オープンソースへの取り組みと展開（後編）

段木亮一¹

著者抄録：本稿は、2回にわたりオープンソースの実践的導入とその展開方法について解説している。前編（Vol.46, No.12, p.785-796）は、採用が活発になりつつあるオープンソースを概観し、導入の利点、また注意すべき点に触れ、各種サーバの構築方法について解説し、当社の新ネットワーク/サーバ群設計・構築事例をベースに、インターネット向け、および複数拠点間通信を含むイントラネット向けサービス・サーバ群の構築手順を紹介した。後編は、パッケージ導入・活用から独自システム開発に目を向ける。まず、一般的なLAPPシステム開発に必要なパッケージ群の導入とそれらの環境設定、アプリケーション開発環境の整備と履歴管理方法を取り上げ、そして具体的な独自開発アプリケーションとして、経営サポート手法のひとつとして注目されるナレッジ・マネジメント・システムを紹介する。

キーワード：オープンソース、Linux、システム環境設定、アプリケーション開発環境、ナレッジ・マネジメント、創発

Practical use of the open source and taking the new turn

DANGI Ryoichi¹

Author Abstract: This article is the second one dealing with practical introduction of open sources and how it should be deployed. The first article outlined open sources which increasingly become used frequently, advantages of the introduction, and points to be attended in introduction. Besides it covered how to develop various servers. Based on design and develop case of new network/servers in the company the author belongs to the author has already described procedures of develop service servers for the Internet as well as for intranet including communications among plural numbers of nodes. This article, the second one is concerned with introduction and use of packages, and development of a customized system. First the author describes introduction of packages necessary for development of general LAPP system, the environment setup, arrangement of application development environment and the system operating management. Then the author introduces knowledge management system as concrete development application, which is highly paid attention as a management support method.

Key words: Open Source, Linux, system setup, application development environment, knowledge management, emergence

原稿受理 (2004-06-14)

(情報管理 47 (5): 321-337)

1. はじめに

オープンソースの活用は、投資費用の削減と、RAPS^{注1}としての高安定性・高性能・高セキュリティ・高品質、そして豊富なネットワーク機能との両立を目的とするだけではなく、オープンソース本来の「ソースコードが一般に公開され、自由に改変

できる」点に着目すべきである。

前編では、活用の視点でオープンソースを概観し、「User Own Risk」にかかわるいくつかの課題を取り上げ、オープンソースを用いたシステム構築・運用の是非をTCOも含め、効果とリスクとのバランスの事前評価・判断の下に実施することが肝要であることを指摘した。

¹ 川鉄テクノリサーチ株式会社 知的財産・技術情報事業部（〒260-0835 千葉県千葉市中央区川崎町1 技術研究所別館）
Tel. 043(262)4175 Fax.043(268)5493 E-mail: dangi@ktec.co.jp URL: <http://www.ktec.co.jp>

¹ Patent & Technical Information Business Division, KAWASAKI STEEL Techno-research Corporation (1, Kawasaki-cho, Chuo-ku, Chiba-Shi, Chiba, 260-0835)

これに対し、「ソースコードの公開と自由な改変」の視点では、オープンソース活用の本来の意義が見えてくる。後編ではオープンソースのこの特徴に着目し、オープンソースの活用展開について理解を深めていくこととする。

2. オープンソースによるシステム開発

ユーザは、パッケージ間の連携を取りつつ、独自アプリケーションの仕組みを構築していくことが要求される。また、オープンソースは、パッケージごとにおのおのユーザ会等^{1),2),3),4)}で独立に開発・提供されており、パッケージ間の連携は必ずしも保証されるものではない。一方、ユーザ側は、独自システム構築の必要性から、ソースプログラムの改変を避けて通ることができないケースも多い。これらの場面でオープンソースを活用することの利点が見えてくる。一方、ソースプログラムを解析することで、優れたプログラミング技法を習得する手段としても活用できる。

