

教室内でサーバプログラミングの学習を可能にする 教材システムの開発

本多 佑希¹ 兼宗 進¹

概要: 2020 年度以降に実施される小中高の新教育課程において、中学校技術・家庭の技術領域では「双方向コンテンツのプログラミング」で、高等学校に新設される情報 II では「情報システムとプログラミング」において、ネットワーク通信を利用したプログラミングが扱われる予定である。そのような背景から、ネットワークプログラミングを容易に体験できる環境が求められている。そこで本発表では、中学校、高等学校、大学等の授業で活用できるサーバプログラミング学習環境を提案する。

1. はじめに

2020 年度から段階的に開始される予定の新教育課程の中で、小学校ではプログラミング教育が必修化され、中学校と高等学校においてはネットワークを利用したプログラミングが扱われる予定である。

中学校では技術・家庭の技術領域で「双方向コンテンツのプログラミング」として、文字や画像などのコンテンツを、サーバなどを介してユーザー間でやり取りするプログラミングを扱う [1,2]。題材としては、クライアント間でのメッセージ交換のようにクライアントで動くプログラムの作成や、Web の掲示板のようにサーバで動くプログラムの作成などが考えられる。クライアント間で通信するプログラムについては、ドリトルがネットワーク上の共有メモリ空間に対応した P2P のプログラミング環境を提供しており、すでに中学校の授業で利用されている [3-5]。サーバ側で動くプログラムについては、どのように学習環境を用意するかは今後の課題である。

高等学校の共通教科情報に新設される選択科目の情報 II では「情報システムとプログラミング」が扱われる [6]。情報システムでは一般的に、サーバ側のデータベースやプログラムとクライアント側のプログラムが協調して動作する。このようなプログラミングでは、プログラムの設計やデバッグなどの難易度が高くなることに加え、使用する言語もサーバ側とクライアント側で異なる場合も想定される。

そこで本研究では、高等学校での「サーバとクライアントが協調するプログラミング学習」、中学校での「サー

バ側でのプログラミング学習」などを想定した、ネットワーク通信を利用したプログラミングの学習を行うための学習環境を提案する。提案する教材システムを使用することで、特別なサーバなどを用意することなく、教室内の PC を用いてネットワークを利用したプログラミングの学習が可能である。具体的には、生徒のクライアント PC の Web ブラウザからプログラムを編集し、教員 PC や生徒 PC 上でサーバプログラムとして実行することを可能にする。

2. 関連研究

2.1 学習環境

ネットワークを使ったプログラミング学習は、古くは C 言語などでソケット通信等を題材として使っていたが、難易度が高く学校教育での利用は容易ではなかった。

その後、Web の普及により、Perl などの言語を利用した CGI プログラムや、PHP などを利用したサーバ側のプログラミング教育が大学等で行われている [7,8]。

高等学校でのサーバプログラミングの事例としては、間辺らは共通教科情報の授業において、ドリトル/JavaScript/PHP という複数の言語を用いることで、クライアントプログラムと Web 画面で動くプログラム、サーバ側で動くプログラムを体験できる授業を行った [9]。ドリトルによるクライアントプログラムでは、センサで計測したデータをサーバに送信して記録する IoT を意識した学習を扱い、JavaScript による Web プログラミングではゲームを作成してスマートフォンで動作させる学習を扱い、PHP によるサーバプログラミングでは情報システムの体験としてメッセージ交換とショッピングサイトを扱っている。

¹ 大阪電気通信大学
Osaka Electro-Communication University

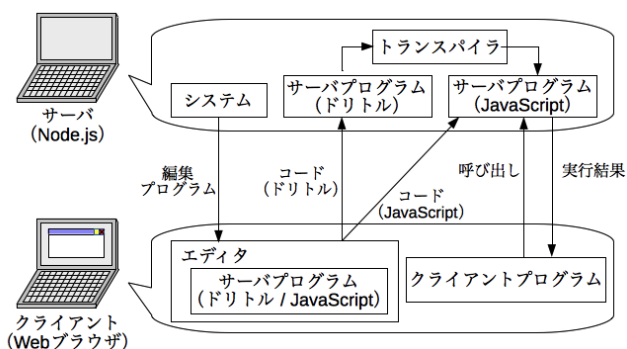


図 1 教材システムの概要

クライアント側の P2P 的なネットワークプログラミングでは、ドリトル [10] がクライアント間で共有できるメモリ領域を提供している。Scratch [11] にも MESH や Scratch Offline Editor [12] により同様の機能が存在する。

