高度なメカトロニクス教育のための分散処理システム

藤田　和友　　生産システム工学専攻 2 年　124007
指導教員　　前田　弘文

Distributed processing system for advanced mechatronics education

Kazutomo Fujita and Hirofumi Maeda (Adviser)

ABSTRACT

This paper describes about distributed processing system "Marionette" for advanced mechatronics education. The origin named a "Marionette" is from the marionette of a puppet play. In developing a program, this system can modularize each function. Therefore, it is possible to shorten the time spent on the design and development of the system. As a result, it can learn more efficiently by using this free time. The effect was actually verified by the experiment using a robot.

1. 序　論

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普及している．また，多くの産業用機械でこれらのコンピュータが制御部分に組み込まれている．そのため，企業では複雑なシステムを開発するために高度なメカトロニクス分野に強い人材が求められている．メカトロニクスは機械技術や電子技術，ソフトウェア，制御技術を含む総合技術であり，ロボット制御のような高度なメカトロニクス技術を学ぶためには多分野の技術の修得が必要不可欠である．高等専門学校においても，様々なメカトロニクス教育が実施されている．本校もまた，「創造性実験」と称する科目が開設されており，本科 4 年生を対象に組み込みマイコンを用いたシステム開発の実験実習が行われている．しかし実際に行われている実習は，設計や部品の発注などを実習に取り組む点において有効であるが，実際に製作したシステムは簡単なセンサの読込やモータの制御等，電子工作のレベルに近いシステムが大半を占める．メカトロニクス教育の目的は，ハードウェアの改良やソフトウェアのデバッグ等の試行錯誤を通して，高度な複合技術と工学的センスを修得することである．これは評価結果をフィードバックし，完成度を高めることができるシステムを構築する体験がなければ成し得ない．

そこで，高度なメカトロニクス教材を開発するために，過去に開発されたレスキューロボットに搭載されている分散処理システム(以下，Marionette)に着目した[1]~[3]．この Marionette は，ロボットに搭載される多種多様な機能を統合管理するために，プログラムを機能ごとのモジュールに分割し並列に処理を行うシステムである．このシステムを利用して，より高度なメカトロニクス教育を行うための教材の検討と開発を行う．

研究では，教材システムとして Marionette を使用するために，レスキューロボット専用のソフトウェアである Marionette のソースコードの汎用化を行い，Marionette を他のシステムに利用するための改良を行った．さらに，Marionette のプログラム開発の効率化を図るために，ソースコードの自動生成プログラム Marionette Would Creator(以下，MWC)を作成した．また Marionette と MWC の再利用性と利便性を確認するために，実際に MWC を用いてモビリティロボットの制御プログラムを作成し，メカトロニクス教材としての有用性について評価した．

2. メカトロニクス教育

2.1　メカトロニクス教育

メカトロニクスとは，機械技術や電子・電気技術，情報処理技術等を統合した総合技術分野である．近年の半導体技術の発展に伴い，多くの産業機械や製品などに小型で高性能なコンピュータが組み込まれ，メカトロニクス技術を用いた製品が開発されている．そのため，それらを開発・管理するために要素技術を修得できるメカトロニクス教

育の重要度が増加している．また，ロボット製品が産業に深く関わることから，より高度なメカトロニクス技術の修得が必要である．

2.2　メカトロニクス製品の開発サイクル

メカトロニクス製品の開発は，目的とする製品の検討から始まり，制御システム・制御対象の設計および構築を経て製品として完成する．それらを分析した結果を踏まえて，検討・改善を行うことでシステム全体の完成度を高めていく（図 1）．この開発サイクルは，マネジメントにおける PDCA サイクルに類似しており，設計＝計画（Plan），構築＝実行（Do），分析＝評価（Check），検討＝改善（Action）に対応させることができる．

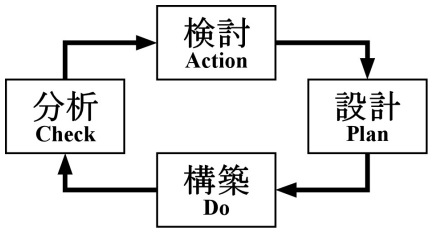

図 1：システムの開発サイクル

メカトロニクス教育において，高度なメカトロニクス技術の工学的センスを身に着けるためには，分析・検討に重点を置き，トライアンドエラーを繰り返しながら，開発サイクルをより多く回すことが重要である．しかし，メカトロニクス教育の現状は設計や構築の部分に多くの時間が費やされており，期待したほどの効果が得られていない．

