

# 授業評価システム開発におけるオープンソース・ソフトウェアおよびフリー・ソフトウェアの利用

湯川治敏\*・蔣湧\*・龍昌治\*\*

\* 愛知大学経済学部 \*\* 愛知大学短期大学部

## 1. はじめに

2004年4月、愛知大学学内理事会に於いて授業評価システムプロジェクトの発足が正式に認められた。このプロジェクトは、これまで紙ベースで行ってきた授業評価アンケートを Web ベースで行うためのシステム開発プロジェクトであり、その大きな特徴として中心的な開発メンバーが本学教員3名と学生3名で構成されていた点が挙げられる。また、もう一つの特徴として開発環境および実行環境をできる限りオープンソース・ソフトウェアあるいはフリー・ソフトウェアを利用しているという点である。本稿では2点目の、特に開発環境におけるオープンソース・ソフトウェアおよびフリー・ソフトウェアの利用について、その選定や利用方法、設定等について解説するが、ある種の備忘録的な内容になってしまうであろう事をあらかじめお断りしておく。

## 2. オープンソース・ソフトウェア、フリー・ソフトウェアとは？

「オープンソース・ソフトウェア」という言葉はコンピュータ関連の分野ではすでに定着しつつあるが正確な定義については、Open Source Initiative (OSI) によって1998年に定義された[3]。それによると、「オープンソース」とは単にソースコードを得られるだけでなく、自由な再配布を認める事や個人、グループに対しての差別を行わない事等を含む10項目によって定義されている。しかし、これよりもっと以前、emacsに代表されるGNU（「グニュー」と発音する）プロジェクトの主宰者であるRichard M. Stallmanによって1985年に発足したFree Software Foundation (FSF) によると、「フリー・ソフトウェア」は、ユーザが実行する自由、プログラムを書き換える自由、無料あるいは有料でコピーを再配布する自由などの4項目によって定義している[5]。一見、これらは同じ事のように思われるがRichard M. Stallmanによれば「オープンソース運動にとって、ソフトウェアがオープンソースであるべきか否かという問題は、倫理的な問題ではなく、実用的な問題である」としている。また、「オープンソースは開発の方法論であって、フリーソフトウェアは社会運動である」という他人の言に同調している。一方、FSFによる「フリーソフトウェア」の定義を知らない人々にとって

導入した。実験サーバの導入目的は実行環境にシステムを移行する前のまさしく実験運用と実行環境が本格稼働した後にシステムの修正を行う際、実行環境を止めずにほぼ同じ環境で実験をした後にシステムの入れ替えをする必要があったためである。実験サーバの環境は開発環境のシステムソフトウェア環境と全く同じである。

開発環境および実験環境のシステム・ソフトウェアを以下に示す。

**開発／実験サーバ環境：**

OS：Fedora Core

Web Server：Apache 2.0.52

JSP コンテナ：Tomcat 5.0.28

データベース：PostgreSQL 7.4.6

さらに開発環境では以下の開発用ソフトウェア群もインストールした。

**開発用ソフトウェア：**

統合開発環境（SDK）：Eclipse

データベースツール：phpPgAdmin, PgAdmin

グラフィックライブラリ：JFreeChart

### 3.3 その他開発段階で用いたソフトウェアについて

実行環境、開発環境以外で用いたソフトウェアの中にはグループウェア、負荷テストツールなどがある。

グループウェアは以下に示す<sup>(株)</sup>トップマネジメントサービスおよび<sup>(株)</sup>システム・コンサルタンツが無償提供する LalCooda WIZ を用いた。このグループウェア用に以下のマシンを用意した。

**グループウェア用マシン環境：**

OS：Windows 2003 Server

データベース：mysql 3.23.58

スクリプト言語：php 4.22

ソフトウェアの詳細については後述する。

## 4. システム開発に用いた各ソフトウェアについて

前述した授業評価システム開発環境および開発には直接関与しないものの補助的なツールとして使用したそれぞれのソフトウェアについての簡単な紹介と具体的な利用方法、設定、問題点等を解説する。

