

平成 30 年 6 月 19 日現在

機関番号：56203

研究種目：基盤研究(C) (一般)

研究期間：2015～2017

課題番号：15K00504

研究課題名(和文) アドホック/センサーネットワークの技術を活用したタブレット教育環境の構築

研究課題名(英文) Construction of education environment utilizing tablet devices by applying Ad Hoc / sensor network technology

研究代表者

高城 秀之 (TAKAJO, Hideyuki)

香川高等専門学校・通信ネットワーク工学科・准教授

研究者番号：30261118

交付決定額(研究期間全体)：(直接経費) 3,600,000円

研究成果の概要(和文)：本研究では、タブレットを一種のセンサーとみなし、タブレットを用いた小テストにより、授業中における学生の学習状況を逐次モニタリングすることで、リアルタイムに学生の理解度を把握し、授業中に再度説明したり補足したりすることを可能とするシステムの開発を行った。  
研究の結果、既存のe-learningシステムでは見られない双方向通信や高速データベースを利用することで、この目的を満たすシステムを構築できた。また、実際の授業で実証実験を行い、その有効性を確認した。

研究成果の概要(英文)：In this study, we regard a tablet as a kind of sensor, and have developed a Web-based e-learning system by which teachers can monitor student's understanding level through tablet-based quiz test in real time during class and explain again in the same class depending on the level.  
As a result of this research, we have build the system that satisfies this objective by using two-way communication and high-speed database that are not found in existing e-learning systems. In addition, we carried out demonstration experiments in actual classes and confirmed its effectiveness.

研究分野：IT技術を活用した教育システムの開発

キーワード：小テスト タブレット e-learning リアルタイムモニタリング WebSocket 双方向通信

### 1. 研究開始当初の背景

近年、タブレットの教育現場への導入が進んでいる。その一方で、導入当初はその目新しさから一時的に学生の興味を引くものの、やがて学生が飽きてしまうという問題も指摘されている。また、書籍の電子化が進み、今ではタブレット上で電子書籍の閲覧もできるようになったものの、タブレットが持つ通信機能や情報処理機能を十分活かしているとは言えず、単に書籍の媒体が紙から電子機器に変わっただけのものも見られる。

また、教育現場によっては、無線 LAN のアクセスポイントを設置することが困難な場合がある。そのため、ネットワークのインフラが不十分な環境でもタブレットを活用した授業を実施できる環境の構築が求められるようになった。

### 2. 研究の目的

現在普及している e-learning システムは、小テスト機能、教材コンテンツ提供機能、成績管理機能などを有しており、学生は授業中や授業以外の時間に同システムを利用して学習できるようになっている。一方、これら既存の e-learning システムでは、教師は小テストの成績を参照し、学生の理解状況を把握することはできるものの通常、それを参照するのは授業終了後である。

そこで、タブレットで閲覧および実行可能な電子教科書および小テストを用いて授業中に小テストを行い、その結果をリアルタイムに教師用端末に表示させることができれば、教師が学生の理解状況をその場で把握できるようになり、その結果を即座に授業に反映させることで、これまでより質の高い授業が可能になると考えた。

本研究の目的は、このためのシステム開発を行い、そのシステムを実際の授業で用いてその有効性と問題点を検証することである。合わせて、ネットワーク環境が不十分な場合でのシステム運用方法の検討を行なった。

### 3. 研究の方法

本研究では、小テストの解答状況をリアルタイムに教師に提示する。つまり、本研究では、小テストを学生の理解状況のモニタリングのために行う。これは、センサーを用いて監視対象の状態をリアルタイムにモニタリングする仕組みと同一である。ここでは、学生（の理解状況）をセンシング対象とし、タブレットを一種のセンサーと見立てている。そして、タブレットを通してセンシングした情報はサーバに集められ、それを教師のタブレット上に表示することで、教師はセンシング対象の状態をリアルタイムに把握することが可能となる。