2.3　創造性実験

メカトロニクス教育の具体例として，本校で行われている創造性実験を挙げる．創造性実験とは，本校の情報工学科 4 年生の授業で行われる組み込みマイコン Arduino（Arduino Software 社製）を用いた実験実習である．この創造性実験では，Arduino を用いたシステムの設計・開発を行い，実践的なプロジェクト管理を体験することを目的としている．実験は 1 グループあたり 2～4 人で構成され，年度後期に週 4 時間のプロジェクトに取り組んでいる．Arduino を用いたシステムを開発目標に，プロジェクトの計画，システムの概要設計，予算見積もり，部品の発注を行う．その後に計画や設計に応じてシステム開発を行い，最後に成果発表を行う．

このように，企画から開発，評価に至るプロジェクト開発に近い形で製品開発の実習を行うことができるが，実習期間が 13 週間と短いため，開発に必要な予備知識を習得することが難しい．そのため，完成した試作品は，センサの読み込みやモータ，圧電ブザー，LCD の出力制御等の電子工作レベルのシステムが大半を占める．また，成果発表で授業を終えるため，成果の分析を踏まえた改良を行う時間を確保できない等の問題点が挙げられる．

このことから，工学的センスを身に着けるためには図 1 の開発サイクルにおける設計・構築を事前に行い，分析・検討に重点を置いたトライアンドエラーをより多く行えるシステムを開発することができる教材が求められる．

そこで，多くの複雑な機能を持つレスキューロボットの開発に使用されている Marionette システムに着目した．本研究ではこの Marionette を利用することにより，開発目標のロボットシステムを機能ごとのモジュールに分割し，学生が開発に挑戦する機能を限定的に絞ることで，短期間により多くの開発サイクルを繰り返すことのできる教材を開発する．

3.　レスキューロボットの概要

Marionette を用いたシステムの実用例として，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されたレスキューロボット UMRS-2010（以下, UMRS-2010）を図 2 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である[4].

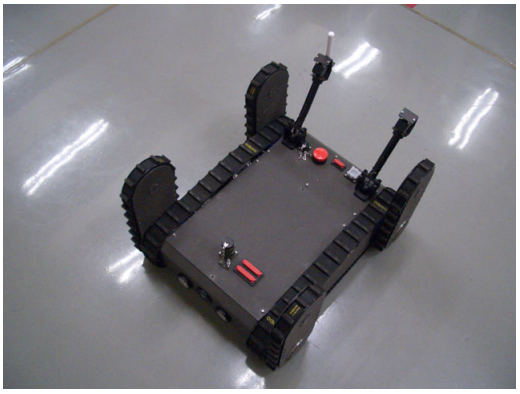

図 2：UMRS-2010

Marionette の開発・動作環境には Linux の頒布形式(ディストリビューション)の一つである Ubuntu(カノニカル社製)を使用している．そのため，本ロボットシステムのソフトウェア環境は，操作卓，ロボット本体ともに開発当時の最新の長期保証版である，Ubuntu 10.04 LTS を搭載している．また，開発言語には C 言語を用いる．なお，Marionette は操作卓側のプログラムに使用されている(図 3)．コントローラには若手のレスキュー隊員に馴染みが深いことから Wii リモコン(任天堂株式会社製)が使用されており，Bluetooth を用いて操作卓と接続する．また，操作卓と UMRS-2010 は無線 LAN によって接続する．

次に図 3 の Marionette をベースとした操作卓プログラムの構成を図 4 に示す．Marionette によってプログラムをモジュール化することにより，それぞれのモジュールの機能がシンプルな物となる．そのため，central_processing モジュール等の制御系と camera_connection モジュール等のカメラ画像処理系はほぼ独立した形を構成でき，操縦者との Graphical User Interface(以下，GUI)において統合される．console_gui は GUI を画面に出力するための外部プログラムであり，marionette によって管理されているプログラムではない．

実際に Marionette を使って作成した UMRS-2010 の操作卓画面(GUI 画面)を図 5 に示す．図 4 で実装された通信状況やカメラ画像の出力，ロボットの状態が GUI によって表示される．

