

WordPress を用いた大学研究所向けモバイル対応 Web サイト開発

常盤 祐司[†]

モバイル端末の普及に対応するために Contents Management System のひとつである WordPress を用いて大学研究所向け Web サイトを開発し 2013 年 4 月に公開した。本稿では既存 Web サイトからの移行, WordPress functions を活用したカスタマイズ, プラグインによる機能追加, 国際化など, システム構築にて直面した様々な課題への対応を通じて得られた大学情報基盤システムとしての WordPress の可能性と課題について報告する。

Development of a Mobile Compatible Web Site for University Research Institutes using WordPress

YUJI TOKIWA[†]

Based on WordPress, a mobile compatible Web site for our research institute was developed and released on April, 2013. This paper describes on the feasibility of WordPress as IT infrastructure of universities from the viewpoint to resolve the issues such as a migration of existing Web site for PC, a capability of customization by WordPress functions and plugins, and i18n.

1. はじめに

大学生のスマートフォン保有率はここ数年急激に増加している。マイナビが 2014 年卒のマイナビ会員の大学生に対して行った調査[1]によると学生のスマートフォン保有率は 83.9%となっている。これは、前年度に行った 2013 年卒の大学生に対する同様の調査[2]による 59.3%という結果から 24.6 ポイントも増加している。こうした状況に対応するため大学が提供する IT 関連サービスについては PC だけでなくモバイルに対応する必要性が年々増している。

このような環境下で著者が所属する研究センターでは 2012 年度に Web サイトのリニューアルが計画され、言うまでもなくモバイル対応は要件のひとつとなった。研究センター向けの Web サイト開発については先に Drupal を用いた開発について報告したが[3], その開発においてモバイル対応は要件になく、結果としてモバイル対応の Web サイト開発は本研究センターにとって初めての試みとなった。ただし、大学向けモバイル対応のフレームワークの先行事例として、MIT の iMobileU を起源とする KurogoTM[4]あるいは Jasig コミュニティの uMobile プロジェクト[5]などがあった。また、Contents Management System (以下, CMS) の WordPress がバージョン 3 から Responsive Web のテクノロジーを取り入れ、スクリーンサイズが小さいスマートフォンにも対応できるようになりつつあった。

本研究センターでは IT を活用した教育および研究の推進を事業目的としており、実験的な試みをいち早く実施し、そこで得られた知見を学内システム構築にフィードバックする役割も担っている。そのため、今後広く利用される可能性を持つ WordPress を使った Web サイト構築は、取

り組むべき研究・開発の一環として位置づけられた。本稿では 2012 年末から開発に着手し 2013 年 4 月に公開されたモバイル対応 Web サイト開発基盤となった WordPress の可能性と課題について報告する。

2. モバイルサイト実装と課題

モバイル対応が叫ばれ始めた 2011 年に開催された EDUCAUSE 年次カンファレンスでは意識的にモバイルに関連するセッションに参加した。そこでは MIT が開発した iMobileU フレームワークに関連したモバイルサイト構築に関する多数の報告があった。その後の調査で iMobileU のプロジェクトに携わっていたメンバが設立した Modo Labs という会社で KurogoTM という PHP ベースのモバイル対応のオープンなフレームワークを継続して開発していることがわかった。ソースおよびドキュメンテーションは公開されているので、本研究センターのサーバに KurogoTMを導入し検証を行った。その結果、KurogoTMはスマートフォン本体に導入する Apps ではなく Web ベースのサーバアプリケーションとして構築されており、スマートフォンからは付属の Web ブラウザで利用するアーキテクチャであることがわかった。また、後日導入した Jasig の uMobile も同様のアプローチを採用していた。スマートフォン用アプリケーションをクライアント側の Apps にて実装するアプローチは、iPhone 用 Apps および Android 用 Apps を開発する必要があり、Android 用ではさらに Android バージョンも意識する必要がある。そのため Apps による実装は望ましいアプローチと考えるはなかったが、KurogoTM および uMobile が採用するサーバアプリケーションで対応するアプローチは本研究センターにとって望ましい方式であった。

[†] 法政大学 情報メディア教育研究センター
Research Center for Computing and Multimedia Studies, Hosei University

^a モバイルはスマートフォンとフィーチャーフォンに大別されるが本稿ではスマートフォンを想定する。



図 1 Kurogo™によるモバイルサイト実装事例
 Figure 1 An example of mobile site by Kurogo™.