### 4.1 Red Hat Enterprise Linux 3 (ES)、Fedora Core (オペレーティングシステム)

Linux とは、1991 年に当時フィンランドのヘルシンキ大学在学中であった Linus Torvalds が開発を

開始した OS の中核部分であるカーネルを指す場合が一般的であるが、現在ではそのほかのソフトウェアも併せた OS 全体を指す場合もある。OS 全体を指す場合は GNU/Linux [9] と呼ぶべきであると FSF の Rechart M. Stallman らは主張している。この GNU/Linux にさらに様々なソフトウェアやインストールや設定を容易にするツールなども含めて配布されているものをディストリビューションパッケージと呼び、そのひとつが Red Hat 社が提供する Red Hat Enterprise Linux である。Linux はそもそもフリー・ソフトウェアとして開発されたものであるが、Red Hat Enterprise Linux の場合は保守契約を結ぶことでセキュリティパッチの配布等の情報提供を含むサポートを受けることができる。その必要のない場合は同じ Red Hat 社からサポートは無い代わりに無償で利用できる Fedora Core と呼ばれるディストリビューションを利用可能である。3.1 ですべに述べたように授業評価システムの Web サーバおよびデータベースサーバ用には Red Hat Enterprise Linux を用い、開発・実験環境では Fedora Core を用いた。これは当初から本格運用開始時にはシステムのメンテナンス等は外部業者への委託を検討していたこと。そのためには最もポピュラーなものうちのひとつであり、委託する業者にとってもメンテナンスの実績があるディストリビューションであることが望ましいことなどから総合的に判断した。また、開発・実験サーバには本格運用とできるだけ同じディストリビューションで開発あるいは実験を行うことが望ましいと考えられたため、さらにはコストの削減も考慮し、同じ Red Hat 社と関係のある Fedora プロジェクト [8] で開発している Fedora Core を選んだ。

環境設定としては日本語文字コードを EUC とした。これは、開発開始当時でもそろそろ UTF が使われ始めてきてはいるがまだまだ EUC で作られるシステムが多く、開発コンサルティングを依頼したサーバ納入兼システム保守業者でも UTF を使ったシステム開発はその当時は行っていないとのアドバイスによる。従って、システム内の文字コードはデータベースも含め、全て EUC とした。しかし、開発環境である Fedora Core のデフォルト漢字コードは UTF であり、設定変更をしないままインストールし、そのまま開発を行ったために特にグラフィックスでの日本語表示等でかなり不都合が生じてしまった。このような事を避けるためにも実行環境と開発環境では同じ環境となるよう、特に日本語文字コードについては開発初期段階に於いてどの文字コードを使用していくかについて慎重に選ぶ必要があることを痛感した。

## 4.2 Apache (Web サービスソフトウェア)

Apache Software Foundation (ASF、Apache ソフトウェア財団) [6] という非営利団体のプロジェクトで開発された Web サービスを行うソフトウェアで、現在世界で最も多く稼働している Web サーバソフトウェアといわれている。ほとんどの Linux ディストリビューションにはデフォルトで組み込まれており、Linux と同じくオープンソースの代表的なソフトウェアのひとつである。Red Hat Enterprise Linux および Fedora Core のディストリビューションにも組み込まれているため、OS インストール時にこのソフトウェアも併せてインストールし、サーバ管理ツールなどでソフトウェアを起動するのみで Web サービスを簡単に開始できる。ただし、授業評価システムでは Java 言語をベースにしたサーバアプリケーションを開発するため、デフォルトの設定のみでは十分ではなく、次の