このようなモニタリングを実現するためには、学生が小テストの1つの問題に解答するたびに、即座にその情報をサーバに送信し、その情報を教師が一目で把握できるような仕

組みが必要である。

以上を踏まえ、本研究では、e-learning システム上に設置したコンテンツと連携可能なリアルタイムモニタリングシステムの開発を行った。なお、本研究ではタブレット端末として iPad を用いるが、Android 系や Windows 系のタブレットでも利用できるよう、タブレットの OS に依存しない Web 技術を用いた。

### 4. 研究成果

#### (1) 開発システムの構成

本システムは図1のように、e-learning システム(以下、e-Learning 部)とリアルタイムモニタリングシステム(以下、リアルタイムモニタリング部)から構成される。e-learning 部では、学生のユーザー管理、成績管理、コンテンツ管理、小テスト管理などを行う。実装は一般的な e-Learning システムと同様である。一方、本システムの特徴でもあるリアルタイムモニタリング部は、通信プロトコルに WebSocket を用い、サーバのバックエンドに置くデータベースにも NoSQL タイプを使用するなど、従来の Web アプリケーションとは構成が異なる。

システム構成図

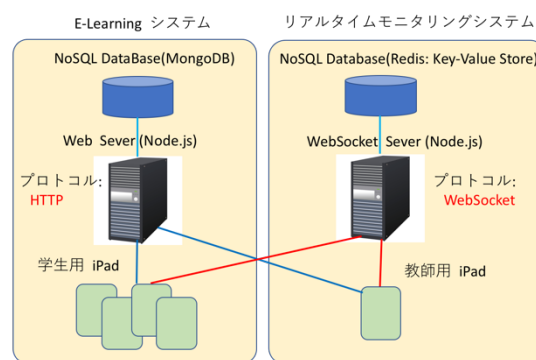


図1 システム構成

本システムではデータベースとして、NoSQL 型の MongoDB と RedisDB を採用した。

MongoDB は NoSQL 型のデータベースの中でも「ドキュメント指向」のデータベースとされ、階層構造のデータを格納したり、JSON 形式のデータの処理を得意とする。JSON (JavaScript Object Notation) 形式とは JavaScript 言語のオブジェクト表現をそのままデータ形式としたフォーマットであり、本システムの中核にある Node.js で使用する場合、データ形式の変換が不要になるというメリットがある。本システムでは、JSON 形式の小テストを格納するために MongoDB を用いた。

RedisDB は Key-Value-Store 型のデータベースであり、複雑なデータ構造を格納するには向かないが、非常に高速なアクセスが可能である。本システムでは、学生の小テストの解答状況をリアルタイムかつ高速にデータベースに保存する必要があることから、同データベースを採用した。

次に図1に従って、本システムの動作の流れを説明する。

まず、学生はタブレットから e-learning 部にログインする。e-learning 部は既存の多くの e-learning システムと同様な構成となっている。e-learning 部のコンテンツから小テストへのリンクをクリックし、小テストの Web ページに遷移すると WebSocket プロトコルに切り替わりリアルタイムモニタリング部に接続する。以後、小テストの解答はリアルタイムモニタリング部の WebSocket サーバに転送され、そこから、RedisDB に保存されると同時に教師の iPad 上へも転送される。

教師は、学生同様まず e-learning 部にログインし、そこからリアルタイムモニタリング部に接続する。以後、学生の iPad から送られてくるデータはグラフィクスに変換され iPad 上に表示される。

## (2)リアルタイムモニタリングについて

以下では、本研究で開発したシステムについて詳説し、最後に同システムを実際の授業で使用した結果について述べる。

図2および図3はそれぞれ、学生が iPad 上で小テストに解答している様子と、学生の iPad 上に表示された小テスト画面である。

学生は、まずタブレットから本システムにログインする。小テストはあらかじめサーバ上に複数用意されており学生はそこから、授業中に教師から指定された小テストを選択する。

