

1 研究テーマ

オープンハードウェアロボットの多用途化を目指した制御プログラミング環境の再構築

2 研究の背景

オープンハードウェアロボット「Mugbot」（図1、図2）は、安価で入手が容易なハードウェア（100円ショップなどで入手可能な部材、Raspberry Pi と量販マイコンボード）に、オープンソースのソフトウェアを用いて製作可能なロボットである。

製作の方法についても工夫がされており、専門的な知識がないユーザが、容易かつ安価にソーシャルロボットの製作を可能とすることを目指した研究プロジェクトである。ここでいうソーシャルロボットとは、人間の生活の中で実際に働くロボットのことで、Mugbot を介して人々が集まり、人間同士のコミュニケーションを豊かにする役割（※1）を目指したロボットである。全ての設計図と製作方法、関連ソフトウェアはWeb上に無償公開されており（<http://www.mugbot.com/>）、この情報を参照することで、誰でも自由にロボットの製作と改良が行える日本発の「オープン」なロボットを普及させることを目標としている。



図1：Mugbotの外観

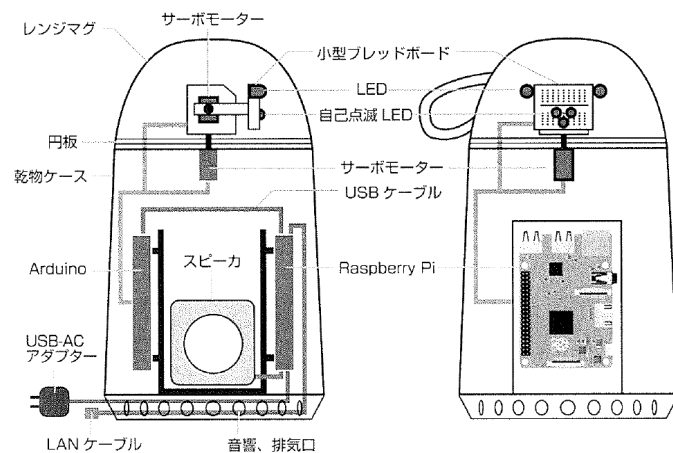


図2：Mugbotのハードウェア（※1より抜粋）

※1：おしゃべりロボット「マグボット」-ラズパイとArduinoで電子工作, 小池星多著, リックテレコム, 2016

Mugbot に関連する研究プロジェクトは、Mugbot の発案者である東京都市大学の小池星多教授によって主導されている。筆者は Mugbot の動作と発話を制御するためのソフトウェア「Mugbot Action Designer」(以下、MAD と略す) の研究開発を分担して担当している。

MAD (図 3) は、専門的なプログラミングの知識がなくとも容易に Mugbot の制御プログラムが記述できるように工夫してある。MAD はマサチューセッツ工科大学 メディアラボが開発をしている子ども向けのプログラミング環境である「Scratch 1.4」をベースに開発してきた。Scratch もオープンソースで無償公開 (https://scratch.mit.edu/scratch_1.4/) されているソフトウェアである。Scratch は、マウスを使って命令が記述されたブロックを組み合わせてプログラミングができることが特徴の一つである。これにより、プログラミング言語の文法に関する専門知識の事前学習が不要という点が優れている。

