

Plone4 による熊本大学公式 Web サイトの構築

永井 孝幸¹, 杉谷賢一¹, 久保田 真一郎¹, 木田 健¹, 松葉 龍一², 坂本 瑞穂², 伊澤 睦²,

岩永 菜穂子³, 中村 直美³, 谷口 勝紀⁴, 上田 誠⁴, 後藤 正三⁵, 河津 秀利⁵

熊本大学総合情報基盤センター¹, 熊本大学 e ラーニング推進機構²,
熊本大学マーケティング推進部広報戦略ユニット³, 熊本大学工学部技術部⁴,
熊本大学運営基盤管理部情報企画ユニット⁵

tnagai@cc.kumamoto-u.ac.jp¹

概要: 大学 Web を通じた情報発信を活発に行うには Web の更新作業を各部署で行えるようにすることが望ましいが, 学外へ公開するコンテンツについては編集ワークフローに基づいた承認手続きや不正アクセスへの対処など様々な要求を考慮する必要がある. 熊本大学では既存の業者製 CMS からオープンソース CMS への移行を検討し, 要求を満たす CMS として Plone4 を選定した. 本稿では Plone4 を用いた大学 Web サイトの構築方法について述べる.

1 はじめに

大学 Web を通じた情報発信を活発に行うには Web の更新作業を各部署で行えるようにすることが望ましいが, 学外へ公開するコンテンツについては編集ワークフローに基づいた承認手続きや不正アクセスへの対処など様々な要求を考慮する必要がある. 熊本大学では既存の業者製 CMS からオープンソース CMS への移行を検討し, 要求を満たす CMS として Plone4 を選定した. 本稿では Plone4 を用いた大学 Web サイトの構築方法について述べる.

本稿の構成は以下の通りである. まず 2 節で大学公式 Web サイトへの要求事項について述べ, 3 節で大学公式 Web サイト用 CMS の選定基準と選定過程について述べる. 4 節では大学公式 Web サイト用システムの構築結果について述べる. 5 節では大学 Web の構築運用体制について述べ, 最後に 6 節で大学 Web のリニューアルに関する今後の予定について述べる.

2 大学公式 Web サイトへの要求

本節では熊本大学公式 Web サイトの現状と, リニューアルに伴う要求事項について述べる.

2.1 熊本大学公式 Web サイトの現状

現在の熊本大学公式 Web サイトではコンテンツの編集を CMS 上で行い, コンテンツをいったん静的 HTML として出力したものを Web サーバで配信している. この方式は Web サーバでセキュ

リティ上の問題が起きにくいだけでなく, HTML エディタで作成したコンテンツも混在させて配信できるため, CMS エディタの機能不足を補えるというメリットがある. その代わりに, コンテンツの統一的な管理ができずに古いコンテンツが残ったままになる, コンテンツを時限公開できない, といった点が問題になっていた.

こういった不便をふまえながらもこの方式を用いてきた理由として, 過去(2005 年 4 月)に熊本大学の公式 Web サイトがクラッキングにより改ざんされた^{1,2}ことが挙げられる. この時の被害は利用していた CGI の脆弱性が直接の原因であったが, 以後, 熊本大学の公式 Web サイトでは動的コンテンツは用いないという状態が続いている.

2.2 熊本大学公式 Web サイトへの要求

熊本大学では平成 22 年の事務組織改編の一環として大学公式 Web サイトとは独立に教職員向け Web サイトを Joomla で構築し, 担当者自身によるコンテンツの軽微な修正を行えるようにしていた. 大学公式 Web サイトについても同じように各部局の情報発信者によるコンテンツ修正を可能にしたい, という要望が広報戦略ユニットから出され, この要望を受ける形で熊本大学公式 Web サイトのリニューアルを開始した.

