

## 授業支援システムのリッチクライアント化と評価

## A class support system using rich client and its evaluation

柳下 慶輔<sup>\*1</sup>  
Keisuke Yagishita山口 大輔<sup>\*2</sup>  
Daisuke Yamaguti片山 富美代<sup>\*2</sup>  
Fumiyo Katayama高橋 宗雄<sup>\*2</sup>  
Muneo Takahashi<sup>\*1</sup> キャッツ株式会社  
CATS Co., Ltd<sup>\*2</sup> 桐蔭横浜大学工学部  
Faculty of Engineering,  
Toin University of Yokohama

Recently, a technology so called Rich Client is drawing attention and being introduced as a technology to enable high usability of Web application systems. Rich Client is getting to be used to improve the low usability of Web application systems developed by HTML, which is widely used for client side software. Ajax (Asynchronous JavaScript + XML) supports asynchronous transmission techniques which is a fundamental technology of Rich Client. In this paper, we improve existent Class Support System made with Web 1.0 technique (CGI, etc.) based on Rich Client and evaluate its usability. We also discuss the degree of usability, operability and satisfactions.

## 1. はじめに

現在、コンピュータはオフィスや家庭に広く普及し、それ自体使っている事を感じさせない様な時代になってきた。また、インターネットは通信設備の高速化にともない、伝達手段としての利便性の高さから世界的に広まっており、近代社会においてなくてはならないものとなっている。こうした中で近年、教育現場ではe-ラーニングが盛んに行われている。本大学においてもプログラミング演習の授業で支援ソフトが作成され活用されている(図1)。既存のプログラミング演習 Web サイトは、コンピュータ、携帯電話からでも利用できるように Web1.0 ベースの技術によって作成されている。そのため、構造がシンプルで、機能面では便利ではあったが、操作性の面で問題があった。

本研究では、Ajax(Asynchronous JavaScript + XML)を使って Web サイトを構築し、操作性の改善を試みる。そして、Web2.0 ベース技術(Ajax)を使用する事で既存の Web1.0 ベースのシステムと比べて、利便性が向上するか評価する。



図1 既存プログラミング演習 Web サイト

## 2. Web システム

## 2.1 Web2.0 とは

2005 年頃から Web2.0 というキーワードが急速に広がり始めた。Web2.0 とは、インターネット開設時を Web1.0 と見たときの次世代 Web 技術をいう。Web2.0 にする事により、情報の送り手と受け手が対等の立場でやりとりできるネット環境へと変わり、高い操作性が実現可能となった。

Web2.0 は、Web の新しい利用法を提案する概念であり、いくつかの共通要素を共有する事が特徴である。代表的なサービスとしてブログ、Wikipedia、Google Maps などが挙げられる。

## 2.2 リッチクライアントとは

リッチクライアントとは低コスト、高操作性を実現する技術の総称であり、最低限、以下の 2 つ条件を満たしていればリッチクライアントと呼ぶことができる。

1. 高い操作性を実現していること
2. クライアントへのソフトの配布が容易であること

リッチクライアントを実現する技術の一つとして Ajax がある[1]。リッチクライアントの出現により Web アプリケーションの開発において、従来一般的であった、品質の良いソフトウェアをいかに安く短期間で開発するかという生産性重視の開発に対して、低コスト、高操作性に重点を置いた開発ができるようになった。

## 2.3 Ajax (Asynchronous JavaScript + XML)

Ajax は、JavaScript の組み込みクラスである、XMLHttpRequest を利用して、ページ全体の読み込み、非同期にサーバーと通信し、ダイナミック HTML を利用してページの必要な部分だけを書き換える Web アプリケーションの開発手法である。従来の Web アプリケーションでは、アクションのたびに画面遷移(ページ全体の再読み込み)が発生していた。そのため通信量が増え、読み込み中はユーザの作業が中断されてしまうという問題があり、これが HTML クライアントの操作性を大きく低下させる原因となっていた。しかし、Ajax を使用した Web アプリケーションでは

データのみを送受信して、表示を更新することで、無駄な画面遷移を排除し、飛躍的に操作性を向上させることができる。

つまり Ajax では、ユーザが意識せずにシステムの裏側で通信を行い、動的に Web ページを動作可能とすることにより、高い操作性を得る事ができる。例えば、Google Map や Google Suggest では、ページ全体で読み込むことなく、地図を拡大したり、マウスで地図を移動させたりすることができる。

### 3. 授業支援システムの構築

#### 3.1 授業支援システムとは

本大学では、Web をいろいろな形で使用し、授業の支援に役立てている。例えば、緊急時の連絡、試験の日程、問題・課題の提示、レポート提出などである。Web の最大の利点は、送る側(教員)、受ける側(学生)の双方とも、時間、場所の制約を受けずに更新、参照が可能な点である。これによって、教員・学生間で迅速且つ柔軟な情報提供が可能となる。

