

# オープンソースによるクリッカーシステムの提案と試作

大見 嘉弘<sup>1</sup> 水谷 正大<sup>2</sup> 永井 保夫<sup>1</sup>

**概要:** クリッカーを安価かつ簡便に利用できることを意図した、オープンソースであるクリッカーシステムを提案し試作した。試作では、教員がPC1台とWi-Fiルータを用意し、学生がそのルータに接続し、PC上のサーバとWi-Fi通信する方式を前提とした。本システムは、学生のユーザ認証を行わないことなどにより、学生個人と回答との紐づけを一切排除した。これは、学生が気軽かつ正直に回答することを期待したものである。また、Wi-Fi通信が混雑する恐れや高い双方向性を実現するために、学生が回答する時の通信方式にWebSocketを採用した。本システムの動作試験を行ったところ、少なくとも10名強の回答者であれば、クリッカーシステムとして十分に利用できることが分かった。

## Proposal and Trial Implementation of Open Source Clicker System

YOSHIHIRO OHMI<sup>1</sup> MASAHIRO MIZUTANI<sup>2</sup> YASUO NAGAI<sup>1</sup>

**Abstract:** We propose and implement an open-source clicker system which is intended to use as a handy classroom tool inexpensively. This system is composed by a single PC and a Wi-Fi router if needed brought by a teacher, and student's devices connectable to the router communicating to a server on the PC. This clicker system is carefully designed to avoid to leave individual signatures of students inferable from these responses for sake of collecting honest and quick feedback of students. The system server is implemented by means of WebSocket for communication with students devices, because of the rapid coding of rich interactivity and the relaxing of congestion which might be caused by Wi-Fi communication. The rough performance test using popular Wi-Fi routers suggests this system is of practical.

### 1. はじめに

学生の授業への関与が消極的であると指摘されることは多いが、それは日本の教育固有の理由もあり、人前でわからないと表明したり質問をすることがはばかれるという事情も絡んでいるのではないだろうか。クリッカーは授業中に教師が投げかけた3択程度の質問にボタンを押すだけで、赤外線や電波などの帯域を使って集計機に届けられ、それぞれの回答数を即座に集計表示する機器+システムである。

クリッカーの応答に固有番号を含ませて、学生の出欠管理や応答記録を保存できるシステムもある。一方、それぞれの応答（たとえば、分かった/分からない）が個人に紐付けされるようなことがなければ（さらにクラス人数が多ければ）、学生は気楽にしかも正直に応答を返すことが期待

できる（これがクリッカーの当初の姿であった）。

クリッカー導入による教育効果はクラス全体の応答傾向（たとえば理解度）がリアルタイムで教師と学生とで共有可能であることにあり、この観点に立った成果研究が多数報告されている [1-9]。学習ポートフォリオなど学生の学習行動を記録する目的のためにクリッカー用途を拡大しようとする、その要求に即した新たなクリッカー機器が必要となる。それぞれの用途に応じてパッケージ化されたクリッカー機器は扱いやすいのだが、基本機能を備えたクリッカー単体の経費は比較的安価であるとはいえ、クラス単位でクリッカーを体験するにはそれなりの導入コストが必要となる [3,10]。

実際、筆者らはクリッカー利用の教育的意義を感じながらも、実際のクラスへの導入はできないでいた。そこで、クリッカーに興味を持つ教員たちが安価にしかも簡便に自分のクラス内で手軽に利用できるソフトウェアクリッカーの開発を構想してきた。

<sup>1</sup> 東京情報大学  
Tokyo University of Information Sciences

<sup>2</sup> 大東文化大学  
Daito Bunka University

## 2. クリッカーアプリケーションの要件

クリッカーアプリケーションの開発において、まず最初に考慮されるべきことは学生が「クリック」する機器として何を想定するかである。クリック結果はサーバに集められて集計処理されるわけだが、このクリックの機器の想定はどのような経路でクリックすべきボタン（選択肢）をクリック機器に送信し、クリック結果をどうやってサーバに伝送するかの問題でもある。質問に答える学生は指定された URL (IP アドレス+ポート) によってサーバにアクセスして、クリック機器に表示された質問とその応答番号（選択肢）をクリック（タップ）するのである。