¹熊本大学のホームページがハックされてた訳ですが (<http://blog.hiroaki.jp/2005/04/000253.html>)

²大変だ!! 熊本大学のトップページがクラックされている (<http://d.hatena.ne.jp/softether/20050414>)

今回のリニューアルでは、単に大学公式 Web サイトに用いる CMS を更新するだけでなく、大学公式 Web サイトと教職員向け Web サイトを統合することで Web サイトの更新・運用の利便性を図る、ということが大きな目標として挙げられた。これは両サイトで利用する CMS が異なると、職員の異動等が行われた際にコンテンツの管理・変更作業を行える担当者が不在になることがあり、安定的な情報の発信が難しくなるためである。このため、採用する CMS はユーザ権限に応じたコンテンツの編集・閲覧制限機能を備えるだけでなく、人事異動に伴う権限割り当ての変更にも対応可能なことが求められた。

これらの要望を満たすには静的 HTML を配信する現在の方式をやめ、CMS をネットワークに公開する方式に切り替えることが不可欠であるが、セキュリティを確保することが大前提である。大学公式 Web サイトは攻撃対象となりやすいため、CMS の選定においては機能面だけでなくセキュリティ面にも大きな比重を置くこととした。

3 大学公式 Web サイト用 CMS の選定

本節では今回の大学公式 Web リニューアルに伴う CMS の選定基準と選定経緯、Plone4 採用のメリット/デメリットについて述べる。

3.1 CMS 選定基準

大学公式 Web サイトに対する要求を CMS の技術要件に直した結果、次のようになった。

1. CAS 統合認証に対応すること
2. LDAP 等のユーザディレクトリを用いたグループ管理が可能なこと
3. ページ毎に編集・閲覧権限を設定できること
4. 階層化コンテンツの制作が容易であること
5. Web ブラウザ上でコンテンツの更新が可能なこと
6. コンテンツの版管理機能を有すること
7. 編集ワークフロー機能を備えること
8. 編集フローメールの自動送信ができること
9. 管理画面でコンテンツを更新日順にソートできること
10. 入試合格発表時の一斉アクセス(1 万リクエスト/分)に対応できること
11. 公開開始・終了日時にもとづくコンテンツの时限公開が可能なこと
12. 添付ファイルに日本語ファイル名が利用可能なこと

13. コンテンツの日本語全文検索が可能なこと
14. 多言語コンテンツの作成に対応していること
15. UI の言語を容易に切り替えられること
16. CSS によるデザインのカスタマイズが可能なこと

大規模・多人数運用に必要な要件が多く含まれていることが分かる。

3.2 候補 CMS 調査結果概要

3.1 節で述べた基準に基づき、商用およびオープンソースの CMS から候補システムの調査を行った。50 程度のシステムについて調査を行ったが、統合認証・編集ワークフロー対応等の大規模運用に必要な機能を備えているシステムはごく少数であった。調査を行った時点³で(多少のカスタマイズを前提とした上で)利用可能なシステムとして、候補に残ったのは以下のシステムである(表 1)。

表 1 大学 Web 用候補 CMS 一覧

順位	候補システム
1	Plone 4
2	Joomla 1.6
3	Joruri 1.1
4	ez Publish
5	Drupal 7

候補 CMS から最終候補を選定するにあたっては、システムの機能面だけでなく導入・運用面も考慮して判断を行った。導入・運用費用の面からはオープンソースであることが望ましいが、システム構築・改修を内製で行う必要があり、システム変更は最小限にとどめたい。Joruri についてはグループ別に閲覧権限を設定する機能がないため候補から外し、Drupal 7 についてはリリース直後で日本語化が不十分であることから候補から外れた。学内では Joomla によるサイト構築の経験があったため、「Plone 4 と Joomla 1.6 であれば Joomla の方が良い」との意見が優勢であった。ところが、Joomla 1.6 が長期サポート対象から外れ半年ごとにバージョンアップが行われる、という開発ロードマップが公表されたことから長期運用を前提とする大学 Web サイトに用いるのは適切

³ 候補 CMS の調査は 2011 年 1 月に実施

でないという結論になった。

以上の経緯から、最終的に Plone4 を用いて大学公式 Web サイトを構築することに決定した。

3.3 Plone4 採用のメリット・デメリット

Plone4 採用のメリット・デメリットをまとめると次のようになる。

(1)Plone4 採用のメリット

Plone を用いることのメリットとして大学 Web に必要な機能を満たすことの他に、高セキュリティであることが挙げられる。Plone については報告されるセキュリティ脆弱性の数も毎年 10 件未満と少なく、政府系機関での採用実績が知られている⁴。また Plone は Zope を基盤として構築されているため、CMS 内の全ての素材に対してアクセスコントロールリストを用いた統一的なアクセス制御が可能である。個々のプラグインで独自にアクセス制御を実装する方式では、場面によってアクセス制御が適用されない、プラグイン側のセキュリティホールによって意図しないアクセスが可能になるといった問題が発生するが、Plone/Zope の場合は Zope 側で統一的なアクセス制御が行われるためこの問題は発生しない。

