Vol.2010-GN-75 No.1
2010/3/18

ノンプログラミングによるWebアプリケーション開発支援システムの提案

水越悠太[†]　服部哲[††]　速水治夫[†,††]

近年，様々なWebアプリケーションが提供されている．個人でもWebアプリケーションを開発・公開することができるが，Webアプリケーションの開発には知識や技術が必要であり，こういった知識や技術を持たないユーザとっては大きな障壁が存在していると言える．本論文では，この障壁を環境構築・設計・言語の３つに分類し，それぞれの障壁を軽減するためにノンプログラミングによるWebアプリケーションの開発支援を中心としたシステムを提案，試作した．

An environment for supporting non-programming Web application development

Yuta Mizukoshi[†]　Akira Hattori[††]　Haruo Hayami[†,††]

In recent years, various Web applications have been provided. Web application development is possible with not only company but also individual. But Web application development is requiring a lot of know-how, therefore, a lot of users who do not have this know-how, have a barrier for Web application development. In this paper, we classify wall for Web application development to three types; those are constructing environment, design, and language. For reducing barriers, we suggest an environment for supporting non-programming Web application development.

1. 研究対象の現状

1.1 Webアプリケーション

近年，Webアプリケーションが広く普及している．また，固定的な表示が多く機能面でも様々な制約のあったWebアプリケーションであるが，動的な表示が可能で機能面でもデスクトップアプリケーションと大きな差のないRIA(Rich Internet Application)も増えてきている．

Webアプリケーションが身近なものになったことで，一般のユーザにとってもWebアプリケーションは欠かせないものとなってきている．日常の生活の中でも，Webを検索する，電車の経路を調べる，チケットを予約するなど，様々な場面でWebアプリケーションが利用されている．

1.2 Webアプリケーション開発

1.2.1 開発方法

Webアプリケーションはサーバ等の動作環境，開発環境，開発に用いる言語，フレームワークなどを目的に応じて組み合わせて開発を行う．例えば，サーバにはApacheやInternet Information Services(IIS)，開発環境にはEclipseやVisual Studio，開発に用いる言語にはJSP/Servlet，PHP，ASP，Perl，Python，Rubyといったものがある．

Webアプリケーション開発の流れは，①Webアプリケーションを動かすための環境であるサーバソフトウェアやデータベースソフトウェアのインストール，②Eclipse等の開発環境のインストール，③プログラミング，④テスト，⑤公開用サーバの用意やネットワークの設定等の公開準備のようになる．

1.2.2 開発支援ツール・技術

Webアプリケーション開発を支援する様々なツールや技術が存在している．

サーバ構築においては，必要なサーバ・データベース・開発言語をパッケージ化し，一括してインストールできるXAMPPやVertrigo Servなどが存在する．これらを利用することによって，複数のソフトウェアを個別にインストールする必要が無くなり，動作環境の構築が容易になる．

Webアプリケーションの開発においては，EclipseやAptanaといった統合開発環境により，コーディングやテストに対する支援が充実してきたことも大きな助けとなっている．また，様々な機能を持ったフレームワークやライブラリが登場したこともWebアプリケーション開発が容易になってきた要因の一つである．フレームワークは，Rubyにおける Ruby on Rails，PHPにおける Zend Framework，CakePHP，Symfony，Java

† 神奈川工科大学大学院工学研究科
Graduate School of Engineering, Kanagawa Institute of Technology
†† 神奈川工科大学情報学部情報メディア学科
Information Media Faculty, Kanagawa Institute of Technology

Servlet における Struts，Java Server Faces(JSF)，Spring Framework など様々なものが開発されている．Ajax を実現するための JavaScript のライブラリも prototype.js，jQuery，Dojo などが開発されている．

また，それまで企業等でしか開発できなかったような機能や多くのデータを扱うような Web アプリケーションを幅広い層が開発できるようになった背景に WebAPI の存在がある．WebAPI は通常の API(Application Programming Interface)と同様に，アプリケーション開発を容易にするための機能をまとめたものである．WebAPI では，インターネットを経由して用意された機能を利用することができる．Google，Amazon，Yahoo!をはじめとした様々な企業が WebAPI を公開し始めたことで，高度な機能を持つ Web アプリケーションを開発することがこれまでと比べ容易になってきた．