学生のクリック機器として、PC またはスマートフォンなどの IEEE 802.11 無線 LAN (Wi-Fi) で通信可能な機器、または公衆電話回線のみで Wi-Fi 接続のできないフィーチャーフォン（通称「ガラケー」）が考えられる（赤外線は現状のスマートフォン仕様では考察対象外であるとした）。そして、開発の効率や難易度を考慮し、クリッカー機器に適合したネイティブアプリケーションを開発するのではなく、Web アプリケーションとして開発することにした。ここで考慮すべきことは、ソフトウェアクリッカーシステムを利用する教員には学校から格別の（設備的または資金的）支援が期待できないこと、しかも可能な限り ICT スキルを要求せず、クリッカーシステムがだれでも簡単に利用できることである。

これに対する一つの在り方は、クリッカーシステムをインターネット上の Web サービスとして稼働させることである。クリック応答を処理する Web サーバにアクセスするための障壁として次の3つが考えられる：1) クラス内または学校内で LAN 環境が整っていない、2) LAN 環境がつかえたとしてもクリック機器とサーバホスト間の通信が禁止されている設定である場合（筆者の一人が勤めている大学の環境がそうで、互いに ping すら通らない）、3) クリッカー利用を考えている教員の Web アプリケーションを構成・設定するためのスキルがない（協力が得られない）。インターネット上の Web サービスはこれらの課題を解消できる。このサービスを使用したい教員のために、サーバ側はサービスアカウントを発行したうえで、Web 経由で質問項目を登録したり学生からの応答結果集計をダウンロードできるようなサービス提供するのである。Web サーバへのアクセスについての課金問題はあるとしても、インターネット上の Web サーバへのアクセスであるならば LAN 環境が一切なくとも学生のスマートフォンやフィーチャーフォンからおおむね問題なくクリック結果を送信することができる。

ただし、インターネット上の Web サービスは、学生の応答結果のみを保管し機器データなどをシステム上に残さない仕様にしたとしても、授業に関するデータを第三者が用意したサーバ内に置いておくことに躊躇を感じる教員は少なくないと思われる。それでも、こうして複数クラス（または学校）にサービスを提供し得る Web サービスでは、利用教員（と学生）の許可をうけた集計結果を利用者間でシェアするということが可能になり、クリッカー利用の新しい局面を切り拓く可能性があることを指摘しておきたい。

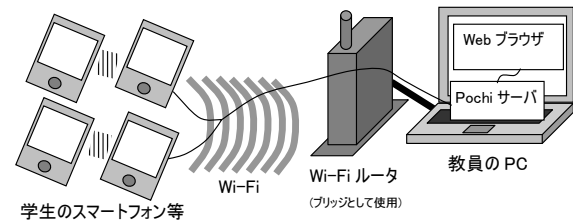


図 1 Pochi のシステム構成

本研究では、LAN 環境下にある教員（持参）の PC 上で Web アプリケーションを稼働させるクリッカーシステムを提案する。LAN 環境が存在しないまたは設定に問題がある教室では持参した Wi-Fi ルータ（基地局）に教員の PC を接続し、ローカル IP アドレスを公示して学生のクリック機器を Web サーバに接続するのである。5 章で検討するが、このクリッカーシステムは Wi-Fi 接続のできないフィーチャーフォンでは利用できない。教員パソコンで稼働させる Web アプリケーションは、汎用の HTTP サーバなどは利用せずに、クリッカーシステムとして当面必要な機能を持たせたスクリプトプログラムを書いて、最小限の ICT スキルさえあればシステムを起動し運用できることを目指した。それでもインストール作業が必要になるが、その部分については誰かの協力をおおぐとし、クリッカーシステムの起動の仕方さえ教えてもらえれば、質問の登録作業や応答処理作業などは全て Web から行えるように企画した。システムの検証は Windows および Mac OS で行っている。

