

HTML5 を用いたリアルタイム性のある Web ベースの教育支援システムの開発

栗島聖大[†] 香川考司^{††}

Web ベースのプログラミング学習支援環境は、インストールが不要で、編集やコンパイルがブラウザから行えるため学習者の負担を軽減することができる。この他にも Web ブラウザをプラットフォームとする事により、様々な拡張が可能になる。現在策定が進められている HTML の新基準である HTML5 では、従来の方法に比べ高速なデータのやり取りを可能とする WebSocket という機能や、様々な図形や文字を描画することのできる Canvas などの機能が加えられている。本研究では、この HTML5 を利用して、従来のプログラミング学習支援用 Web アプリケーションには無かったリアルタイム通信機能を持つシステムを開発する。

Development of a Web-based Educational Environment for Programming using HTML5

Kiyohiro Kuwajima[†] Koji Kagawa^{††}

Web-based education environments for programming do not require installation and can reduce burden of learners by enabling them to edit and compile programs on Web browsers. Since they use Web browsers as the platform, they are open to various extensions. HTML5, which is the next version of the HTML standard currently on progress, includes several extensions such as WebSocket and Canvas. WebSocket provides for fast and bi-directional communication between browsers and Web servers. Canvas allows for rendering various 2D shapes and characters by scripting. We develop a Web-based educational environment for programming with real-time communication capabilities using HTML5.

1. はじめに

通常、プログラミング学習環境を構築するためには、学習するプログラミング言語に関するコンパイラやエディタのインストールや環境設定が必要で、学習を始めるまでに学習者の負担となる作業が多々ある。また、簡単な CUI での操作法の習得も必要で、これらはプログラミング学習を行う初心者にとって大きな壁となる。それに対し Web ベースのプログラミング学習環境を用いる場合、学習者は Web ブラウザから所定の URL にアクセスするだけでプログラミング学習を始めることができる。コンパイラ等の必要なソフトウェアはサーバで提供されるので、学習者が準備、設定を行う必要はない。その為、学習者は簡単にプログラミング学習を開始することができるという大きなメリットがある。また、指導者側としても学習者が同じ環境で学習することになるので、コンパイラのバージョンの違いや環境設定のミスといった学習環境の導入時に起こりうる個別のミスを事前に防げるという利点にもなる。

Web ベースの学習用プログラミング環境構築のためのアプリケーションフレームワークに Wappen[i][1], WappenLite[ii][2][3]がある。Wappen は Eclipse RCP に基づいており、Web ブラウザから所定の URL にアクセスしてリンクをクリックすることによって学習に必要なソフトウェアがダウンロードされ、起動する。ユーザインタフェース (以下、UI) は Eclipse ベースとなっている。

これに対し WappenLite は、HTML や Ajax, Flash といった通常の Web ブラウザで用いられるユーザインタフェースを用いており、SWT のようなプラットフォーム依存の技術を使用しない。これによって、Wappen と比べて UI の起動の高速化や、カスタマイズ性の向上、配備が容易になるといった特徴が挙げられる。UI は図 1 WappenLite 実行例 1 のように通常の HTML ベースとなっており、必要なソフトウェアをダウンロードする必要があった Wappen に比べ UI の起動が高速になっている。

[†] 香川大学大学院工学研究科
Graduate School of Engineering, Kagawa University

^{††} 香川大学工学部
Faculty of Engineering, Kagawa University
i Wappen: Web ベースのプログラミング学習プラットフォーム , <http://guppy.eng.kagawa-u.ac.jp/Wappen/>
ii WappenLite マニュアル , <http://guppy.eng.kagawa-u.ac.jp/WappenLite/manualJ.html>