Tomcat と連携させた Web サービスの運用が必要である。Apache の詳細は Tomcat と併せて次節で解説する。

### 4.3 Tomcat (JSP (Java Server Pages) コンテナ)

#### 4.3.1 Tomcat とは？

Tomcat は前述の ASF のサブプロジェクトである Jakarta プロジェクトで開発された JSP (Java Server Pages) コンテナであり、Apache 同様オープンソースである。通常の Web サービスの場合はクライアントからのリクエストに対して Web サーバ上にある html ファイルを http プロトコルに従ってレスポンスとして返すだけである。このような Web ページを静的なページという。つまり、どのクライアントからいつ見ても同じページが表示されるという意味であり、このような場合には Apache のみで十分である。しかし、最近の Web ページはクライアントからある情報を入力してもらい、その情報に応じてレスポンスを変化させる動的なページを扱う Web サイトが非常に多い。いわゆる Web アプリケーションが非常に普及しているが、これを実現する方法には大きく分けてクライアントサイドプログラミングとサーバサイドプログラミングがある。前者は JavaScript などのスクリプトを html のファイルに直接埋め込み、それをクライアントの Web ブラウザが解釈して動的なページを実現する方法である。この方法は非常に簡単に実現でき、また、Web サーバとしても Apache 単体で実現可能である。しかし、データベースと連携したアプリケーションや複雑な処理を行う大規模プログラミングには不向きである。一方、後者はクライアントからの情報を Web サーバで受け取り、サーバ内部で処理を行った後にクライアントに加工した html ファイルを返す方式であり、PHP や Perl などのスクリプト言語を用いた CGI (Common Gateway Interface) などがこれまで用いられてきた。しかし、CGI の利用を許可する事によりシステムのセキュリティが脆弱になるという指摘や CGI はひとつのリクエストに対してひとつのプロセスを起動するので同時アクセスが多数あると想定される大規模サイトには不向きであるとされている。また、PHP や Perl などのスクリプト言語は本来大規模プログラミングには不向きである。これに対して Tomcat はプログラミング言語として Java を使用できる為、完全なオブジェクト指向プログラミングが可能である事が特徴である。つまり、大規模プログラミングにも耐えられ、また、クライアントからのリクエストに対してはプロセスよりも負荷の軽いスレッドが起動されるだけなのでシステムの負荷も小さく、大規模システムに適している。同じような製品に Microsoft の ASP があるが ASP の場合はプラットフォームが Windows に限られるだけでなく、Web サーバも IIS でなければならないという制約がある。これに対して Tomcat は Linux はもちろん、Windows でも稼働するプラットフォーム非依存であり、Web サーバも Apache 以外のサーバソフトウェアとも連携可能である。また、Tomcat 自体も Web サーバとしての機能を持ち合わせている。さらに大きな特徴は Tomcat は無償利用が可能な点である。今回開発した授業評価システムのシステムとしての一番の特徴はこの Tomcat を用いた点であろう。

#### 4.3.2 Apache との連携と各ソフトウェアのバージョン

前節で述べたとおり、Tomcat は非常に優れたソフトウェアではあるが単体で Web サーバを兼任さ

せた場合には Apache と比較するとその処理能力は十分とは言えない。従って、大規模サイトでは Apache と Tomcat を連携させて動作させる事が一般的である。つまり、通常のリクエストの場合には Apache が対応し、JSP として動作する必要があるときは Tomcat がリクエストを受け付けて処理をする。この連携を行うソフトウェアのことを jk-Connector と呼んでおり Tomcat と併せて Jakarta プロジェクトが提供している。Apache, Tomcat, jk-Connector とも様々なバージョンがあるがこのプロジェクトで用いた各ソフトウェアのバージョンは以下の通りである。また、開発当初から Tomcat の最新バージョンは 5.5.x 系であったが、対応する Java が J2SE5.0 であり、開発スタッフが J2SE1.4.2 を使っていたことから Tomcat のバージョンをあわせた。

Apache : 2.0.46

Tomcat : 5.0.28

jk-Connector : 1.2.6

Java : j2sdk-1.4.2

#### 4.3.2 Apache-Tomcat 連携とその設定

Apache-Tomcat 連携をするために Web サーバ上で以下の設定を行った。

- 1) /etc/httpd/modules ディレクトリに jk-connector のバイナリ・リリースである mod\_jk.so をインストール
- 2) /usr/lib/httpd/modules/mod\_jk.so から上記 mod\_jk.so にシンボリックリンクを張る
- 3) /etc/httpd/conf/httpd.conf の 208 行目に

```
LoadModule jk_module modules/mod_jk.so を追加
```

- 4) /etc/httpd/conf/httpd.conf の最下部に下記 IfModule ブロックを追加

```
<IfModule mod_jk.c>  
    JkWorkersFile "/etc/httpd/conf/workers.properties"  
    JkLogFile logs/mod_jk.log  
    (セキュリティ保護の為、中略)  
</IfModule>
```

- 5) /etc/httpd/conf に下記内容の workers.properties ファイルを追加

```
worker.list=ajp13  
worker.ajp13.lbfactor=1  
worker.ajp13.port=8009  
worker.ajp13.host=wbces  
worker.ajp13.type=ajp13
```

