
オープンソースソフトウェア（OSS）を使用した ポータルサイトの構築

（株）メイテツコム

■ 執筆者 Profile ■



鏡 志圭生

1992年 株式会社メイテツコム 入社
文教向けシステム担当。
主に、私立大学事務系システムの
開発・運用・維持を担当

2008年 現在 第2システム事業部
サービス事業担当 所属

■ 論文要旨 ■

学校法人梅村学園中京大学様では学生個人への情報発信ツールとして、学生用ポータルサイトシステムの構築を決定し、当社がその開発を受託した。このポータルサイトは、オープンソースソフトウェア（OSS）のCMS「Plone」を使用して構築を行った。

稼動後、約1年経過するが9割以上の学生が利用しており、学内の掲示板も廃止され、学生への情報発信ツールとして定着したと思われる。

本論文では、

- ・「Plone」の採用理由及び選定方法
- ・システム概要
- ・開発の実際
- ・大学からの要求の実現及び問題点の解決 など

について記述した。

今回の構築を通じて、我々のような情報サービス事業者にとって、適切なOSSの利用は、商用パッケージ製品への対抗手段となり得ることが実感できた。また、OSSを利用したシステム提供には、ソースプログラムレベルの理解が必要であり、問題解決にあたっては、Webでの情報収集が必須であることも実感できた。

■ 論文目次 ■

1. はじめに	《 4》
1. 1 当社の概要	
1. 2 システム化の背景	
2. オープンソースソフトウェア利用の経緯	《 4》
3. 利用オープンソースソフトウェアの概要	《 6》
3. 1 Python について	
3. 2 Zopeについて	
3. 3 Ploneについて	
4. システム概要	《 7》
4. 1 システムの目的	
4. 2 利用者	
4. 3 ポータルサイトの構成要素	
4. 4 情報提供のイメージ	
4. 5 ハードウェア構成	
4. 6 ソフトウェア構成	
5. 開発について	《 10》
5. 1 開発スケジュール	
5. 2 開発手法	
5. 3 開発環境	
5. 4 開発体制	
6. 要求・問題の解決	《 12》
6. 1 ユーザ認証	
6. 2 コンフリクトエラー	
6. 3 ロードバランス	
6. 4 パフォーマンスの悪化	
7. 稼働状況と今後の拡張	《 14》
8. おわりに	《 16》

■ 図表一覧 ■

図1	学内に設置されている掲示板	《 4》
図2	KIOSK端末	《 8》
図3	プラズマ表示装置	《 8》
図4	情報提供のイメージ図	《 8》
図5	ハードウェア構成（サーバ）	《 9》
図6	ハードウェア構成（クライアント）	《 9》
図7	webアプリケーションサーバ ソフトウェア構成	《 10》
図8	データベースサーバ/バックアップサーバ ソフトウェア構成	《 10》
図9	開発体制	《 11》
図10	PloneでのセッションIDの埋め込み	《 13》
図11	ポータルサイトトップページ	《 14》
図12	携帯サイトトップページ	《 14》
図13	プラズマディスプレイ（休講ボード）	《 14》
図14	ポータル稼動後の掲示版	《 15》
図15	時間帯毎のログイン数	《 15》
表1	オープンソースソフトウェアを使用したCMSの比較	《 5》
表2	開発スケジュール	《 10》

1. はじめに

1. 1 当社の概要

株式会社メイテツコム

設立：1976年9月27日

所在地：名古屋市中村区名駅南一丁目21番12号 名鉄協商コンピュータビル

従業員：271名（2008年7月現在）

名古屋鉄道グループの情報システム会社として誕生。

創立30周年を機に『SCOPE INNOVATION』（視界革新）というブランディングステートメントを設定、ITコンサルティングからシステム開発、システム運用、データセンタによるネットワークとハードの保守・運用およびセキュアなデータ管理まで、ワンストップでITの全領域をカバーする体制と人財を用意し、高度で上質なサービス提供を目指している。

学校法人梅村学園中京大学様（以下 中京大学）とは1986年10月より入試をはじめとする学籍・教務・財務・就職・校友会等、基幹事務システムの開発・運用を受託。

2002年にメインフレームからWEB系へシステムリプレースを機に、大学—当社データセンタ間を専用線で接続してサーバのハウジングを開始、引き続き基幹事務システムの運用を実施している。

1. 2 システム化の背景

中京大学では、学内に設置されている掲示板（図1）を利用して学生に情報を発信していた。これでは、学生が大学に来なければ情報の収集ができず、また、掲示されているさまざまな情報から自分に必要なものを取捨選択しなければならない。

たとえば、「月曜1限〇〇科目履修の学生へ」という掲示物があった場合、該当科目を履修している学生のみが対象だが、当然関係ない学生も掲示を見る事ができる。逆に該当する学生が必要な掲示を見逃すことも考えられる。

これらの問題解決にあたり、中京大学では学生個人への情報発信ツールとして、学生用ポータルサイトシステムの構築が決定され、当社がその開発を受託した。

このポータルサイトは、オープンソースソフトウェアのCMS「Plone」を使用して構築を行った。本論文は、アプリケーションプロダクトとしてのオープンソースソフトウェアを使用したシステム構築の事例として記述する。