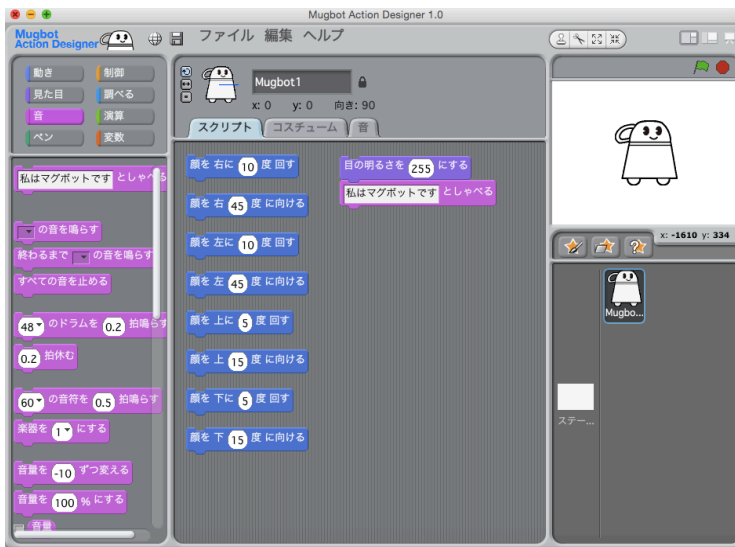


図 3 : MAD の実行画面

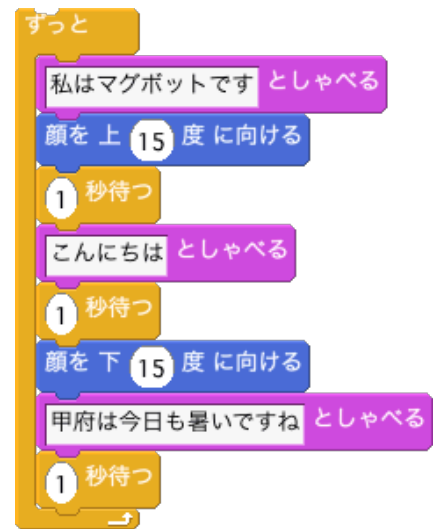


図 4 : 制御プログラムの例

図 4 に示したのが、MAD を使って記述した制御プログラムの例である。図 4 のブロックの内部にある白抜きの部分はユーザーが自由に数値や文字を入力することができ、顔の方向の制御に必要な角度や、発話の内容を指定する。発話に関しては Mugbot に内蔵された音声合成ソフトウェアを使って、ユーザーが発話したい文章を入力しておけば (図 4 の例でいうと「甲府は今日も暑いですね」など)、Mugbot に接続もしくは内蔵されたスピーカーから自由に発話を行うことができる。

MAD を開発することで、非専門家である Mugbot のユーザーが、用途に合った制御プログラムを独学で容易に記述できるようになり、ユーザーが Mugbot を主体的に活用するという事例が飛躍的に増加した。例として、①子供向けの教育に教材として導入する、②公共施設等での案内役として Mugbot を活用する、③障がい者のケアへ活用する、といった試みが行われ

ている。②の受付や展示案内の事例として、東京都多摩市総合福祉センターの受付での案内（図5）、岩手県宮古市「薬師塗漆工芸館」での展示案内（図6）などが挙げられる。



図5：東京都多摩市総合福祉センター



図6：岩手県宮古市「薬師塗漆工芸館」

③の例として、重症心身障害児施設である島田療育センターでは、重症心身障害児のケアのために Mugbot を活用するための研究グループが活動している（図7）。身体的な介護などを補助するわけではなく、人と人とのコミュニケーションを円滑にするという目的で、Mugbot をケアに活用できないかが検討されている。



図7：MAD を使って制御プログラムを記述する島田療育センターの職員

こうした活用事例では、コミュニティの外部から Mugbot の活用アイデアを持ち込む場合もあるものの、多くの場合、それぞれのコミュニティに属するユーザが、現場の課題を解決するために Mugbot の活用方法を試行錯誤しながら検討する。プログラミングの専門的な知識がなくとも、ユーザの用途に合わせた制御プログラムを手軽に記述し、試行錯誤できるようにすることが、様々なコミュニティにおける Mugbot の活用を推進する仕掛けとして重要である。

3 研究の目的

本研究では、筆者が開発している Mugbot の動作と発話を制御するためのソフトウェアである MAD の再構築をテーマとした。

MAD を開発したことで、非専門家である Mugbot のユーザが、用途に合った制御プログラムを独学で容易に記述できるようになり、ユーザが Mugbot を主体的に活用するという事例が飛躍的に増加した。しかし、MAD を利用しているユーザからの要求には「インターネットから取得した情報を制御プログラムに活用できれば、更に Mugbot の利用用途が広がる」というものが多かった。本研究では、こうした改善要求に応えるため、現状の MAD の実行環境と実装基盤を再構築し、Mugbot の活用形態をさらに多様化し、これまで以上に様々なユーザ主体の活用が生まれる可能性を広げることをねらいとした。

4 研究の方法

Scratch 1.4 をベースに開発していた従来の MAD では、インターネット上の情報を取得し、それを制御プログラムに活用することはできない。そこで、現在の Scratch の最新バージョンである Scratch 2.0 に MAD の実装基盤を変更し、従来の MAD と同じ機能を持った新しい MAD を開発することにした。Scratch 2.0 は Flash で開発された Scratch 1.4 とほぼ同等の機能や互換性を持ち、Web ブラウザ上で動作する Scratch 1.4 の後継アプリケーションである (図 8)。