これからの大学は、講義室内での授業だけでなく、学外、企業といった場での演習や実習など、教育形態が多様化するとともに、幅広い教員・学生層を考慮しなければならないため、Web 授業支援のシステムの機能は、益々、高度なものになると思われる。

現在、本大学の授業で実際に使用されている、プログラミング演習 Web サイトでも、Perl + HTML1.1 + Java Script1.2 で構築されている。

#### 3.2 リッチクライアント化

OpenLaszlo4.0.2 を用いて、既存プログラミング演習 Web サイトをリッチクライアント化した(図 2)。XML ベースの LZX 言語と JavaScript で構築した追加機能を表 1 に示す。

既存プログラミング演習 Web サイトのページの操作で、新ページを見たいとき、書かれている言葉などをクリックしてサーバーにアクセスして新ページを取得するが、新プログラミング演習 Web サイトではサーバーにアクセスして新ページを取得する必要はない。ドラッグ&ドロップ・リモコンの機能を使ってページを移行することができる。

本大学のプログラミング演習の授業は、情報演習室で行われている。Ajax ホワイトボード機能は、授業を受けている学生のタイピングをサポートするために作成した。授業では、「配布資料のダウンロード」だけでは、間に合わず急にプログラムのソースを書いたり、出席学生の全員に同じことをタイピングしてもらったりすることが多々ある。以前は、このような場合、教員が打った文字を、学生は前のスクリーンを見て、自分が使用しているコンピュータにタイピングしていた。しかし部屋が広いと、前のスクリーンでは文字が小さかったり、建物の構造上見えづらい席にいたり、や光の反射によって見えなかったりした。これにより、タイピングに遅れる学生などが出て、授業への支障があった。ホワイトボード機能の追加により、教員が打っている文字を全学生が自分が使用しているコンピュータで同じ条件で閲覧できるようになった。

Ajax ホワイトボードは、学生側は閲覧のみ可能である。教員側は名前、文字の色、メッセージを入力することができる。この Ajax ホワイトボードの最大の特徴は、教員は送信ボタン、学生は更新ボタンを押さなくていいことである。通常では、メッセージを出す側はメッセージを送るたびに送信ボタンを押し、メッセージを受ける側は新メッセージを取得するたびに更新ボタンを押

していた。しかしそれでは、教員はメッセージを更新するたびに、学生に更新ボタンを押すように伝えなければならない。これでは、学生も自分のタイピングに集中できなくなってしまい意味がない。Ajax ホワイトボードでは、教員はメッセージを入力して Enter キーを押せば学生側も勝手にメッセージが更新される。学生はそれを見て、自分の作業に集中できるようになっている。



図 2 新プログラミング演習 Web サイト

表 1 追加機能

項目	内容
サーバとの通信	新ページ取得の不要
ドラッグ&ドロップ	任意にページ、window を移行
リモコン	1 ページずつ次のページに移行
ホワイトボード	教員がタイピングした文字を Web サイトでリアルタイムに閲覧
Window	Window に書かれている項目をクリックすることにより、別 Window を生成して、項目の詳しい情報を取得
受講者一覧 (新ページ)	受講者の情報(出欠、課題、教員からコメント)を Window 機能の形で表示

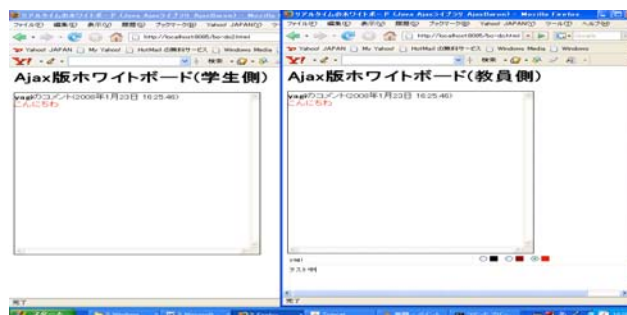


図 3 Ajax ホワイトボード

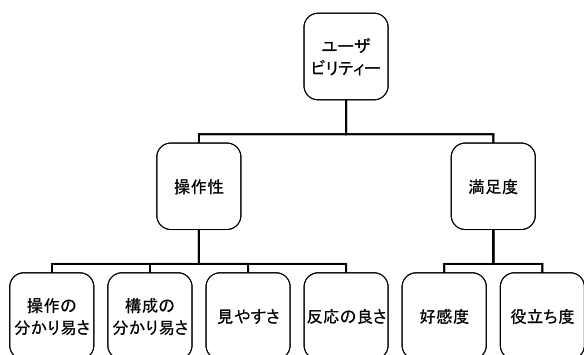
### 4. ユーザビリティ評価

#### 4.1 ユーザビリティとは

最近では、製品開発をユーザの視点で行い、新しい機能やデザインを考慮した製品開発プロセスの検討や実現など、ユーザにとっての製品の「使いやすさ」が追求されている。このような