## 3. オープンソース クリッカーシステム Pochi の開発

### 3.1 開発方針

開発するクリッカーシステム Pochi は、教員が提示する質問に対して学生が所有するスマートフォンなどのモバイル端末<sup>\*1</sup>を用いて回答するという使用法を想定する。ネットワーク接続については、教員が PC 1 台と Wi-Fi ルータ<sup>\*2</sup>を用意し、学生はモバイル端末をそのルータに接続し、PC 上のサーバと Wi-Fi 通信することでクリッカーシステムを実現する（図 1）。また、教室でモバイル端末を Wi-Fi 接続できる環境が既にあり、そこから教員 PC 上のサーバにアクセスできれば、その環境を利用することもできる。さらに、今後開発する携帯電話回線を通じてインターネット上の Web サーバにアクセスする方式も考慮した設計とする。

今回はフィーチャーフォンを排除したことにより、従来のクリッカーシステムをそのまままねるのではなく、スマートフォンならではの高性能を活用することにした。まず jQuery および jQuery Mobile などの JavaScript による技術を用いてネイティブアプリに近い操作性を実現する。また、Wi-Fi 通信が集中する恐れや双方向性の高いアプリケーションを意図して WebSocket [11] を用いた通信を多用する。

本システムは、クリッカーの当初の考えを尊重し、学生

\*1 タブレット端末や Wi-Fi 機能のある PC でも利用可能である

\*2 Wi-Fi アクセスポイント（ブリッジ）として同一セグメント（LAN）内で使用する。ルータ機能は使用しない。

の応答を個人に紐づけしないことにする。まず、本システムでは、学生は一切のユーザ認証を行わない。学生の回答時の個別認識は原則として IP アドレスを使用するが、最終的に保存されるデータには IP アドレスを含めず、選択肢ごとの回答数だけを残すことにする。これにより、例えば問1で2を選んだ者が問2でどれを選んだというような質問毎の回答の関連を知ることが一切できなくなる。しかし、学生が安心して回答できることを優先し、敢えてそのような関連を知ることが断念した。

### 3.2 サーバ側の開発

本システムは、Apache などの汎用 HTTP サーバを使用せず、サーバプログラムを JavaScript で記述する Node.js を用いた。これによりサーバ、クライアント共に JavaScript で統一し、高い開発効率を狙った。特にデータ表現方式を JSON 形式に統一できることが効果的であった。

### 3.3 サーバ - クライアント間通信

サーバ - クライアント間の通信には、WebSocket を多用した。違う Web ページに遷移する場合は、従来の HTTP アクセスであるが、学生の回答や質問開始時の通信などに WebSocket を用いた。

WebSocket は TCP/IP におけるソケット通信に代表される非同期双方向通信を HTTP を拡張することで実現したものである [11]。World Wide Web の普及でインターネットは発展したが、Web の通信プロトコルである HTTP には多大な冗長性がある。また、HTTP は原則として、クライアントからサーバへの要求が原則であり、その逆はできない。このため、クライアントからサーバに定期的にポーリングを行うなどして、サーバからクライアントへの通信（いわゆるサーバプッシュ）を実現する Ajax が登場し普及した。しかし、ポーリングを行う Ajax はトラフィックの増大や応答速度に大きな弱点を抱えており、効率の良いプッシュを実現する手段が望まれていた。

以上の状況から、WebSocket が提案された。WebSocket は最初に HTTP リクエストの拡張によりソケット通信のモードに切り替えられる。このため HTTP の標準である 80 番ポートがそのまま使用でき、ポート制限による通信不能の問題が起りにくい。また、WebSocket は軽量プロトコルとして設計されており、効率の良い非同期双方向通信が可能である。本システムでは、学生の多数の端末を Wi-Fi 接続するため、有線 LAN に比べ格段に輻輳が起りやすい。WebSocket を用いることで HTTP に比べ、パケット数を少なく、パケット長も短くなることで、輻輳が起る可能性を減らせ、同時アクセス数を増やせることが期待できる。