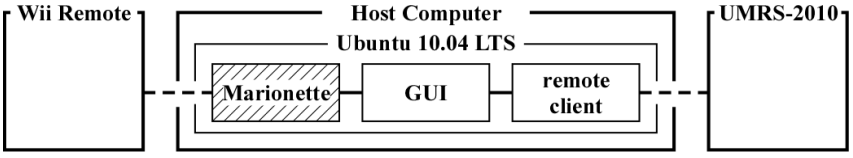

図 3：ロボットシステムのプログラム構成の概略

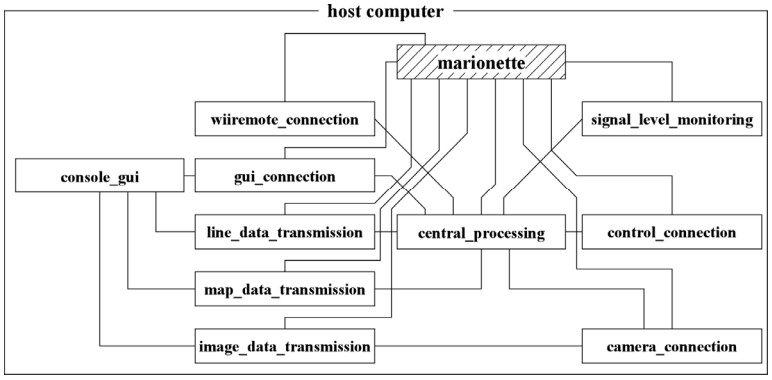

図 4：操作卓プログラムの概観図

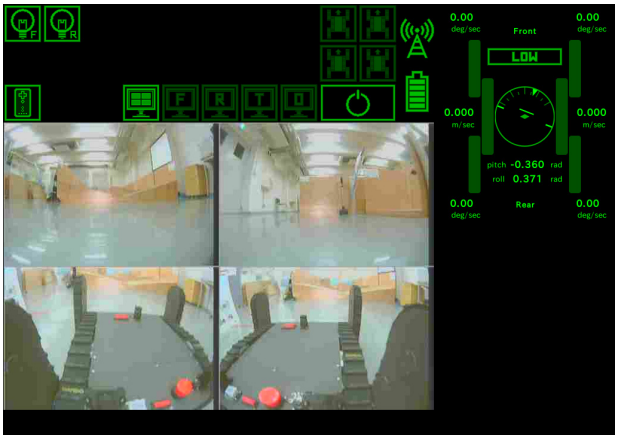

図 5：操作卓画面

4. 分散処理システム

本研究では分散処理システム Marionette を使用した教材システムを開発するために，Marionette の汎用化とソースコードを自動生成するプログラムの開発を行った．本章では分散処理システムの概要と，研究内容の詳細を述べる．

4.1 分散処理システムの概要

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を操るための縦糸に分かれる(図 6)．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 も操り人形を模擬しており，一つの大規模なシステムを手や足のような機能ごとのプログラムに分割し，Marionette によって一括管理を行う手法を採用している．そのため，各プログラムと Marionette は縦糸で，各プログラム同士は横糸で接続された形をとる(図 7)．

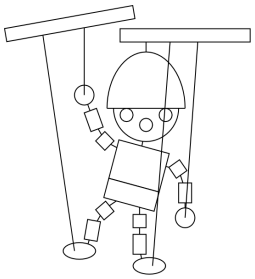

図 6：操り人形

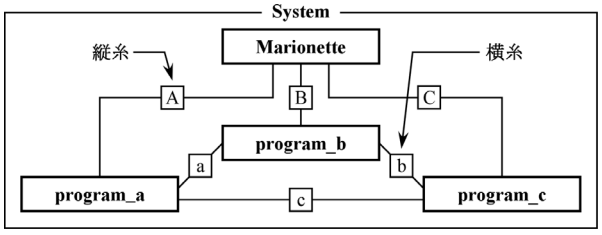

図 7：Marionette の概観

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る．また，プログラム同士間，プログラム Marionette 間はそれぞれ 1 対 1 の関係となっている．共有メモリとは複数のプログラムがアクセスできるメモリである．これによってプログラム間のデータの受け渡しを行っている.プログラム同士が 1 対 1 で接続されているため，共有メモリの受け渡す値が誤っているなどのバグが発生した場合，コードの記述ミスを捜索する範囲が接続されている二つのプログラムに限定されるため，デバッグ作業の時間を短縮できる．また，プログラム同士が独立しているため，過去に開発したプログラムを有効に活用できる．

以上の利点により，メカトロニクス教材に Marionette を利用することで，システムの設計と開発に費やされる時間を短縮することが期待できる．

4.2 分散処理システムの汎用化

レスキューロボット専用のシステムである Marionette をメカトロニクス教材として利用するためには，他のシステムに使用できるように汎用性を高める必要性がある．