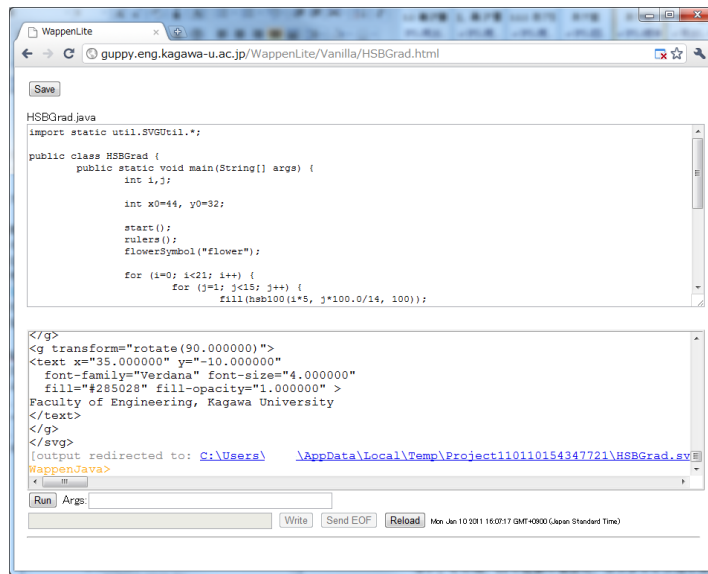


図 1 WappenLite 実行例 (Google Chrome 上での実行)

いずれも Web ベースで構築された学習環境であるが、コンパイラやエディタといった必要なソフトウェアのインストールを学習者が行う必要はない。その為、学習者各人がそれぞれのクライアント PC に対して必要なソフトウェアのインストールや設定をする必要のある非 Web ベースのプログラミング学習環境と比べると、導入が非常に容易といえる。

さらに Web ベースの環境は、Web ブラウザをプラットフォームとすることにより、様々な拡張が可能になるという利点がある。現在策定が進められている HTML の新基準である HTML5 [iii] では、従来の方法に比べ双方向で高速なデータのやり取りを可能とする WebSocket という機能や、様々な図形や文字を描画することのできる Canvas などの機能が加えられている。

一般的に、プログラミングを学習する環境は少数の指導者に対して多数の学習者が集まるクラスであることが多い。プログラミングをある程度経験したことがある者が新しい言語の学習を行う場合は、その言語特有のコーディング方法や、他言語と大きく異なるポイントといった、間違えやすいと考えられる点を重点的に指導すれば良い。しかしプログラミングを初めて学ぶ学習者の場合、経験者が考えもしないようなこと

iii HTML5.JP, <http://www.html5.jp/>

ろで躓く可能性が非常に高い。

大学では、TA (Teaching Assistant) という指導者の手助けをする上級生が学習者ひとりひとりの質問に答えて回することで、学習者が躓いているポイントを指導していることがある。しかし、同じような質問を数人の学習者から質問を繰り返される場合もあり、多数の学習者が躓く傾向が見られる場合にはクラス全体に向けて一括した指導を行う方がより効率的である。近年のプログラミング学習を行う高校や大学では、ネットワークに接続された PC を各個人が使い、座学と実習を並行しながら学習している場合が多い。このようなネットワークに接続されている環境では、直接指導者を呼んで質問に答えてもらうだけでなく、ネットワークを介した質問と回答のやり取りを行うことにより指導を受けることも可能である。しかし、現在多く用いられている掲示板やメールのような形式でのやり取りではリアルタイム性に欠け、即座の対応には向いていない。そこで、HTML5 を用いた Web ベースでリアルタイム性を持った学習支援システムについて検討する。中でも、WebSocket に着目し、指導者が学習者からのフィードバックをリアルタイムに集約、監視できるシステムの構築を行う。

学習者側からプッシュされたデータを活かす先行研究として、筆者らの研究室の川ノ上の研究 [4] がある。これは Comet というアプリケーションサーバを介して学習者と指導者のコミュニケーションや不正監視を行うシステムに関する研究である。Comet サーバを利用することによって、サーバサイドからのプッシュ送信による学習支援に対応し、指導者側から学習者側のエディタ操作やポップアップ表示による注意喚起を可能とした。しかし、Comet を利用していたため、一度サーバから情報をプッシュした後にクライアントサイドからリクエストの発行を毎回行わなければならないという問題があった。その為、複数の連続したデータを断続的に送信するアプリケーションの設計には向いていなかった。本研究では川ノ上の研究で用いられていたアプリケーションサーバの Comet ではなく、データのプッシュ毎にリクエストの発行を必要としない WebSocket を利用する。