WebSocket は策定途中であるが、現在、主要な Web ブラウザのほとんどが対応している。ただし、Android の標準ブラウザは対応していない。本システムでは、JavaScript を用いて WebSocket 通信を行うために Socket.IO ライブラリを用いている。Socket.IO は WebSocket 以外のプロトコルに対応しており、このため本システムは、Android の標準ブラウザでも利用できる。しかし、応答速度やトラフィックについては期待できない。

WebSocket により通信する場合、独自にプロトコルを規



図 2 アドレス表示機能

定する必要がある。本システムでは、特にパケット長を短くしたい学生の回答は、イベント名と回答番号のみを送ることで極力短くした。その他の通信では、伝えたいデータを JSON 形式にして送ることにした。本システムはサーバ、クライアント共に JavaScript で開発するため、JSON は親和性が高く、高効率で信頼性の高い開発が実現できたと考える。

### 3.4 クライアント側の開発

本システムは、Web ブラウザ上でリッチアプリケーションを実現するため、jQuery、jQuery Mobile などの JavaScript ライブラリを用いる。これにより、スマートフォン上のネイティブアプリに近いデザインと操作感で本システムを利用できる。また、WebSocket を用いた通信により、双方向性の高い操作感を実現した。例えば、教員が本システムで質問開始の操作をすると、学生の端末に即座に質問内容が表示される。

以下に本システムの典型的な利用手順を示す。

- 1) まず、Wi-Fi アクセスポイントを立ち上げ、学生は端末をアクセスポイントに接続する。アクセスポイントの設定は、本システムと独立しており、SSID や暗号化方式やパスワードの設定は、教員の自由である。大勢の回答者がいると、Wi-Fi の 1 チャンネルでは輻輳し、本システムが満足に機能しない場合がある。その場合、複数のアクセスポイントを設けたり、5GHz 帯も利用するなどの対策が必要となる。
- 2) 学生は端末の Web ブラウザを起動し、指定した URL を入力して本システムに接続する。URL は、アドレス表示機能で教員画面に URL とその QR コードを表示することができる<sup>\*3</sup>(図 2)。
- 3) 教員は、あらかじめ設定した質問集から一つを選ぶ。そうすると最初の質問が開始される。その時の教員画面を図 3、学生画面を図 4 に示す。従来のクリッカーでは、教員の PC 画面をプロジェクト等で学生に見せることで質問していた。本システムも同様の形態で使えるが、学生の端末に質問と回答の選択肢が表示されるため、教員画面を見せなくても機能する。このため、プロジェクト等が常設でない教室

<sup>\*3</sup> URL は教員 PC のネットワークインタフェース (NIC) に割り当てられている IP アドレスから求めているため、複数の NIC が存在する場合は、間違った結果となる場合がある。



図 3 質問画面 (教員)

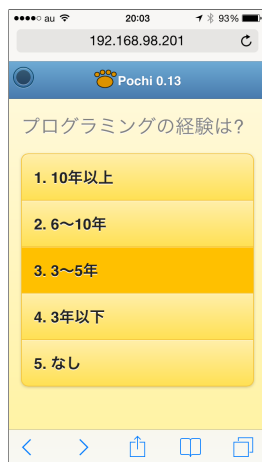


図 4 質問画面 (学生)

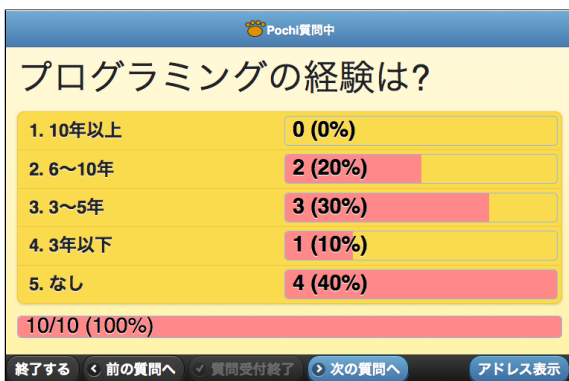


図 5 結果画面 (教員)