2. 問題点

2.1 Web アプリケーション開発における問題点

Web アプリケーションを利用するなかで「こういう機能があったら便利だ」というような要望を持ったり，デスクトップアプリケーションを利用するなかで「これが Web 上で使えたら良いな」というような要望を持ったりすることがある．また，企業や研究室などにおける日常的な活動の中で，「こういう Web アプリケーションがあったら良いな」というようなアイデアを思いつくこともある．そう思ったときに，プログラミングや Web アプリケーション開発の経験があるユーザであれば実際に Web アプリケーションを開発することもあるが，そうでないユーザは実際に開発することは稀であると思われる．

プログラミングや Web アプリケーション開発の経験が無いユーザが Web アプリケーションの開発を行わないのは，良いアイデアを思い付いても敷居の高さから Web アプリケーション開発をあきらめてしまっているためであると考えられる．一般的な方法で Web アプリケーションを開発するためには，多くの複雑な手順が必要であり，それぞれの手順で様々な技術と知識が求められることが，経験や知識の無いユーザが手を出しづらくなっている要因であると考えられる．本研究では，Web アプリケーション開発においてユーザの障壁となる要因を「環境構築」「設計」「言語」の 3 つに分けて考える．

2.1.1 環境構築

Web アプリケーションを開発するためには，まず開発環境を構築する必要がある．一般的には，開発用のコンピュータを用意し，そこにサーバソフトウェアやデータベースソフトウェアをインストールしてテストを行うための環境を構築する．また，必要に応じて Eclipse や Visual Studio といった統合開発環境やエディタを用意する．Web アプリケーションの開発を行った後はそのアプリケーションを運用することになるが，その際には運用環境を構築する必要がある．開発に使用したコンピュータや，LAN 上で運用する場合は開発環境でそのまま運用することも可能であるが，Web 経由での利用や Web 上で公開する場合はネットワークの設定なども必要になる．

こうした手順は複雑であり，Web アプリケーション開発経験の無いユーザには難しい．特に Web アプリケーションについての知識の無いユーザの場合は Web アプリケーションの動作原理の理解から始めなければならないため，その負担は大きくなる．

ただし，様々なソフトウェアやホスティングサービスなどにより，こうした環境構築の負担を軽減することが可能である．実行環境の構築では，サーバやデータベースを一括してインストールすることが出来るソフトウェアや，実行環境を併せ持った統合開発環境などを利用することができる．また，運用環境についてはホスティングサービスなどを利用することでネットワーク設定のような負担を軽減することができる．しかし，このような方法で負担を軽減することはできるが，ソフトウェアのインストールや設定，ホスティングサービスの選定などにある程度の知識は必要であり，全てのユーザが簡単に行えるというようなものではない．

2.1.2 設計

アプリケーションの設計はプログラミングの経験のあるユーザでも難しい．一般ユーザはアプリケーションがどのように動いているかは想像が難しく，アプリケーションを開発しようとしてもそう簡単にできるものではない．Web アプリケーションではサーバサイド・クライアントサイドといった概念の理解やデータベースとの連携なども必要であるため，特に大きな障壁となる．

2.1.3 言語

ユーザにとって最も難しく，拒絶反応を起こすのがプログラミングであると思われる．文法や機能等の学習はもちろん，変数などの概念の理解も一般ユーザにとっては大きな障壁である．特に Web アプリケーションは通常，HTML/JavaScript/PHP/SQL というように複数のプログラミング言語で開発されており，それらを学習することは初学者にとって大きな障壁となる．

また，言語の学習はプログラミングに初めて触れる多くのユーザにとっては「とりあえずやってみる」というような意識で取り組むものではなく，書籍や Web ページなどを見ながら腰を据えて取り組むような問題であると思われることも，障壁を大きくしている要因の一つであると考えられる．

さらに，タイピングが得意でないようなユーザの場合，多くの文字を入力しなければならないコーディングも負担となり得る．

2.2 問題点解決の利点

これらの問題点を解決することができれば，多くのユーザが Web アプリケーション開発を行えるようになる．ユーザが自らの要望に沿った Web アプリケーションを開発することで，様々な Web アプリケーションが提供されるようになる．様々な Web ア