後編では、上記、オープンソースゆえの特徴を活かした活用展開について解説する。

2.1. システム構成

オープンソース・ベースの独自開発システムとして、規模の大小を問わず、LAPP、あるいはLAMPによるWebアプリケーション・システムがさまざまな場面で採用され、普及してきている。

LAPP/LAMPとは、Linux OSをベースとするシステムである。このOS上にWebサーバ・パッケージのひとつであるApacheを搭載し、Webページを制御するサーバ・サイド・スクリプト言語のひとつとしてのPHP組込みを基本とし、おのおのの頭文字からの“LAP”と、さらにPHPと親和性の高いデータベースとしてpostgreSQL、あるいはMySQLを組み合わせたシステム構成おのおのの頭文字“P”，あるいは，“M”より、LAPP/LAMPシステムと呼ばれる。

この2つのデータベースについて、前者はトランザクションに優れるという特徴を有し日本を中心として、また後者は高速性に優れ欧米を中心とし

て、それぞれ広く普及している。

本稿では、前者LAPPを用いたシステム環境構築・設定から、開発環境の構築、そして、独自アプリケーション構築の一例としてKnowledge Managementシステムを取り上げ、オープンソースで構築したアプリケーションの活用例を紹介する。

2.2. LAPPシステム

一般にオープンソースで構築されるLAPPシステムは、図1に示す構成を採る。

図1 LAPPシステム基本構成

同図中のApache⁵⁾はNCSA⁶⁾ (The National Center for Supercomputing Applications) httpd1.3⁷⁾をベースに、1995年に開発が始まった世界で最も広く用いられているWebサーバである。

OpenSSL⁸⁾は、SSLライブラリをベースにEric A. Young/Tim J. Hudsonによって開発された世界標準の暗号通信プロトコルであるTLS v1 (Transport Layer Security) を実装したWebブラウザ・サーバ間の通信セキュリティ・プロトコルである。また、ApacheとOpenSSLとの連携⁹⁾により、https (Hypertext Transfer Protocol Security) 暗号化通信を確保することができる。ただし、図2に示すTCPデータのみを暗号化¹⁰⁾し、受信メッセージの改竄 (かいざん) の有無を検証する手段としてメッセージ認証コード (MAC¹¹⁾: Message Authentication Code) を有する。

図2 通信パケット構成

一方、PHP(Hypertext Preprocessor)¹²⁾は、動的なWebページを構成するWebサーバの拡張機能のひとつであり、HTMLソースプログラム中に記述されるスクリプト言語である。WWWブラウザからの要求に対し、サーバ側でスクリプトを実行し、そ

注1) 基幹系システムのRAS (Reliability, Availability, Serviceability) に対するLinux表現

Reliability: 高信頼性

Availability: 高可用性

Portability: 可搬性

Scalability: 拡張性

の実行結果を返すフレームワークとして機能させるためには、Apacheから起動されるモジュール¹³⁾として動作させるべくPHPのインストール設定を行う必要がある。特に、日本語処理を行うためには、マルチバイトを扱うオプション設定が必要となり、その詳細は後述する。

postgreSQL¹⁴⁾は、カリフォルニア大学バークレー校で開発されたデータベースシステムPOSTGRESをベースとするオブジェクト・リレーショナル・データベース管理システム(ORDBMS)である。ただし、postgreSQLをPHPで制御するには、PHPインストール時にpostgreSQL関数を利用するオプションを設定する必要がある、その詳細は後述する。

以上、OpenSSL-Apache-PHP-postgreSQLの連携により、例えばclientPCからの入力情報がPHPを介してpostgreSQLにDBデータとして格納されるなどの連携機能を実現することができる。

2.3. システム管理ツール

上記LAPPシステムを開発・保守運用していくうえで用いられる、図3に示すシステム管理ツールの例を紹介する。

図3 システム管理ツール (例)