(図1) 学内に設置されている掲示板

2. オープンソースソフトウェア利用の経緯

当開発案件は、中京大学の作成した要求仕様を含む提案依頼に基づいてコンペティションが開催された。要求仕様書は、非常に詳しく記述されており、新規参入会社にとって不利にならないような配慮がなされていた。言い換えれば、システム運用業務を通じて中京

大学の事情を熟知している当社の優位性が失われるほどであった。

このとき、提案依頼を受けた業者は当社以外にも数社あったが、全てパッケージ製品を持ち、中には大手メーカーも含まれていた。

パッケージに対抗する手段として検討した結果、まず中心プロダクトとして CMS を使用してシステムを構築することにした。これは、紙書類の掲示物を Web 画面上のコンテンツとして捉えた結果といえる。また、製品として販売されている、商用の CMS も存在するが、高価なものが多く、価格面での競争力向上を意識して、オープンソースソフトウェアの CMS を使用することにした。

比較検討を行ったオープンソースソフトウェアの CMS は次の通りである。（表 1）

（表 1） オープンソースソフトウェアを使用したCMSの比較

CMS	言語	Webサーバ	データベース	特徴
Plone	Python	Zope内蔵のZserver Apacheなどとの連携が可能	Zope内蔵のZODB Oracle、PostgreSQL などとの連携可能	大規模Webサイトでの導入実績がある 日本語の情報も結構ある 欧米ではもっとも有名
Drupal	PHP	Apacheなど	MySQL(推奨) PostgreSQL	比較的規模が小さく、処理動作が軽い あまり普及していない
XOOPS	PHP	Apache(推奨)	MySQL	日本でもっとも普及、多数の実績あり
Mambo	PHP		MySQL	ニュース掲示板などを備えた小規模なWebサイトを構築、管理できるソフトウェア
Joomla	PHP	Apache IIS	MySQL	Mamboからフォーク
Geeklog	PHP	Apache(推奨)	MySQL	ブログ&コミュニティポータル支援ツール
WordPress	PHP		MySQL	日本語の情報がない
OpenCms	Java			日本語の情報がない

今回、中京大学の要求仕様には、

- ・サーバOSはUnix または Linuxを使用すること
- ・データベースは、Oracle または PostgreSQL を使用すること

とあり、表中でこの条件を満たすものは、上位2つであるが、システム規模等を考慮した結果、Ploneの採用を決定した。また、国内でも多数の利用実績があり、日本語での情報が多数存在することも採用を決定した要因の一つである。

更に提案書の作成にあたっては、実際にPloneを使った現実味の感じられるサンプル画面を盛り込んだ。また、ハードウェア提案においては、提案書に「ハードウェア機器についてはお客様ご指定の製品への変更が可能です。」との注記をした。これは、当社はユーザー系システム子会社であり、ハードウェア選択に制約が無いことをアピールし、かつ提案競合会社対策の意味合いを持つ。この結果、

「本学のことを良く判っておられ、メイテツコムに発注した場合、リスクがほとんど無い」

との評価を受け、見事に受注できた。

本開発では契約書に

「今後5年間、愛知県内の学生数5,000人以上の大学に対して本件ソフトウェアを利用させる場合は、乙（当社）は甲（中京大学）の了承を得なければならないものとする。」との条項が盛り込まれた。

おそらく、大学側としても他大学と同じパッケージよりも大学オリジナルの物を作成し、他大学と差別化を図りたかったと思われる。

3. 利用オープンソースソフトウェアの概要

オープンソースソフトウェアの代表例としては、Linux や Apache であるが、今回利用した他のオープンソースソフトウェアについて紹介する。

3. 1 Python について

Python は Guido van Rossum 氏によって作られたオープンソースのプログラミング言語である。オブジェクト指向スクリプト言語の一種であり、文法が簡易で可読性に優れる一方、拡張モジュール（ライブラリ）が豊富に用意されており、テキスト処理に限らず多様なアプリケーションの開発に利用できる。Perl とともに欧米で広く普及している。

Red Hat 系 Linux ではインストーラや環境設定ツールの開発言語としても利用されている。

3. 2 Zope について

Zope は、「Z Object Publishing Environment」の略で、Python によって書かれた、Web アプリケーションサーバであり、同時に CMS でもある。

Python が動作する環境ならどこでも動作し、自身がウェブサーバ、データベースサーバとしての機能を持っているため、利用環境の構築が簡単である。Zope には、導入時点で、Blog や Wiki といったウェブアプリケーションが同梱されており、これらを組み合わせて容易にウェブサイトを構築できる。

Zope 自身の拡張は、Zope 上で Python を使って行うことができ、プログラマーのスキルに応じた開発、運用が可能である。

3. 3 Plone について

Plone は Zope 上でのプロダクト（アドオン機能）として動作する。

CMS にも様々な種類が存在するが、機能の充実度や完成度の高さでは Plone は非常に優れている。Plone は標準機能だけで完結できるほど完成度が高く、使いやすくなっている。さらに、標準機能だけでなく、追加機能も豊富に準備されており、かつ、他のシステムとの連携も柔軟に行うことができる。

Web ワークフロー機能（コンテンツ作成→承認→公開）も標準で備えている。

また、世界中の開発者がさまざまなプロダクト（機能がモジュール化されたもの）を提供しており、標準機能にない、ブログや投票といった機能を追加する事も容易にできる。

Plone の使用事例を次に示す。

- ・国内

JETRO (<http://www.jetro.go.jp/indexj.html>)