また、プログラミング学習を行う際にタブレット PC を用いたアノテーションシステムを利用することで学習意欲の向上や、指導者に対するフィードバック精度が向上する傾向にあるということはこれまでの研究で報告されている [5]。現在では、タブレット PC 以外にも、Apple 社の iPad や SAMSUNG 社の Galaxy Tab に代表されるようなタブレットデバイスが広く使われるようになってきている。また、HTML Canvas や WebSocket に代表される、リッチコンテンツを取り扱うことのできる HTML5 についても、これらのタブレットデバイスで利用することが可能で、今後はより広く広がっていくものと考えられる。

タブレットデバイスの利点としては、サイズや重量、形状の面から従来のノート PC に比べて持ち運びやすく、Web ベースのアプリケーションを利用するには十分なスペックと直感的な操作性を有していることが挙げられる。このようなデバイスの利用も

考慮し、Web ベースのシステムの構築を検討する。

2. 技術的背景

2.1 Jetty について

Jetty とは、Java のオープンソースのサーブレットコンテナで、他のサーブレットコンテナと比較するとサイズが小さく組み込み型の Java アプリケーションで利用しやすくなっている。Jetty 7 以降、HTML5 の WebSocket がサポートされているため、WebSocket を利用するためのアプリケーションサーバとして Jetty を用いる。

2.2 通信時の方法

Web の従来の方法としては、クライアントのリクエストに対してサーバが答えるというものであった。クライアントがリクエストをするたびにサーバ間に HTTP コネクションを生成し、ウェブサーバはクライアントにデータを返した後にコネクションを閉じるという方法である。この方法では、ページ全体を転送するために長い時間を要し、ページのリフレッシュは莫大な遅延を生み出していた。

2.2.1 Ajax を用いた解決方法

従来の方で問題となっていたページのリフレッシュによる多大な遅延の解消に取り入れられた技術で、ブラウザ内の変更があった部分だけをリクエストし、更新させる方法である。しかし、クライアントがデータを取得する前にはリクエストを出さなければならないという問題は依然残っている。これはチャットのようなアプリケーションを作成する際にポールループによる解決を行うことに繋がるため、処理の大半をサーバ上のイベントの確認に費やしてしまうことや、ネットワークの帯域を多く消費してしまう問題がある。

2.2.2 Comet による解決方法

Comet を利用した Web アプリケーションでは、サーバはクライアントからのリクエストに対してすぐに応答せず保留状態にしておくことで、サーバ上でイベントが発生した時にレスポンスを返すことができる。これは、Ajax の方法で述べたポールループが解決されるため有効な方法であるが、Comet を利用するためのアプリケーションサーバを構築しなければならないなど、面倒な点が存在する。

2.2.3 WebSocket による解決方法

WebSocket は ws という http を拡張したプロトコルでコネクションを張ったままの状

態でデータをやり取りする。利点としては、Ajax と違い、無駄なトラフィックが発生しない点や、Comet と違ってシンプルで HTTP ヘッダが少なく済み、ブラウザとサーバの双方から同じメッセージフォーマットを使うことができるという点などが挙げられる。最近では Google Chrome をはじめとした一部の Web ブラウザで対応がされている他、スマートフォンで用いられている Web ブラウザでも対応が始まっており、今後の発展が期待される。

2.3 Canvas について

Canvas とは、ブラウザ上に図を描写するために策定された仕様である。従来の HTML による規定では、HTML を用いて Web ブラウザ上で図を表現するためには JPEG や GIF、PNG といったフォーマットの画像を用意する必要があった。また、設定された数値や条件、操作による図の変化を表現するためには Flash や Java アプレットが使われてきた。Canvas は、Flash や Java アプレットのようなプラグインを使わずに、HTML に組み込まれた JavaScript によって図を描写することが出来る。複雑な図形の描写やアニメーションの表示も行える Flash の代替になるものではないが、使用する Web ブラウザが対応していれば Flash が動作しない端末に対してもコンテンツの提供を行える為、今後の発展が期待される。