中学校でのネットワークを利用したクライアントプログラミングの事例としては、西ヶ谷らは技術科の授業において、ドリトルを使用してチャットプログラムを作成する授業を実施した [13]。生徒は画面に入力したメッセージをネットワーク上の共有領域に送信し、他の生徒のメッセージを共有領域から読むことで、双方向のメッセージを取りするプログラムを作成した。

3. 教材システムについて

3.1 概要

教材システムの概要図を図 1 に示す。サーバーは、生徒のグループの 1 台、または教員の PC で起動する。クライアントから Web ブラウザで接続すると、サーバー上のプログラムファイルを編集するエディタ画面が開く。記述したプログラムはサーバーに転送されて保存される。

サーバーの実行画面を図 2 に示す。図の (A) にサーバーの IP アドレスが、(B) にはポート番号が記載されており、クライアントからサーバーにアクセスする際には、これらの情報を基にアクセスする。また、図の (C) はプログラムエディタである。サーバー用 PC からサーバープログラムを編集する場合には、このエディタ部で編集を行う。

サーバー上のプログラムは Node.js により、JavaScript のプログラムとして実行される。記述したプログラムがドリトルなどの JavaScript 以外の言語の場合には、トランスパイラによって自動的に JavaScript に変換される。

サーバー上のプログラムはクライアントからの呼び出しによって実行される。呼び出しは Web の標準的なインターフェースである「http://サーバー/プログラム.js」の形で呼び出すことができる。このことから、クライアントのプログラムとして、HTML やドリトルなどの複数のプログラミング言語を利用することを可能とした。

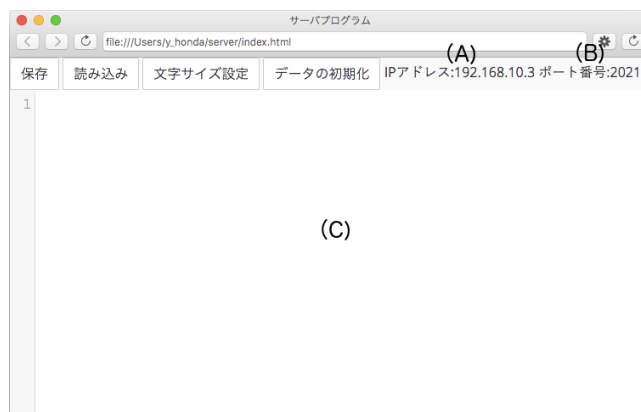


図 2 教材システムの実行画面



図 3 サーバープログラムの編集画面

4. 設計と実装

4.1 提供する機能

教材システムは、3つの機能から構成される。

- (1) プログラム編集環境は、クライアント PC の Web ブラウザからサーバー上のプログラムを編集する機能を提供する。
- (2) トランスパイラは、クライアントで記述したプログラムを JavaScript に変換する。
- (3) サーバープログラムの実行環境は、クライアントプログラムからの呼び出しを受け付け、サーバー上のプログラムを実行する。

4.2 プログラム編集環境

教材システムでは、サーバープログラムを図 2 のサーバー画面で編集できるほか、Web ブラウザを利用してクライアント PC から編集が可能である。

クライアントの Web ブラウザから編集する画面を図 3 に示す。この図の A 部は、図 2 と同様にプログラムエディタになっており、このエディタ部にプログラムを記述して保存を行う。

表 1 アクセス時の URI と対応した環境

URI	環境
http://ipadress:port/edit	編集画面
http://ipadress:port/A.js	A.js を実行 (A.js の部分は可変)