プリケーションが提供されるということは二つの点において有益であると考えられる．

一つは，単純にユーザがより自分の目的に合った Web アプリケーションを選択することができるようになるということである．

もう一つは，Web アプリケーションを利用したユーザが自分の要望などを盛り込んで新たな Web アプリケーションを開発することで，Web アプリケーションの選択の幅がさらに広がるということである．このようなことは，YouTube やニコニコ動画，Pixiv といった UGC(User Generated Contents)サービスにおいて実際に起こっている．YouTube やニコニコ動画は動画を，Pixiv はイラストをコンテンツとして扱う UGC サービスである．こうしたサービスを利用するユーザは，他のユーザが作成した多数のコンテンツの中から自分の好みに合ったコンテンツを選択して視聴・閲覧することができる．更に，視聴・閲覧したコンテンツの影響を受けたユーザが新たなコンテンツを作成し，サービスに投稿することがある．こうして投稿されたコンテンツを更に他のユーザが閲覧・視聴することで，更に多くのコンテンツが作成されていくというようなことが起こっている．

3. 解決方針

3.1 対象ユーザ

本研究で対は Web アプリケーション開発のスキルを持たないユーザが思いついたアイデアを実現するというようなものを対象としており，企業等における業務での Web アプリケーション開発のようなものは対象としない．

本研究において主な対象とするユーザは Web アプリケーション開発スキルを持たない，あるいは Web アプリケーション開発スキルの低いユーザである．スキルの低いユーザはスキルの高いユーザに比べ，Web アプリケーション開発における障壁の影響を大きく受けるため，障壁を取り除くことによる恩恵も受けやすいことから本研究の主な対象とした．

また，Web アプリケーション開発の経験やスキルのあるユーザでも，Web アプリケーション開発に要する時間を現在の手法に比べ短くすることができれば，プロトタイピングなどに利用することもできると考えられる．

3.2 解決方針

環境構築・言語・設計の３つの障壁について，次のようなアプローチで解決を図る．

3.2.1 環境構築

開発環境と実行・公開環境を併せ持ったシステムを Web アプリケーションとして提供することで，ユーザの環境構築にかかる負担を軽減する．通常はサーバの構築やソフトウェアのインストールといった手順が必要になるが，ブラウザからシステムにアクセスするだけで環境構築が整うため，ユーザの環境構築にかかる負担はほとんど考慮する必要がなくなる．

3.2.2 設計

ユーザが開発したアプリケーションやアプリケーションの一部であるモジュールを登録し，他のユーザがそれを利用できる仕組みを用意する．これにより，ユーザがゼロからアプリケーションを開発せずに済むようにする．

ある程度の機能を持ったモジュール同士を組み合わせたり，手を加えたりすることの方が，ゼロからアプリケーションを設計することに比べて設計が簡単になると考えられる．また，クライアントサイド・サーバサイドを意識せずに Web アプリケーションを開発できるようにすることでも，設計を単純化することが可能であると考える．

3.2.3 言語

GUI ベースのノンプログラミングなアプリケーション開発を実現することで，Web アプリケーションの構造や動きを理解しやすくし，ユーザのプログラミング言語学習及びコーディングに対する負担を軽減する．また，サーバサイド・クライアントサイド・データベース等の開発を全て GUI 上でシームレスに行えるようにすることで，複数の言語を学習するといった負担も軽減される．

4. 関連研究・類似システム

4.1 Visual Meta Groupware

グループウェアを構築するためのグループウェアである Visual Meta Groupware(以下 VMG)が研究されてきた [1), 2)]．VMG では，ユーザが簡単にグループウェアを構築するために，ホワイトボードを使って情報共有ツールを構築していることに着目している．ホワイトボードに磁石やビニールテープを張り付けるように，Web 上の疑似ホワイトボードに様々な部品を配置することで情報共有を目的としたツールを構築できるようにしている．

VMG が部品の位置情報などを利用した情報共有ツールを構築できるのに対し，本研究では様々な機能を持つ Web アプリケーションを開発することができるという点が異なっている．

本研究においてノンプログラミングによる Web アプリケーション開発を実現する手法として，疑似ホワイトボードに部品を配置していくという VMG の概念を取り入れる．

4.2 開発支援ツール

開発支援ツールとして，統合開発環境や Web オーサリングツールといったものがある．これらは，アプリケーションや Web サイトの開発をさまざまな面から支援する．Web アプリケーションの開発を支援するという点では本研究と重複するが，これらの開発支援ツールではプログラミングは基本的に不可欠であるという点が大きな違いで

ある．

4.3 マッシュアップツール

Microsoft Popfly や Yahoo! Pipes[3]といった，ノンプログラミングでマッシュアップを行うことのできる Web アプリケーションが存在している．

Yahoo! Pipes では，何らかの処理の出力が別の処理の入力に接続されている状態が視覚的に表現されており，パイプを構築する際に情報の流れを理解しやすいようになっている．