でも手軽に利用できる。

- 4) 学生は回答選択肢の一つをクリック (タップ) する。サーバに回答が伝わると選んだ選択肢がハイライトされる。図 4 は、学生が 3 を選択した様子である。質問中は、教員画面に回答した人数がグラフと数字で即時表示される。教員はそれを見て全員が回答したと判断して質問終了の操作を行う。そうすると、図 5 のように教員画面に選択肢ごとの回答者数がグラフ表示される。

教員が質問する内容は、複数の質問を 1 つのリストとして管理するようになっており、このリストを質問集と呼ぶ。教員はあらかじめ作成した質問集のうち 1 つを選び、順に

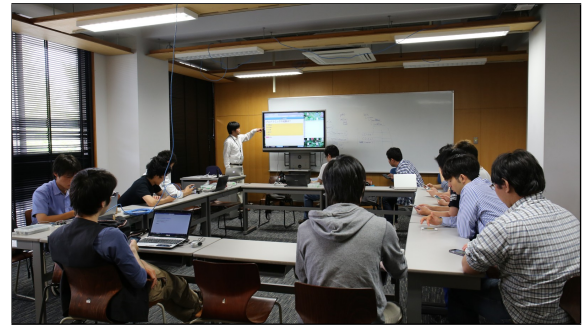


図 6 動作試験の様子

質問と回答を繰り返すことが典型的な使用法である。質問集の編集は、本システム上の編集画面で行うことができる。また、質問集は CSV ファイルとしてインポート、エクスポートが行えるようにしている。

最後の質問とその回答を終了するか、途中で中断操作を行うと、一連の回答結果が自動的に保存される。過去の回答結果は、結果表示機能ですべて確認できる。また、個々の結果を CSV ファイルとして保存する機能も設けた。

### 3.5 本システムの配布

本システムは、広く利用していただけるようオープンソースとして公開している [12]。インストーラは設けていないが、必要とする Node.js などは手順通りであれば、比較的容易<sup>\*4</sup> にインストールできる。

本システムのライセンスは、極力幅広く使ってもらえるように、利用制限が緩い MIT License とした。

## 4. Pochi の適用と評価

### 4.1 動作検証

本システムが、要求通りに機能するかを検証するために、実際の授業を想定した環境で動作試験を行った。学生として大学生 11~18 名<sup>\*5</sup> が参加した (図 6)。参加した学生 18 名のうち、17 名がスマートフォン (iPhone: 7 名, Android: 10 名) を携帯しており、フィーチャーフォンを携帯している 1 名はタブレット端末 (Nexus7) で実験を行った。また、Android OS のメジャーバージョンはすべて 4 であった。

ネットワーク構成は、据置型の Wi-Fi ルータ (NEC WR8750N) を Ethernet でノート PC に接続し、学生に Wi-Fi 接続させた。

最初の実験では、質問を提示した後、各自自由なタイミングで回答するようにした。その結果、大きな遅延もなく本システムは機能した。ところが、タイミングを計り全員同時に回答するようにしたところ、応答に 2 秒以上かかる者が 2 名あった。さらに、教員画面に表示される回答者数の表示が最大 4 秒程度遅れていることが分かった。この原因として、以下の 2 つを疑った。

- 1) **WebSocket** に対応していないブラウザを使用している

当初は、Android の標準ブラウザを使用する学生が多いため、Chrome をインストールさせ使用させた。そ

<sup>\*4</sup> Node.js インストーラのダウンロードとインストール、npm によるパッケージのインストールという 2 つの作業だけで済む。

<sup>\*5</sup> 数日に渡り試験したため、日により参加人数が異なる。

表 1 WebSocket と HTTP の比較

	WebSocket	HTTP Get
パケット数	4	8
総バイト数	277	1254

の後は、遅くとも 2.0 秒以内に全員が回答を完了できるようにになった。

## 2) 教員画面の表示が追いついていない

当初、0.2 秒毎にサーバから Web ブラウザ上の教員画面に回答者数を送っていた。そのグラフの描画に時間がかかり、描画の遅延が累積していることが推察された。このため、0.5 秒ごとに回答者数を送るようにしたところ、1.5~2.0 秒程度で全員が回答したと表示されるように改善された。

