

<p>研究テーマ</p>	<p>共有メモリを用いた分散処理システム Marionette の研究開発 ～第1報：ソースコードの一般化～</p>
<p>学 生 名</p>	<p>百垣愛弓</p>

1. 緒言

特定非営利活動法人国際レスキューシステム研究機構(以下, IRS)は、近年のロボットシステムの複雑化に伴い、共有メモリによる分散処理システム Marionette の開発を行ってきた<sup>(1)</sup>。このシステムは、実際にレスキューロボットに搭載された実績を持ち、初級プログラマでも容易に理解できる構造となっている。しかし Marionette は、レスキューロボットに特化した専用ソフトであり、Marionette 部分のソースコードがむき出しとなっていることから、汎用性があるとは言えない。そこで本研究では、この 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足がかりとする。なお本発表では、Marionette の概要とプログラム構造、およびソースコードの一般化について述べる。

2. 分散処理システム

分散処理システム Marionette は、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。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。なお、Marionette はシステムを複雑化しないために、以下の3つの制約がある。

- ・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。

最後に Marionette システムの実用例として、IRS 製 UMRS2010 の操作画面を Figure 1 に示す。

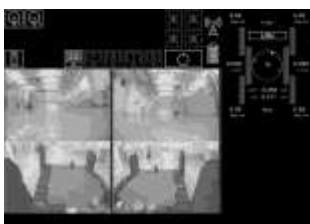


Fig. 1 Operation screen

3. プログラム構成

今回プログラムを作成するにあたって、Marionette を含む4つのプロセス間通信を行うものとした

(Figure 2)。また、それぞれのプロセス間通信は共有メモリを用いた。なお、それぞれの共有メモリにはセマフォを設けるとともに、型宣言も行っている (Figure 3)。

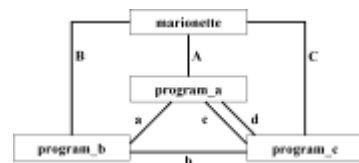


Fig. 2 Process configuration

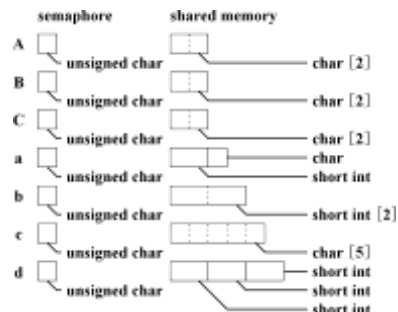


Fig. 3 Memory configuration

4. ソースコードの一般化

ソースコードの一般化に伴い以下の4つの作業を行った。これは、Marionette の汎用性を高める他に、Marionette を熟知していないユーザでも簡単に扱えるようにするためである。

- ①marionette 部分のソースコード抽出
- ②セマフォの配列部分の切り離し
- ③ソースコードの関数化
- ④マクロによるソースコードの隠蔽化

5. 結言

今回、Marionette のソースコードを一般化することで、他のシステムに適用できる状態を構築した。しかし、他のシステムに実装するためには、ソースコードを手動で記述していかなければならない。そのため、多くの時間を要することとなる。そこで今後は、ソースコードを自動生成するプログラムを構築していく予定である。

文献

- (1) 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, 紀要第34号, pp.48~153, 2012