これらはノンプログラミングによる Web アプリケーション開発の支援を目指す本研究と非常に近い．ただし，これらのマッシュアップツールは Web 上に存在しているコンテンツを組み合わせて新たなコンテンツを開発することを目的としているが，本研究ではそれも含めた Web アプリケーション全般を対象としているという違いがある．

5. 提案システム

5.1 システム概要

解決方針に基づき，次のようなシステムを Web アプリケーションとして提供することにより，環境構築・設計・言語の 3 つの障壁の解決を図る．

本システムの中心となるのは，ノンプログラミングで Web アプリケーションの開発を行う機能及び，開発した Web アプリケーションの実行を行う機能である．これらにより，開発環境と言語に起因する障壁を回避する．GUI ベースのノンプログラミングによる Web アプリケーション開発は，Web 上の疑似ホワイトボードに部品を配置するという VMG の概念を継承することで実現する．本研究においては疑似ホワイトボードを「ボード」，その上に配置する部品はそのまま「部品」と呼ぶ．ボードに配置した部品に対して Web アプリケーションの振舞いである「動作」を設定することで，Web アプリケーションを開発していく．この Web アプリケーションの開発・実行を行う部分をまとめて本システムでは「実行・開発部」と呼ぶ．

設計に起因する障壁を回避するためのモジュール共有機能を，本システムでは「カタログ」により実現する．カタログはユーザが開発したモジュールを登録し，他のユーザが参照することを可能にする機能である．カタログを実現するための「カタログ管理部」で提供するのは主にモジュールの検索などであり，モジュール化やボードへのモジュールの読み込みは実行・開発部との連携によって実現する．

その他，本システムを稼働させるために必要なユーザ登録・管理やボードの作成・管理等の機能を持つ部分を「管理システム部」と呼ぶ．

5.2 利用手順

提案システムでの Web アプリケーション開発は①システムへアクセスしボードを作成，②ボードに部品を配置，③部品の情報・動作を設定，④Web アプリケーションの実行・テストというような流れで行う．また，必要に応じて作成した Web アプリケーションや部品群をモジュールとしてカタログに登録する．

6. 試作システム

6.1 使用技術

提案システムの実現可能性を確認するために一部を試作した．試作したのはノンプログラミングによる Web アプリケーションの開発・実行を実現する部分である．したがって，モジュール化やモジュールの登録・読み込みなどに関わるカタログ関連の機能は実装していない．

試作システムは Apache Tomcat, Java Servlet, MySQL といった構成で開発を行った．クライアントサイドでは非同期通信や UI 部品などの実装のために，Ajax フレームワークである Dojo toolkit[4]を使用している．

6.2 試作システム画面

試作システムの画面構成を図 6.1 に示す．試作システムの画面は部品メニュー，ボードおよびボード上に配置された部品から構成される．この他，必要に応じて設定ダイアログなどが表示される．

部品メニューにはボードに追加する部品の一覧が表示される．システム利用者は，このメニューから配置したい部品を選択しボード上をクリックすることで部品を配置することができる．

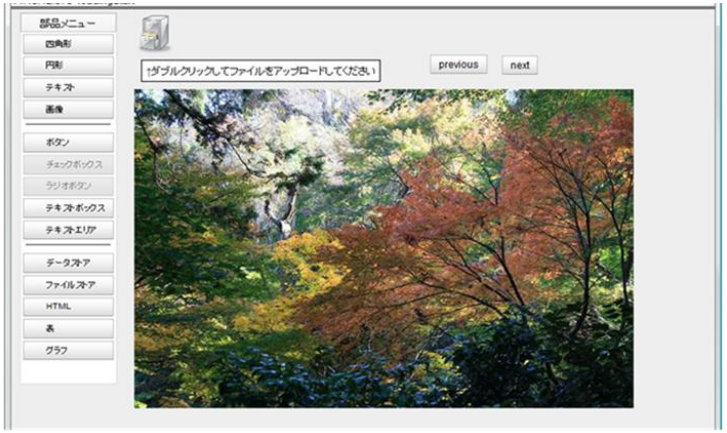

図 6.1 試作システム

6.3 部品

試作システムでは，ファイルストア部品，画像部品，ボタン部品，テキスト部品を実装した．

ファイルストア部品は様々なファイルを格納することのできる部品である．ダイアログ上でファイルのアップロード，並び替え，削除を行うことができる．

画像部品はボード上に画像を表示するための部品である．表示する画像は，画像部品に直接画像ファイルをアップロードする他，ファイルストアに格納された画像を参照することもできる．