ユーザにとっての「使いやすさ」を表すものが、ユーザビリティであり、日本語では「使いやすさ」と訳される。しかしながら、「使いやすさ」を具体的に定義し、評価することは非常に難しい。つまり、製品の良し悪しや使い勝手というものはその製品を使うユーザの利用状況や使用頻度、癖、好みなどさまざまな要素に影響されるからである。

そこで、本研究では、国際規格 ISO 9241 で定義されている「ユーザが目標を達成する際に、正確さと安全性に費やした資源」の効率性と「製品を使用する際の、不快感のなさ、及び肯定的な態度」の満足度の 2 つに焦点をあて、授業支援システムを使用したときのユーザビリティについて評価する。ここでは、仲川ら[2]が開発したウェブサイトユーザビリティアンケート評価尺度を参考に、効率性は、操作性として「操作のわかりやすさ」、「構成のわかりやすさ」、「見やすさ」、「反応のよさ」を、満足度は「好感度」、「役立ち度」を測定する(図 4)。本研究では、客観的評価として、唾液によるストレス値の評価を取り入れ、広義にユーザビリティを捉え、評価することとした。



#### 4.2 ユーザビリティ評価の対象と手順及び内容

今回の評価では、PC 操作能力が同程度になるように、電子情報工学科 3、4 年生 8 名に協力を依頼し、実施した。使用した課題は、既存プログラミング演習 Web サイト(以下、拡張前)と今回開発した新プログラミング演習 Web サイト(以下、拡張後)の 2 つを設定した。評価は、(1)唾液を用いた生理的なストレス評価、(2)Web サイト全体の操作性・満足度の向上に関する使いやすさの主観的評価、(3)表 1 の追加機能の操作性や理解度の向上に関する評価の3点である。

2 つの手法を比較する場合、学習効果や慣れといった要素が唾液によるストレス指標や主観的評価に及ぼす可能性を考慮し、被験者を、拡張前を先に行う Pattern1 と拡張後を先に行う Pattern2 の 4 名ずつに無作為に分けた。また、被験者には、Pattern1 か Pattern2 であるのか、使用している Web が拡張前か拡張後のものであるかは実施前には知らせなかった。分析のためのデータは、それぞれの課題前後の唾液採取によるストレス値、各課題終了後の使いやすさのアンケート結果、全課題終了後に使用した Web サイトのどちらが拡張後であるのかを伝えた上で追加機能(表 1)に関するアンケートの結果である。

分析では、Web サイト使用前後の唾液によるストレス値の差をストレス得点とし、拡張前と拡張後のストレス得点と使いやすさの主観的評価得点を比較した。また、追加機能については回答の分布をみた。

#### (1) 唾液によるストレス評価

測定値には、COCORO METER (NIPRO 社製)を使用した。この測定器は唾液中に含まれるアミラーゼ値を測定する。情報として身体に受けたストレスは交感神経系の視床下部を介して交感神経系の興奮を促し、この興奮が、体外のストレスに対する体内の自己防衛反応としてアミラーゼ活性を高め、アミラーゼを増加させる。すなわち、数値が高いほどストレスが高いと評価できる。

#### (2) 使いやすさの主観的評価

ウェブサイトユーザビリティアンケート評価尺度[2]の 7 つの評価軸のうち、信頼性を除いた 6 因子全 18 項目を、用いた。評価は、「全くそう思わない」から「非常にそう思う」の 5 段階とし、1-5 で得点化して、各因子の合計得点を算出した。

#### (3) 追加機能の評価

表 1 に示した、5 つの追加機能に関して、1~2 項目、計 8 項目を設定し、(2)と同様に、5 段階にて得点化した。

### 5. 結果及び考察

#### (1) ストレス評価(表 2)

唾液によるストレス得点は、拡張前 ( $M=3.5$ ,  $SD=11.59$ )、拡張後 ( $M=4.5$ ,  $SD=28.71$ ) であり、標準偏差は拡張後の方が大きい。これは各被験者の得点の分布にバラつきがある、つまり拡張後に個人のストレス度の差が拡大していることを示している。また、拡張前と拡張後平均値の差を  $t$  検定によって比較した結果、両者に有意差は見られなかった ( $t(7)=0.09$ ,  $p=.925$ )。

個々のストレス度にバラつきがみられるため、個人的要素等による影響を加味して吟味すべきであるが、全体的にみると、客観的評価としてのストレス反応に関しては、どちらの Web サイトを使用しても差はないと判断した。