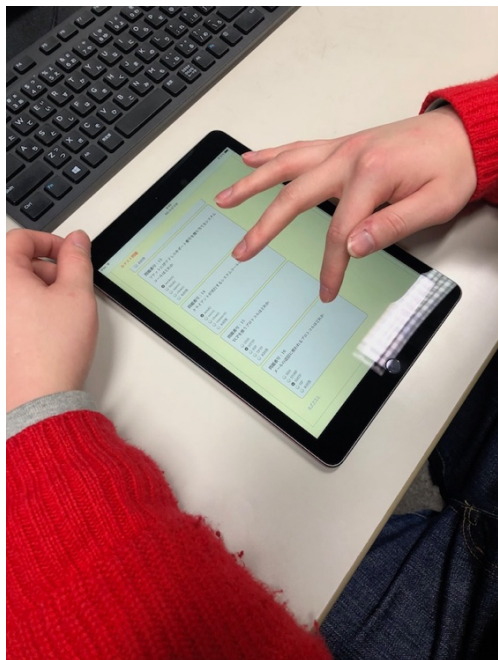


図2 小テストの解答の様子

小テストはタブレットで解答できる4択式となっており、各小問に対してタッチ操作で解答する。ここで、通常の e-learning システムの場合は、全ての小テスト問題に対する解答選択が終了した時点で「送信」ボタンを押

すことにより解答がサーバに送られる仕組みになっている。一方、本システムでは各小問に対して学生が選択肢の中から1つ選ぶたびにその情報は即座にサーバに送られる。従って「送信」ボタンに相当するものは無い。また、この方式により、教師は学生の解答状況をリアルタイムに把握することが可能となった。これは既存の e-learning システムには無い機能である。



図3 小テスト画面

図4および図5は、学生の小テストの解答状況を教師用 iPad 上でモニタリングしているときの iPad 画面である。図4は全体画面、図5はその一部を拡大したものである。

教師はモニタリング用 Web ページで、当該授業の履修者リストと、小テストの問題リストを取得し、モニタリングを開始する。

図5において、IDは学生のID(※個人情報のため、ぼかしを入れてある)を、上部に並んだ1から12の数字は問題番号を表している。問題番号下の1から4の数字は4択式の中から学生が選択した解答番号を表している。ただし、0は未解答を意味する。ここで、この数字が0以外の場合、すなわち、1から4のいずれかを選択した場合、その正誤に応じて、数字の背景色が緑(正解の場合)または赤(不正解の場合)に変化するようになっており、学生の解答状況を一目で把握できるようになっている。

本システムでは、学生が小テストの各小問の1つを選択するたびにその情報は WebSocket サーバを通して、Key-Value-Store 型の高速データベース RedisDB に保存されるようになっている。ここで、学生の解答を RedisDB に保存しているのには2つの理由がある。

1つ目の理由は、学生の iPad と教師用 iPad は WebSocket を介して常に接続された状態に

ある必要があるため、小テストの最中に教師が iPad で別の Web ページを参照した場合、その間に学生 iPad から送信されたデータは教師用 iPad では受信できないためである。そこで、学生の解答を教師用 iPad に送るとの並行して RedisDB に保存するようにした。これにより、学生 iPad と教師用 iPad 間の接続が切れた場合でも、教師がモニタリング用 Web ページに戻り RedisDB から全データを再取得すればモニタリングを継続することができるようになった。図 4 の右上の赤いボタンはこの再取得用のボタンである。



図 4 モニタリング画面

ID	1	2	3	4	5	6	7	8	9	10	11	12
1	4	4	1	1	2	4	3	2	1	1	2	2
2	0	0	0	0	0	0	0	0	0	0	0	0
3	4	4	1	3	2	4	3	2	2	4	2	3
4	4	4	1	3	2	1	3	1	2	3	2	3
5	4	4	1	3	2	4	3	2	2	4	2	3
6	0	0	0	0	0	0	0	0	0	0	0	0
7	4	2	1	1	2	2	3	3	3	2	2	3
8	0	0	0	0	0	0	0	0	0	0	0	0
9	3	2	1	3	4	2	0	0	0	0	2	1
10	0	0	0	0	0	0	0	0	0	0	0	0