そこで，本研究では Marionette の汎用化を行った．汎用化の主な作業として, UMRS-2010 のシステム内に存在する Marionette 部分のソースコードの抜き出しを行った(図 8)．

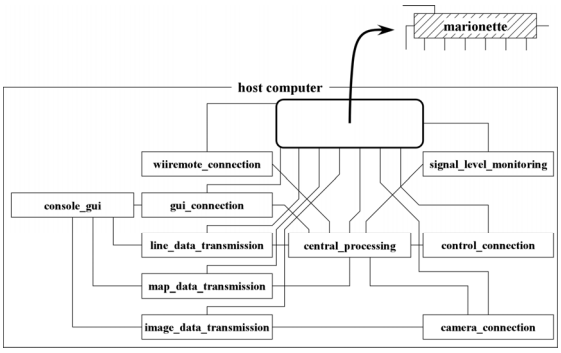

図 8：Marionette 部分のソースコード抜き出し

しかし，他の汎用化処理を行うには，UMRS-2010 のプログラム構成では接続しているプログラム数が多いため，図 7 のプログラム構成に編集し，汎用化処理を行った．

また，他の汎用化処理として，マクロによってソースコードの一部の隠蔽を行った．これにより，並列処理に必要な機能を下記のように記述すれば，ユーザが marionette を意識することなく使用することができる．

READ_[共有メモリ名]：共有メモリのデータを配列に格納する．

WRITE_[共有メモリ名]:配列データを共有メモリに格納する．

INITIALIZATION：初期化，キーの取得，シグナルの設定，共有メモリの接続を行う．

SHUT_DOWN : 終了要求があるか確認し，ある場合にプロセスを終了させる処理に移る．

ENDING : 終了フラグを 1 にセットし，共有メモリの解放を行う．

4.3 分散処理システムの自動生成

前節の汎用化を行った Marionette には，プログラムと共有メモリの定義に関係するコードが多く，プログラムや共有メモリの名前や個数を変更する際に，ソースコードの記述ミスが発生しやすい問題がある．そこで，ソースコードの自動生成プログラム Marionette World Creator（以下，MWC）を開発した．MWC の概観を図 9 に示す．図 9 の上部は MWC 本体を格納したディレクトリ（フォルダ）であり，下部は生成された Marionette である．

メカトロニクス教材として MWC を活用することで，システムの開発時間を短縮することができる．この時，教材システムの Marionette システムにおいて，Marionette 内に含まれる個別のプログラムをモジュールと呼ぶ．また，開発のターゲットとなる機能を有するモジュールをメインモジュール，それ以外の機能を有するモジュールをサブモジュールと定義する．MWC の主な使用手順を以下に示す．

① 図 7 のようなプログラム構成と各モジュールの機能を設計する．

② mwc.ini，marionette.ini，program_*.ini ファイルに①の構成内容を記述する．

③ mwc ファイルを実行すると，tags ディレクトリ下にシステムが生成される．

④ 生成されたシステムの各モジュールに，①で設計した機能を実装する．

⑤ 変更したモジュールを Makefile によってコンパイルする．

⑥ Marionette ファイルを実行することでシステムを稼働させる．

各モジュールにはメインプログラムの main.cpp ファイルや関数を含むヘッダファイル，コンパイルを行うための Makefile を備えている．各モジュールの main.cpp ファイルは共有メモリの初期化等，実行に必要最低限なソースコードが記述されているが，メインループ内は空白であるため編集の必要がある．その時に，過去に開発したモジュールから流用することにより，更に開発時間を短縮できる．