2004 年秋に公開された、日本の Plone 業界の象徴的な存在となっているサイトである。

安部晋三オフィシャルサイト (<http://newleader.s-abe.or.jp>)

いわずと知れた前総理大臣のオフィシャルサイトである。

ダイヤモンド・オンライン (<http://diamond.jp>)

出版社のダイヤモンド社が提供する、ビジネス情報サイトである。

- ・海外

NASA 火星探査シミュレータ (<http://mars.telascience.org/home>)

eBay Developer サイト (<http://developer.ebay.com>)

ニュージーランド電子政府サイト (<http://www.e.govt.nz/standards/>)

ハワイ州 (<http://www.hawaii.gov/gov>)

4. システム概要

今回、構築したシステム概要は次の通りである。

4. 1 システムの目的

①学生の利便性の向上

- ・学内に点在する掲示板を Web 化することで情報収集が容易になる
- ・学内だけでなく、自宅などからでも情報収集が容易になる
- ・個人に必要な情報のみを入手することができるようになる

②職員の業務改善

- ・掲示板の管理作業を軽減できる
- ・必要な情報を必要な学生に容易に周知できる

4. 2 利用者

①システム管理者

- ・情報システム部（1名）、教務課（1名）の計2名

②情報発信者

- ・事務部署職員（約150名）原則全ての事務部署。教員については、当面利用しない

③情報取得者

- ・学部学生（約13,000名）科目等履修生、特別聴講生も含む

4. 3 ポータルサイトの構成要素

- ①ポータル Web サイト
 - ・インターネットに接続されたパソコンにて参照する
- ②ポータル携帯サイト
 - ・携帯電話のインターネット接続機能にて参照する
 - ・NTT Docomo, AU, SoftBank Mobile の第2世代以上に対応
- ③学内情報端末
 - ・KIOSK 端末 (図2) を利用してポータル Web サイトの内容を参照する
 - ・学生証(磁気カード) を挿入することにより容易にログインができる
- ④プラズマ表示装置 (図3)
 - ・多くの利用者が参照することができる大型表示装置
 - ・休講情報、ニュースボード用の2枚1組で配置
 - ・焼付け防止のため、一定時間毎に画面をスクロールさせる
- ⑤携帯用メール配信
 - ・携帯電話用メールに情報を配信する
(休講・補講・教室変更・学生呼出)