3. システム概要

本研究では、多数の学習者の学習の進捗状況の把握や、質問に対する指導を行うことができる Web ベースのアプリケーションの開発を目指す。本研究の鍵となる点は、学習者からの情報を指導者側にプッシュするアプリケーション開発である。今回はその実現のための技術的な題材として現在規格が策定中の HTML5 を用いることにした。

3.1 システム構成

図 2 システム構成 2 にシステム構成を示す。学習者がエディタに入力を行った際、現在エディタ内に入力されている文字情報を ws 通信を用いてサーバに送信する。サーバは、学習者から送信されたエディタに入力されている文字情報を分析し、指導者に対して各学習者の単位時間あたりの入力文字数や、コンパイルを行った際のメッセージ等様々なデータを提供する。指導者は、ポップアップやエディタ操作の他、後に述べる手書き入力による指導支援を用いて学習者に対し指導やフィードバックを行う。

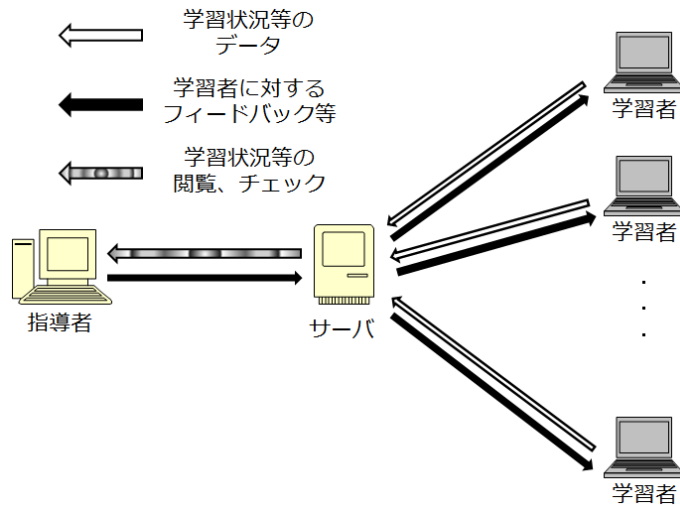


図 2 システム構成

3.2 機能構想

3.2.1 学習状況の把握

学習者の学習状況を把握する機能として、学習者がエディタに入力している内容や、入力速度等の入力状況を取得する。これによって、入力等の操作がしばらく行われていない学習者に対して TA を派遣したり、注意喚起を行ったりすることができる。同様に、突然入力文字数が増減する事を繰り返す学生に対してコピー&ペーストによる不正の監視が行える。

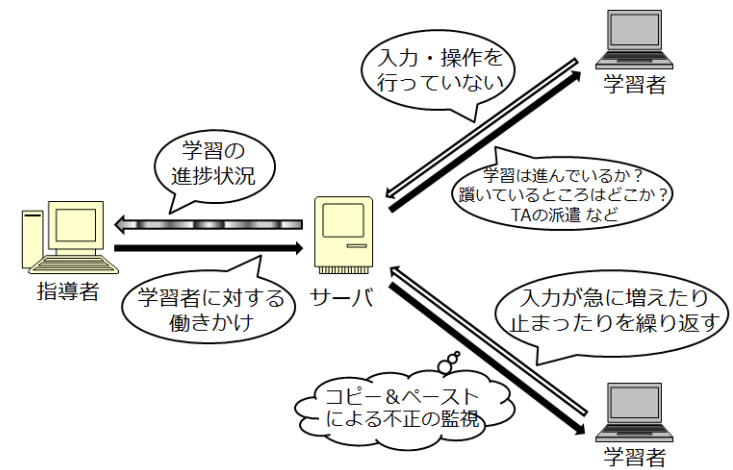


図 3 指導者による注意喚起など

また、学習者向けの教材ページが用意されている場合、該当するページ内に JavaScript を組み込むことで、開かれたページの情報サーバに対して ws 通信で送信する。WebSocket を用いることで、リアルタイムにサーバに送信される情報を解析し、現在学習者達によって開かれているページがどのページであるのかが把握できる。これによって指導者は学習者がどのページを主に参照しているのか、見当違いなページを参照していないか等のチェックを行える。