まず、Webサーバのアクセスログを解析するさまざまな解析ツール¹⁵⁾があり、目的に応じて使い分ける必要がある。ここでは、サーバインストール型の解析ツールとしてAnalog¹⁶⁾を用いることとした。これは、あらかじめ設定した条件でシステムが自動記録するログデータに対し解析を行わせることができ、その結果をWeb表示するツールである。例えば、月別/週別/日別/曜日別/時間別アクセス解析、アクセス先ディレクトリ解析、アクセス・ファイル種別解析、検索語解析、閲覧ブラウザ別解析、アクセスOS別解析、リンク元URL別解析など、さまざまな指標で解析することができ、

図4 Analog解析結果表示例

その表示例を図4に示す。

本ツールの導入方法・環境設定方法に関しては後述する。

通常、データベースを管理するには、コマンド・ベースでの操作が要求されるが、GUIにより操作・管理できるツールも用意されている。ここで開発していくLAPPシステムでは、PHP言語で記述されるPostgreSQLをWebブラウザのGUIで管理するべく、phpPgAdmin¹⁷⁾を活用することとし、その操作管理画面の例を図5に示す。

図5 phpPgAdmin活用例

このツールの活用¹⁸⁾により、システム開発・管理工数を大幅に削減する効果を期待できる。本ツールの導入方法・環境設定方法に関しては後述する。

一方、システム開発のプロジェクト推進、開発アプリケーションの版管理などに用いるファイル管理ツールとしてCVS¹⁹⁾ (Concurrent Versions System) がある。

ここではプログラム開発用の動作検証サーバ、開発者ごとの開発プログラム保存・管理のためのファイルサーバ、および開発プログラムの管理ツールとしてのCVS管理システムの連携を行う。次節ではこれらで構成される開発環境について解説する。

2.4. システム開発環境

大規模システムを構築していくうえでは、開発プロジェクトでのSE間連携が必須となる一方、構築済みシステムのエンハンスを行ううえで版管理は必須である。これらのさまざまな開発・管理業務を一元化し、また管理の自動化を行うことで、これらの全工数を最小化することができる。

ここでは、LAPPシステムを開発・管理・運用するための全体システム環境構築と、その導入・設定事例を紹介する。

通常、開発作業には、開発サーバを開発作業員間で共用する仕組みが用いられる。例えば図6に示すシステム間連携である。

図6 システム開発連携

開発サーバをLinuxベースのLAPPシステムとし、開発用プラットフォームは複数のWindowsパソコンを想定している。開発者は、おのおのの開発PC上でプログラムを開発していくことになるが、プログラムの動作検証は、LAPP環境が開発PCには用意

されていないことから、開発サーバ上に動作検証の時点でプログラムをアップロードする必要がある。これを開発サーバ上で起動し、その結果を同Webサーバを介して、開発者のおおのの開発PC上のWebブラウザを介して検証することとなる。

これらの手順を繰り返し実行することで、開発作業が進められる。しかし、これらの作業は、上述一連の操作手順を延々と繰り返す必要があることから煩雑な作業となる。一方、開発中のプログラムの管理も個人ごとに委ねられることとなり、万一、開発PC障害によるデータ消失の危険性も否定できない。

そこで、図7に示す統合開発環境を構築し、開発・管理運用効率を高めることとした。

図7 統合開発環境

同図中、ファイルサーバは前編で解説したsambaサーバにて構成する。開発PCとファイルサーバとは、windowsOS互換でのファイル共有がなされ、開発PC上で開発されたプログラムは、ファイルサーバ上で管理される運用基準と定める。この環境構築と運用基準により、開発者は、ローカルHDDへのアクセスと同一の操作にて、ファイルサーバ上に開発プログラムを保存・管理し、一方ファイルサーバのバックアップ機能により、ファイル/データ等を保証している。これにより、図6に示すような動作検証ごとにFTP通信手段を用いる制約から解放され、操作環境を向上させることができるとともに、開発プログラム・データ保全を行っている。

ただし、CVSを利用するうえでの留意点として、

同サービス・ポートをあらかじめ開放しておくことが必要となる。これは、クライアントPCと開発サーバ間のCVSサービスにこれが用いられるためである。

このTCP2401を開放するには、例えばiptablesを用い、以下のように設定する。

```
# iptables -A INPUT -i $EXTIF -p tcp --sport 2401 -m state
--state ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -o $EXTIF -p tcp --dport 2401 -m state
--state NEW,ESTABLISHED -j ACCEPT
```