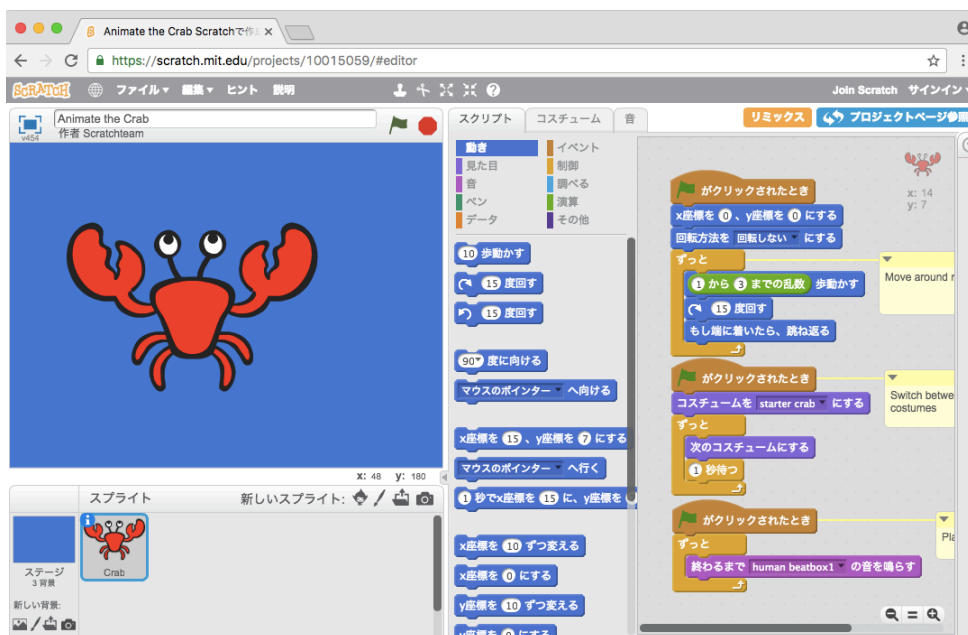


図 8 : Scratch 2.0 の実行画面

Scratch 2.0 上で MAD を動作させるにあたり、Scratch X (<http://scratchx.org/>) という拡張ブロックを実装するための仕組みを利用した。Scratch X によって、Scratch 2.0 上で新しい命令ブロックを実装することができる。Scratch X では、JavaScript を用いてブロックの定義や処理内容を記述し、新しいブロックを作成することができる。例えば、図9のようなコードを記述することで、「my first block」と書かれた新しいブロックを Scratch 2.0 上で動作 (図10) させることが可能となる。

```
(function(ext) {
  // Cleanup function when the extension is unloaded
  ext._shutdown = function() {};

  // Status reporting code
  // Use this to report missing hardware, plugin or unsupported browser
  ext._getStatus = function() {
    return {status: 2, msg: 'Ready'};
  };

  ext.my_first_block = function() {
    // Code that gets executed when the block is run
  };

  // Block and block menu descriptions
  var descriptor = {
    blocks: [
      // Block type, block name, function name
      [' ', 'my first block', 'my_first_block'],
    ]
  };

  // Register the extension
  ScratchExtensions.register('My first extension', descriptor, ext);
})();
```