ボタン部品はボード上にボタンを設置することができる部品であり，ラベルを設定することができる．設定可能な動作は他の部品と共通である．

テキスト部品はボード上に文字列を表示するための部品である．表示するテキストやテキストの色，背景の色を設定することが可能である．

6.4 部品設定

設定項目は「基本設定」と「動作設定」の二つに分かれており，ダイアログ上ではタブで分けられている．「基本設定」では部品の属性を設定することができ，「動作設定」では部品の動作を設定することができる．

6.4.1 基本設定

基本設定では，部品の様々な属性について設定を行うことができる．設定できる項目は部品の種類によって異なるが，ドロップダウンリストやチェックボックス，カラーパレット等の UI を使用してキーボードからの入力を極力減らしている．

6.4.2 動作設定

設定ダイアログによる動作設定の様子を図 6.2 に示す．動作の設定では，「トリガ」「動作の対象」「動作の種類」「詳細設定」の設定を行うが，初期状態ではいくつかの項目は設定できない状態になっている．「動作の対象」を設定することで「動作の種類」が設定可能になり，「動作の種類」を設定することで「詳細設定」が設定可能になる．これにより，ユーザに対してどの項目を設定すればいいのかを明確にしている．また，ドロップダウンリストに表示させる項目を実際に設定可能な項目のみにすることで，適切な項目を選択しやすいようにしている．

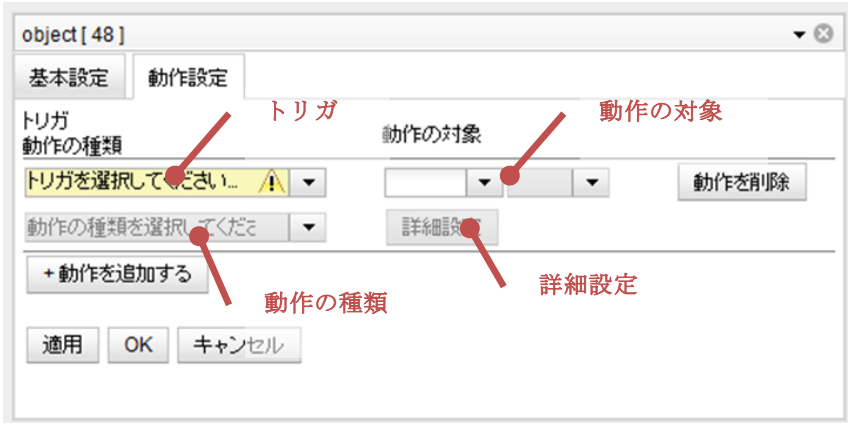

図 6.2 設定ダイアログ

6.5 アプリケーション例

試作システムにおいて実装した部品を用いて構築できる Web アプリケーションの例として画像ビューア（図 6.1）を開発した．画像ビューアはファイルストア部品，画像部品およびボタン部品で構成される．画像部品の参照先にファイルストア部品に格納されたファイルを指定し，ボタン部品がクリックされ時にファイル番号をずらしている．

7. 評価実験

提案システムの実現可能性を確認するため，試作システム使用して評価実験を行った．本実験を通して，試作システムによって環境構築および言語に起因する障壁を軽減できたかを評価する．また，今後の開発に活かすため試作システムによる Web アプリケーション開発についての評価も得る．

7.1 実験方法

実験は実験 1，実験 2 及びアンケートで構成される．実験 1・2 は共に課題に沿って簡単な Web アプリケーションの開発を行ってもらうものである．開発してもらう Web アプリケーションは次のような仕様の画像ビューアである．

- 画像をアップロードできること
- アップロードされた画像を表示できること
- マウス操作によるスライドショー機能を持つこと

この Web アプリケーションを，実験 1 では試作システムを用いて，実験 2 ではプロ

グラミングにより開発してもらう．実験 1 によって本システム自体の評価を行い，実験 1 と実験 2 の比較によって通常の開発方法と比較した際の本システムの評価を行う．評価は開発に要した時間，アンケート，実験時の観察を基に行う．

Web アプリケーションの開発を行うため，実験に要する時間が長くなってしまう．従って，実験 2 は一部の実験協力者に対してのみ行った．

7.1.1 実験 1

実験 1 では試作システムを使用して Web アプリケーションの開発を行う．ボードに接続した状態で作業を開始する．これは試作システムでユーザ登録やボード作成といった機能を実装していないためであり，ユーザ登録やボード作成が終了した状態を想定している．