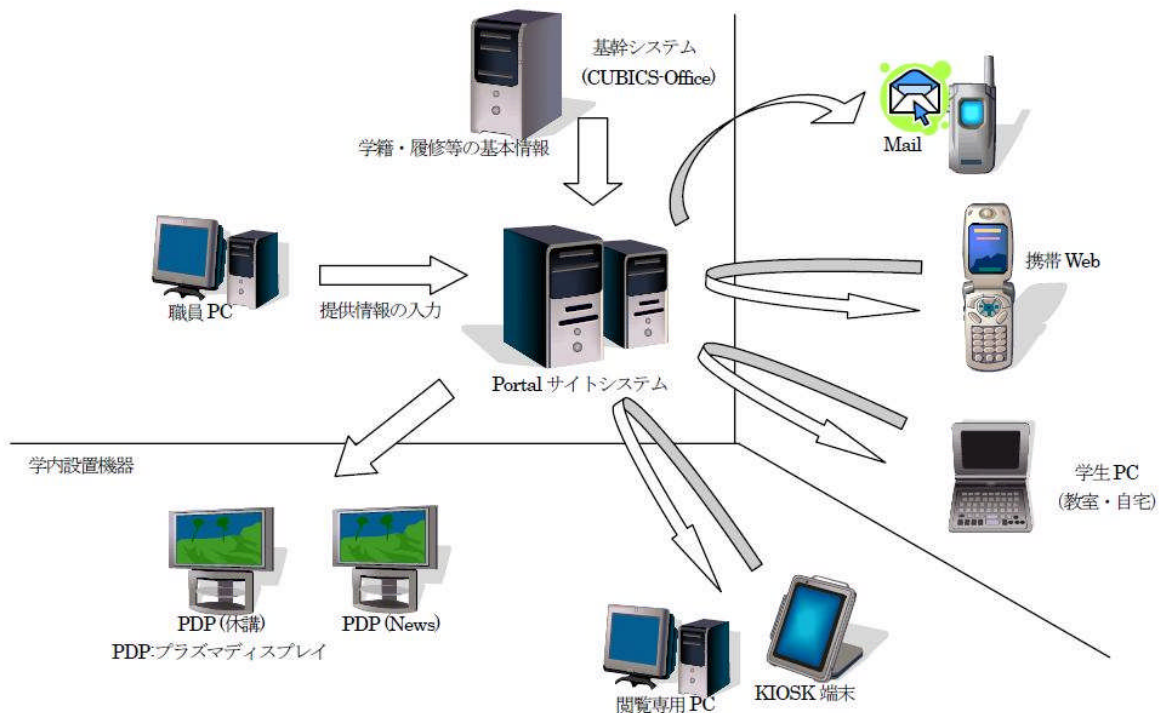
(図2) KIOSK 端末



(図3) プラズマ表示装置

4. 4 情報提供のイメージ

情報提供のイメージは図4のとおりである。



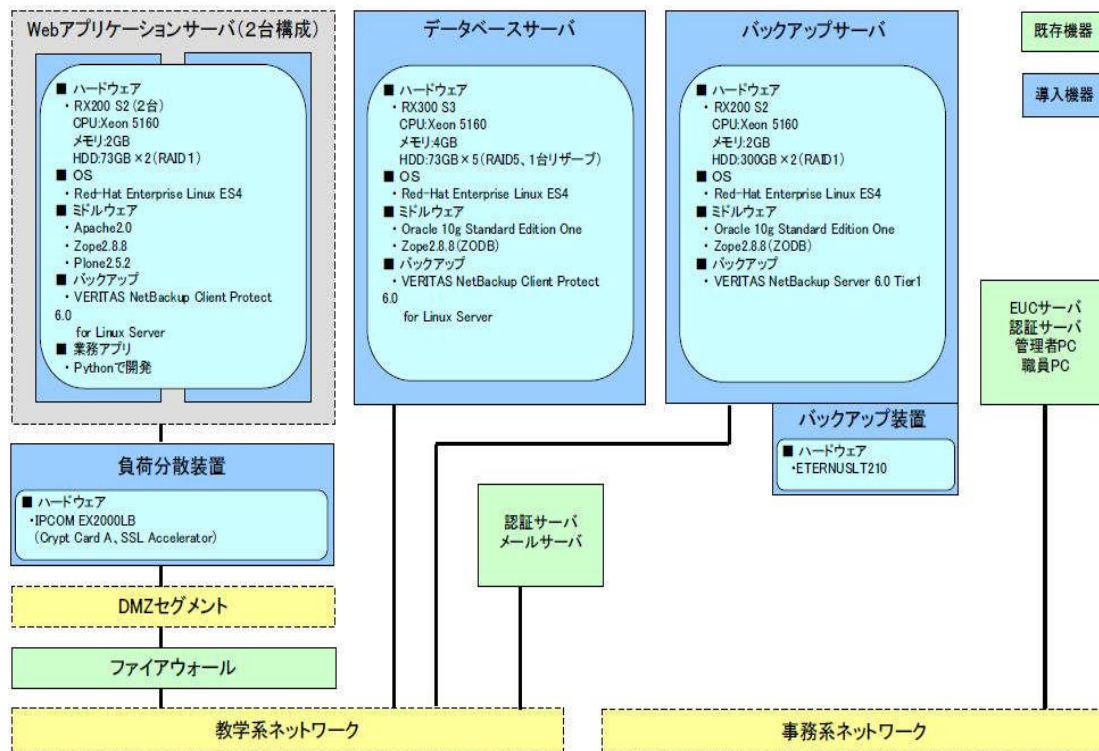
(図4) 情報提供のイメージ図

4.5 ハードウェア構成

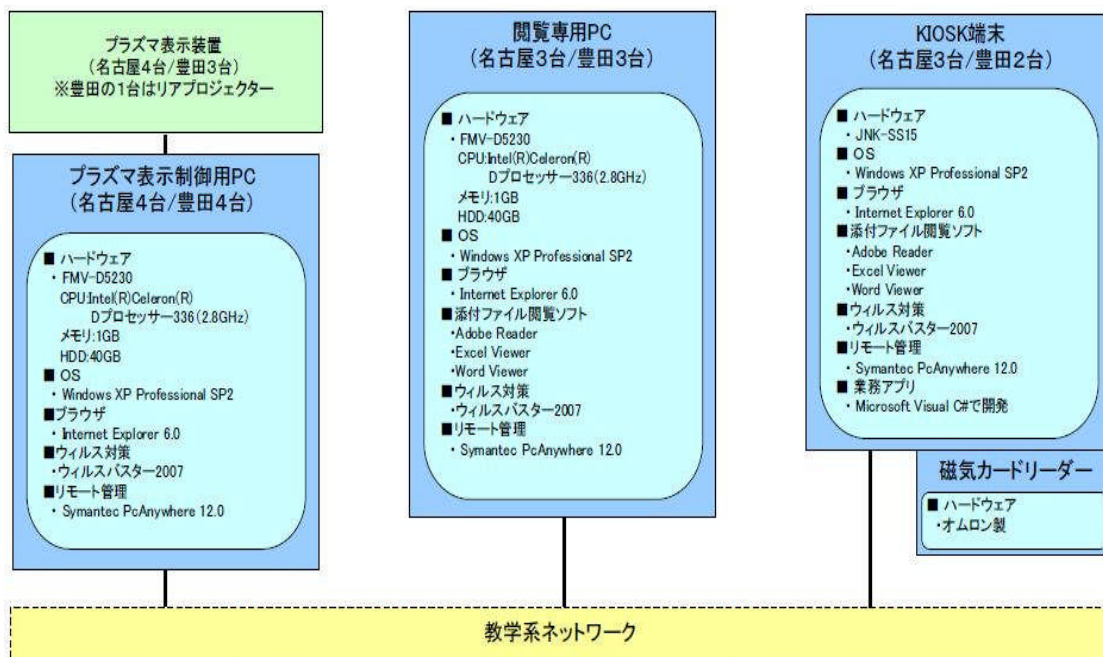
ハードウェア構成は、図5・6のとおりである。

認証・メール・EUCサーバ、ファイアウォールについては大学既存の機器を使用した。

また、データベースサーバ障害時に備えてバックアップサーバが代替機として使用できるようにした。



(図5) ハードウェア構成 (サーバ)



