

共有メモリを用いた分散処理システム Marionette の開発

Development of the distributed processing system "Marionette" using a shared memory

○正 前田 弘文 (弓削商船)

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

Key Words: Rescue Robots, Software, Shared memory, UMRS-2010, Marionette

1. 緒言

近年、レスキューロボットの多くが、操縦者が直接ロボットに接近した状態で操縦を行うのではなく、操縦者が遠く離れた場所から無線を用いて遠隔操縦する形を採用している⁽¹⁾。

しかしこのことは、レスキューロボットを開発する上で、ロボット内部にローカルで複雑な制御システムを構築することを意味する。また、ロボットを遠隔操作する必要があることから操作卓についても、通信を含めた複雑なシステムを構築する必要が出てくる。

そこで我々は、これらの問題を解決する手段として、ロボット内部のデバイス制御を制御ボードの外部に設けるモジュール化を行ってきた。さらに、共有メモリを用いた分散処理システム Marionette の開発を行い、このシステムを用いてレスキューロボットの操作卓プログラム部分の再構築を行ってきた⁽²⁾。今回本論文において、Marionette をレスキューロボット以外にも適用するための一般化を行った。以下に、レスキューロボット UMRS-2010 を例として、Marionette について述べる。

2. ロボットシステム

非営利活動法人国際レスキューシステム研究機構にて開発した UMRS-2010 を図 1 に示す。

UMRS-2010 は、地下街で発生した災害に対応することを目的とした探査ロボット UMRS-2009 の後継機にあたる。

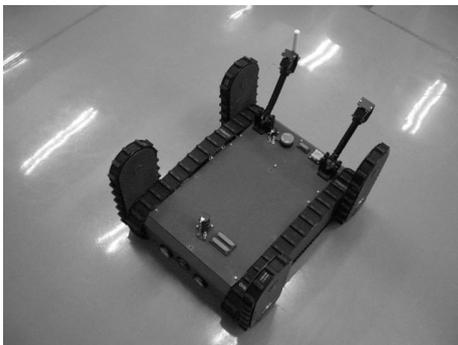


Fig. 1 UMRS-2010

図 2 に UMRS-2010 の内部構造を示す。図 2 において、全てのブラシレスモータ(マッスル株式会社 COOL MUSCLE 2 CM2-C-56B20A-K, CM2-C-60A10A-R)は、デジチェーンによって結合され、シリアル(D-Sub9pin)によってに制御用 PC(ICOP I.T.G. 株式会社 VDX-6314-512)に接続される。また、4 つのカメラ(MAXWIN 小型カラーバックカメラ CAM11A)はビデオサーバ(ACTi ACD-2000QT)を介して、

LAN により接続される。同様に、光ファイバージャイロ(日立電線株式会社 HOFG-OLC-1)についても、組み込み用超小型デバイスサーバ(Lantronix 株式会社 XPort)を用いてシリアルを LAN に変換した後、接続される。さらに、加速度センサ(クロスボー株式会社 CXL04GP3)および LED(三菱電機オスラム株式会社 DP03A-W4-754)については、それぞれ CR フィルタと LED ドライバを介した後、COOL MUSCLE 2(以下、CM2)経由で VDX-6314-512 に接続される。なお、操作卓には NEC Corporation ShieldPRO FC-N22A/BX6SS1B を、無線通信には株式会社パツファロー WLI-UC-GNHP を使用している。

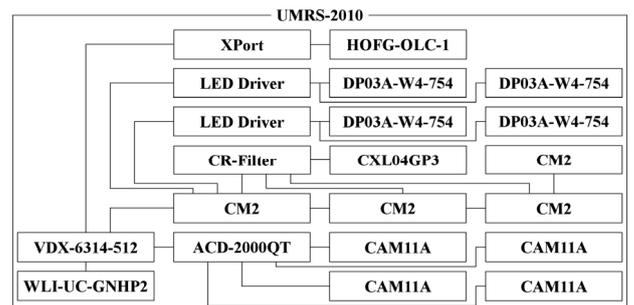


Fig. 2 Inner structure of UMRS-2010

3. ソフトウェア

本ロボットシステムのプログラム構成の概略を図 3 に示す。ソフトウェアの開発には、操作卓、ロボット本体ともに Ubuntu 10.04 LTS 上で動作するものとし、開発言語には C 言語を用いる。なお、操作卓側については、共有メモリを用いた分散処理システム Marionette を使用している。