4.3 トランスパイル環境

サーバー上のプログラムは JavaScript として実行される。そのため、JavaScript 以外の言語をサーバー上でトランスパイルすることで JavaScript に変換するようにした。トランスパイラを用意することで、TypeScript などの AltJS や、ドリトル [10, 14]、C 言語 [15] などが利用可能になる。

4.4 プログラム実行環境

サーバー上のプログラムはクライアントからの呼び出しで実行される。呼び出しは最初に URI の解析を行い、対応するプログラムを起動する。実行されるプログラムは実行ごとのセキュリティを含めた独立性を確保するため、それぞれが別プロセスとして実行されるように実装した。

URI の解析は、Node.js のライブラリである Express を用いて実装した。サーバープログラムの実行や、サーバープログラムを編集する際の URI など、それぞれ表 1 のように分けた。ipadress 部分と port 部分はサーバーの IP アドレス、ポート番号によって変わる。

サーバー上のプログラムでは、標準の JavaScript の命令の他に、ユーザーのプログラムをサポートするためのライブラリを利用することができる。たとえばデータをファイルに保存、読み込みを行う命令を用意することで、簡単にデータのファイル入出力を可能にした。今後はデータベースの機能などの拡張を計画している。

4.5 サーバー側のプログラム例

サーバープログラムの例を図 4 に示す。main として定義されている関数が実行され、戻り値がそのままクライアントへ返る値となる。この例では、"return-value" という文字列が返る。

クライアントがサーバーにパラメータを渡すことも可能である。GET パラメータとして渡された値が連想配列として main 関数の第一引数に渡される。パラメータを受け取って計算を行うプログラムを図 5 に示す。この例のように、「param["p1"]」とすることで、GET パラメータで渡された p1 を参照できる。

サーバープログラムで扱うデータの中には、保存をしたい場合がある。データの読み込みと保存の例を図 6 に示す。Key-Value ストアのモデルで保存されている。read 関数の第一引数に読みたいデータのキーを渡すことで値を読み込むことができる。また、write 関数の第一引数にキー、

```

1 function main(){
2     return "return-value";
3 }

```

図 4 サーバープログラム例 1 - 値の返し方

```

1 function main(param){
2     return param["p1"] * 10;
3 }

```

図 5 サーバープログラム例 2 - パラメータの受け取り

```

1 function main(param){
2     var val = read("val");
3     val += param["p1"];
4     write("val",val);
5 }

```

図 6 サーバープログラム例 3 - 値の読み込みと保存

第二引数に保存したい値を渡すことで、値を保存することができる。この例では、引数で渡された値をサーバー上の val というキーの値に足し合わせている。

現在、read 関数で渡したキーの値が初期化されていない場合はエラーが発生する仕様になっている。サーバー上の値を初期化する場合には、ソフトウェアのメイン画面 (図 2) の「データの初期化」ボタンから変数の初期化を行う必要がある。

4.6 クライアントからのアクセス例

教材システムで開発したサーバープログラムには、HTTP 通信 (GET リクエスト) でアクセスできる。その際には、GET パラメータとして値を渡すことで、サーバーにデータを送ることができる。また、サーバーからの返答は文字列として値が返ってくる。

図 7 にクライアントから非同期でサーバに値を送信し、値を受け取るプログラム例を示す。

5. 授業での利用

5.1 双方向コンテンツ (中学校)

中学校で学習する双方向コンテンツは、チャットプログラムや掲示板を題材としたプログラミングで学習できる。チャットプログラムではクライアント同士が直接双方向に通信を行い、掲示板では、サーバを介してクライアント同士が双方向に通信を行うイメージである。ここでは、掲示板を例に双方向コンテンツの学習を考える。