更に、Zope では全てのデータは ZODB という Zope 独自形式のオブジェクトデータベースに格納されており、ZODB に対するアクセスは全て Zope の API を経由して行うことになる。このため、CMS 側で直接 SQL を発行する余地が無く、SQL インジェクション攻撃に対して強固である。

システム運用の面からは他のメリットもある。Plone4 は buildout という Python 独自のパッケージシステムを用いてインストール・アップデートが可能であり、Plone を動作させているサーバ OS 本体のバージョンアップとは独立に保守を行うことができる。OS のセキュリティアップデートとして CMS のセキュリティパッチが提供されるケースでは、Plone 本体のセキュリティパッチがリリースされた後、さらに OS のセキュリティパッチが提供されるまで待つ必要がある。また、オープンソースシステムの長期運用を考えた場合、サーバ OS のセキュリティパッチが 5 年先まで提供される保証はない。Plone 単体でパッケージシステムを用いてセキュリティパッチを適用可能なことは運用上の大きなメリットである。そのセキュリ

ティパッチ自体も Plone の場合は Plone 起動時に動的にパッチをあてる方式(monkey patch 方式)であり、セキュリティパッチを提供するためだけに Plone 本体のバージョンアップをする必要はない。

(2)Plone4 採用のデメリット

Plone4 を用いることのデメリットとしては第一に日本語資料の不足が挙げられる。Plone の日本語文献としては例えば「Plone 完全活用ガイド」[1]が挙げられるが内容は Plone 3 のものであり、2011 年 10 月時点で Plone 4 の管理運用に対応した内容の日本語書籍が存在しない。そのため、Plone3 の内容から類推して Plone4 を使うか、英語サイト・文献[2][3][4][5]から情報収集する必要がある。Plone 導入時のカスタマイズに関する情報がまとめられた数少ない日本語サイトとしては京都大学工学研究科附属情報センターのホームページ[6][7]が挙げられる。

また、Plone は CMS の一種であるが実体は Zope フレームワーク上に構築された Web アプリケーションであり、設定項目が Plone 内に閉じていない。大規模サイトの構築・運用には Plone だけでなく Zope に関する知識も必要となり導入の技術的なハードルは高い。

大学 Web サイトでは一斉アクセスへの対処も考える必要がある。Plone4 は Plone3 に比べ処理速度が向上しているが、初期インストール状態では毎秒数百リクエストの処理には対応できない。複数 Zope インスタンスを用いた負荷分散やリバースプロキシの導入等により要求性能を満たすようにシステム構築を行う必要があるが、一般に内製で行うには難易度が高いと思われる。

最後に、大学 Web に必要な機能を満たすには必要なプラグインの組み合わせを自分で調べる必要があるだけでなく、4 節で後述するようにプラグインの不具合への対処や機能拡張を自力で行う必要がある。この点で動作保証のある商用システムの導入が適切であるケースも多いと考えられる。

4 Plone4 によるシステム構築結果

本節では大学公式 Web サイト用に構築したシステム構成と Plone のカスタマイズ内容について述べる。なお、本稿で述べるシステム構成はあくまで現時点(2011 年 10 月)のものであり、本運用に入る段階で変更する可能性がある。

⁴ Gov 2.0 guide to Plone
(<http://govfresh.com/2011/03/gov-2-0-guide-to-plone/>)

4.1 システム全体構成

今回大学公式 Web サイト用に構築したシステムの構成を図 1 に示す。Plone を動作させるには Zope の他に MySQL, Apache など複数のサービスを動作させる必要があるため、仮想化環境(KVM)を用いて同一ブレードサーバ上に複数の Linux サーバを動作させている。