図 5 解答状況例

2つ目の理由は、タブレット、ネットワーク、サーバ等でのトラブル発生時の復旧のためである。小テスト実施中にこれらのトラブルが発生した場合、できるだけ速やかにかつ、トラブル発生直前の状態に復旧させることが望ましい。しかし、既存の e-learning システムでは小テストの全ての問題に解答した後に「送信」ボタンを押して解答をサーバに送るようになっている。この方式では、トラブルが発生した時点で送信がまだ済んでいない解答が全て喪失してしまう危険性がある。一方、本システムでは1つの小問に解答するたびに

サーバに自動的に送信され、RedisDB に保存されるため、学生が解答済みの全てのデータを復旧させることができる。なお、本研究では、小テストの目的は授業中に学生の理解度をリアルタイムに把握することにあるが今後、教育現場でパソコンやタブレットを使った定期試験を実施するようになった場合、本研究で開発した方式の採用が有望である。

本システムの特徴である教師用 iPad へのリアルタイムな解答状況表示機能は既存の e-learning システムでは見られない。その理由の1つは既存の Web ベースの e-learning システムでは HTTP プロトコルを使用しているからである。HTTP によるクライアント/サーバ間の通信では、最初のリクエストは必ずクライアント側から行わなければならない、サーバからクライアントに対してプッシュ方式でデータを送信することはできない。また、基本的には1つのリクエストに対してクライアント/サーバ間で接続が張られ、サーバから1つのレスポンスを返した時点で接続は切断されるようになっている。そのため、クライアント/サーバ間の通信量が多い場合、遅延やサーバへの負荷が問題となる。

そこで、本研究ではリアルタイムモニタリング用サーバとの通信に HTTP ではなく、WebSocket プロトコルを用いた。同プロトコルはクライアント/サーバ間で接続を一度張れば、その後はクライアントからでもサーバからでも自由に送信可能なプロトコルであり、接続も一度張るだけで済むのでサーバに対する負荷も小さいからである。

本システムでは教師用 iPad 上に表示するモニタリングページの生成に、近年注目されている SPA (Single Page Application) 方式を採用した。従来の Web アプリケーションではクライアントからのリクエストに応じてサーバから Web ページ全体を返し、それをブラウザ上に表示する方法をとる。この方式では、画面の一部を更新するだけでもページ全体を受信し、さらに画面全体を再生成するため、ネットワークおよびブラウザに対する負荷が大きい。一方、SPA 方式では、サーバから受信するのは更新に必要なデータのみであり、ブラウザ上ではそのデータを用いて画面の一部を再生成する。これにより上記の負荷を低減することができる。

以上の方法でシステムを実装した後、著者の担当する授業で本システムを使用し、その有効性を検証した。

授業中に小テストを実施したところ、図 5 に示されるように、学生がどの部分を理解していないかを授業中にリアルタイムに把握することができた。例えば、図 5 の表示から、問題 10 に該当する部分を学生がよく理解できていないことがわかるので、この部分を再度説明することができた。

なお、多人数で同時に小テストを行った場合の負荷テストとして、学生 iPad 上でランダ

ムな解答を高速かつ自動的に生成し、その解答状況を教師用 iPad に表示した場合に、リアルタイムに追従できるかどうかのテストを行った。そして、多人数が同時に解答した場合でも、通常の人間の解答スピードであれば問題なく追従できるものと思われる結果が得られた。

### (3) テスト駆動型授業の提案

従来の授業では、テスト(試験)は主に学生の成績評価のために行われてきた。そのため、テストの結果が授業に反映されるのは、次回の授業日であったり、次年度であることが一般的である。一方、本研究を通して、授業中に行うテストの結果をその場で確認し、当日の授業に反映させることの有意性が確認できた。これを踏まえ、著者は新しい授業形態として「テスト駆動型」の授業を提案する。