掲示板を制作する場合、サーバーの処理としては以下の二つの処理が必要になる。

- 書き込みを受け付ける処理
- 書き込みの一覧を返す処理

書き込みを受け付ける処理のプログラム例を図 8 に示す。

```

1 //XMLHttpRequest を用いた場合
2 var param = "p1=TEST&p2=10";
3 var url = "http://ipaddress:port/test.js?" +
  param;
4 var request = new XMLHttpRequest();
5 request.open("GET", url, false);
6 request.send(null);
7
8 console.log(request.responseText);
9
10 //jQuery.ajax を用いた場合
11 var opt={
12     type:"GET",
13     url:"http://ipaddress:port/test.js",
14     data:{
15         p1:"TEST",
16         p2:10,
17     },
18     async:false,
19 };
20 var res=$.ajax(opt);
21 console.log(res.responseText);

```

図 7 クライアントからデータの送信と受け取り

```

1 function main(param){
2     var resList = read("resList");
3     resList.push(param["res"]);
4     write("resList",resList);
5 }

```

図 8 掲示板への書き込みを受け付ける

この例では、書き込みの一覧を resList というキーで配列としてサーバーに保存していると仮定している。また、クライアントから GET パラメータの res として書き込みの内容が渡されていると仮定している。この例では書き込みの内容のみが渡されているが、書き込み主の氏名も同時にパラメータに渡し、内容を増やすことも可能である。

書き込みの一覧を返す処理のプログラム例を図 9 に示す。サーバーに保存されている resList を文字列として返している。

クライアントのプログラム例を図 10 に示す。この例では、本質のみを学習できるよう、複雑な処理をラップしたライブラリを事前に用意している。今回のライブラリで用意した HTML タグの操作を補助するオブジェクトを表 2 に示す。また、それらで提供されているメソッドや、複雑な処理をラップした関数とその役割を表 3 に示す。

これらのプログラムを用いて掲示板を開発することが可能である。また、書き込み主の氏名や書き込み時間の追加や、同調を示す「いいね」機能の追加など、発展も可能である。

```

1 function main(){
2     return read("resList");
3 }

```

図 9 掲示板の書き込みの一覧を返す

表 2 クライアント用に提供しているライブラリ - HTML タグの操作を補助するオブジェクト

オブジェクト名	対応するタグ
text	<input type="text">
button	<button>
span	

表 3 クライアント用に提供しているライブラリ - HTML タグの操作を補助するメソッド

関数名、メソッド名	役割
input(str)	文字列をポップアップで表示し、文字列の入力を受け付ける
request(url,data)	第一引数の URL に GET リクエストを送信 第二引数の連想配列をパラメータにする
text.value()	文字列を取得
text.clear()	内容をクリア
button.click(func)	ボタン押下時に実行する関数を登録
span.text(str)	引数の文字列を表示

5.2 情報システムとプログラミング (高等学校)

高等学校で学習する「情報システムとプログラミング」には、IoT の学習に用いることができる。本教材システムは、IoT 機器からサーバーにデータを蓄積する学習や、蓄積されたデータをサーバーから取得し、分析を行う学習に適した環境である。

サーバープログラムに関しては、GET リクエストで受け取ったパラメータを保存するプログラムのみでデータ蓄積が可能のため、実装は容易である。

IoT 機器から蓄積したデータの分析はライブラリが必要であり、JavaScript で学習するのは困難ではないかと考えられる。そのため、標準で統計機能を扱うことができるドリトル [10,16] との連携を検討したい。本研究のソフトウェアは 4.3 章で述べたようにドリトルでの実行も可能であり、連携することでより初学者に適した教材になることが期待できる。

6. まとめ

本研究では、ネットワーク通信を利用したプログラミングを容易に学習することが可能な教材の開発を行った。この教材システムは環境構築、サーバの立ち上げに管理者権限が不要であることや、サーバプログラムに複数のプログラミング言語を使用できることなど、利点が多く存在する。ネットワーク通信を利用してサーバーとクライアントが連

```

1 <html>
2 <head>
3 <script charset="utf-8" src="jquery-2.1.0.js
  "></script>
4 <script charset="utf-8" src="lib.js"></script
  >
5 </head>
6 <body>
7 <input type="text" id="res">
8 <button id="send">送信</button><br>
9 <button id="get">取得</button>
10 <span id="disp"></span>
11 <script>
12 ip=input("サーバーの IP アドレスを入力");
13
14 function send(){
15     res=text("res").value;
16     text("res").clear();
17     name=text("name").value;
18     url="http://xxx.xxx.xxx.xxx:yyyy/A.js";
19     data={res:res,name:name};
20
21     request(url,data);
22 }
23 button("send").click(send);
24
25 function get(){
26     url="http://xxx.xxx.xxx.xxx:yyyy/A.js";
27     disp=span("disp");
28     var resList=request(url);
29     disp.text(resList);
30 }
31 button("get").click(get);
32
33 </script>
34 </body>
35 </html>