図9：Scratch X による拡張ブロックの定義例

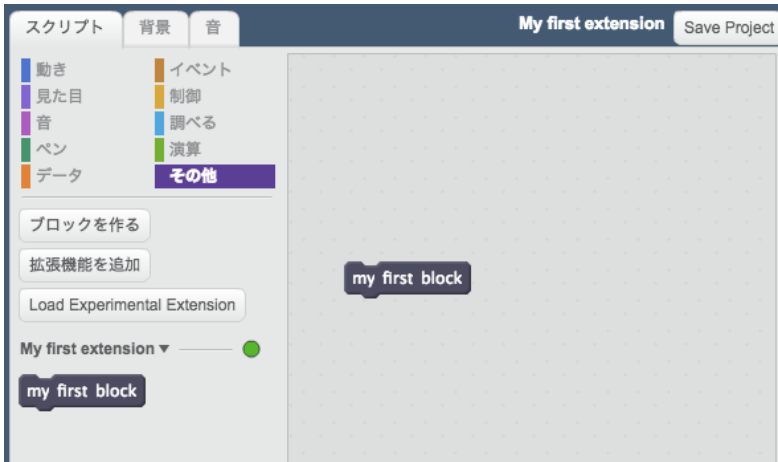


図10：図9で定義された拡張ブロック (my first block) の表示

Scratch X を使い、従来の MAD と同じように Mugbot を制御できる拡張ブロックを定義・追加することで、Scratch 2.0 上で MAD を動作させることができる。一方で、世界中の開発者によって、Scratch X を使った様々な拡張ブロックが既に開発・公開されており、の中にはインターネット上の情報を取得するための拡張ブロックも存在する。これらの既存の拡張ブロックと本研究で再構築した MAD を組み合わせることで、インターネット上の情報を利用した Mugbot の制御プログラムを記述することが可能となる。

さらに従来の MAD では、Mugbot 側で MAD からの命令を受信するサーバについては、MAD の実装基盤の制約から、TCP Socket を利用していた。Scratch 2.0 上で再構築する MAD では、Web Socket を利用することができるため、サーバ側の実装は Web Socket により受信する方式に変更することにした。これにより、小池教授らが開発しているタブレット用の制御パネル（図 1 1）と共通のサーバプログラムが利用でき、ユーザが MAD を使う場合に必要な準備が不要となるという利点も得られる。

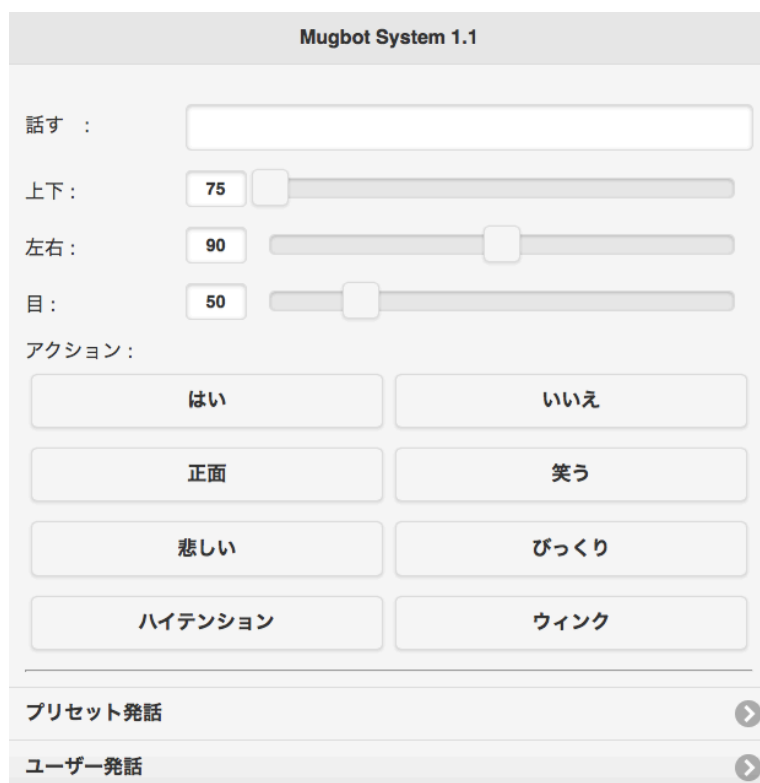


図 1 1 : MAD と同じサーバプログラムで利用できるタブレット用の制御パネル

5 研究の成果

従来の MAD のユーザにも配慮し、命令の種類は変更せずに、Mugbot を制御するための拡張ブロックを Scratch X を利用して実装した。図 1 2 に Scratch 2.0 上で動作する MAD の