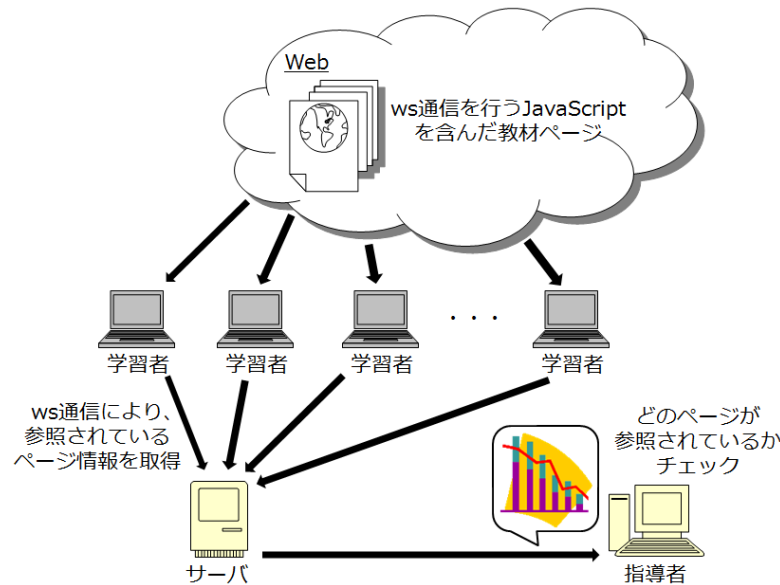


図 4 教材ページの利用頻度確認

3.2.2 手書き入力を用いた指導支援

Canvas の機能を用いて、学習者が作成したソースプログラムやプログラムの出力結果に対して手書き入力による指導の支援を行う。テキストだけの支援に比べ、手書き入力での支援は直感的で分かりやすく、学習者の意欲低下を招きにくい。指導者にとっても、インデントの指導などが文章のみによる場合よりも行いやすいという利点がある。HTML5 の策定によって Canvas と JavaScript により、様々な図形や文字、画像をクライアント側のみの動作によって Web ブラウザ上に表示出来るようになった。従来の方法であれば、ある Web ブラウザ上で操作された手順を Ajax を用いてサーバに一旦保存し、その内容を別の Web ブラウザから再度読み込むことで参照を可能としていた。操作を再現する側のブラウザは、サーバにアクセスしたタイミングで初めて別のブラウザで行われた操作内容を受信することが出来るため、定期的な確認が必要とされていた。

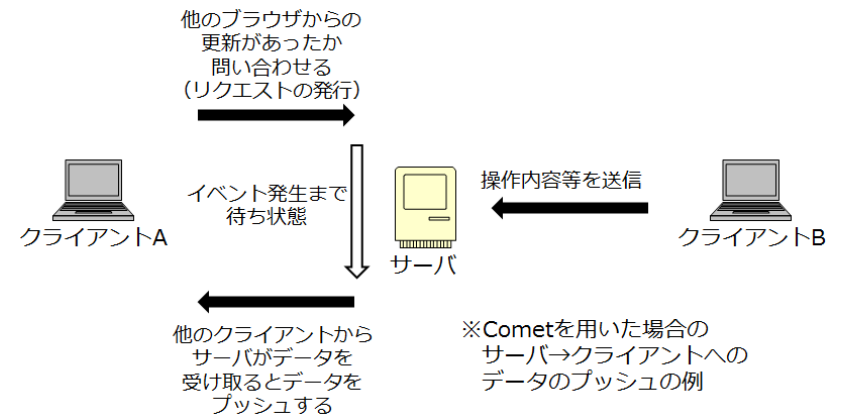


図 5 従来の方法による指導環境

しかし WebSocket を組み合わせることで、ある Web ブラウザ上での操作をサーバと繋がっているクライアント全ての Web ブラウザ上で同時に再現することが可能となった。これにより、Web ブラウザ上でプログラミングソースに関する 1 対 1 の指摘、指導や、1 対多での意見交換なども行えるようになった。