以上の改善後、教員 PC 画面をビデオ撮影することで、応答時間を測定した。その結果、回答者数が 11 名で全員同時に回答した場合、1.17~1.67 秒で全員の回答がサーバに伝わり、1.30~1.67 秒で教員 PC に表示されることが分かった\*6。以上の実験で 10 名強の回答者数であれば、本システムは大きな遅延もなく実用できることを確認した。

また、Wi-Fi ルータとして、小型のルータ (PLANEX MZK-RP150N) を用いて同様の実験を行った。このルータは USB バスパワーであり、ノート PC との組み合わせでコンセントのない場所でも機能するため使い勝手が良いが、同時アクセス時の性能に不安がある。実験の結果、多くの場合は、据置型の Wi-Fi ルータと同等の応答時間であったが、ときどき 3 秒前後まで遅くなることがあった。この結果から、本システムは据置型の Wi-Fi ルータを用いるのが無難であると感じた。

さらに、学生が Wi-Fi 接続に切り替えると、LINE などのアプリが「インターネットに接続されていない」という旨の警告を表示した。それらのアプリが常駐しており、本システムのようなネットワーク構成で警告を発する。この警告は非表示にできるが、本システムを初めて使用する学生が戸惑う一因となり、留意する必要性を感じた。

## 4.2 WebSocket の評価

WebSocket と従来の HTTP による通信について比較を行った。tcpdump コマンドを用い、本システムで学生が回答を行った場合 (WebSocket 使用) と、Apache サーバに HTTP の GET メソッドを用いてアクセスした場合の IP パケットを取得した。その結果を表 1 に示す。

以上から WebSocket は HTTP に比べ、パケット数で半分、パケットの総バイト数で 22% であり、十分に軽量であることが確認できた。

## 5. 授業利用

本研究で開発したクリッカーアプリケーションシステムは、有線接続または Wi-Fi 接続されたパソコンやスマートフォンをクリッカー端末として想定しており、フィーチャーフォンは端末として想定しなかった。大学生を対象としたアンケートでは、大学を問わずにおおむね 9 割以上の学生が Wi-Fi 接続可能なスマートフォンを所有してい

る。本システムでは、学生個人 (または IP アドレス) とそのクリック応答とはシステム上紐付けしておらず、そもそもクリッカー利用の主目的をクラス前提の学習状況の把握としたため、かならずしも全数把握にはこだわらないとした (それでも、将来的には公衆電話回線を経由したインターネット Web サービスとしてのシステム拡張の余地は残しているのだが)。

想定しているクラス利用は、10 名程度のセミナー、40 名程度の情報リテラシー授業、100 人以上の大規模授業であるが、とくに大規模クラスでは Wi-Fi ネットワークの運用 (Wi-Fi ルータの選定、多人数からの同時接続や接続の維持など) に問題が残っており、現在、客観データとしての蓄積に至っていない。しかし、クリッカー導入の必要性を強く感じてきた大規模授業でクラスの趨勢を把握できる程度の Wi-Fi 接続数の達成に向けた工夫・改良には十分な意味があると思われる。

## 6. 関連研究

ここでは、クリッカーアプリやクリッカーシステムの開発に関する研究と、本研究との関連について述べる。

まず、クリッカーアプリの開発に関する研究としては、以下があげられる。児島ら [13] は、スマートフォンを利用してクリッカーの機能を実現したアプリ開発に関する研究をおこなっている。このアプリは Android および iOS を搭載したスマートフォンに対応している。授業でクリッカーアプリが利用できるかを確かめるために、授業中にアプリをダウンロードさせて、クリッカーを体験させた。その後、学生にアンケートをとることで、クリッカーの理解度について調査を実施した。