表 1 Kurogo™モジュールの機能概要
 Table 1 Functions of Kurogo™ modules.

モジュール	機能概要
緊急連絡先 Emergency	緊急連絡先などの電話番号が階層化されて複数登録できる。スマートフォン利用時には番号をクリックするとダイアル動作が行われる。
電話帳 Directory	LDAP クライアントが実装されている。氏名などのキーで検索でき、電話番号、部屋番号などが表示される。
地図 Map	緯度経度情報が複数登録できる。Google Map と連携したキャンパス地図等に利用できる。
時刻表 Transit	テキスト形式の情報が登録できる。バスの時刻表などテーブル形式の情報も登録できる。イメージの挿入はできない。
ニュース News	ニュースサイトもしくは BLOG などの URL を登録できる。サムネイル付の記事インデックスが表示される。
ビデオ Video	YouTube などの URL とデフォルトの検索文字を登録できる。YouTube を検索した結果がサムネイル付のビデオインデックスとして表示される。
リンク Links	他のサイトへのリンク情報を登録できる。
About	アイコンは異なるが、機能は Transit と同様である。
共通サイト Full Website	PC 用の Web サイトを登録できる。

Kurogo™ を用いて構築したシステムにて提供されるホームページを図 1 に示す。システム基盤としては RHEL5, XAMPP for Linux, MySQL といった環境を用いた。Kurogo™ は当時の最新版であった 1.5.1 である。Kurogo™ では各種機能を提供するモジュールは独立したアイコンで表示されており、表 1 にそれぞれのモジュールの機能を示す。なお、Kurogo™ は日本語対応がなされていないため表 1 に記載したモジュール名の日本語は筆者による翻訳である。Kurogo™ が有する機能は Admission, Athletics などを含めて 23 あるが、実際のシステム構築では本学で利用する見込

みのある 9 機能のカスタマイズにとどめた。Kurogo™ で提供される機能は図 1 および表 1 から、これまで大学の研究センターが Web サイトで一般的に公開している情報とは異なる情報の公開を主眼としていることがわかる。このため既存の研究センターの Web サイトを全て Kurogo™ 基盤に移行することは困難ではないかという課題に直面した。

3. CMS 基盤と要件定義

3.1 Why WordPress

本研究センターの次期 Web サイト基盤として Kurogo™ の適用に疑問符がついたため、次に筆者が別研究センターの Web サイトの開発に用いた Drupal の適用を検討した。その際に Drupal を選択した理由のひとつとして「Packt Publishing 社が主催する Open Source CMS Awards にて Drupal が 2007 年、2008 年に 1 位となっていた。」[3]と述べた。今回の開発にあたり改めて Drupal の市場占有率を調査したところ執筆時時点での Drupal および WordPress の市場占有率はそれぞれ 7.1%, 54.8%となっていた[6]。経済学における複雑系の研究によるとソフトウェアは収益通増となり特定のソフトウェアが独占してしまうことが多いことから、50%以上の市場占有率となっている WordPress を基盤として採用することを前提として検討を継続した。

3.2 要件定義

Drupal を用いた Web サイト構築と同様の手順で要求から要件を導いた。その結果を表 2 に示す。表 2 の要件列に記載した要件 3, 5, 7, 8, 9, 10, 11 は Drupal を用いた Web サイト構築と同様の要件であり、表 2 右列に○で示す。これらは参考文献[3]にて詳述しているので、本稿では要件 1, 2, 4, 6 として示した次の新たな要件について検討を行う。

- Responsive Web 対応である。
- 年次で生成されるページを追加するだけで一覧リストが自動的に更新される。
- 市場占有率が高いシステムである。
- リビジョン機能を有する。

上記の要件を検討した結果を順に示す。

Responsive Web 対応はモバイル対応から導かれた要件であるが、前章で述べたように先行事例ではスマートフォンの Apps ではなく、Kurogo™ および uMobile において Web ベースのサーバアプリケーションを用いていたため同様のアプローチを採用する。WordPress で Feasibility Study に着手した時には WordPress 標準のテーマは twentyeleven であり、例えばブラウザの幅を狭めてもメニューバーがメニューボタンに切り替わるようなことはなく完全な Responsive Web 対応ではなかった。そのため Constellation あるいは skeleton という第三者が開発したテーマで Responsive Web を実現していた。2012 年後半にリリースされた WordPress3.5 で標準となったテーマである twentytwelve は

表 2 要件定義と WordPress の対応

Table 2 A requirement definition and a feasibility of WordPress.