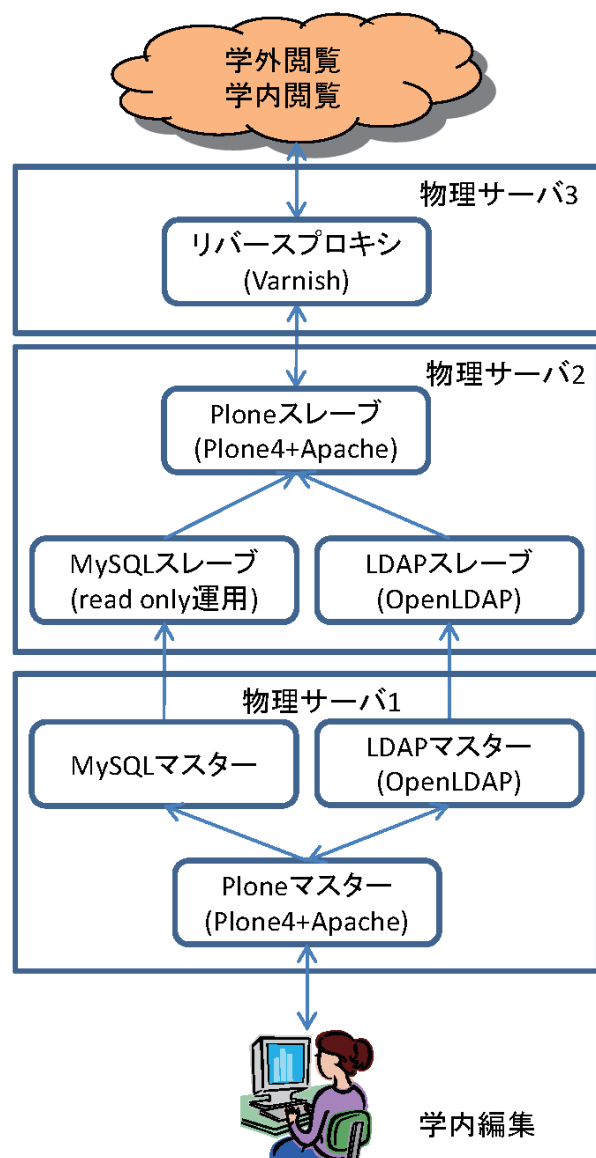


図 1 構築した大学 Web システムの構成

一斉アクセス時の負荷対策と外部攻撃によるコンテンツの改ざん防止のため、Plone を単体で用いるのではなく、更新系の Plone(マスター側)と参照系の Plone(スレーブ側)を設けている。コンテンツ保管用のバックエンドには MySQL を

用い、MySQL のレプリケーション機能を用いることでマスター側 Plone のコンテンツをスレーブ側に配信する。スレーブ側では MySQL・Plone を共に読み出し専用設定で動作させることで、不正な HTTP アクセス・SQL 実行によるコンテンツの改ざんを防止している。

更新系と参照系を分離しているため、参照系 Plone の負荷が高くなっても問題なく Plone の設定・コンテンツ変更を行うことができる。また、大学 Web サイトを止めることなく更新系 Plone の保守作業を行うことができる。

なお不正な HTTP リクエストによる攻撃や Zope 管理画面への不正アクセスを防止するため参照系・更新系ともに Plone を Apache HTTP サーバの背後に置き、Plone 自体は localhost としか通信を行わない設定にしている。

参照系 Plone では更に入試合格発表時の一斉アクセスへの対応としてリバースプロキシサーバ(Varnish)の背後に Plone を設置している。Varnish のキャッシュを用いることで、Plone 単体では 10 リクエスト/秒未満の処理速度しか得られない複雑なページであっても、数百リクエスト/秒まで応答速度を改善できることを確認している。

ユーザ・グループの管理には LDAP を参照する方式を用いている。大学 Web を運用するには数千ユーザ、数百グループを管理の必要があり、Plone の管理画面で直接ユーザ・グループの管理を行うのが難しいことが理由である。特に、今回リニューアルする大学 Web サイトでは人事異動の度にグループ情報を更新する必要があることから、その都度 Plone 上で設定作業を行うのは運用の負担が大きい。

4.2 プラグインによる Plone カスタマイズ内容

Plone を用いて大学 Web サイトを構築するには、機能・Web サイトデザイン両方の面でカスタマイズが必要となる。Web サイトデザインおよび一般利用者向け UI のカスタマイズについて

ては本協議会 2011 年度年次大会の別報告[8]にて発表するため、本節では 4.1 節で述べたシステム構成に必要な機能面でのカスタマイズ内容について述べる。

(1)CAS 統合認証, LDAP 連携

CAS 統合認証対応には `anz.casclient`, LDAP 連携には `python-ldap`, `PloneLDAP` プラグインを利用している。Plone ではユーザ認証, 権限管理, グループ管理にそれぞれ別のプラグインを指定することができ, ユーザの認証にのみ CAS プラグインを指定することで, LDAP によるユーザ・グループ管理と CAS 統合認証を両立させている。