上記設定の後、Tomcat および Apache を再起動すれば Apache-Tomcat 連携が実現できる。

#### 4.4 PostgreSQL (データベース)

オープンソース・データベースの代表格であり、PostgreSQL[4](ポストグレエスキューエルまたはポストGRESキューエルと読む)という名称は1997年のバージョン6.0からであるが元をたどれば1980年代のUCB(カリフォルニア大学バークレー校)におけるIngresというリレーショナル・データベース研究プロジェクトが発端となって現在に至っている。現在の最新バージョンは8であり、UNIX系OSだけでなくWindowsでも稼働するシステムとなっているが、開発開始当初のバージョンは7であったため、本稼働システムおよび開発環境に於いてはバージョン7.4.6を利用した。PostgreSQLは無償利用可能であるにもかかわらず商用データベースに匹敵する高機能、高性能で知られており、何百万人分ものデータを扱うG(ギガ)バイトクラスのデータベースも構築されているという報告がある。授業評価システムにおける学生の履修データの件数としては、全学生数×各学生の履修科目数×春・秋学期の計約8万件に達し、アンケートの回答数も最大で同数が見込まれる。様々な実績報告などでかなり複雑かつ大規模なシステムも取り扱いが可能であることを知ってはいたが、実際に開発チーム内でこの規模のデータを扱った経験が無かったことから、サンプルデータを投入し、実際にデータベースからの読み込み、書き込みが問題無くできるかどうかは非常に不安であった。しかし、実際には全く問題なく稼働している。ただ、PostgreSQLに関して次の2点の問題があった。

1. OSにバンドルされているRed Hat版PostgreSQLでは簡単な問い合わせに対してもレスポンスが極端に遅い
2. データベースのダンプ、リストアが正常に機能しない場合があった

1については未だに原因不明である。PostgreSQLは前述のように世界中で使われているオープンソース・データベースであり、数々のLinuxディストリビューションでもバンドルされている場合が多い。Red Hat Enterprise Linux 3(ES)の場合も当初はPostgreSQL7.3.8をベースにしたRed Hat版PostgreSQLを利用するつもりであり、システム保守を委託した外部業者からもメンテナンスやサポートも受けられるRed Hat版を勧められた。しかし、実際にサンプルデータ投入後、VIEWからの簡単な問い合わせに対するレスポンスが返るまでに2分強掛かるという考えられない状況であった。データベースのチューニングを行わなければ最適なパフォーマンスが得られないとの説明を受けたが、いくらデフォルト設定のままだとしても非常に簡単な問い合わせに2分も掛かるのはチューニング以外の要因が考えられた為、今回はRed Hat版ではなく一般的に使われているPostgreSQL7.4.6を新たにインストールすることとした。こちらの場合はパフォーマンスに全く問題はなく、現在も非Red Hat版が稼働中である。

2の問題の詳細は以下の通りである。システム開発中および本稼働を始めた場合もデータベースは定期的にバックアップを行う。PostgreSQLでは具体的にはpg\_dumpというコマンドを用いてデータおよびスキーマをまとめてひとつのファイルとして保存可能である。特に、PostgreSQLではデータ

ベース稼働中でもバックアップのできるホット・バックアップ機能を持っているので非常に便利である。この `pg_dump` は単にバックアップのみに利用するだけではなく、データベースの名称変更や開発マシンから実稼働マシンへデータベースを移動する場合にも頻繁に用いられる。例えば開発段階でスキーマ (TABLE や VIEW、FUNCTION) の変更があった場合にはそれらを反映させるために変更したデータベースをダンプし、移動先でリストアして全く同じ物を再現する。この操作がかなり行われたが、ある時ダンプしたファイルをリストアしても本来あるべき VIEW や FUNCTION が再現されないという現象が発生した。そこで、ダンプしたファイルはテキストファイルであるので内容を調べてみると原因は VIEW や FUNCTION の変更によるものであることが判った。つまり、次節に述べるデータベース・ツールなどによって VIEW や FUNCTION などは非常に簡単に変更でき、その変更もその時点で使われているデータベースでは即反映される。データベースをダンプする際にはそれぞれのスキーマが定義された順にダンプ、リストアされる為、最初の段階では問題ないものの、スキーマが変更された場合はその変更されたスキーマの定義が最後尾、つまり一番最後に定義されたかのようにダンプファイルに記述されてしまう。このことにより、本来の定義順が狂ってしまう。例えばある VIEW b を定義する為に他の VIEW a を参照していたとすると、本来ならばダンプファイルの中では VIEW a、VIEW b の順に定義されていなければならないが、VIEW b の定義後に VIEW a を変更するとダンプファイルの中では VIEW b、VIEW a の順になってしまい、VIEW b を定義する際に必要になる VIEW a が参照できず、結局、VIEW b は未定義のままになってしまう。これは `pg_dump` の仕様なのかバグなのかかわからず、また、コマンドの利用方法が間違っているのかあるいは別の方法があるかも判らなかつた為、解決策としては依存関係が正しく参照されるようダンプファイルを直接編集した後、リストアを行うことで解決した。なお、一度定義順が修正されたダンプファイルをリストアすればその後は通常のダンプ、リストアが行えることを確認した。