	要求	要件	WordPress	参考文献[3]と 同様の要件
			評価	
方針	モバイル対応としたい。	1 Responsive Web 対応である。	適合	
	既存 Web サイトの内容を引き継ぎかつ課題を改善したい。	2 年次で生成されるページを追加するだけで一覧リストが自動的に更新される。	PHP にて開発可能	
	オープンソースとしたい。	3 システム開発が計画的および組織的に行われるコミュニティソースである。	適合	○
	長期間に渡って利用したい。	4 市場占有率が高いシステムである。	適合	
IT	システム基盤の開発に多くのワークロードをかけたくない。	5 CMS を利用する。 日本語書籍がある。	適合	○
	更新履歴を記録したい。	6 リビジョン機能を有する。	適合	
	国際的に公開したいので多言語でコンテンツを制作したい。	7 多言語機能を有する。	適合	○
プロセス	研究センターメンバによる管理分散型で管理したい。	8 複数の管理者が設定できる。	適合	○
		9 コンテンツの作成, 更新, 削除権限はコンテンツ作成者自身およびグループに対して設定できる。	プラグインにより可能	○
組織	研究内容および体制が流動的なので容易に変更できるようにしたい。	10 プラグインまたはモジュールが利用でき拡張性に富むとともに, 余剰機能を削減できる。	適合	○
		11 コンテンツに変更を加えることなくレイアウトあるいは構造の変更ができる。	適合	○

先に述べたメニューバーのメニューボタン化をはじめとして, より Responsive Web を意識したデザインになった。WordPress にて標準的に利用されているテーマであれば WordPress 本体との整合性が高いと考え, テーマとして twentytwelve を採用することとした。

次の要件は研究センターというよりむしろ年度単位で活動をしている大学ならではの要求から導かれている。既存 Web サイトでは毎年発行される紀要である研究報告をオンラインジャーナル形式で公開している。既存 Web サイトは DreamWeaver™ を用いて HTML によりコンテンツを制作しており, iframe などを利用していない構造であったため, その年度の研究報告を発行するたびに関連するページのリンクを追加・更新する作業が発生していた。また, 研究プロジェクトを説明するページあるいは研究成果を公開するページについても年度毎に追加・更新作業が発生していた。この点が既存 Web サイト管理上の大きな課題であった。この課題については WordPress functions を用いた開発で対応することとし, その詳細は後述する。

次の市場占有率の高いシステムは継続性の観点から導かれた要件であり, 前節にてその現状を述べた。

最後のリビジョン機能は更新履歴の要求から導かれた要件である。研究センターのシステムはそれ自体を対象として実験的な試みを行ったり, 研究者自身が管理運用をするため, はじめにページを制作したユーザが一貫してそのページを管理していく保証がなく, そのためにこうした要件を加えた。WordPress では標準的にリビジョン管理機能を有している。

以上より WordPress を基盤とした Web サイト構築の目処がたったのでシステム構築に着手した。

4. システム構築

4.1 立ち上げ

WordPress は PHP で実装されデータベースとして MySQL が利用できるため CentOS6 上にミドルウェアとして XAMPP for Linux 1.7.7(PHP 5.3.8, MySQL 5.5.16)を用い, 当時最新であった WordPress 3.5 を使ってシステムを立ち上げた。WordPress は Drupal と同様, 導入時には最小限の機能しか提供されていないのでプラグインを追加した。追加したプラグインとその機能概要を表 3 に示す。プラグインはパフォーマンスの観点から最小限の利用にとどめることを推奨するガイドがあるが, 対象としている Web サイトは

表 3 追加したプラグイン一覧

Table 3 A list of added plugins.