一方、ファイルサーバ上で管理される開発ファイル群のディレクトリは、LAPP開発サーバ上のディレクトリからマウントされるように構成することで、ファイルサーバ上のファイル群がLAPP開発サーバ上に存在するように機能させることが可能となる。その結果、開発エンジニアは、プログラムを開発しつつ、同時にその動作機能を検証することができるため、あたかも開発PC上ですべての開発作業を完結することと同一の環境を提供できる。この開発サーバ、ファイルサーバ、および開発PC間連携を図8に示す。

図8 サーバ間連携

まず、同図中、開発サーバのPHP部分に着目する。PHPにて記述される開発プログラムはCVSにて管理され、同管理対象ディレクトリとしてのリポジトリが存在する。プログラム開発者は、この

領域にある編集対象ファイルをチェックアウト（図8の中の①）し、各自の開発PC上での開発作業が開始される。このチェックアウトには、CVSベースのファイル取得手段が用いられる。チェックアウトされたプログラムは、開発が完了するまでの間、担当者ごとにそれぞれの開発PC上で管理され、その開発途上のプログラムはファイルサーバ上に保管される。

一方、プログラムを開発するうえでの開発環境の一例として、PeggyPro²⁰⁾がある。同ツールはCVS連携²¹⁾が可能であり、その編集画面例を図9に示す。

図9 PeggyPro編集画面例

また、同CVS操作の管理例を図10に示す。

図10 CVS管理画面例

開発途上のプログラムの動作を確認するには、ファイルサーバ上のあらかじめプログラム開発領域として定義されるディレクトリに当該プログラム（図8の中の②）を保存する。この保存操作は、

windowsOS上での操作方法を取ることができる。

ファイルサーバ上のプログラム開発領域は、開発サーバ上のPHP動作領域にマウントされ(図8の中の③)、postgreSQLとの連携動作により(図8の中の④)、その実行結果がopenssl経由でWebサーバ(図8の中の⑤)から開発PC上のWebブラウザ上に表示される。

ファイルサーバへの保存(図8の中の②)以降の操作を繰り返し、開発を完了した時点で、完成プログラムを開発サーバ上のCVSリポジトリに対しコミットし、開発を終了する。

一方、以上一連の手順により完成したプログラム群を新しいリリース版として新規稼働させる場合、ライブラリ管理者は開発版テスト環境にて総合試験を行う。最終動作確認が終了した後、正式にリリースされ、以降、版管理が行われる。

上記開発フェーズでは、プロジェクトによる連携開発の形態が採られることが多い。例えば、デザイナーによる画面設計とSEによるプログラム開発との連携作業がある。作業を効率的に進めるうえで、デザイナーに対しては、できるだけPHPコーディング規約²²⁾を意識させず自由にデザインできる環境を提供する必要がある。例えば、図11に示すようなPHPで生成されるデータをHTMLプログラム内で表示するプログラミング・ルールを定める。

```
<html>
<head>
<title><?= $title ?></title>
</head>
<body>
<? = $title ?>
<p>
<table>
<? foreach($name_list as $value) { ?>
<tr>
<th>名前</th>
<td><?= $value ?></td>
</tr>
<? | ?>
</table>
</body>
</html>
```

図11 PHPプログラミング・ルール(例)

上記PHPコーディング規約を適用することで、例えば、Dreamweaver(Macromedia²³⁾)等のWebデザイン・ツールを用いてデザイナーによる開発作業を行わせることが可能となる。

同デザイン・ツールを用いたPHPプログラムの編集例を図12に示す。同図上に示された[PHP]マーキング部分がPHPプログラムとして自動抽出・表示され、デザイナーは画面デザインに専念することができる。

図12 Dreamweaver編集画面例

図13 SE/デザイナー同時開発手順例

コーディング・ルール²⁴⁾と開発環境の整備により、例えば、SEとWebデザイナーとによる連携作業として、図13に示す並行作業が可能となる。さらに、PHPプログラミングもまた、複数のSE間の連携作業も可能であり、同一プログラムの同時編集に対する整合化は、CVSを利用することで可能となる。