Web アプリケーションを開発するために必要な情報をまとめた手引書を用意し，実験協力者には手引書を参照しながら Web アプリケーションの開発を行ってもらう．手引書は次のような構成になっている．

- 試作システムの概要
- Web アプリケーション開発手順
- Web アプリケーション仕様
- Web アプリケーション開発手引き
- 部品情報

試作システムの概要では，試作システムを利用するにあたって理解する必要がある画面構成や部品の概念についての簡単な説明を行い，Web アプリケーション開発手順で試作システムを利用して Web アプリケーションを開発する流れを説明している．Web アプリケーション仕様では実験において開発する Web アプリケーションの機能，完成イメージ，部品構成などを説明し，Web アプリケーション開発手引きで Web アプリケーション仕様において提示した Web アプリケーションを開発する具体的な手順の説明を行っている．また，部品情報では使用する部品の概要，設定可能な項目，使用方法などの説明を行っている．

7.1.2 実験 2

実験 2 ではプログラミングにより Web アプリケーションの開発を行う．開発に用いた環境は次のとおりである．なお，開発環境は構築済みであり，データベースも事前に定義されている．

表 7.1　実験 2 実施環境

開発言語	Java Servlet, SQL, HTML, JavaScript
開発環境	Eclipse(+ Web Tools Platform)

実験協力者に実際に行ってもらうのは次のようなことである．

1. プロジェクト作成
2. プロジェクトの設定（パッケージの追加，ソースファイルの作成）
3. コーディング
4. サーバへの配備
5. デバッグ

実験 2 でも実験 1 と同じように手引書を用意し，実験協力者に手引書を参考にしながら開発を行ってもらった．手引書は次のような構成になっている．

- 概要
- Web アプリケーション開発手順
- Web アプリケーション仕様
- Web アプリケーション開発手引き
- API

概要では Web アプリケーションの概念および実験 2 において使用する技術の概要を，Web アプリケーション開発手順では Eclipse を利用した Web アプリケーション開発の流れを説明している．ここまでで，Web アプリケーション開発に関する基礎的な知識を理解してもらう．次に，Web アプリケーション仕様で，開発してもらう Web アプリケーションの機能やシステム構成，完成イメージ，データベース構成の説明を行っている．Web アプリケーション仕様で説明した Web アプリケーションを実際に開発するための具体的な手順と開発するために必要な Servlet/SQL/HTML/JavaScript のコードを，Web アプリケーション開発手順と API でそれぞれ説明している．

実験 1 の手引書と実験 2 の手引書は，ほぼ同じ構成になるように作成してある．実験 2 の手引書は Web アプリケーション開発に必要な情報を Web 上から探し，それを参照しながら開発を行うことを想定して作成した．ただし，手引書に記載されているソースコードはほぼそのまま入力すれば目的の Web アプリケーションを開発することが可能なものであるため，実際に Web で情報を探しながら開発する場合と比べるとやや簡単であると言える．

8. 評価・考察

8.1 実験協力者

4 名に実験に協力してもらい，そのうち実験協力者 a 及び実験協力者 b の 2 名については実験 2 も実施した．実験協力者の内訳を表 8.1 に示す．実験協力者 a は情報技術に関する教育を受けておらずスキルレベルが低いため，実験協力者 4 名のうちでもっとも本研究の対象とするユーザ像に近い．

表 8.1　実験協力者

	職種	スキルレベル	実験 1	実験 2
a	幼稚園教諭	低	○	○
b	情報系学生	中	○	○
c	情報系院生	高	○	—
d	情報系学生	高	○	—

8.2　実験結果

8.2.1 開発環境の構築

実験 2 ではサーバソフトウェアや開発環境のインストールは完了した状態から Web アプリケーションの開発を行ったが，コーディングを開始するまでにプロジェクト及びファイルの作成といった手順が必要である．本研究ではこの部分も「環境構築」の一部であると考える．この作業に，実験協力者 a は 17 分，実験協力者 b は 12 分ほどの時間を要していた．通常の Web アプリケーション開発ではサーバソフトウェアや開発環境のインストールなども必要になるため，実験 2 よりも環境構築にかかる時間は長くなる．

一方，試作システムを用いた実験 1 では，システムにアクセスすればすぐに開発を行うことができるので，この手順は必要ない．試作システムでは実装していないユーザ登録やボード作成が必要になるが，それらを加味しても実験 2 においてコーディング開始までに要したほどの時間はかからないと考えられる．

