794-2506, Japan

**Key Words:** Distributed Processing System, Shared memory, Marionette, Robot System, Rescue Robots

### 1. 緒言

近年の飛躍的な半導体技術の発展により、小型で高性能なコンピュータが広く普及している。また、多くのロボットシステムにおいて、これらのコンピュータが制御部分として使用されることが多い(組み込みシステム)。そのため、システムが複雑化し、プログラムや電子回路、機械要素といった広い総合的な知識が必要とされる。そこで、システムの複雑化を回避するために、それぞれの要素をモジュール化する手法が用いられる。

我々もこれまでにレスキューロボットを中心としたモジュール化を行ってきた<sup>(1)(2)(3)</sup>。レスキューロボットに用いられているシステムは、共有メモリを用いた分散処理システム(以下、Marionette)であり、メンテナンス性を意識したものとなっている。そのため、各モジュールが独立したプログラムとなっており、それぞれの機能はいたってシンプルなものとなっている。またこれまでに、分散処理システムとレスキューロボットシステムが混合した状態で存在していたソースコードを一般化するとともに、他のロボットシステムに転用できる状態まで開発を進めてきた。

本研究では、Marionette を容易に流用するための Marionette 自動生成プログラムについて述べる。また、モビリティロボットを用いた実装例についても述べる。

### 2. Marionette システム

Marionette は、糸を使った操り人形を語源としている。操り人形の糸は、各部位を繋ぐ横糸と人形全体を操るための縦糸に分かれる。また、縦糸は人形を操作するために一本もしくは複数の棒によって接続されている。そのため、棒を操作することで人形全体を操ることが可能となる。

Marionette も操り人形を模擬しており、各プロセスと Marionette は縦糸で、各プロセス同士は横糸で接続された形をとる(図1)。これにより、一つ一つのプロセスは、人形の腕や足のように、それぞれの担当する小規模な機能を実装するだけで済み、プログラムをシンプルに構成することが可能となる。

以下に、Marionette における制約を示す。

- ・プロセスは1対1の関係を維持する。
- ・共有メモリはセマフォによって管理する。
- ・プロセスの起動および終了は同期する。

これらの制約を実装するために、Marionette 本体では、そ

れぞれのプロセスを同時に起動するとともに、終了の際も同期を取って同時に終了を行う。この管理を行うものが、先に示した縦糸に該当する共有メモリである。また、プロセス間のデータ通信に用いる共有メモリについても Marionette 本体が管理する。そのため、Marionette 以外が共有メモリを管理できないよう、Marionette は各セマフォと共有メモリのキーをそれぞれ2個ずつ、プロセスの起動時に必要なプロセスのみに受け渡す形を取る(図2)。これによってシステム全体におけるバグの発生を抑制している。

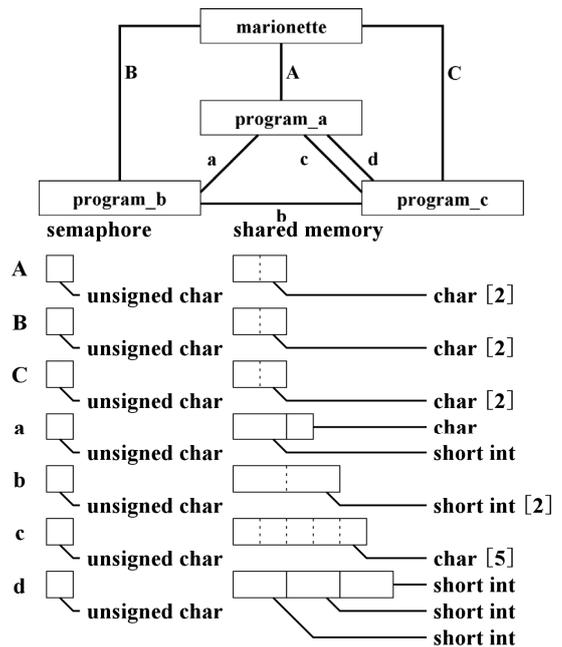


Fig. 1 View of Marionette

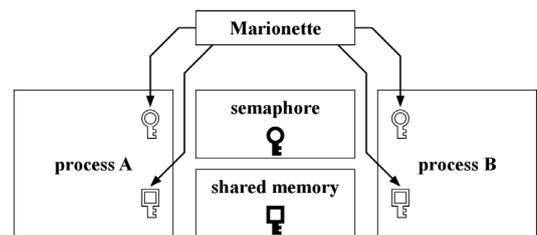


Fig. 2 Delivery of the Key in Marionette

### 3. 自動生成プログラム

図3に Marionette のソースコードを自動生成するプログラム (Marionette World Creator : MWC) のファイル構造を示す。mwc が自動生成プログラムの実行ファイルであり、init ディレクトリに収められた ini ファイルの情報をもとに、data ディレクトリのデータファイルと合成することで、ソースコードを生成する。src ディレクトリ内は mwc のソースファイル、doc ディレクトリ内はドキュメントファイルが格納されている。実際に自動生成されたソースコードは全て tags ディレクトリに格納され、ユーザはそれぞれのディレクトリにある main.cpp を変更することでプログラムの改良を行う (marionette ディレクトリを除く)。