プラグイン	機能
Breadcrumb NavXT	パンくずリストを生成する。
Collapse-O-Matic	オンラインジャーナルの抄録をアコーディオンの開閉で表示する。
Google Analytics	Google Analytics のためにデータを Google に転送する。
Google XML Sitemaps	Google Web Master Tool にて Sitemap を表示する。
Link Manager	サイドナビで別サイトへのリンクを設置する。
One Click Close Comments	ページに設置されるコメントを無効にする。
TinyMCE Advanced	HTML エディタの機能を高める。
WP Favicon	Favicon の設定をする。
WPML Multilingual CMS	複数言語で固定ページを作成する。
WPML String Translation	メニューを翻訳する。
WP Multibyte Patch	WordPress を日本語化する。

多くても日に数百程度のアクセスであり、表2で示した要件10にあるように機能の追加・削除を簡便に行うことを優先してプラグインで対応できる機能追加はすべてプラグインを適用した。

4.2 年次リスト表示用ウィジェット開発

表2要件2で示した年次ページリストの自動生成機能はプラグインが存在しないために新たに開発を行った。具体的には毎年発行する研究報告を追加した際に図2にて矢印で示すような年次リストを自動的に生成するプログラムである。本稿ではスマートフォンで表示した際にこの年次リストがページ最下部に移動できるようにウィジェットとして作成した。

WordPressのカスタマイズに関しては様々な情報が提供されているので詳細な説明は省くが、カスタマイズをする



図2 年次リスト事例

Figure 2 An example of list of yearly page.

表4 年次リスト表示用ウィジェット概要

Table 4 A widget to list yearly pages.

```

/* 過年度の研究報告などをリストするための Widget */
//
class HoseiWidgetList extends WP_Widget {
    function __construct() {
        $widget_ops = array('classname' => 'widget_PageList', 'description' => '【法政大学】カスタムキーでリストする。カスタムキーは単一であることが条件。');
        parent::__construct('PageList', '*Hosei Widget* Year List', $widget_ops);
    }
    function widget( $args, $instance ) {
        extract( $args );
        // カスタムキーを獲得する。 '_'で始まるシステム変数を取り除く。
        // 複数のカスタムキーがある場合にはそれぞれのリストを表示する。
        $customkeys = get_post_custom_keys($post->ID);
        // _ のあるシステム用のキーを取り除く。
        $keys = array_filter($customkeys, create_function('$val', 'return (substr($val,0,1)=="_") ? FALSE : $val ;'));
        // 設定されているカスタムキー全ての年次リストをそれぞれ表示する。
        foreach($keys as $key) {
            // リストされる対象は研究報告: bulletin, 研究プロジェクト: project, 公表論文: paper とする。
            if(!($key=='bulletin' | $key=='project' | $key=='paper')) continue;
            if($key=='bulletin') {$title=__('Previous Bulletins','twentytwelve-Hosei');}
            if($key=='project') {$title=__('Previous Projects','twentytwelve-Hosei');}
            if($key=='paper') {$title=__('Previous Papers','twentytwelve-Hosei');}
            $out = wp_list_pages( apply_filters('widget_pages_args', array('meta_key' => $key, 'sort_order' => 'DESC', 'sort_column' => 'post_name', 'title_li' => "", 'echo' => 0)));
            if ( !empty( $out ) ) {
                echo $before_widget;
                if ( $title ) echo $before_title . $title . $after_title;
                echo "<ul>" . $out . "</ul>";
                echo $after_widget;
            } //if( !empty( $out ) end
        } //foreach end
    }
    function update( $new_instance, $old_instance ) {
        $instance = $old_instance;
        return $instance;
    }
    function form( $instance ) {
    }
}
add_action(
    'widgets_init',
    create_function("", 'return register_widget("HoseiWidgetList");')
);
    
```

場合には、標準のテーマに子テーマを追加して、子テーマにカスタマイズした PHP および CSS を格納する方法が一般的である。これは将来 WordPress 本体をバージョンアップする際にカスタマイズした部分に影響なく WordPress 本体だけを入れ替えることができるという理由による。