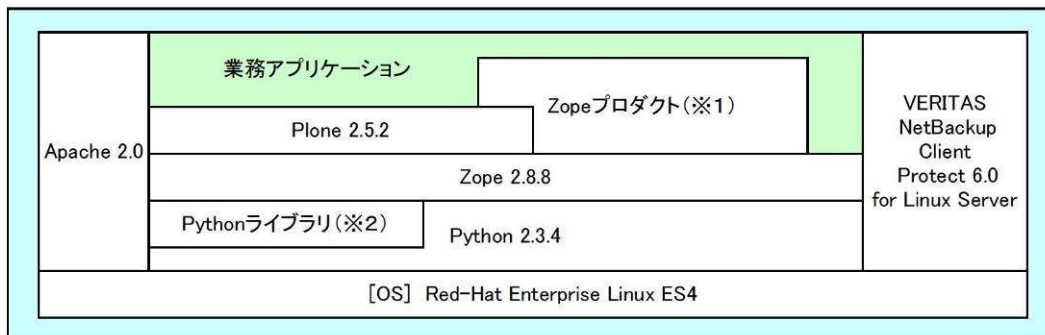
(図6) ハードウェア構成 (クライアント)

4. 6 ソフトウェア構成

サーバのソフトウェア構成は、図7・8のとおりである。

サーバ側の業務アプリケーションはすべてPythonで開発している。

クライアントについては KIOSK 端末のみ、Visual C#を使用してアプリケーション（磁気カード読込ソフト）を開発した。



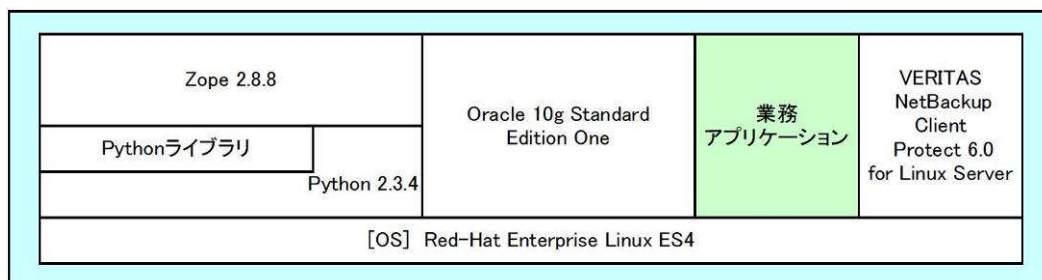
(図7) web アプリケーションサーバ ソフトウェア構成

※1. 追加で使用する Zope プロダクト

LDAPUserFolder 2.7, LDAPMultiPlugins 1.4, Session Crumbler 0.2, CMFContentPanels 2.4, Five 1.2.6

※2. 追加で使用する Python ライブラリ

python-ldap 2.2.1, JapaneseCodecs 1.4.11, Python ImageLibrary 1.1.6



(図8) データベースサーバ/バックアップサーバ ソフトウェア構成

5. 開発について

5. 1 開発スケジュール

開発スケジュールは表2のとおりである。8月のシステムテストからは本番機を使用した。

(表2) 開発スケジュール

2007年	1月	2月	3月	4月	5月	6月	7月	8月	9月
ソフトウェア	要件定義		基本設計		製造			システムテスト	運用テスト
ハードウェア	▲	▲			▲	▲	▲	▲	▲
	デモ機セットアップ				本番機発注	本番機セットアップ	機器移設		
					KIOSK・PC発注				

5. 2 開発手法

開発手法にはプロトタイピングを応用した手法を使用した。

これは、大学側の要求仕様が詳細に決定しており、視覚的なデザイン・操作性が最大の関心事によるためである。

5. 3 開発環境

開発環境は、提案時より協力をいただいている協力会社側に構築した。

更に当社内にはデモ環境を構築し、直接、中京大学のネットワークに接続した。これは、大学と当社間は専用線（光ファイバ）ですでに接続されており、大学側ネットワークの一部が当社内に存在したため容易にできた。

また、デモ環境の構築には、社内の既存の PC に CentOS (Red Hat Enterprise Linux と互換することを目指したフリーの Linux ディストリビューション) をインストールし、そこに Zope / Plone をインストールした。つまり、OS から使用するアプリケーションまでまったく費用をかけずに調達することができた。

その後、協力会社で開発したシステムをデモ環境に移行することにより大学職員がいつでも画面レイアウトや操作性を確認することができ、バグや仕様不足などを随時ご指摘いただけた。