スクリーンショットを、図 1 3 に移植した制御ブロックの一覧を示す。

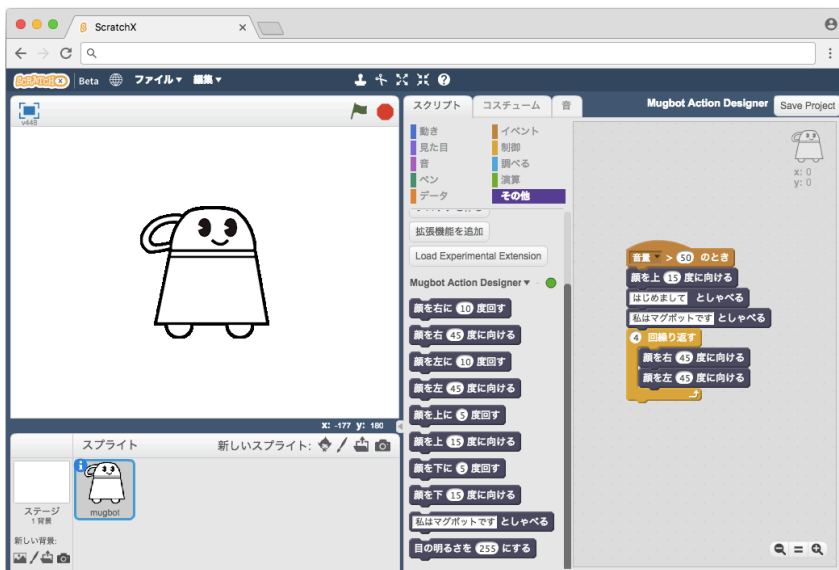


図 1 2 : Scratch 2.0 上で動作する MAD



図 1 3 : 制御ブロック一覧

これにより、制御プログラム中にインターネット上の情報を活用することができるようになった。例えば、Scratch X の拡張ブロックの一つとして Weather Extension がある。この拡張ブロック (<https://khanning.github.io/scratch-weather-extension/>) を使うことで、OpenWeatherMap (<http://openweathermap.org>) という Web サービスから世界各地の気象情報をリアルタイムに取得するためのブロックを Scratch 2.0 に追加することができる。この拡張ブロックと MAD のブロックを組み合わせることで、図 1 4 のような制御プログラムを作ることが可能となった。

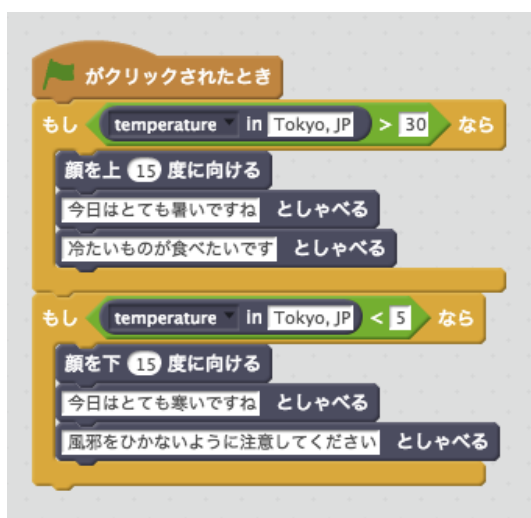


図 1 4 : 現在の気温によって動作を変更するプログラム例

この制御プログラムの例では、東京の現在の気温を OpenWeatherMap から取得し、その気温によって Mugbot の動作（反応）を変更している。気温が 30 度以上（暑い）場合は、顔を上に向け「今日はとても暑いですね。冷たいものが食べたいです。」と発話する。一方、気温が 5 度以下の（寒い）場合は顔を下に向け「今日はとても寒いですね。風邪をひかないように注意してください。」と発話する。

図 1 4 のプログラム例では、Mugbot が現在置かれている場所（プログラム例では東京）を指定して気温を取得しているが、遠く離れた海外の気象情報を取得し、その結果を制御プログラムで利用したり、温度に限らず、現在の天候や湿度に応じた発話や動作をしたりといった、様々なバラエティに富んだ制御プログラムを作成するための基盤を構築できた。

6 今後の予定

本研究で開発した Scratch 2.0 上で動作する MAD は、マニュアルの整備などを追加で実施した後で、オープンソースソフトウェアとしてインターネット上に公開する予定である。今年度は研究成果で述べたように、MAD の再構築に焦点を絞り、開発作業を中心に実施した。今後は再構築した MAD を活用し、山梨を含めた地域での活用事例を増やすための活動を行いたいと考えている。

Mugbot は高齢者などをはじめとした様々なケア施設での活用だけでなく、地域の活性化にも活用できる余地が残されている。これらの課題は全国的なものではあるが、とくに山梨県でも解決すべき重要課題であろう。地域での Mugbot の活用の方策を探ると共に、山梨英和大学に所属する心理学の専門家との共同研究なども検討していきたい。さらに、再構築した MAD に関する様々な活用事例に関する知見を加えて考察を行うことにより「ユーザ自らが作るソーシャルロボットを、どのように地域コミュニティの中で活用していくか」というテーマの論文を執筆していきたいと考えている。