この作法に従って、子テーマに `functions.php` を追加し、そこに `class` を記述して新たにウィジェットを作成した。表 4 にそのプログラムの一部を示す。WordPress では変更が少ないページを作成するときに固定ページのテンプレートを用いて作成する。その固定ページ用の機能として、ページ属性ともいえるメタデータを設定できるカスタムフィールド機能があり、変数とその値を任意に設定できる。年度毎に作成する研究報告では、変数として `bulletin`、その値として年度を指定する。これにより作成された固定ページは `bulletin` というキーに対してその年度を指定することによって検索が可能となる。

固定ページにはこうした設定がされているという前提で表 4 に示したプログラムの処理の概要を述べる。

それぞれのページで設定されているカスタムフィールド値は WordPress に用意されている `get_post_custom_keys` 関数で獲得できる(①)。そこで得られたカスタムフィールド値 `$customkeys` は配列形式になっておりシステム変数も含まれているので、それらを `array_filter` 関数にて取り除いて変数とデータからなる `$keys` を得る(②)。`$keys` には発行されたすべての年度の研究報告が含まれ、その年度と固定ページの URL が配列として関係づけられている。それらを `wp_list_pages` 関数によって `$out` に取り出し(③)、その後図 2 矢印で示す形式になるように HTML タグを設定する(④)。

5. モバイル対応

WordPress で標準的に使われているモバイル対応の仕組みは CSS の Media Query による。従ってブラウザで Web サイトにアクセスし、ブラウザ幅を狭めていくとテーマとして設定した `twentytwelve` ディレクトリ内にある `style.css` の `@media` にて設定されたピクセル値にてレイアウトが変更される。また、イメージも幅に合わせて縮小される。WordPress 3.5 から標準のテーマとして採用されている `twentytwelve` では PC 向けにはメニューバーで表示されているメニューがスマートフォン向けにはメニューボタンになる。その状況を図 3 に示す。テーマとして設定した `twentytwelve` では以前のテーマの `twentyeleven` に比較してページ上部の大きなロゴがなくなっているが、これは表示領域の狭いスマートフォン画面においてロゴにより専有される領域を削減するためである。また、図 2 および図 3 で示したように通常左側にレイアウトされることが多いサイドバーが右にレイアウトされている。スマートフォン利用時でもユーザの視線が集中するメイン領域は PC 利用時と同様に中央に配置されるがスマートフォン利用時には補助的な情報を提供するサイドバーはメイン領域の下に回りこむように設定され、ユーザに違和感を与えないようになっている(図 3)。

6. 考察

WordPress を用いて PC 対応の既存 Web サイトを PC に加えスマートフォンでも利用できるように Web サイトに移行した。その結果、既存 Web サイトをベースとした WordPress によるモバイル用 Web サイトとスマートフォン専用の Kurogo™ による Web サイトの両システムを比較することができるようになった。ここではそれらの比較からモバイルサイトで発信すべき情報について考察してみたい。

WordPress による Web サイトでは過去の Web サイトが提

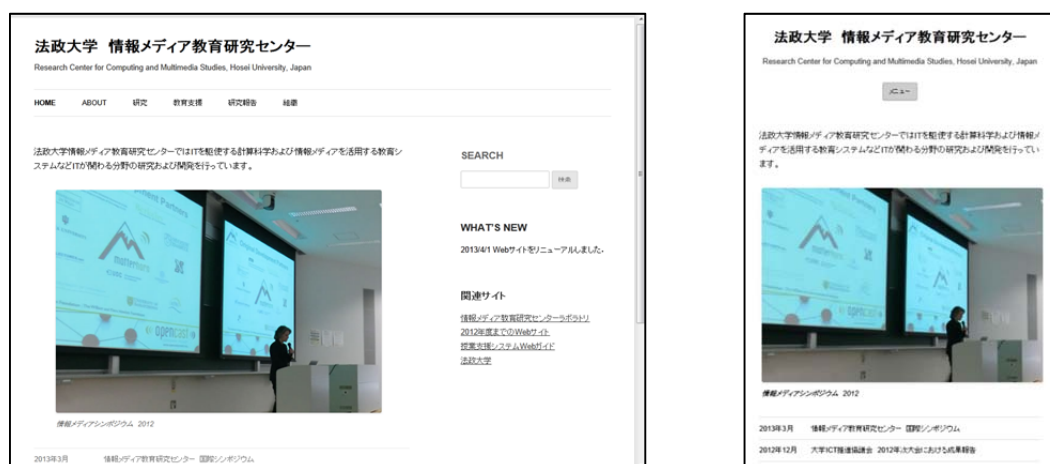


図 3 Responsive Web による PC およびモバイル用画面事例

Figure 3 An example of screenshots of PC and mobile to show a technology of responsive Web.