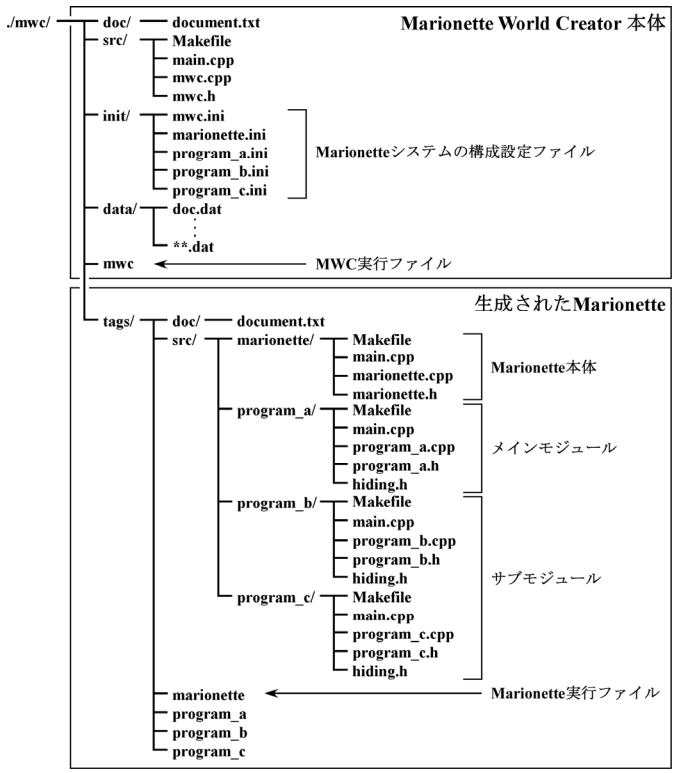

図 9：MWC の概観

5. モビリティロボットを用いた実装実験

高度なメカトロニクス教材を開発するために，モビリティロボットに Marionette を実装し，その動作実験を行い教材化の検討と評価を行った．

5.1 モビリティロボットの概要

図 10 に実験対象となるロボット（以下, Orange-Caracara）を示す[5]．Orange-Caracara は PlayStation Portable（PSP：ソニーコンピュータエンタテイメント社製）で動作する．この Orange-Caracara は中学生向けの広報のために開発されたロボットであり，操作に慣れている PSP を利用することで，ロボット技術への興味を持たせることを狙いとしている．また，操作を簡単にするために，機械部品としてモータと車輪を搭載し走行機能のみを実装している．この Orange-Caracara の操作プログラムは，PSP 用のソフトウェア開発キット（以下，PSP-SDK:PSP-Software Development Kit）によって C 言語で開発されている．

図 10：Orange-Caracara の外観図

図 11 に Orange-Caracara のシステム構成を示す．PSP からの制御命令は無線 LAN（IEEE 802.11b）によりアクセスポイントとルータを経由して，Orange-Caracara の WiPort へ送信する．WiPort とは組み込み用無線 LAN 対応デバイスサーバーである．WiPort により，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 Pulse Width Modulation（PWM）に変換し，それぞれのサーボモータを動作させる．

図 11 に Orange-Caracara のシステム構成を示す．PSP からの制御命令は無線 LAN（IEEE 802.11b）によりアクセスポイントとルータを経由して，Orange-Caracara の WiPort へ送信する．WiPort とは組み込み用無線 LAN 対応デバイスサーバーである．WiPort により，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 Pulse Width Modulation（PWM）に変換し，それぞれのサーボモータを動作させる．

5.2 実装実験

Marionette の再利用性を確認するために，UMRS-2010 と Orange-Caracara の二つの異なるロボットシステムの，Marionette による結合を行う．そこで，Orange-Caracara のコントローラを PSP から UMRS-2010 に使用されていた Wii リモコンに変更した．また，MWC をシステム構築の際に利用することで，MWC の利便性の確認も行った．図 12 に Marionette による Orange-Caracara のシステム変更の概要を示す．図 11 の PSP と無線 LAN アクセスポイント，ルータを Wii リモコンと PC に変更している．Wii リモコンは Bluetooth を介して Marionette を搭載した PC に接続し，無線 LAN によって Orange-Caracara に接続している．また，Orange-Caracara 本体の構成は従来の Orange-Caracara と同じである．

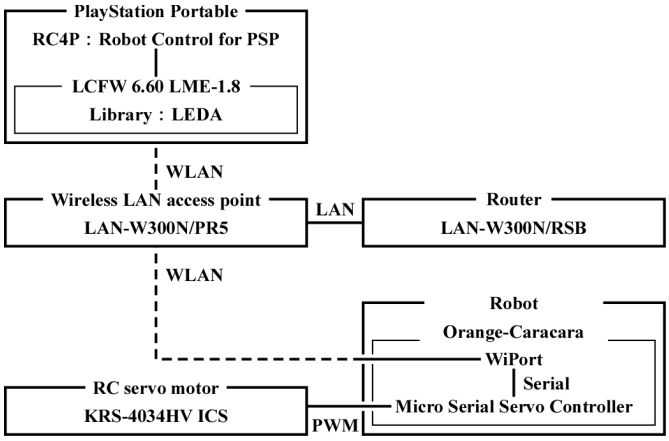

図 11：Orange-Caracara のシステム構成

図 12 において PC 内で実行している Marionette のプログラムを図 13 に示すように，MWC によって，marionette, controller connection，robot connection，central processing の 4 つのモジュールに分割した． controller connection モジュールは第 3 章で述べた UMRS-2010 のプログラムより，Wii リモコンと PC 間の通信プログラムを移植した．また, robot connection モジュールは Orange-Caracara の PSP とロボット