今回は問題解決までそれほど時間が掛からず開発に大きな影響を及ぼすことは無かったが、データのダンプ、リストアというデータベースの管理にとっては最も重要な機能のひとつが意図したように動作しない状況に陥った際、オープンソース・ソフトウェアの場合は基本的には自らが問題解決に取り組まざるを得ない。これが商用データベースであればメーカーあるいは販売店等に直接問い合わせることができるのだが、オープンソースの場合はメーリングリストや BBS、提供サイトの FAQ などを参考にするしかない。多分、ほとんどの問題がこのような方法で解決可能であるだろうが、時間的な制約がある場合などはすぐに問題解決ができるかどうかは非常に大きなファクターとなるだろう。今回の問題により改めてオープンソース利用の利点だけでなく、利用の際の注意点、問題解決手段の確保の必要性を学んだ。

#### 4.5 PgAdmin, phpPgAdmin (データベース操作ツール)

PostgreSQL はオープンソースであるにも関わらず、非常に多機能、高性能である事はすでに紹介したが、PostgreSQL 単体だけでは CUI (キャラクター・ユーザ・インターフェイス) のみしか提供していない。つまり、TABLE や VIEW の定義、データ入力などはコマンドラインで入力するか、

SQL文を書いたファイルを読み込ませることによって行われる。しかし、大規模なシステム構築にはGUI（グラフィカル・ユーザ・インターフェイス）によるデータベース操作ツールが望まれる。幸い、PostgreSQLにはいくつかそのようなソフトウェアが存在し、本プロジェクトにおいてもPgAdmin および phpPgAdmin の双方を利用した。どちらも似たような操作性であるが一長一短があるので両者について紹介する。

PgAdmin は PostgreSQL の開発チームとは直接関係ないようであるが、phpPgAdmin と共に PostgreSQL 本家の Web サイトに Graphical Clients として紹介されている。現在のバージョンは PgAdmin III であり、Linux を含む UNIX 系 OS だけでなく、Windows、MacOSX でも動作するマルチプラットフォーム対応のアプリケーションである。授業評価プロジェクトチームでも Linux および MacOSX 上で動作させデータベース操作を行った。データの入力や編集作業も GUI ででき、また、スキーマの定義を確認し、変更する事が非常に簡単にできる事が特徴である。ただ、大きな欠点は CSV (Comma Separated Values) ファイルなど外部ファイル読み込み機能が無いことである。これに対して phpPgAdmin は CSV ファイルの読み込み、書き出し機能がついており、Excel ファイル等からのデータ移行が非常に楽であった。ただし、phpPgAdmin は PgAdminIII の様な単独のアプリケーションではなく、php による Web アプリケーションとしてインプリメントされているため、実行するにはあるマシンで Apache などの Web サーバを動かす、スクリプト言語 php を利用できる環境が必要である。これは単独のアプリケーションである PgAdmin と比較すると大きな欠点であるかのようにも思えるがネットワーク上のどこかのマシンで phpPgAdmin を実行可能であり、そのマシンへのログインが可能であればリモートで利用することが可能である。例えば、授業評価のデータベースサーバは学内マシンからのアクセスしか受け付けない設定にしてあるので、学内のどこかのサーバで phpPgAdmin を動かす事により学外からデータベースサーバにアクセスすることも可能となる。同様に PgAdmin ではネットワーク経由でのサーバ接続も可能であるが授業評価データベースサーバは学外からの直接のアクセスを許可していないので学外からの接続はできない。