3. システム開発環境の構築

以上のシステム開発環境を構築するための手順を以下に解説する。

3.1. opensslの導入

/usr/local/src/にopensslのソース・プログラムをダウンロード²⁵⁾し、root権限にて下記操作を行う。

```
cd /usr/local/src
tar zxvf httpd-2.0.45.tar.gz
cd httpd-2.0.45
./configure --enable-so --enable-ssl
make
make install
```

configureオプションは、インストールディレクトリを「--prefix=/usr/local」と指定するとともに、OpenSSLが使用する設定ファイルのディレクトリを「--opensslldir=/usr/local/openssl」と指定する。また、apacheのモジュールmod_sslをDSO(Dynamic

Shared Object 動的共有オブジェクト) として使用するためのオプションとして「-fPIC」を指定している。

3.2. apache2の導入

本システムに導入するWebサーバであるapache2のソース・プログラムを、`/usr/local/src/`にダウンロード²⁶⁾し、root権限にて下記操作を行う。

```
cd /usr/local/src
tar xzvf openssl-0.9.7b.tar.gz
cd openssl-0.9.7b
./config --prefix=/usr/local --openssldir=/usr/local/openssl -fPIC
make
make install
```

configureオプションは、opensslと同一である。

次に、`/usr/local/apache2/conf/httpd.conf`を編集する。デフォルト設定は、ISO-8859-1で規定される文字コードであることから、以下のとおりoffに設定する。

```
AddDefaultCharset off
```

日本語文字判定コードは、第一優先順位として設定する必要があるため、以下の位置にjaを定義する。

```
LanguagePriority ja en da nl et fr de el it ko no pl pt pt-br ltz ca es sv tw
```

さらに、`/etc/rc.d/init.d/httpd`を以下のとおり定義し、起動スクリプトを作成する。

```
#!/bin/sh
# chkconfig: 345 95 16
# description: Apache2 httpd
# Source function library.
./etc/rc.d/init.d/functions
[ -x /usr/local/apache2/bin/apachectl ] || exit 0
#[ -x /usr/local/apache2/bin/httpd ] || exit 0
RETVAL=0
# See how we were called.
case "$1" in
  start)
    echo -n "Starting httpd: "
    daemon /usr/local/apache2/bin/apachectl $1
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/httpd
    ;;
  stop)
    echo -n "Shutting down httpd: "
    daemon /usr/local/apache2/bin/apachectl $1
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/httpd
    ;;
  status)
    status httpd
    RETVAL=$?
    ;;
  *)
    echo "Usage: $0 {start|stop|status}"
    exit 1
esac
exit $RETVAL
```

起動スクリプトに実行権限を与え、シンボリックリンクを設定した後、下記コマンドにてWebサーバを起動し、client PCよりWebブラウザ上でApacheの起動状況を「http:// (同サーバのIPアドレス)」にアクセスすることで動作を検証する。

```
service httpd start
```

上記動作を確認したうえで、ドキュメントルートの設定を行う。新たに`/var/www/html/`を作成したうえで、`/usr/local/apache2/conf/httpd.conf`を下記のとおり編集し、Apacheを再起動し、設定作業を完了する。

```
#DocumentRoot "/usr/local/apache2/htdocs"
DocumentRoot "/var/www/html"
```

3.3. postgresSQLの導入

postgresSQLデータベース・サーバのソースプログラムを`/usr/local/src/`にダウンロード²⁷⁾し、下記手順でpostgresSQL管理者と同管理ディレクトリを作成したうえで、ダウンロードファイルを解凍する。

```
useradd postgres
passwd postgres
mkdir /usr/local/pgsql
chown postgres.postgres /usr/local/pgsql
cd /usr/local/src
tar xzvf postgresql-7.3.1.tar.gz
```