(2)リバースプロキシ連携

Plone と独立にリバースプロキシを設置する場合, Plone 側でコンテンツを更新してもしばらくの間リバースプロキシ上のキャッシュに古いコンテンツが残ったままになるという問題がある。このキャッシュ不整合問題を避けるために, `plone.app.caching` プラグインを用いている。

このプラグインを用いることで, 更新のあったコンテンツのキャッシュを破棄するための PURGE リクエストを Plone からリバースプロキシに自動的に送ることができる。また, コンテンツの種類に応じてキャッシュポリシー(キャッシュの有効期間など)を変更することもできる。

(3)MySQL バックエンドへの対応

Plone のコンテンツは Zope 独自の ZODB と呼ばれるオブジェクトデータベースに格納されているが, このデータベースの保存形式を Zope 側のプラグイン設定で変更することができる。

ZODB をリレーショナルデータベースに保存するための Zope 用プラグインとしては `RelStorage` が利用可能である。今回は `RelStorage` を MySQL 用に設定することで MySQL のレプリケーション機能を用いたマスター・スレーブ間のコンテンツ同期を実現している。`RelStorage` の標準設定では Plone にアッ

プロードされるバイナリファイルをサーバ上のローカルファイルシステムに保存するようになっているが, 今回は MySQL のレプリケーションを通じて全てのデータを同期させるため, バイナリファイルもデータベースに保存するように設定している。

(4)編集ワークフローのカスタマイズ

Plone は初期インストール状態でも編集ワークフローのカスタマイズを行うことができるが, 状態遷移表を記入するという UI のため, 複雑なワークフローを構築することは困難である。大学 Web 用の編集ワークフローを作成するにあたり, Ajax による分かりやすい編集 UI を備えた `uwosh.northstar` プラグインを導入している。また, 編集フローメール内でマクロを用いてユーザ情報やコンテンツ情報を埋め込むために `stringinterp` プラグインも利用している。

(5)コンテンツの時限公開への対応

Plone は標準機能としてコンテンツの時限公開機能を備えているが, 今回の用途では機能が不十分であることが分かった。標準機能の動作は公開状態のコンテンツをメニュー表示・検索結果から除外するという動作であり, 直接コンテンツの URI にアクセスした場合には期日前でも公開状態コンテンツの参照ができてしまう。

入試の可否発表においては合格発表時刻よりも前に Web 上で結果を配信することは許容されず, 「合格発表ページの URI を予想してアクセスすれば事前に可否発表を閲覧できてしまう」という動作には問題がある。

そこで, Plone 標準の時限公開機能を利用する代わりに, 指定時刻にコンテンツを非公開状態から公開状態に切り替える自作の Python スクリプトを動作させる方式[9]を採用した。この自作スクリプトを定時実行させるために `cron4plone` プラグインを用いている。

4.3 ソース改変による Plone カスタマイズ内容

今回のシステム構成で Plone を運用するにあ

たり、一部の動作実現にはソースコードレベルでの改変が不可避となった。

(1) 読取り専用状態での不具合対策

スレーブ側 Plone を読取り専用状態で動作させるには、基本的には ZODB を read-only に設定すればよい[10]。しかしながら、この状態で Plone にログインするとログイン履歴の書き込みができないためにエラーが発生し、事実上 Plone にログインできないことが判明した。

リニューアル後の大学 Web サイトでは、教職員向けコンテンツは認証済みの教職員ユーザに対してのみ閲覧を許可することを想定しており、スレーブ側 Plone へログインできない不具合は致命的である。この現象への対策として、read-only 状態では ZODB への書き込みに失敗してもエラーを出さないように RelStorage プラグインの改修を行った。

また、スレーブ側 Plone にログインした際、コンテンツの編集権限を持つユーザについては、自分が編集権限を持つコンテンツを閲覧した際に編集メニューが表示されてしまうという問題があった。参照専用の Plone で編集メニューが表示されるのはユーザに混乱を招くことから、Plone が読取り専用状態に設定されている時は編集メニューを表示しないよう Plone 本体の改修を行った。