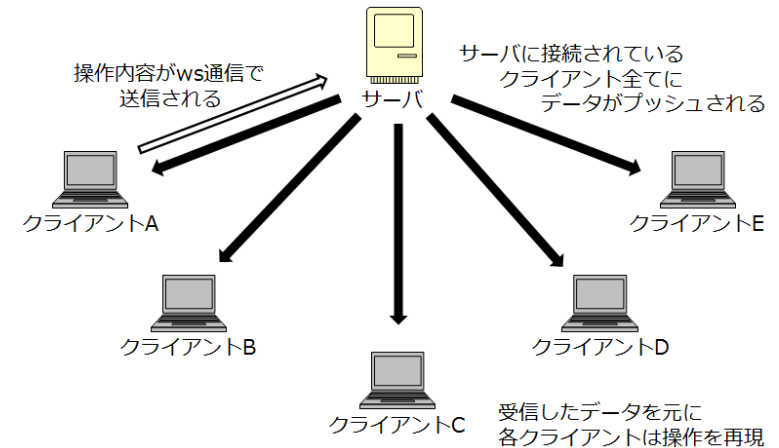


図 6 WebSocket を用いた場合の指導環境

図 7 手書き入力の実装に示すのは、実際に実装した手書きによる描写の様子である。WebSocket により、あるクライアントで行われた入力操作がサーバに接続されている別のクライアントに配信され、その入力操作を再現している。

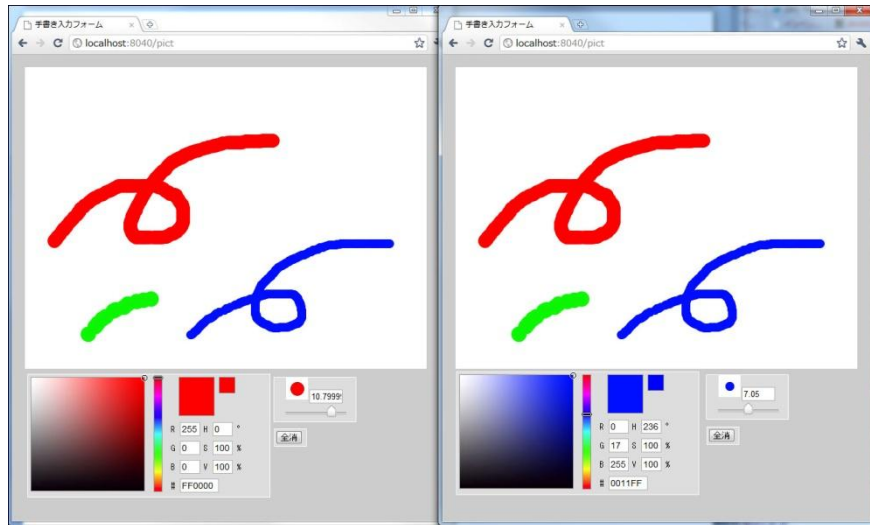


図 7 手書き入力の実装



図 8 iPhone4 での表示例

図 8 iPhone4 での表示例 8 に示したのは、iPhone4 での表示例である。iPhone4 は Flash の表示に対応していない端末であるが、HTML5 に対応したブラウザを使用することが出来るので、WebSocket を用いた Web アプリケーションを利用することが出来る。