引き続き下記手順に従って、インストール作業を実行する。

```
Cd /usr/local/src/postgresql-7.3.1
./configure --enable-multibyte=EUC_JP
make all
make install
```

--enable-multibyte=EUC_JPで指定するConfigureオプションにて、マルチバイト文字の標準文字コードを日本語EUCとする。

また、`/home/postgres/.bashrc`を編集し、「# User specific aliases and functions」行の次に以下の記述を追加し、`/usr/local/pgsql`下ディレクトリへのパス²⁸⁾を通す。

```
export PATH="$PATH":/usr/local/pgsql/bin
export POSTGRES_HOME=/usr/local/pgsql
export PGLIB=$POSTGRES_HOME/lib
export PGDATA=$POSTGRES_HOME/data
export MANPATH="$MANPATH":$POSTGRES_HOME/man
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH":$PGLIB
```

標準の文字エンコードに日本語EUCを用い、日本語ソート処理時の不具合を回避のためにロケール不設定を行うため、以下を実行する。

```
initdb --encoding=EUC_JP --no-locale
```

また、`/usr/local/pgsql/data/postgresql.conf`を以下のとおり設定する。


```
silent_mode = on
```

下記手順で起動を確認した後、rootユーザ権限にて自動スクリプト設定を行う。

```
pg_ctl -w start
cp /usr/local/src/postgresql-7*/contrib/start-scripts/linux
                                     /etc/rc.d/init.d/postgres
chmod 755 /etc/rc.d/init.d/postgres
chkconfig --add postgres
```

共有ライブラリ設定²⁹⁾にて、/etc/ld.so.confに以下を追加、保存した後、設定を有効にする。

```
/usr/local/lib
/usr/local/pgsql/lib
```

以上の設定作業を完了した後、postgresユーザ権限でApacheの実行ユーザとしてのnobodyを下記コマンドによりユーザ登録し、すべての設定を完了する。

```
createuser nobody
```

3.4. PHPの導入

PHPプログラミング言語³⁰⁾のソースプログラムを/usr/local/src/にダウンロードし、下記の手順により、rootユーザ権限でインストール作業を実行する。

```
cd /usr/local/src
tar zxvf php-4.3.1.tar.gz
cd php-4.3.1
./configure --with-apxs2=/usr/local/apache2/bin/apxs
--with-pgsql=/usr/local/pgsql --enable-mbstring --enable-mbregex
--enable-zend-multibyte --without-mysql
make
make install
```

Configureオプションについて、--with-apxs2では、Apache 2.0のDSOについて、Apache拡張モジュールであるapxsへのパス指定として、「=/usr/local/apache2/bin/apxs」を、またPGSQLによるpostgreSQLサポートを有効とするためのパス指定として、「--with-pgsql=/usr/local/pgsql」を行う。

また、マルチバイト文字列のサポートを有効とする「--enable-mbstring」の指定、マルチバイト正規表現のサポートを有効とするための「--enable-mbregex」の指定、PHPスクリプトをShift_JISで記述可能にする「--enable-zend-multibyte」の指定、そしてmysql不使用の「--without-mysql」の指定をそれぞれ行う。

引き続き、php.iniの設定で以下を実行する。

```
cp php.ini -dist /usr/local/lib/php.ini
```

すべて行頭のコメントを外した後、次のとおり編集する。

```
output_handler = mb_output_handler
mbstring.language = Japanese
mbstring.internal_encoding = EUC-JP
mbstring.http_input = auto
mbstring.http_output = SJIS
mbstring.encoding_translation = On
mbstring.detect_order = auto
mbstring.substitute_character = none
mbstring.func_overload = 0
mbstring.script_encoding = auto
```

ApacheとPHPとを連携させることを目的に、/usr/local/apache2/conf/httpd.confを次のとおり編集し、作業完了後Apacheを再起動する。

```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule php4_module modules/libphp4.so
#
# AddType allows you to add to or override the MIME configuration
# file mime.types for specific file types.
#
AddType application/x-tar .tgz
AddType image/x-icon .ico
AddType application/x-httpd-php .php
DirectoryIndex index.html index.php
```