(2) CAS 認証ユーザに対する Cookie の設定

リバースプロキシを用いる際、同じ URI でもユーザによって表示内容が異なるページにはキャッシュを用いないようにする必要がある。Plone/Zope でこの動作を実現するには、Plone にログインした際に Cookie に設定される属性 (ac) の有無を調べるのが一般的な方法であるが、CAS 認証でログインした場合にはこの属性が設定されない。そのため、CAS ログイン時に Cookie の属性を設定するよう anz.casclient プラグインの改修を行った。

(3) 代理ログイン(SU)機能の CAS・LDAP 対応

大学 Web サイトのように大勢のユーザが利

用するシステムでは、不具合報告があった際の状況把握のために管理者権限で指定ユーザとして代理ログインする機能(Unix での SU 機能)が不可欠である。

Plone4 では Niteoweb.LoginAs プラグインが利用可能であるが、このプラグインが LDAP 連携に対応しておらず、LDAP 上のユーザとしてログインしようとする「該当ユーザなし」という扱いになる。そこでユーザ検索ルーチンを改修し、LDAP 上のユーザに対しても代理ログインを可能としている。また、セッション内に保持された CAS の認証情報を代理ログイン時に破棄する処理も追加している。

5 大学 Web 構築運用体制

本節では Web サイトリニューアルの組織体制と編集ワークフローについて述べる。

5.1 Web サイトリニューアル組織体制

大学公式 Web のリニューアルは大学全体の広報に関わる事項であるため、大学ホームページに関する最上位の組織である「広報推進会議」の下、学内の各部局と連携しながら作業を行っている。Web サイトリニューアルに伴うページのデザイン変更については、基本的に現行 Web サイトのデザイン(図 2)を踏襲するものとし、Plone への移行に伴う外見上の変化が少なくなるようにしている(図 3)。



図 2 現行の大学 Web トップページ



図 3 リニューアル後のトップページ(予定)

Web サイトリニューアルのため、e ラーニング推進機構兼務職員を中心に事務部・技術部から総勢 23 名により作業チームを構成し、メンバーを小規模のワーキンググループに分けて作業を進めている。各グループの進捗を月に一度の定例ミーティングで報告し、検討事項・問題点の整理やスケジュール調整を行っている。システムの技術的な側面を総合情報基盤センター・工学部技術・情報企画ユニットが担当し、Web サイトデザインのカスタマイズ・コンテンツ作成を e ラーニング推進機構が担当している。また、Plone 導入時の教職員研修に必要な研修用コンテンツの作成も e ラーニング推進機構が担当する予定である。

5.2 編集ワークフロー

熊本大学では大学公式ホームページに掲載できる情報に関して基準を設けており、学外に公開するコンテンツについては全て広報戦略ユニットのチェックを得た後に公開することになっている。

編集ワークフローの策定にあたって広報戦略ユニットと検討を重ね、リニューアル後の大学 Web サイトでは「制作者」「承認者」「総合管理者」の 3 種類のユーザ区分を設けることとした(図 4)。「制作者」は各部局でホームページ用コンテンツの作成を行うスタッフに対応し、「承認者」は各部局の課長に相当する。「総合管理者」は広報戦略ユニットの学外ホームページ責任者

に対応し、コンテンツ公開に関する最終権限を持っている。

コンテンツの作成はまず部局レベルで行い、「制作者」はコンテンツ作成後に「承認者」に承認依頼を行い、コンテンツに不備があればこの時点で「制作者」に差し戻される。不備がなければ「承認者」は該当コンテンツの「公開依頼」を「総合管理者」に対して行う。「総合管理者」は「公開依頼」のあったコンテンツの内容を確認し、問題がなければ「公開」とする。