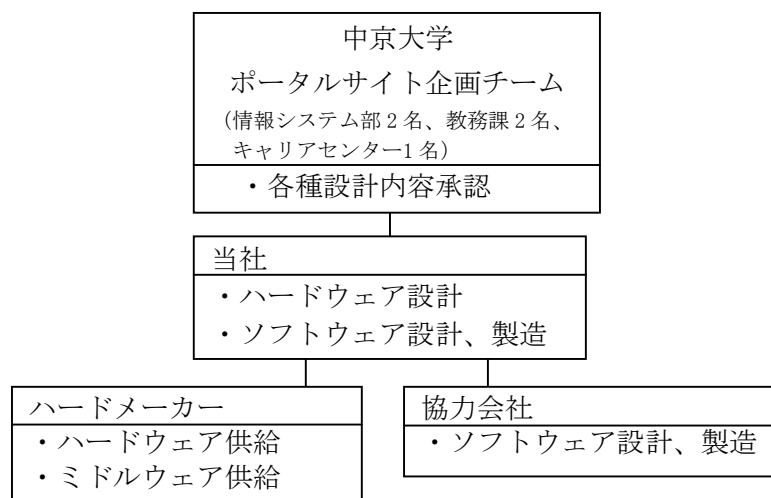
5. 4 開発体制

開発体制は、図9のとおりである。

打ち合せは、月2回程度行った。メンバーは、中京大学ポータルサイト企画チーム（5名）と当社・協力会社のSE各1名である。

意見交換については主にメールを使用した。その結果、少ない打ち合わせ回数で開発を進めることができた。

開発が始まってからは、当社の PG 1 名は、協力会社に出向いて、協力会社の PG と一緒に作業を行った。



(図9) 開発体制

6. 要求・問題の解決

ここでは、大学からの要求および開発時に発生した問題をどのように解決したかを述べる。

問題解決にあたり、オープンソースであるがゆえ当然メーカー等のサポートは無く、全て書籍や Web にて情報収集を行った。Web での情報収集も海外のサイトを調べることも多かった。

当然、ソースコードも解析し、時にはログを書き出すようにして修正箇所を特定することもあった。

6. 1 ユーザ認証

認証サーバは大学に設置されている既存の認証サーバ (LDAP) を使用する。しかし、学生 (CNC ドメイン) と職員 (MNG ドメイン) では認証するサーバが異なる。

Plone は複数の認証サーバ (LDAP) を使用することができるが、動作は次のようになる。

ユーザログイン時、Plone に登録されている認証サーバを順次検索し、ユーザが見つかった時点でパスワードの確認を行い、一致すればログイン可とする。

ここで問題なのが、教員と職員で同じアカウントが存在することがある。教員は CNC ドメインに登録されており、職員は MNG ドメインに登録されている。

このままでは、教員と同じアカウントを持つ職員 (実際には別人である) がログインすると CNC ドメインに同じアカウントが存在するためパスワードが不一致となりログインができなくなる。そこで、大学側と協議し、職員がログインする場合は、アカウント名の後に「@mng」を付加し、Plone 側で MNG ドメインにパスワードの確認を行うときは、アカウントの後に「@mng」がある場合は、除去した後、確認を行うよう Plone の認証部分 (Zope プロダクト) をカスタマイズした。

6. 2 コンフリクトエラー

システムテストにて同時アクセスのテスト (100 人の学生が同時にメールアドレスの登録を行う) を行ったところ、コンフリクトエラー (資源の衝突) が多発した。これは、学生のメールアドレスを全て同じコンテンツ内に登録をしている関係で、ある学生がメールアドレスを登録している時は別の学生はそのコンテンツの待ち状態になってしまうからである。実際には、待ちが発生したら一定時間後、リトライを繰り返す。このリトライが 3 回発生した時点でコンフリクトエラーとなる。

この件については、メールアドレスのコンテンツ内にさらに学生ごとのコンテンツを作成し、そこに、メールアドレスを登録するようにし、なるべくコンフリクトエラーが発生しないようにした。

その結果、確かにコンフリクトエラーの発生頻度は少なくなったがまったく無くなったわけでは無かった。そこで、リトライの回数を 3 回から 6 回に増やすことで発生頻度をさらに減らすようにした。これは、Zope プロダクトよりも OS 側にある Python ライブラリをカスタマイズした。

6. 3 ロードバランス

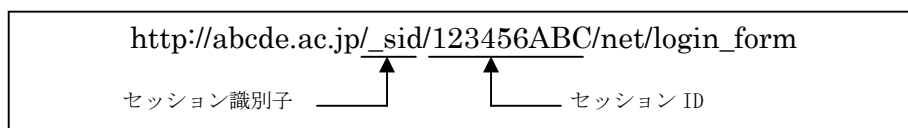
Plone でのセッション維持は、データベースサーバにて行うため、2台ある web アプリケーションサーバが途中で入れ替わっても問題が発生しない設計であったが、テスト中、まれにセッションが切れることがあった。

そこで、負荷分散装置で URL に埋め込まれたセッション ID を使用してセッション維持を行おうとしたが、Plone での URL へのセッション ID の埋め込み方が特殊なのかうまくセッション維持ができなかった (図 10)。やむを得ず、接続元の IP アドレスを使用してセッション維持を行うこととした。

この設定でよいと思われたが携帯サイトのテスト中、NTT DOCOMO の携帯の場合、アクセスのたびに IP アドレスが変わることが判った。また、開発中に大学内のネットワークの更新があり、学内の PC の IP アドレスがグローバルアドレスからプライベートアドレスに変更された。その結果、大学内の PC からポータルサイトにアクセスすると NAT により全て同じ IP アドレスで接続されることになり、負荷分散できない状態になってしまった。