```

図 10 掲示板のプログラム例 (クライアント)

携しながら動くプログラムは実務で使われているシステムに近いモデルであり、学ぶ意義は十分ある。

中学校の双方向コンテンツの学習例として掲示板を例に双方向コンテンツの学習への例を示したが、今後も授業例の検討を進めていきたい。また、高等学校の情報システムとプログラミングの学習について、ドリトルとの連携も含め、IoTの学習などへの応用なども検討を進めていきたい。

参考文献

- [1] 文部科学省: 中学校学習指導要領 (2017).
- [2] 文部科学省: 中学校学習指導要領 (平成 29 年告示) 解説技術・家庭編 (2017).
- [3] 兼宗進, 久野靖: ドリトルで学ぶプログラミング [第 2 版], イーテキスト研究所 (2011).
- [4] 兼宗進, 村松浩幸, 上野耕史: ドリトルによるプログラミング学習, 東京書籍 (2017).

- [5] 竹野英敏, 浅水智也, 安藤明伸, 大村基将, 木村浩之, 紅林秀治, 藤原英治, 宮内智, 上野耕史: やってみようプログラミング, 開隆堂 (2018).
- [6] 文部科学省: 高等学校学習指導要領 (2017).
- [7] 兼宗進, 長慎也. 文科大学におけるサーバーサイドプログラミング授業の試み. 情報処理学会 コンピュータと教育研究会, CE(83), pp.141-148, 2006.
- [8] 兼宗進, 白井詩沙香, 竹中一平, 長瀧寛之, 小林史弥, 島袋舞子, 田邊則彦. データベースと情報システムを学習する授業の提案と実践. 情報処理学会論文誌「教育とコンピュータ」, Vol.3, No.3, pp.18-28, 2017.
- [9] 間辺広樹, 長島和平, 並木美太郎, 長慎也, 兼宗進. 高等学校における複数言語によるプログラミング教育の提案. 情報処理学会論文誌「教育とコンピュータ」, Vol.3, No.3, pp.29-41, 2017.
- [10] プログラミング言語「ドリトル」.
[http://dolittle.eplang.jp/\(2018/05/17参照\)](http://dolittle.eplang.jp/(2018/05/17参照)).
- [11] Scratch - Imagine, Program, Share.
[https://scratch.mit.edu/\(2018/05/17参照\)](https://scratch.mit.edu/(2018/05/17参照)).
- [12] Scratch - Scratch Offline Editor.
[https://scratch.mit.edu/download\(2018/05/17参照\)](https://scratch.mit.edu/download(2018/05/17参照)).
- [13] 西ヶ谷浩史, 兼宗進, 紅林秀治. 中学校技術科におけるドリトルを利用した通信プログラムの実践. 情報処理学会 コンピュータと教育研究会, CE(134), 2016.
- [14] 本多佑希, 長慎也, 大村基将, 久野靖, 並木美太郎, 兼宗進. Dolittle のオンラインプログラミング環境の開発. 情報処理学会 コンピュータと教育研究会, CE(134), 2016.
- [15] 長慎也, 長島和平, 堀越将之, 兼宗進, 並木美太郎. オンラインプログラミング環境 Bit Arrow を用いた C 言語プログラミングの授業実践. 情報処理学会, 情報教育シンポジウム (2017), pp.121-128, 2017.
- [16] 小林史弥, 白井詩沙香, 山澤昭彦, 兼宗進. ドリトルでのデータ処理機能とグラフ描画機能の開発. 情報処理学会, 情報教育シンポジウム (2017), pp.178-181, 2017.