手書き入力での支援機能の一つとして、現在操作している画面のスクリーンショットを送信し、その送信された画像に手書きによる描写を行うことも検討した。これには、java.awt.Robot クラスを用いて指定した範囲のスクリーンショットを取得し、WebSocket により送受信する方法で実装した。現在の WebSocket の仕様では ws 通信でバイナリデータを送受信できない為、取得した画像データを base64 化して送受信する方法をとった。この方法によってアイコンのような小さな画像の送受信が出来ることは確認できたものの、画面全体を映したような大きな画像の送信はできなかった。これは画像データを base64 化した際にとっても長い文字列に変換されることが原因と考えられ、データの分割、結合、また分割するサイズの最適化等の問題があり完全な実装には至らなかった。

3.2.3 アラート機能

大学の講義のような学習環境下で、学習者が解決できない問題がある場合に指導者に対して意思表示をする機能を実装する。形式としては、学習者と指導者全体が参照できるチャットのようなものである。WebSocket を用いてリアルタイムに状況を把握できるため、指導者は学習者がどのような問題で躓いているのか、現在どの問題に取り組んでいるのか等が把握しやすくなる。このような機能と学習者の位置関係を結びつけることでアラートの役割を持たせ、TA の派遣などに役立てる。

3.2.4 学習者間の情報共有

学習者間で情報をやり取りできる場を設けることで、学習者同士の支援や、過去の質問に対する回答の閲覧なども可能とする。学習者同士の支援環境を作ることで、課題をこなすだけでなく、他の学習者が躓いている問題の解決法を考える機会を与え、より深い学習へとつなげる。

4. おわりに

4.1 まとめ

HTML5 を用いた Web ベースの教育支援システムの構想を述べ、現在の実装状況と課題について報告した。WebSocket を用いることで、学習者が行っている学習の進捗状況の取得や、学習者と指導者間のやり取りがリアルタイムに行うことが可能となった。さらに、複数のクライアントによる 1 対多の相互の通信を同時に行うことが可能となった。これによって、特に手書き入力による支援の実装では、1 対 1 の指導だけでなく 1 対多によるリアルタイムなコミュニケーションが可能となったので、複数の学習者を交えたディスカッション等が出来るようになった。今回主に利用した WebSocket や Canvas の他にも HTML5 には様々な仕様が追加されるので、Web ベースの学習支援環境はさらに充実したものとなると考えられる。

4.2 今後の課題

画面のスクリーンショットの送信が可能となれば、指導者は学習者が躓いている状況をより詳しく知ることができる。また、受信した画像に対して手書きによる描写を行うことが可能になれば、指導者からの指摘をより的確に学習者は把握することができる。また、学習者や指導者を交えたディスカッションにも効果的に利用出来ると考えられる。このようなことから、画面のスクリーンショットの送受信が今後の課題として挙げられる。しかし WebSocket の仕様は現在も策定中であり、現在の仕様では、ws 通信ではテキストデータしか送受信できない。しかし、今後仕様の策定が進むにつれ、バイナリデータをはじめとした様々な種類のデータのやり取りが可能となると考えられるので今後の動向に注目したい。

謝辞

本研究は科研費 (20500836) の助成を受けている。

参考文献

- 1) Yusuke Mimoto, Koji Kagawa: A Framework for Web-based Applications for Learning Programming using Eclipse RCP, World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-Media 2008), Vienna, Austria, pp.2253—2258 (2008)
- 2) Koji Kagawa : WappenLite: a Web Application Framework for Lightweight Programming Environments, 9th International Conference on Information Technology Based Higher Education and Training (ITHET 2010), April 2010, Cappadocia, Turkey, pp.21—26
- 3) Zhe Chen and Koji Kagawa : WappenLite and its Flash-based User Interface for Novice Learners, The 2010 IEEE International Conference on Information and Automation (ICIA 2010) , June 2010,

Harbin, China, pp.840—845

4) 川ノ上 真進: Comet を用いたプログラミング学習支援環境の構築, 香川大学大学院工学研究科平成 21 年度修士論文 (2010)

5) Kimberle Koile and David Singer : Improving Learning in CS1 via Tablet-PC-based In-Class Assessment, ICER'06, September 9—10, 2006, Canterbury, United Kingdom.