供していた情報を引き継ぎ、研究センターの沿革、理念、研究内容、研究成果、体制などが公開されている。一方 Kurogo™ によるモバイルサイトでは緊急連絡先、電話帳、地図、時刻表、ニュース、ビデオなどが公開されており、既存 Web サイトを継続した WordPress による Web サイトとは提供する内容が全く異なっている。実際 Kurogo™ によるモバイルサイトを構築した直後に、この理由で次期 Web サイト基盤として Kurogo™ を採用することを断念した経緯がある。

こうした点を踏まえて Kurogo™ を開発している Modo Labs の Web サイトにある事例 [7]を参照すると、Harvard University, Brown University などほとんどの大学が大学ドメインに m という mobile の頭文字であるサブドメインの m を配し、m.*.edu といった URL にてサービスを提供していることがわかる。これから、Kurogo™ を使ったモバイル用 Web サイトは大学の Web サイトを代替するものではなくモバイル専用の Web サイトとして位置づけられていると考えられる。しかしながら米国の大学のように十分な IT スタッフを抱えられない日本の大学では複数の Source による情報提供は大学をあげて取り組むべき課題だと思われる。

スマートフォンの利用の歴史はまだ浅くそのユースケースについても確立できていない状況ではそのユースケース自体が研究対象になり、これは新たな研究課題として別途取り上げるべきだと考えている。

7. おわりに

PC ユーザを前提とした HTML ベースの既存 Web サイトをスマートフォンユーザの要望に応えられるように Responsive Web 対応の WordPress 3.5 を用いて新たな Web サイトに移行し、一つのソースを PC およびスマートフォンから利用できる One Source Multi Devices の環境を構築し

た。また、これまで課題となっていた年度毎に出版されるオンラインジャーナル登録時の煩雑な作業を解決するために、関連リンクをマニュアルで更新することなく、自動的にリストを生成するウィジェットを開発し、WordPress の拡張性についても事例を示した。

また、当面の課題としてモバイル専用のサイトの位置づけを示した。これについては Responsive Web の機能を利用すると図 4 で示すようにスマートフォンでアクセスした場合にのみ Kurogo™ にジャンプする“Mobile Only”というボタンを表示することができるので、こうした実験的なサイトにてユースケースを今後検討して行きたい。

今後大学では BYOD に関連する要求により様々なデバイスに対応することになるだろうが、WordPress を基盤としたシステムも選択肢として検討される際に、本研究の成果が参考となれば幸いである。

参考文献

- 1) マイナビ: 2014 年卒マイナビ大学生のライフスタイル調査, マイナビ(オンライン), 入手先
<http://saponet.mynavi.jp/enq_gakusei/lifestyle/data/lifestyle_2014.pdf> (参照 2013-04-18)
- 2) マイナビ: 2013 年卒マイナビ大学生のライフスタイル調査, マイナビ(オンライン), 入手先
<http://saponet.mynavi.jp/enq_gakusei/lifestyle/data/lifestyle_2013.pdf> (参照 2013-04-18)
- 3) 常盤祐司: Drupal を用いた大学情報基盤システム開発, 情報処理学会研究報告教育学習支援情報システム (CLE), 2012-CLE-7(7), 1-7 (2012-05-19)
- 4) Modo Labs: Kurogo Mobile Middleware (online), available from
<<http://kurogo.org/home/>> (accessed 2013-04-17)
- 5) Jasig Community: uMobile (online), available from
<<http://www.jasig.org/umobile>> (accessed 2013-04-17)
- 6) W3Techs: extensive and reliable web technology surveys (online), available from <<http://w3techs.com/>> (accessed 2013-04-15)
- 7) Modo Labs: Kurogo Examples (online), available from
<<http://kurogo.org/examples/>> (accessed 2013-04-17)



図 4 モバイル専用ボタンの配置事例

Figure 4 An example of “Mobile Only” button.