そこで、最終的には次のような解決方法をとった。

携帯サイトアクセス時は常に Web アプリケーションサーバ 1 にアクセスするよう負荷分散装置を設定し、Web サイトについては Plone の設定を変更し、cookie を使うようにし、さらに負荷分散装置にて携帯サイト以外なら、cookie を使用してセッション維持を行うようにした。



(図 10) Plone でのセッション ID の埋め込み

6. 4 パフォーマンスの悪化

システム負荷テスト中に著しくパフォーマンスが悪化する現象が発生した。これについては Oracle の設定を変更することで対応した。

実際に変更した内容は、CURSOR_SHARING パラメータを SIMILAR に変更をした。

通常、VB 等で SQL 文を発行する場合、バインド変数を使用して WHERE 句の値が変更されても実行計画は同じにする事ができるが、Plone の場合、バインド変数が使用できない。しかし、CURSOR_SHARING パラメータを変更する事により一部リテラルが異なる SQL であっても Oracle がリテラル値をバインド変数に置換するため、同じ実行計画を使用するようになる。

その結果、次のようなパフォーマンスの改善が図られた。

テストケース：トップページ表示のリクエスト開始からページ表示までを測定
(50 秒間に 200 リクエスト)

変更前 最小：2.1 秒、最大：67.4 秒、平均：44.7 秒

変更後 最小：0.4 秒、最大：20.3 秒、平均：5.1 秒

7. 稼働状況と今後の拡張

前章での問題を解決し、期日通りリリースした。

また、当ポータルサイトは「**CHUKYO ALBO**」と名付けられた。ちなみに ALBO とは、イタリア語で掲示板の事である。

構築した画面例を以下に示す。(図 11・12・13)



(図 11) ポータルサイトトップページ



(図 12) 携帯サイトトップページ

日付	曜限	教員名	科目名	ALBO 休講ボード 1/1
12/08	木 1	浅野 豊美	政治学B	
12/08	木 2	浅野 豊美	政治学B	
12/08	木 4	三宅 なほみ	認知科学 1	
12/07	金 2	田中 良三	障害児教育 2	
12/07	金 2	寛多 康弘	貿易論	
12/07	金 4	釜田 公良	ゲーム論	
12/10	月 5	境 賢三	英語科教育法 II	
12/11	火 2	山本 茂義	コンピュータ処理論B	
12/11	火 4	山本 茂義	コンピュータ処理論B	
12/11	火 5	山本 茂義	基礎ゼミ	
12/12	水 1	寺岡 寛	中小企業論 II	
12/12	水 2	山本 茂義	コンピュータ処理論B	

(図 13) プラズマディスプレイ (休講ボード)

稼動開始から約1年となるが、稼動直後によくありがちな、顧客との認識の違いによるシステム改修や初期不具合は、まったくと言って良いほど発生していない。あまりにスムーズに稼動したため周りの人に本番稼動が認知されていないほどであった。

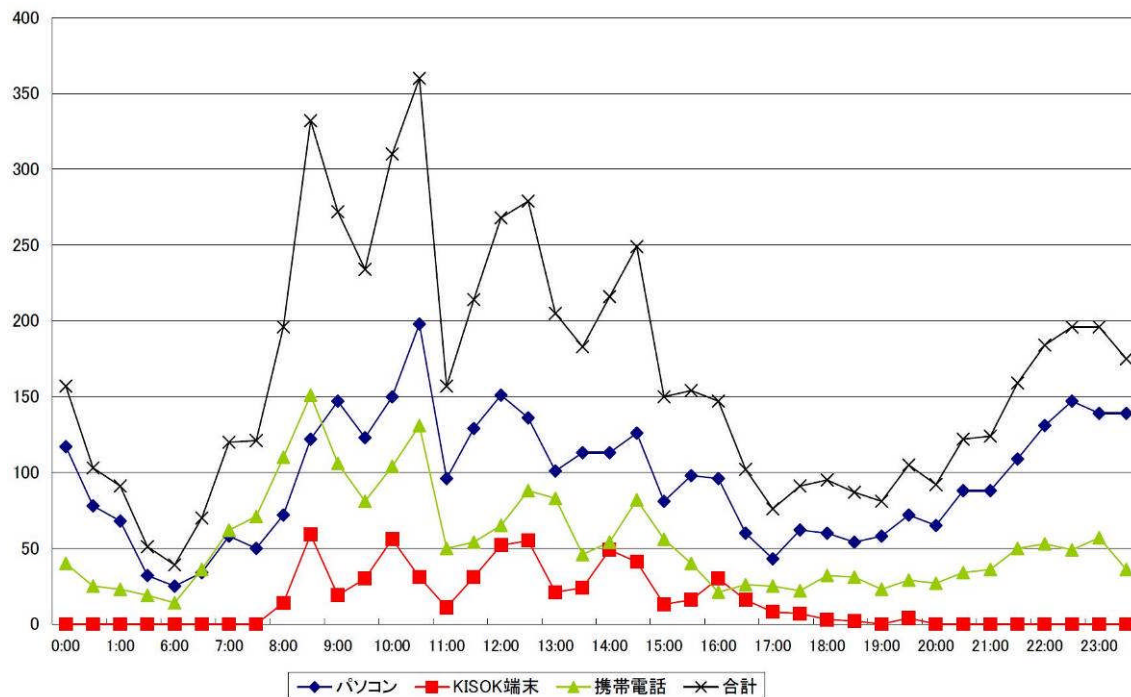
稼動後には学内に設置されていた掲示板も、すべて廃止された。(図14)