間の通信プログラムを使用し，Marionette システムに導入できる形に編集した．

controller connection は central processing へ Wii リモコンから所得したジョイスティックの座標値を共有メモリに介して受け渡す. central processing では受け渡された値を，前進，後退，左右旋回の命令と座標値に合わせたモータ速度の値に変換し，robot connection へ受け渡す．最後に robot connection では受け渡された命令と値に応じた PWM 信号を，Orange-Caracara に送信する．以上の流れで命令が伝搬していくため，共有メモリの値は一方通行になっている．また，実行中は marionette モジュールが各モジュールから送られる終了命令を監視しているため，値の受け渡しは発生しない．このようなプログラム構成により，central processing をメインモジュールとした場合，controller connection からの出力値と，robot connection への入力値を理解し，それらの値を変換するプログラムを記述するだけでロボット制御が可能となる．

また第 4.2 節の汎用化処理により，各モジュールのソースコードは Marionette の実行に関する記述を隠蔽化されていることから，各モジュールのメインループは非常に簡単なソースコードとなる．そのため，条件分岐やループ処理等のプログラミングの基礎知識のみで，システムの学習が可能である．

この実験の結果として，実際に本研究室で行った Orange-Caracara の PSP との通信制御プログラムの開発では，PSP-SDK の理解，通信手段や通信フォーマットの決定に，約 2 ヵ月間(約 240 時間)も費やしたことに対し, Marionette システムを用いた新しいロボットシステムは 4 日間(約 16 時間)で開発を終えることができた．この実験を評価したメンバーは教員 1 名と学生 1 名である. 両者とも Marionette の知識があり，学生は本校の生産システム工学専攻 1 年生である．また，完成したプログラムの central processing モジュールのプログラムコードは 200 行程度であり，メインループ内に条件分岐が記述されているだけの，非常に簡単なプログラムになっている．このことから，Marionette の再利用性と MWC の利便性を確認できた．