山田 [10] は、授業において利用することを目的として、自作のクリッカーシステムを作成し、個人で所有できる安価なクリッカーシステムを実現している。ここでは、赤外線受信部を自作することにより、赤外線式のリモコンを利用したクリッカーシステムを自作している。しかしながら、このシステムでは、他の人のリモコンの信号を重ねた場合に認識されなかったり、一人で何度も投票できるなどの問題点が出ている。さらに、赤外線方式のため、受信可能距離が短く、受信可能角度も狭いということも明らかになっている。

古賀ら [14] は、スマートフォンデバイスでの利便性を追求した Web システム版クリッカーの開発に関して研究をおこなっている。具体的には、回答送信リモコン、回答受信装置、集計結果表示機能を Web アプリケーション化し、スマートフォンからリモコン Web アプリにアクセスさせることで、ハードウェアコストの低減をおこなっている。これにより、教員と学生との双方向コミュニケーションを可能とした Web システムが実現されている。

次に、クリッカーシステムの利用による研究としては、以下があげられる。佐藤 [15] は、クリッカーやワイヤレスデバイスを用いて、教員の自由な移動性を向上させた学生参加型授業に関する研究をおこなっている。これにより、ICT 活用授業においても板書と机間巡回を同時に行うことが可能と示されている。飯高 [16] は、参加型授業の実現方式を検討するために、クリッカーの機能を携帯電話の端末を用いて Web アプリケーションとして実現する研究

\*6 ビデオ撮影の測定のため、30 分の 1 秒の精度である。

をおこなっている。教員の質問に対して受講者が応えることで、回答の情報をリアルタイムで集計し、教員側の PCなどで表示をおこなっている。また、飯高 [17] は、クリッカーに代表される参加型・相互的授業支援を実現するために、CMS である NetCommons の小テストモジュールに、受講者が携帯電話からのアクセスをおこなう研究を実施している。

さらに、川上ら [18] は、多機能教育用応答システム端末を実現するために、クリッカー端末を発展させることで、Web ブラウザに FeliCa を用いた認証のための拡張を施した多機能教育応答システムを提案している。

以上のようなクリッカーアプリやクリッカーシステム開発ならびにそれらの利用に関する研究と比べ、我々の研究ではオープンソース化することで、幅広い利用や有志による機能拡張が期待できる。例えば、学習履歴分析などのように学生の学習行動を記録し分析する目的のためにクリッカー用途を拡大できる。また、他の Web アプリケーション開発による研究に比べ、WebSocket を採用したことで、学生の同時最大接続数において有利であることが期待できる。この取り組みにより、クリッカーに興味を持つ教員たちが安価にしかも簡便に自分のクラス内で手軽に利用可能となることを目指している。

## 7. おわりに

クリッカーを安価かつ簡便に利用することを意図したオープンソースであるシステムを試作し、少なくとも 10 名強の回答者であれば十分に利用できることを確認した。

最後に本研究の今後の課題について述べる。

一番の課題は本システムを利用するための平易な案内文書を用意することである。本システムの中核となる Web システムの導入や運用を案内する文書も不足しているが、それよりも特に Wi-Fi ルータを適切に設定することの技術的障壁が高いと感じた。

本システムでは、Wi-Fi ルータ上で DHCP サーバを機能させ、教員 PC と学生端末の IP アドレスを自動的に割り当てることを原則としている。その代わりに教員 PC 上で DHCP サーバを機能させる場合、OS 毎に事情が違い複雑であり、DHCP サーバの停止を忘れて他のネットワークに接続した場合に多大なトラブルを起こす懸念がある。

Wi-Fi ルータの多くには、ルータモードとアクセスポイント(ブリッジ)モードに切り替えられる機能があるが、アクセスポイントモードでは、DHCP サーバを有効にできない機種が多い。したがって、本システムで使用する場合は、ルータモードにして LAN ポートのみ使用するという、ルータとしては不自然な使い方を強いることになる。このような具体的なルータの設定に関する案内はまだ用意しておらず、一番の課題と考える。しかし、Wi-Fi ルータには様々な機種があり、設定方法もメーカーや機種により様々であるため、具体的な説明が難しい。このため、まずは推奨するルータに絞って具体的な設定方法を文書化する予定である。