これは、ソフトウェア開発の手法の1つである「テストファースト」型の開発手法を教育に応用しようというものである。「テストファースト」型のソフトウェア開発では、最初にテスト項目を定義し、そのテストに合格するようにソフトウェアをリファクタリングしながら徐々にソフトウェアの品質を高めていく。そのため、ソフトウェアの開発工程の中で繰り返しテストを行う必要がある。この手法を教育に適用するには、予め教える内容を定めておき、その内容が理解できているかどうかを授業中に繰り返し行う小テストで確認しながら、徐々に学生の理解度を高めていけばよい。本研究で開発したシステムはこのような授業形態に適したものとなっている。

### (4) アドホック型ネットワークについて

本研究では当初、十分なネットワーク環境が整っていない教室でのタブレットの使用を考慮して、タブレット同士が直接通信し、アドホックネットワークを構成する方法を検討していた。その後の研究で、例えば iPad には「インターネット共有」という機能があり、これを使えば、iPad が一種の無線 LAN アクセスポイントのように働くことが分かった。よって、授業中に iPad を複数配置し、これらを介して通信すればアドホックネットワークを構成しなくても、当初の目的を達成できることがわかった。

### (5) 電子書籍との連携について

近年、これまで紙媒体で提供されてきた教科書や問題集の電子書籍化が進んでいる。電子書籍には、紙に比べ製造コストが安い、内容の修正や配布がしやすい、インタラクティブを導入しやすい、音声や動画、3D グラフィクスを埋め込むことができる、などのメリットがある。

本研究で開発したシステムも、電子教科書と連携させることにより、教育効果が高まることが期待されることから、電子教科書との連携方法について考察した。

現在、電子書籍にはいくつかのフォーマットがあるが、今後は EPUB 形式が主流になると思われる。EPUB 形式は、XHTML、CSS、JavaScript などの所謂 Web 技術をベースとしていることから、Web との親和性が高い。本研究ではオープンソースの EPUB ビュアーである Radium-js-viewer を用いて本システムとの連携をテストした。

現在、EPUB の規格では JavaScript もサポートされているが、XHTML や CSS に比べ、JavaScript のサポートはオプションとなっており、現在 Web で使用されている JavaScript の機能の全てが利用できるわけではない。総じて Radium-js-viewer は他の EPUB ビュアーに比べ JavaScript のサポートは良好であったが、本システムで採用した SPA を利用するには Radium-js-viewer のソースコードの変更が必要となる。なお、他の EPUB ビュアーの中には本システムで開発した JavaScript プログラムが動かないものもあった。この原因は分からなかったが、セキュリティを考慮してビューアーの方で JavaScript の動作を制限しているのではないかと思われる。

以上より、現状では、電子教科書と e-learning システムやリアルタイムモニタリングシステムは独立した構成とし、EPUB フォーマットが持つ XHTML のリンク機能によりこれらを連携させるのが最も簡単な方法であるが、既に Radium-js-viewer をはじめとするオープンソースの電子書籍ビューアーも存在するので、これらのソースコードを元に WebSocket や SPA に対応させる方法も有効であると考えられる。

### (6) 情報公開について

本研究で開発したソフトウェアを以下の GitHub のサイト(下記、5. の [その他])を参照)で公開した。ライセンスは BSD ライセンスである。

ただし、公開にあたって、学生のユーザ ID などの個人情報は削除した。また、実際に使用したサーバの IP アドレスも削除してある。

本システムでは Node.js 上の各種ライブラリを使用しているが、同ライブラリは公開ソフトウェアには含まれていない。ただし、これらのライブラリは、公開ソフトウェア一式中の package.json ファイルを参照することによって再インストールすることができる。

## 5. 主な発表論文等

[その他]  
ホームページ等  
[https://github.com/takajohideyuki/quiz\\_monitor](https://github.com/takajohideyuki/quiz_monitor)

## 6. 研究組織

(1) 研究代表者

高城 秀之 (HIDEYUKI, Takajo)  
香川高等専門学校・通信ネットワーク工学  
科・准教授  
研究者番号：30261118