このようにこの2つのツールは相互補完的な機能および、長所、短所があるため両ソフトウェアを使い分ける事で効率よく作業を行うことができた。

#### 4.6 Eclipse (統合開発環境)

Eclipse は IDE (Integrated Development Environment、統合開発環境) と呼ばれるアプリケーション開発ツールのプラットフォームであり、2001年11月にIBM社がオープンソース・コミュニティ、のちの Eclipse Foundation[7] に寄贈したことからオープンソースとしての開発が始まった。当初からこのソフトウェアには4000万ドル相当の価値があるとされてきたが、プログラマー、特にJava開発者の間で急速に広がり、Java開発のデファクト・スタンダードと言っても過言ではない存在となった。基本的な機能だけ挙げてもJavaコンパイラ、デバッガ、Antビルドツール、CVSツールなどである。また、様々な機能をプラグインの形で拡張可能であり、Eclipseのプラグインのみを紹介するWebサイトがある程、無数のプラグインが世の中に存在する。もちろんその中には有償のもの

も含まれるが今回のシステム開発では無償利用可能な Tomcat プラグインを用いた。このプラグインを用いることにより、開発マシン上では Apache - Tomcat の連携をする必要が無く、必要な設定さえすれば Eclipse の画面から Tomcat を単独で起動、再起動、停止を操作できる。Tomcat で動作するアプリケーションをコーディングしてはブラウザで確認、Eclipse に戻って修正後、ブラウザで再確認、という作業が非常に効率良く行えた。また、デバッガも非常に使いやすく、デバッグ効率も非常に良かったと思われる。もし、Eclipse を利用せずと同じシステムを開発しなければならないとしたら開発に掛かる期間は多分 1.5 倍から 2 倍程度に延びていたのではないかとと思われるくらい開発効率向上に貢献したソフトウェアである。ただ、あまりに高機能すぎてデフォルトで提供されているにもかかわらず使い切れなかった機能がある。そのひとつが CVS (Concurrent Versions System) である。これはプログラムやドキュメントのバージョン管理を行う機能であり、いつ誰がどのように変更したかを逐一記録し、「どの時点のプログラムやドキュメントに戻したい」などの要求に応える事ができる。本格的なチーム開発には欠かせない機能である。今回のシステム開発ではエースプログラマがほとんどひとりでコーディングを担当していたため、大きな混乱は無かったが、それでもコードの修正を同時期に複数人が担当することもあったため、本来であれば CVS の機能を用いてコードのバージョン管理をすべきであった。これが今回の Eclipse 利用に関する最大の反省点である。

#### 4.7 JFreeChart (Java グラフィック・クラス・ライブラリ)

JFreeChart[10] はフリーの Java グラフィックス・クラス・ライブラリであり、2D および 3D の非常にきれいなグラフを簡単に生成することができる。様々なグラフを PNG 形式や JPEG 形式で保存することが可能であり、また、JSP や Servlet からの利用も可能である。授業評価システムで用いるグラフは積み上げ棒グラフと、棒グラフ+折れ線グラフの 2 種類のみであり非常に単純であったが、それらを awt などの Java のグラフィックス・ライブラリを用いて直接書いていたとしたらグラフィックスの部分に非常に多くの時間を割かなければならなかったであろうし、時間を掛けたとしても必ずしも良い物ができるとは限らない。今回、かなり早い段階でこの JFreeChart を見つけることができたことは開発効率の向上に大変役になったと思われる。Java がオブジェクト指向であり、コンポーネントの組み合わせにより現存するソフトウェア資源を有効に活用できることを身をもって体験できたソフトウェアである。

#### 4.8 LalCooda WIZ (グループウェア)

システム開発に直接関係するソフトウェアではないが、プロジェクト全体として非常に有効利用したソフトウェアのひとつがこの LalCooda WIZ[12] である。このソフトウェアは(株)トップマネジメントサービスおよび(株)システム・コンサルタンツが無償提供しているグループウェアであり、以下の機能を提供している。

- ・ファイル共有
- ・スケジュール

- 伝言メモ
- 掲示板
- 回覧板
- 施設予約
- アドレス帳

このうち開発プロジェクトで特に頻繁に用いたのがファイル共有である。プロジェクト構成メンバーの都合上、全員が同じ場所に集まれる機会は概ね1週間に1度だけであった。従って、会わない間の成果物の共有、議事録の配信、事務的な文書のやりとりなど、プロジェクト全体の進行にとって非常に重要な役割を果たした。