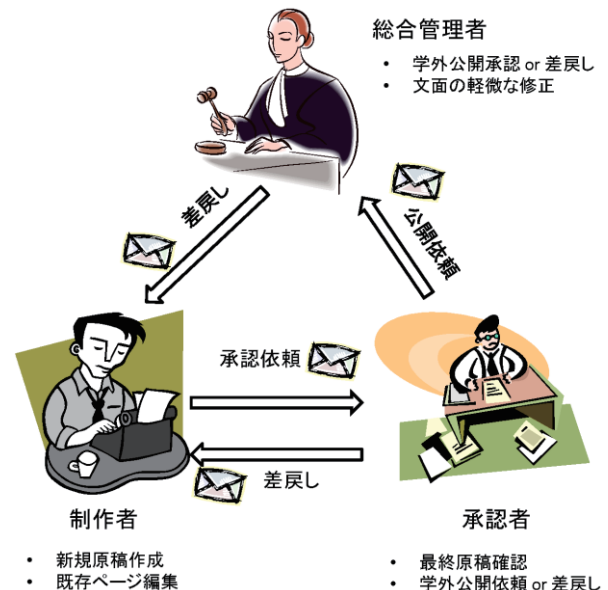


図 4 大学 Web の学外公開ワークフロー

基本的には各部局の承認者の決済を通った時点で学外に公開しても問題ない状態にコンテンツが仕上がっていることを想定しているが、予定されていたイベントが急遽とりやめになった等の例外的な事態が発生した場合は総合管理者によって学外公開申請を却下することになる。

「制作者」「承認者」「総合管理者」間のやりとりには Plone の編集フローメール機能を利用しており、コンテンツ編集画面にある決済用メニュー(「承認申請」「学外公開申請」「却下」など)を選択するだけで該当者にメールで知らせるようになっている。

6 今後の予定

Plone によるシステム構築についてはほぼ目処がたったところであるが、実運用に入るまでにはまだ準備が必要である。本節では実運用に入るまでに実施を予定している作業について述べる。

6.1 既存コンテンツの移行

Plone4 を用いて大学 Web サイトをリニューアルするにあたっては、既存コンテンツの移行作業が大きな課題となる。Plone には他 CMS のデータを一括で取り込む機能はないため、数千ページに及ぶ既存コンテンツを手作業で Plone のコンテンツとして作り直す必要がある。

WYSIWYG エディタ上のコピー・ペーストで済む作業もあるものの、外部 HTML エディタによって作られた複雑なレイアウトや図表には個別に対処する必要がある。特に事務手続き関係のページでは書式見本等の図表・PDF が大量に使われているため、学外コンテンツに比べてより多くの作業時間がかかると見込まれている。

6.2 セキュリティ監査

利用しているソフトウェアにセキュリティホールがないことと、システム全体が適切に機能していることは話が別であり、リニューアルした Web サイトの本運用を開始する前にセキュリティ監査を実施する予定である。

6.3 教職員研修

大学 Web をリニューアルする目的は各部局の情報発信者によるコンテンツ修正を可能にすることであり、各部局のコンテンツ製作者・承認者に対する利用者研修を行うことが不可欠である。Plone の一般的な利用方法については市販の書籍が利用可能であるが、今回構築した編集ワークフローや情報発信ガイドライン等の熊本大学固有の内容についても合わせて研修を行う必要があり、学内で研修用コンテンツを作成する予定である。熊本大学では授業や研修にお

いて e ラーニングの活用を進めており、今回の研修にも e ラーニングを活用する予定である。

謝辞

Plone4 を使った Web サイトの構築・カスタマイズにあたっては、京都大学工学研究科附属情報センターのホームページで公開されている技術情報[6][7]が大変参考になりました。この場を借りてお礼を申し上げます。

参考文献

- [1] 柴田 淳, 鈴木 たかのり, 木下 文史他, Plone 完全活用ガイド, 技術評論社, 2008
- [2] Martin Aspeli, Professional Plone Development, Packt Publishing, 2007
- [3] M. Delmonte, D. Moro, A. Narduzzo, F. Reale, A. McKay, The Definitive Guide to Plone, Apress, 2009
- [4] Robert J Nagle, A User's Guide to Plone 4, Enfold Systems, Inc., 2010
- [5] Martin Aspeli, Plone 4 Professional Development, Packt Publishing, 2011
- [6] Plone3.3 サイトの構築,
<http://info.t.kyoto-u.ac.jp/ja/docs/plone33>
- [7] Plone4.0 サイトの構築 ,
<http://info.t.kyoto-u.ac.jp/ja/docs/plone40>
- [8] 坂本 瑞穂, 伊澤 睦, 久保田 真一郎, 永井 孝幸, 松葉 龍一, 熊本大学公式 Web サイトの構築—CSS 等のカスタマイズによる Web サイトデザイナー, 大学 ICT 推進協議会 2011 年度年次大会
- [9] Time-based publishing in Plone with higher security,
<http://valentinewebsystems.com/en/blog/time-based-publishing-with-higher-security>
- [10] Notes on a More Secure Plone Deployment,
<http://nathanvangheem.com/news/notes-on-a-more-secure-plone-deployment>