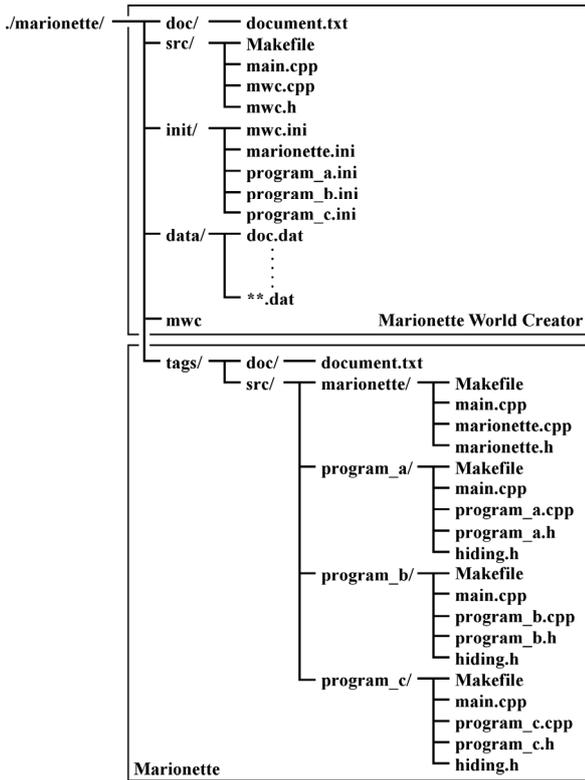


Fig. 3 File Composition of MWC

### 4. 実装例

図4に Marionette システムの実装を行うモビリティロボット (Orange-Caracara) を示す。Orange-Caracara は PlayStation Portable (PSP : ソニーコンピュータエンタテインメント社製) で動作する。ここで、Marionette システムを用いて、コントローラを PSP から Wii リモコン (任天堂株式会社製) に変更する。



Fig. 4 Orange-Caracara

図5にシステム構成を示す。Wii リモコンは PC を解して、無線 LAN (IEEE 802.11b) から Orange-Caracara の WiPort に命令を送る。WiPort は受け取ったデータをシリアルデータとして、マイクロシリアルサーボコントローラに送信することで、PWM に変換しそれぞれのサーボモータを動かす。なお、KRS-4034HV ICS (近藤科学株式会社製) は、無限回転が行えるモータである。

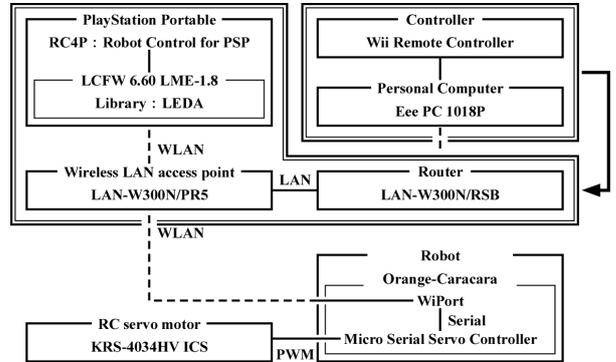


Fig. 5 System Configuration

PC 内のプログラムは図6に示すように、marionette を含めて4つのモジュールに分割した。controller connection モジュールは以前作成したレスキューロボットのプログラムより移植し、robot connection モジュールは PSP のプログラムを流用した。結果、central processing モジュールのみの作成だけでロボットの制御が可能となり、Marionette および MWC の再利用性と利便性を確認できた。

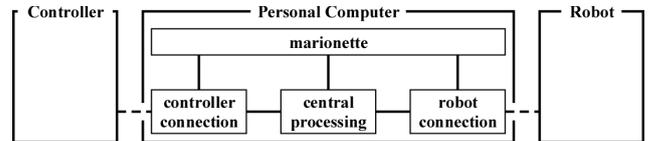


Fig. 6 Program Configuration

### 5. 結言

本研究では、MWC について述べた。また、モビリティロボットを用いた実装例についても述べた。今後は、自動生成プログラムの設定部分に GUI を用いるなど、よりユーザに使いやすいものに改良していく予定である。

### 文献

- (1) 石井 良典, 大坪 義一, 小林 滋, 小林 泰弘, 山本 祥弘, 梅田 栄, 海藻 敬之, 前田 弘文, 高森 年, 田所 諭, “閉鎖空間内探索ロボットのための遠隔操縦システムの開発”, 第11回システムインテグレーション部門講演会(2010), pp.1238~1241.
- (2) 前田 弘文, 小林 滋, 高森 年, “レスキューロボットにおけるデバイス管理を容易にするためのシステム開発”, 弓削商船高等専門学校紀要第34号(2012), pp.48~53.
- (3) 藤田 和友, 百垣 愛弓, 前田 弘文, “メカトロ教育のための分散処理システム”, 第13回システムインテグレーション部門講演会(2012), pp.1213~1218.