このようなことから，環境構築に関係する障壁を取り除くことができたと考えられる．

8.2.2 Web アプリケーション開発を容易にできたか

実験 1 では，全ての実験協力者が課題の Web アプリケーションを開発することができた．特に実験協力者 a はプログラミング等の経験が全くない初心者であったが，手引書を参考にすることで開発を行うことができていた．それぞれの実験協力者の Web アプリケーション開発に要した時間を図 8.1 に示す．なお，実験協力者 a は実験 2 においては途中でリタイアしており，プロジェクトの作成から HTML ファイルを編集した時点までの時間である．実験の観察から，実験協力者 a の開発速度は実験協力者 b に比べが遅かった．仮に実験 2 を最後まで行った場合，その所要時間は最低でも実験協力者 b の所要時間である 2 時間 29 分以上，事によるとその倍の 5 時間あるいはそれ以上かかる可能性もあった．

実験協力者 b は Java Servlet による開発経験は無いものの，学部の演習において Web アプリケーションの開発をした経験がある．このような経験者の実験 2 の結果に比べ，実験協力者 a の実験 1 における所要時間が短かった．また，実験 2 の準備のために著者が課題の Web アプリケーションを開発するのに要した時間がおよそ 90 分であった．この時間には設計や開発に必要な技術の調査なども含まれている．著者は試作システムの開発に実験 2 と同じような技術構成を用いており，ある程度のスキルレベルにあると思われる．本研究におけるスキルレベルの分類では比較的高いレベルのユーザである筆者と比較しても実験協力者 a の実験 1 の所要時間は短かった．

このようなことから，試作システムを使用することで，スキルの無いユーザでも比較的容易に Web アプリケーションを開発することができたと言える．

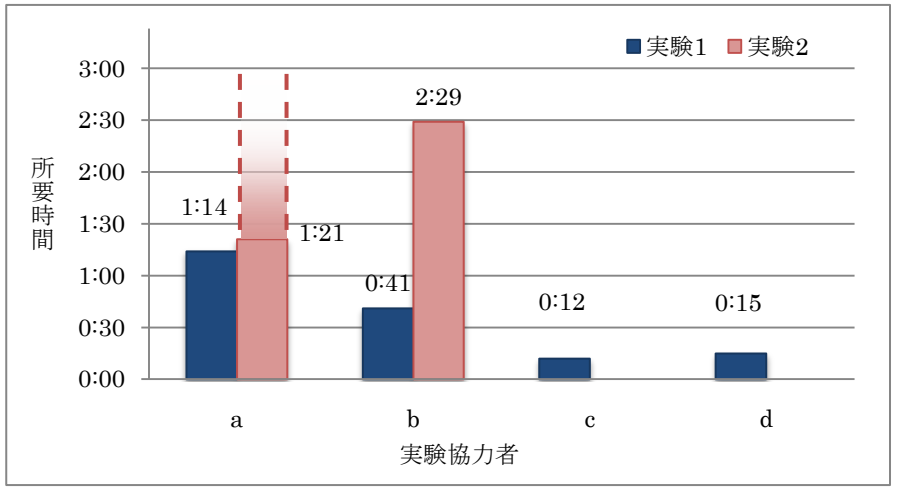

図 8.1　実験所要時間

8.2.3 試作システムの評価

本システムについて次の項目についてアンケートにより調査した．

- 部品の役割は理解できたか
  理解しやすかった／理解できた／理解できなかった
- 部品の設定は簡単だったか
  簡単／普通／難しい
- 動作の概念は理解できたか
  理解しやすかった／理解できた／理解できなかった
- 動作の設定は簡単だったか
  簡単／普通／難しい
- 部品と部品の関係は理解できたか
  理解しやすかった／理解できた／理解できなかった

アンケートの結果を図 8.2 および図 8.3 に示す．理解の度合いをたずねた質問（図

8.2）では，全てにおいて「理解できた」あるいは「理解しやすかった」との回答を得た．簡単さをたずねた質問（図 8.3）では，動作の設定が難しいという声があった．動作の設定が難しいと思った理由は,「動作の対象などの選択が分かりにくかった」というものだった.試作システムでは部品を選択する際に部品 ID を表示していたため，どの ID がどの部品を指すのか理解しづらかったようである．実験を観察するなかでも，多くの実験協力者が動作対象の設定でつまずいていた．より分かりやすくするために，部品に名前を付けられるようにし，動作の対象などを部品の名前によって選択できるようにするというような改善策が考えられる．