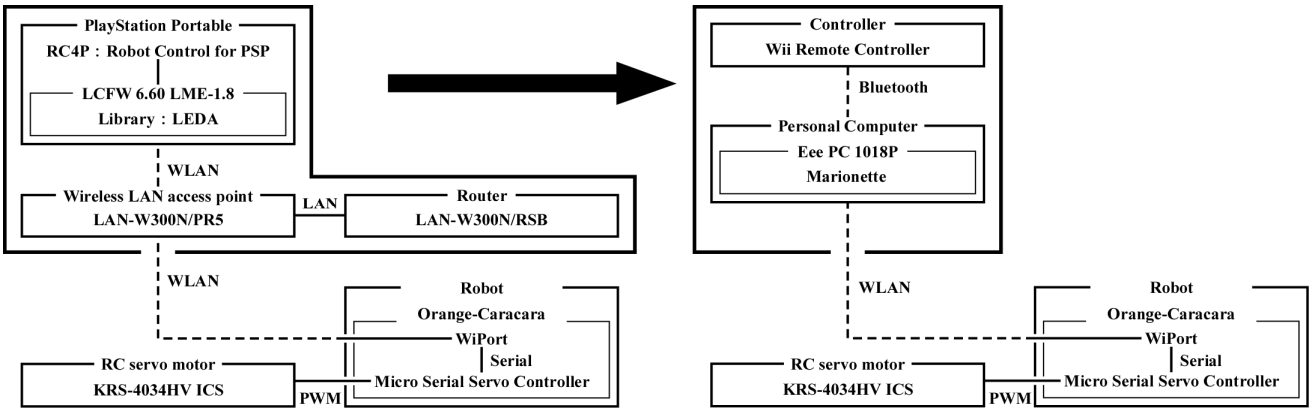

図 12：Orange-Caracara のシステム変更

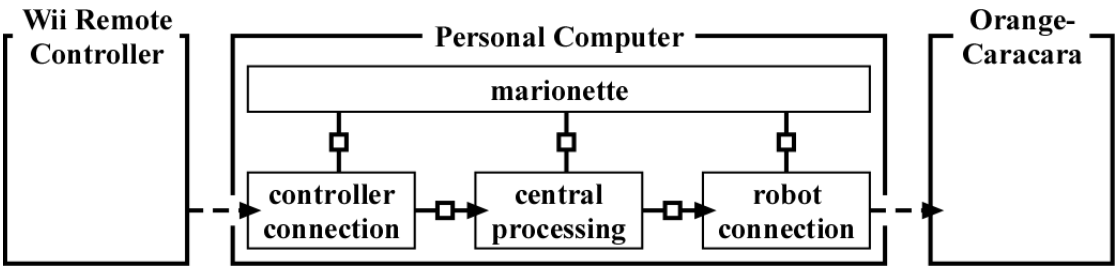

図 13：プログラム構成

5.3 分散処理システムによる教材システム

前節の実験に使用したロボットシステムを踏まえて，Marionette を利用したメカトロニクス教育のための教材システムについて考案する．

この教材はメカロトニクス分野の応用であるロボット開発に重点を置いているため，メカトロニクス分野の基礎を有する高等専門学校専攻科生や工学部 3 年生以上の学生を対象としている．この教材システムによって習得することのできる技術は，ロボットシステムを簡易的に開発することができることから，通信技術やモータ制御，演算処理，画像処理，センシングなどの情報工学分野が中心となる．また，ソフトウェアの開発時間を短縮できることから，電子回路設計や機械設計の開発に重点を置くことでハードウェアの完成度を高めるための時間を確保できる．

次にこの教材システムを使用した実験実習のモデル授業を図 14 に示す．

教員は実験実習の準備段階として，実習内容や専門分野に合わせた開発目標を立て，MWC によって教材システムを生成する．この際，生成された教材システムはプログラム構成が構築されただけの雛形の状態であるため，事前にサブモジュールを教員が開発しておく．システムの全サブモジュールが完成した時点で学生は実習を開始する．自動生成したメインモジュールを学生に配布し，実習としてメインモジュールのメインループ内処理のプログラム開発を行う．開発が終了したメインモジュールを教員が回収し，システムに組み込み実行する．完成したシステムの分析を行い，完成度を評価し，必要に応じて再度修正開発を行う．この修正を繰り返すことで，適正な制御値の算出や最適なアルゴリズムの探索を行い，より高い完成度を持つシステムの開発を短期間で行うことができる．

配布されたメインモジュールを開発するためには，Marionette 自体を完全に理解する必要はなく，メインモジュールに関係するサブモジュールとの入出力や，共有メモリの読み書きのための記述法など，Marionette の概要を勉強するだけでよい．そのため，プログラミング経験を有する学生であれば，20 分程度で開発を開始することができる．以上のことから，プロジェクト開発の授業や卒業研究等，プログラムに費やされる時間の短縮が求められる用途においても使用できる．また，各々の機能をモジュール化していることから，複数人で個別にプログラムを分担して開発することも可能となる．

最後に，メカトロニクス教材として，情報工学だけでなく機械工学や電子工学等を複合した実験実習を想定し，既存のハードウェアシステムに機能を追加する実験テーマについて考える．

機能の追加例として，図 13 のロボットシステムに距離センサとカメラユニットを追加し，障害物検知機能とカメラ画像表示機能を持たせたロボットシステムを図 15 に示す．斜線の入っているモジュールとユニットが新しく追加した機能である．PC とロボットの通信は既に完成したものであり，修正が難しいことから，range sensor と camera unit はロボット本体とは異なる Bluetooth 等の通信方法を採用することとする．また，ロボットへのユニットの追加は，ロボットの外部にユニットを設置するためのパーツの設計を行うものとする．この機能を追加する時，再び MWC によって Marionette を再構築するが，controller connection と robot connection は従来のモジュールを使用することができる．また，central processing は従来のモジュールにセンサ値に対する処理とカメラの入出力の機能を追加するだけでよい．さらに，camera connection と display connection は UMRS-2010 に使用された同種類のモジュールを使用できる．そのため，この例では sensor connection のみを新しく開発するだけでよいことが分かる．

以上のことから，プログラムの修正やデバッグ作業等の試行錯誤を通して，工学的センスやメカトロニクス技術を身に着けることのできる Marionette を使用した教材システムの実用性を確認することができた．