(図14) ポータル稼動後の掲示板

学生の利用状況は、2008年7月末までで、12,705人。これは、対象学生13,283人の約95.6%にあたる。1日あたりでは、約8,000人の学生よりアクセスがある。

時間帯毎のログイン数は図15のとおりである。ログイン数のピークが昼間帯にいくつか出てくるが、これは、授業の休憩時間と一致しており、大学向けシステムの特徴が良く現れている。



(図15) 時間帯毎のログイン数

メールアドレスの登録は、稼働後3ヵ月で4,000人が行ったが、これは、大学が予想した人数よりも少なかった。当システムは、登録時のメールアドレスチェックをRFC準拠に行っており、次のルールに該当するアドレスはエラーとしている。

1. 先頭が「ピリオド」・「アンダーバー」・「ハイフン」である。
2. “@” の直前が「ピリオド」・「ハイフン」である。
3. “@” より前に「ピリオド」・「アンダーバー」・「ハイフン」の連続がある。

このことが、最近の学生が好むバラエティ豊かなメールアドレス

(例：y_y.aaa@xx.ne.jp や abc.....@xx.ne.jp 等)

を登録できなくしており、結果として予想を下回ったと想定される。

しかし、これらのアドレスが登録できないことをポータルログインページにあるFAQやお知らせの機能を利用して学生に案内した結果、現在では、約7,700人の学生がメールアドレスの登録がされている。

今後の展開であるが、機能追加の要望があり、来年度にかけて追加開発を予定している。

現行機能を情報の流れでみると、職員⇒学生の単一方向のみであるが、これに学生⇒職員といった逆方向の流れを実現させる機能追加を行う。具体的には、各種申し込み機能やアンケート調査機能である。

また、教員も参照したいとの要望があり、現在、職員のみログインできる画面を教員もログインできるように変更する予定である。

ハードウェアについても、Webサーバ(携帯専用)・KIOSK端末の増設の話もいただいている。

参考までに、学生サイトログインページのURLを示す。ログインページにはFAQの参照や利用マニュアルのダウンロードもできる。

<https://albo.chukyo-u.ac.jp>

8. おわりに

今回の構築が成功した要因を以下に示す。

- ・要求仕様の明確化

大学からの要求仕様がはっきりしており、打ち合せを進めていってもこの内容から大きく逸脱することがほとんどなかった。

- ・デモ環境の構築

開発時、いち早くデモ環境を構築し、大学から自由に参照できるようにしたことにより、開発途中でも随時、大学から不具合等をご指摘いただけた。つまり、知らない間に大学職員もデバッガになっていた。

- ・提出資料の確認

こちらから提出する資料に対して大学職員が必ず確認をしていただけた。

たとえば数百ページのシステム設計書を提出したがかなり細かいところまで確認をしていただけた。

- ・業務内容の熟知

長年、当社が基幹事務システムの運用をしており、業務内容及び各データ内容について熟知していた。今回、基本データ（学生・履修等）は基幹事務システムより取得するが各データの特異性についてもよく理解しており、システムを構築する上で戸惑うことがなかった。

- ・信頼性・技術力のある協力会社の確保

提案段階から協力会社と共同で作業を行ったが、協力していただいた会社の技術力が高く、Plone・Zope についての調査、カスタマイズ等が極めてスムーズに実施できた。また、設計段階でもこちらが気づかないような細かい点も指摘を受けられた。

- ・オープンソースソフトウェアを利用するに必要な知識修得

欧米では非常に実績のあるオープンソースソフトウェアを使用したが、幸い、日本語の詳しい書籍（Plone・Python 共に）が出版されており、基本的知識を得ることができた。

使用したオープンソースソフトウェアはすべて Python 言語で書かれており、Python を修得する事により、さまざまなカスタマイズができた。

書籍だけでは解らないことは、Web (<http://plone.org> や <http://plone.jp> など) を使用する事により情報を得ることができた。

今回、オープンソースソフトウェアを使用して開発を行ったが、業務アプリケーションとしてのオープンソースソフトウェアの利用事例が増えてきているといっても、国内ではそれほど事例がないのが現状である。しかし、使用するオープンソースソフトウェアによっては、安価にかつ短期間にシステムが構築できるのも事実である。

留意点としては、オープンソースであるが故に、不具合があった場合、全て自己責任で対処しなければならない。その際、情報収集も Web が中心となるであろう。ともすれば、ソースコードも一から全て追わねばならなくなる。今回採用した Plone は、幸い欧米ではかなり実績のあるプロダクトであったため、不具合はまったく無かった。

当社のように業務パッケージ製品を多く持たない会社においても、適切なオープンソースソフトウェアを選択できれば、十分、パッケージ製品に対抗できるシステムを構築でき得るとの感触を実感できた。更に、文教・公共分野では、「ベンダロックイン」（特定メーカー製品や特定ベンダー技術でのユーザの囲い込み）を脱却しようとする傾向が見られ、オープンソースソフトウェアの利用はこの動きに合致するといえる。

今後のシステム構築において、オープンソースソフトウェアの利用を選択肢に含めることにより、より良いシステム提案ができ、より多くの受注に繋がるとの期待を持つものである。