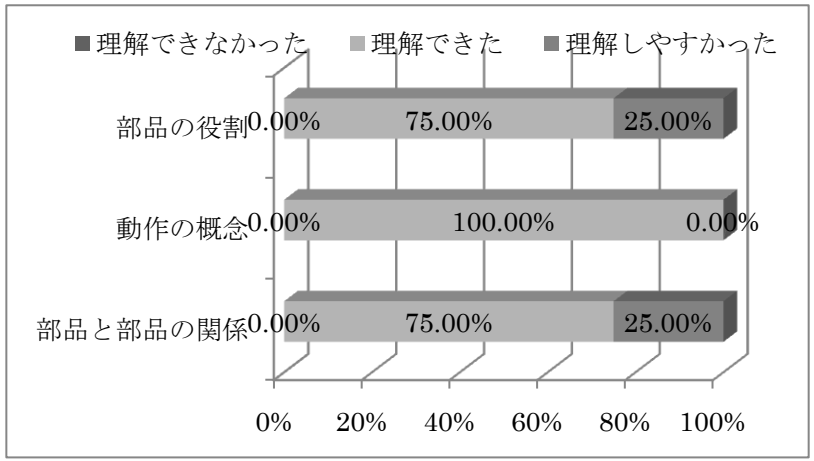

図 8.2　試作システムの理解に関するアンケート結果

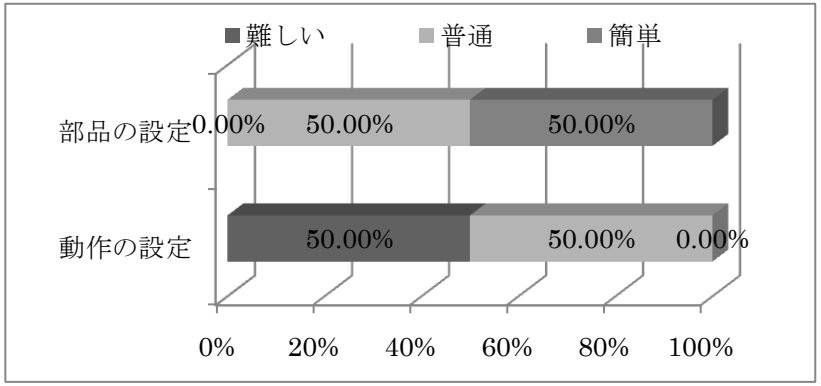

図 8.3　試作システムでのアプリケーション開発関するアンケート結果

9. むすび

本研究では，Web アプリケーション開発において知識や技術の無いユーザが Web アプリケーション開発を行おうとする際に障壁となる要因を環境構築・言語・設計の３つに分け，それらを解決するためのシステムを提案した．また，提案したシステムの一部を実際に開発し，障壁を解決できていることを確認するため実験を行った．

実験の結果，技術や知識のレベルに関係なく，本研究で試作したシステムを使用することで，通常のプログラミングによる開発に比べて開発に要する時間が短くなったことが確認できた．また，本研究で主な対象とするユーザである知識・技術レベルの低いユーザでは，プログラミングによる開発は途中で棄権したが，試作システムを使用した場合は開発を完了できたことからも，試作システムにより言語に起因する障壁を解決できたと考えられる．

また，試作システムを利用した場合は，実行・開発に必要なソフトウェアのインストールやプロジェクトの作成,ファイルの作成といった手順を必要としないことから，環境構築に起因する障壁も解決できたと考えられる．

設計に起因する障壁については，カタログ機能を試作システムでは実装していないため，本研究では解決策として提案するにとどまった．

今後は，システムとして運用するための機能やカタログ機能，部品を実装し，本格的に運用を行い，設計に起因する障壁の解決を図るとともにシステムとしての評価を行っていきたい．また，システム開発の経験が無いユーザが理解しやすいような部品や部品同士の関連の表現方法を検討するほか，より分かりやすい設定方法なども検討していく．

参考文献

1)　世古将洋，グループウェアをノンプログラミングに構築するシステムの提案，平成 13 年度神奈川工科大学修士論文（2002）
2)　松本義隆，自動処理可能なビジュアルメタグループウェア，平成 19 年度神奈川工科大学修士論文（2008）
3)　Yahoo! Pipes，http://pipes.yahoo.com/
4)　Dojo toolkit，http://www.dojotoolkit.org/