他の課題としては、本システムの GUI の改善が挙げられる。特に質問集の編集画面を分かりやすくする必要がある。また、画面描画の遅さについては、jQuery Mobile とグラフを表示するために使用した jQuery Easy UI ライブ

の性能の問題と考えられる。jQuery Mobile は最近のバージョンで性能向上を果たしているため、今後置き換えを図りたい。また、グラフ描画は別の手段を試し性能向上を図りたい。

本システムは、現状でもインターネット上に公開された Node.js が使用できるサーバがあれば、携帯電話回線を通じて使用可能である。ただし、サーバー一台当たり、1 クラス(教員 1 名)での使用に限られる。今後はユーザ登録機能を設け、複数教員で同時利用できるシステムに発展させたい。

## 参考文献

- [1] Caldwell, J E: Clickers in the large classroom: Current research and best-practice tips, CBE Life Sc Edu, 6(1), pp.9-20 (2007).
- [2] Margle, M: Clickers in the classroom an active learning approach, Educause Quartely, 30(2), pp.71-74 (2007).
- [3] 鈴木久男, 武貞正樹, 引原俊哉, 山田邦雅, 細川敏幸, 小野寺彰: 授業応答システム”クリッカー”による能動的学習授業 -北大物理教育での 1 年間の実践報告-, 高等教育ジャーナル-高等教育と生涯学習- 16, pp.1-17 (2008).
- [4] 武田直仁: クリッカー(授業応答システム)の効果的な活用法を求めて, 平成 25 年度 教育改革 ICT 戦略大会 (2013).
- [5] 石田健一, 古井陽之助: クリッカーの導入による双方向性教育の試み, 九州産業大学総合情報基盤センター COMMON Vol.33 (2013).
- [6] 金子劭榮, 新村知子, 稲葉宏和, 桑村佐和子: 双方向性の高い授業を目指して—クリッカーの可能性を探る—, 石川県立大学年報 21, pp.29-37 (2010).
- [7] 山田邦雅: クリッカーの有効性と効果拡大に向けた開発, 物理教育第 59 巻 第 1 号, pp.40-43 (2011).
- [8] 籠谷隆弘: 授業応答システムと学習管理システムを活用した授業実践, 仁愛大学研究紀要 人間生活学部篇 1, pp.83-88 (2009).
- [9] 猫田泰敏: 疫学講義におけるクリッカーの使用と学生の反応, 日本看護研究学会雑誌 Vol.35 No.1 pp.137-143 (2012).
- [10] 山田邦雅: 自作クリッカーシステムによる授業, 高等教育ジャーナル -高等教育と生涯学習- 16, pp.19-29 (2008).
- [11] I. Fette and A. Melnikov: The WebSocket Protocol, RFC6455 (online)  
入手先 (<<http://tools.ietf.org/html/rfc6455>>)
- [12] 大見嘉弘: Pochi (online)  
入手先 (<<https://github.com/YoshihiroOhmi/Pochi>>)
- [13] 児島完二, 三輪冠奈: クリッカーアプリの開発と試用, 2012 PC Conference pp.355-358 (2012).
- [14] 古賀掲維, 柳生大輔, 野崎剛一: スマートフォンデバイスでの利便性を追求した Web システム版クリッカーの開発, 大学情報システム環境研究, VOL.16, pp.43-50 (2013).
- [15] 佐藤宏介: ワイヤレス ICT 機器を活用した机間巡回可能な学生参加型授業, 平成 21 年度工学・工業教育研究講演会講演論文集, pp.574-575 (2009).
- [16] 飯高敏和: 参加型授業を実現する CMS モジュール, 信学技報 IEICE Technical Report ET2011-96 (2012).
- [17] 飯高敏和: NetCommons の小テストモジュールを用いた双方向講義の実現, 信学技報 IEICE Technical Report ET2010-99 (2010).
- [18] 川上孔明, 鈴木真: 多機能教育用応答システム端末と対応する Moodle プラグインについて, 情報処理学会 研究報告 IPSJ SIG Technical Report Vol.2011-CE-112 No.3 (2011).