#### 4.9 Mailman (メーリングリスト管理システム)

同じような機能を持つソフトウェアにメーリングリストがある。プロジェクトが発足してから2006年1月で約1年半になるが、この間、プロジェクトメンバー内での電子メールのやり取りは1200件を超える。1通のメールを複数人に送るという操作は些細な事かもしれないが、これだけの数の電子メールによって情報共有と情報伝達が必要だった事を考えるとメーリングリストが提供するグループ内への一括送信という機能は非常に大きな省力化につながったのではないかと考えられる。このプロジェクトで用いたメーリングリスト管理システムはGNUプロジェクトで開発されたMailman[2]でありメーリングリストサーバは大学が管理しているサーバを利用させて頂いた。

#### 4.10 JMeter (負荷テストツール)

前節のグループウェアと同様、システム開発に直接的には関係しなかったものの、実際にシステムを稼働させる前に必ず行うべきテストのひとつが負荷テストである。授業評価システムは本学学部生および短大生全員を対象としているため、ユーザ数は約1万人である。これらのユーザが一斉にシステムを使う可能性はほとんど考えられないが、50人規模のコンピュータ実習室が3キャンパス併せて9存在するので、その1/3の約150人程度の同時アクセスは可能性として否定できない。実際にそれだけの人数を使って人海戦術による負荷テストも考えられるが、時間と労力のコストが掛かりすぎる。最終的には80名同時アクセスを試行し、問題なく接続できたが、その準備段階として負荷テストツールによって最大150同時アクセスを実験し、認証、ログイン後の表示まで問題なく動作することを確認した。負荷テストツールとしてはApacheのサブプロジェクトであるJakartaプロジェクト(Tomcatと同じである)で開発されたJMeter[1]を用いた。JMeterはJavaアプリケーションとしてインプリメントされているため、プラットフォームを選ばない事、またApache、Tomcatとともにオープンソースであることが特徴である。負荷テストの設定、実行はGUI画面で行うことができ、結果もリアルタイムでグラフ化されるなど非常に使い易く作られている。実際にどれくらいの同時アクセスに耐えることができるのか?あるいは特定の同時アクセス数の場合のレスポンスタイムはどれくらいになるか?など最終的な人海戦術を行う前にある程度システムの信頼性を確認できた事は有益

であった。

#### 4.11 iモード HTML (携帯電話シミュレータ)

授業評価システムではアンケートに回答できる機会をできる限り増やそうとする意図から様々な工夫を行った。そのひとつが携帯電話からの回答を可能にしたことである。この開発途中では実際に各携帯キャリアの電話機を用いてアクセステストを行ったが、それ以前の画面表示テストなどでは携帯電話シミュレータを用いて画面設計や修正などを行った。

## 5. まとめ

授業評価システムプロジェクトが始動して約1年半が経過した。実際には現在も修正を要する機能、画面等が存在し、プロジェクトからなかなか抜けきれない状態が続いているが、今後もこれまで紹介してきたオープンソース・ソフトウェア、フリー・ソフトウェアを用いての修正作業が続くことであろう。時間の経過とともに、それらのソフトウェアが洗練され、ますます、使いやすくなっていくことを期待するとともに、このプロジェクトで得た経験を今後の研究、教育活動に生かしていきたい。

#### 引用・参考文献、関連 Web サイト

- [1] Apache JMeter, <http://jakarta.apache.org/jmeter/>.
- [2] Mailman, the GNU Mailing List Manager, <http://www.list.org/>.
- [3] Open Source Initiative, <http://www.opensource.org/>.
- [4] PostgreSQL Global Development Group, <http://www.postgresql.org/>.
- [5] Richard M. Stallman、長尾高弘訳、フリー・ソフトウェアと自由な社会 : Richard M. Stallman エッセイ集、株式会社アスキー、2003。
- [6] The Apache Software Foundation, <http://www.apache.org/>.
- [7] The Eclipse Foundation, <http://www.eclipse.org/>
- [8] The Fedora Project, <http://fedora.redhat.com/>.
- [9] The GNU Operating System, <http://www.gnu.org/>.
- [10] The JFree software projects, <http://www.jfree.org/>.
- [11] The Jakarta Project, <http://jakarta.apache.org/>.
- [12] ウェブグループウェア La!cooda WIZ, <http://wiz.syscon.co.jp/>