#### (2) 使いやすさの主観的評価(表 2)

ウェブサイトユーザビリティアンケート評価尺度の 6 因子の得点を拡張前後で比較した。その結果、「好感度」の因子では拡張前 ( $M=5.88$ ,  $SD=2.37$ ) に比べて拡張後 ( $M=10.75$ ,  $SD=1.30$ ) の得点が高く、有意な差が見られた ( $t(7)=5.85$ ,  $p<.001$ )。他の因子に関しては、有意差は見られなかったが、「反応の良さ」を除き、平均値が拡張前に比べて高かった。

「反応の良さ」の因子で平均値が低下した理由としては、拡張後の Web サイトはアニメーション化にしているため、ユーザの操作に対して、ページ移行・表示が遅かったことが原因であると思われる。しかしながら、全体的に拡張した内容はユーザにとって良い結果をもたらしたと判断した。

#### (3) 追加機能の評価(図 5)

追加機能に関しては、「ドラッグ&ドロップ」を除き、多くの被験者が「そう思う」、「大変そう思う」と回答している。つまり、「サーバーとの通信」、「リモコン」、「ホワイトボード」、「Window」の機能に関しては、改良により便利、操作しやすいなどのプラスの効果があったと思われる。しかし、「ドラッグ&ドロップ」のみ、半数以上が「そう思わない」、「全くそう思わない」と回答している。総合してみると、この結果は、使いやすさの主観的評価の結果

と同様な傾向を示しており「ドラッグ&ドロップ」の操作は、「反応の良さ」に関連してユーザに負担をかけることが、わかった。

## 6. まとめと今後の課題

ユーザビリティ評価の結果、プログラミング演習 Web サイトのリッチクライアント化は、サイトの反応性に若干の課題を残すが、より使いやすい、良い印象をもってもらえるものとなった。しかしながら、新プログラミング演習 Web サイトは、反応性、操作性に改良の余地があることがわかった。これはアニメーション機能との関連性があり、これを修正することで、満足度の低下が懸念されるため、更なる検討が必要である。

今回は評価方法に、アンケートによる主観的評価に加え、客観的指標として唾液を用いてストレス評価を試みた。しかしながら、ストレス値は被験者による数値のバラツキが多く、このデータのみで判断するのは難しいことが明らかになった。この数値のバラつきは個人差と測定前の状況による影響を受けている可能性も考えられるため、データ採取前の一定時間の被験者の活動内容の統制も必要であると思われた。また、追加機能に関する評価項目については、設問の適切性や評価方法の検討が残された。今回は定量的に評価をしたが、行動観察やインタビュー法を用いることにより改善点がいずれもより明確になる可能性がある。

以上のように、今後は、評価方法として被験者数、評価対象・方法の吟味をし、改善点を明確にすることで、ユーザにとって使いやすい授業支援システムを構築していきたい。

## 参考文献

- [1]Tim O'Reilly, What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, 2005.
- [2]仲川薫, 須田亨, 善方日出夫, 松本啓太, ウェブサイトユーザビリティアンケート評価手法の開発, 第10回ヒューマンインターフェース学会紀要, 2001.

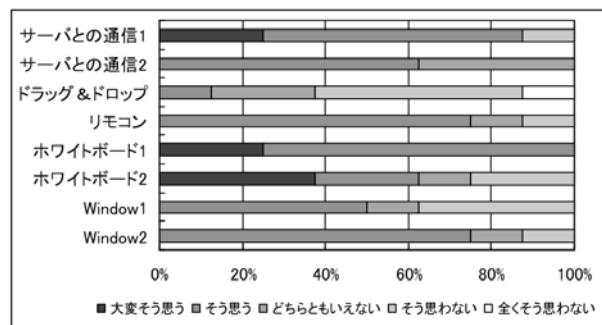


図5 追加機能に関する評価

表2 ストレス評価およびユーザビリティの6因子の拡張前後の得点差

評価項目	拡張前 $M(SD)$ / 拡張後 $M(SD)$	t 値	自由度	有意確率
唾液	-3.5(11.59) / -4.5(28.71)	0.09	7	0.925 ns
操作の分かり易さ	9.6(2.06) / 10.0(1.94)	0.81	7	0.442 ns
構成の分かり易さ	9.3(2.11) / 10.8(1.85)	1.11	7	0.303 ns
見易さ	10.6(2.23) / 11.0(1.50)	0.33	7	0.750 ns
反応の良さ	10.8(1.39) / 9.6(1.22)	1.47	7	0.186 ns
好感度	5.9(2.37) / 10.8(1.30)	5.85	7	0.001 拡張前<拡張後**
役立ち度	8.9(2.37) / 9.5(2.50)	1.26	7	0.250 ns

\*\* $p < .01$