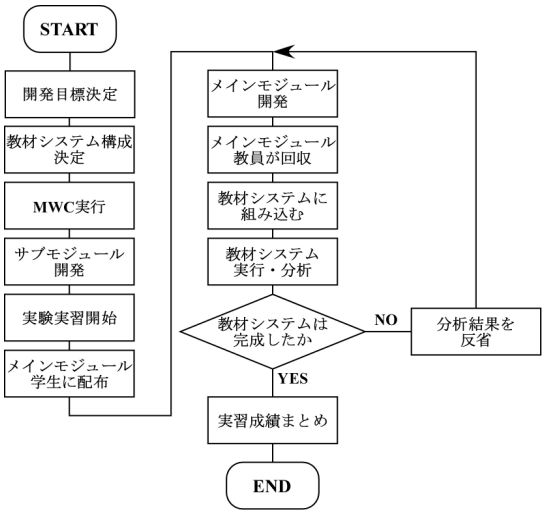

図 14：教材システムを使用した授業の流れ

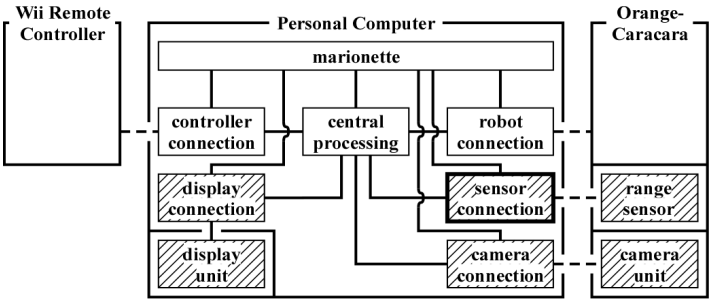

図 15：新機能を追加したロボットシステム

6. 結 論

本研究ではメカトロニクス教育に分散処理システムを利用して，高度なメカトロニクス技術を学ぶための教材システムを開発するために，UMRS-2010 に搭載されている Marionette システムの汎用化と自動生成プログラムの開発，モビリティロボットを利用した実装・評価実験を行った．

教材システムとして Marionette を使用するには，UMRS-2010 専用のソフトウェアである Marionette を汎用化する必要がある．そのため，UMRS-2010 のシステム内に存在する Marionette 部分のソースコードの抜き出しを行った．また，他の汎用化処理として，マクロによるソースコードの一部の隠蔽を行った．これらの汎用化処理により，並列処理に必要な機能をユーザが marionette を意識することなく使用することができる．

しかし，汎用化処理を行った Marionette はプログラムと共有メモリの定義に関係するコードが多く，プログラムや共有メモリの名前や個数を変更する際に，ソースコードの記述ミスが発生しやすい問題がある．そこで，ソースコードの自動生成プログラム MWC を開発した．これによって，Marionette システムの開発時間を短縮する上に，ソースコードの記述ミスが減少させることができる．

また Marionette 及び MWC の再利用性と利便性を確認するために，UMRS-2010 と Orange-Caracara の二つの異なるロボットシステムを結合したシステムを構築し，その動作確認を行った．Orange-Caracara へ指令を送信する PC 内のプログラムは marionette を含めた 4 つのモジュールで構成し，通信モジュールには UMRS-2010 のプログラムや Orange-Caracara の操作プログラムを使用した．その結果，一つのモジュール(central processing)の作成だけでロボットの制御が可能となった．また，Marionette によって一つの新しいロボットシステムを構築できたことから，Marionette および MWC の再利用性と利便性を確認することができた．

以上の結果を踏まえて，ロボットを使用したメカトロニクス教材の検討を行った．実験実習のモデル授業の流れや，前述した Orange-Caracara のシステムをベースに機能を追加する実験テーマについて考察を行った．これにより，専攻科生や学部 3 年生以上を対象とした，様々な分野の学生がロボット開発を行える高度なメカトロニクス教材の実用性を確認することができた．

参考文献

[1] 前田 弘文, 五百井 清, 大坪 義一, 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ミドルウェア開発，日本機械学会講演論文集 No.1110，p.123～124，2011

[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, pp.48～153, 2012

[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(SI2012)，pp.48～153，2012

[4] 前田 弘文，藤長 大祐，五百井 清，大坪 義一，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)，pp.60～63，2011

[5] 山崎 歩惟，藤田 和友，前田 弘文，携帯ゲーム機を用いたロボットのモジュール化，中国四国学生会第 43 回学生員卒業研究発表講演会，703，2013