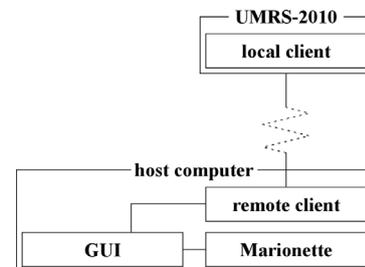


Fig. 3 Overview diagram of program for robot system

3.1 ロボット本体プログラム

本ロボットは台車(ベース)と呼ばれている部分に該当し、現場調査において必要とされる必要最低限の機動力を確保するものである。以下に搭載機能を示す。

- ・操作卓4カメラ画像表示
- ・ロボット本体の姿勢検知
- ・ロボット本体の速度, 角速度
- ・モータ制御
- ・電波強度検出
- ・バッテリー残量検出
- ・自己位置推定
- ・LED点滅制御
- ・現在位置表示(未実装)
- ・通信切断時の対処

3.2 分散処理システム

分散処理システム 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(図4). 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.

本 Marionette も操り人形を模試しており, 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5). 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なり, 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.

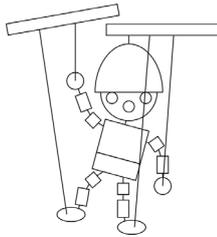


Fig. 4 Overview diagram of the puppet

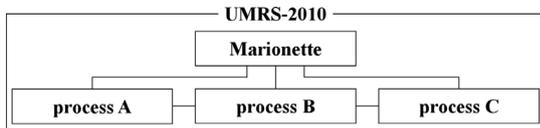


Fig. 5 Overview diagram of Marionette

実際にプログラムを実装する際は, 縦糸と横糸は共有メモリとなり, プロセス間同士, プロセス・Marionette 間はそれぞれ1対1の関係となっている. 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を使った排他制御によって管理する.
- ・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処理を行う. 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6). 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.

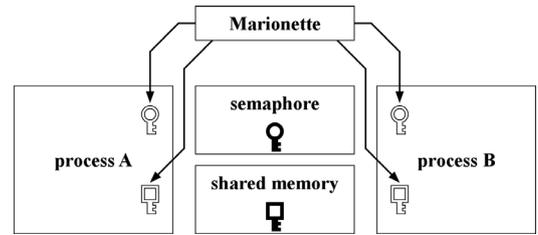


Fig. 6 Delivery of a key

3.3 操作卓プログラム

Marionette をベースとした操作卓プログラムの構成を図7に示す. 制御系とカメラ画像処理系は, ほぼ独立した形を取り, 操縦者との GUI 表示部分において統合される. 最後に Marionette を使って作成した UMRS-2010 の操作卓画面(GUI画面)を図8に示す.

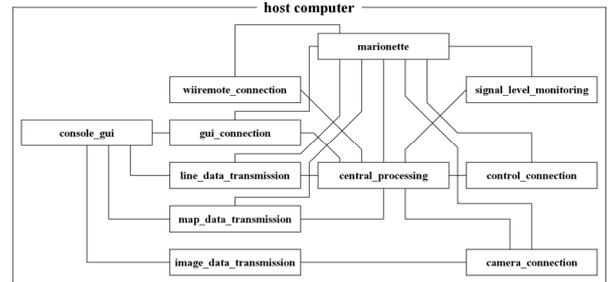


Fig. 7 Overview diagram of program for console

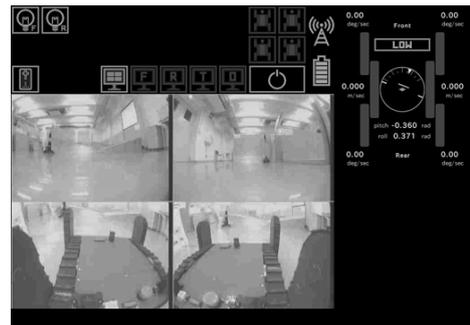


Fig. 8 Console screen

4. 結言

今回我々は, 分散処理システム Marionette をソースレベルで一般化し, 他のシステムにも容易に適用できる状態を構築した. また, 実装例を上げ, Marionette の内部構造についても触れた.

しかし, 今回の一般化だけでは十分とはいええず, Marionette 実装にかなりの時間を費やすという問題点が残る. そこで, 今後はこれらのソースコードを自動生成するソフトウェアの開発を行っていく予定である.

文献

- (1) 永谷 圭司, 岡田 佳都ら, “火山探索を目的としたクローラ型移動ロボット Kenaf による桜島での遠隔操作実験”, 第10回システムインテグレーション部門講演会(SI2009)(2009), pp.1943~1946.
- (2) 前田 弘文, 藤長 大祐ら, “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分散処理システムの開発”, 第12回システムインテグレーション部門講演会(SI2011)(2011), p.60~63.