以上が正しく設定されていることを確認する手段として、例えば、/var/www/html/index.phpを下記内容で作成する。

```
<? phpinfo(); ?>
```

作業完了後、clientPCブラウザからURLに「http://サーバIP/」を指定しアクセスすることで、図14に示すPHP設定確認画面が表示されることを確認し、作業を終了する。

図14 php設定確認画面

3.5. phpPgAdminの導入

postgreSQLのWeb管理ツールであるphpPgAdminのソースプログラムを/usr/local/src/にダウンロード³¹⁾し、Webアクセスが行われるディレクトリに、下記操作にてファイルを解凍する。

```
cd /var/www/html/
tar /usr/local/src/xzvf phpPgAdmin-3.1.tar.gz
```

あらかじめphpPgAdmin/conf/config.inc.phpファイルのコメントを外した後、以下のとおり編集し、日本語対応に変更する。

```
$conf['default_lang'] = 'japanese';
```

一方、日本語文字コードをEUCで取り扱うことができるよう、phpPgAdmin/lang/japanese.phpを以下のとおり編集する。

```
$lang['aplang'] = '日本語(EUC-JP)';
$lang['appcharset'] = 'EUC-JP';
$lang['aplocale'] = 'ja_JP';
$lang['apdbencoding'] = 'EUC_JP';
```

以上一連の操作を完了した時点で、clientPC側でWebブラウザを起動し、URLに「http://サーバIP/phpPgAdmin/」を入力し、図15に示す画面が起動されることを確認する。

図15 phpPgAdmin起動画面例

3.6. Analogの導入

Webアクセス解析パッケージのひとつとして、Analogを活用することとし、同ソースプログラム

を/usr/local/src/にダウンロード³²⁾する。

引き続きroot権限にて以下の手順にてインストール作業を行う。

```
cd /usr/local/etc
tar xzf /usr/local/src/analog-5.32.tar.gz
mv analog-5.32 analog
cd analog
make
```

設定ファイル「/usr/local/etc/analog/analog.cfg」を以下のとおり編集³³⁾し、保存する。

```
# If you need a LOGFORMAT command (most people don't -- try it
without first!),
# it must go here, above the LOGFILE commands.

LANGUAGE JAPANESE-EUC

FULLDAILY ON
BROWSERSUM ON
REQFLOOR 1r

APACHELOGFORMAT (%h %l %u %t V"%rY" %>s %b
V"%{Referer}iY" V"%{User-Agent}iY")

LOGFILE /usr/local/apache2/logs/access_log
OUTFILE /var/www/html/analog/index.html
HOSTNAME "CILXKTECWK"
```

次に解析結果の出力先ディレクトリの作成、および結果表示用画像ファイル保存操作を以下の手順にて作成する。

```
mkdir /var/www/html/analog
cp -R /usr/local/etc/analog/images/ /var/www/html/analog
```

Apache設定ファイル/usr/local/apache2/conf/httpd.confをroot権限で下記のとおり編集し、Apacheを再起動する。

```
#
#CustomLog logs/access_log common
CustomLog logs/access_log combined
```

以上、一連の設定結果についての動作確認を行うため、下記コマンドを実行する。

```
/usr/local/etc/analog/analog
```

clientPCにてWebブラウザよりURLに「http://サーバIPアドレス/analog」を入力し、図16に示す実行結果が表示されることを確認し、すべての環境設定作業を完了する。

図16 アナログ解析結果例

4. 独自アプリケーションの開発

3章で紹介したLAPP開発環境を用いて、さまざまなアプリケーションの開発、また保守・運用管理を行うことが可能となる。当社では、これらの仕組みを用いてさまざまなシステムを構築し、またエンハンスを行ってきた。

本章は、その開発事例として、新しい経営サポート手段のひとつとして着目され、近年急速に普及しつつある知識管理サポート手段としてのナレッジ・マネジメント・システムを紹介する。

4.1. ナレッジ・マネジメント

ナレッジ・マネジメントは、1990年代後半から米国で体系化された経営理論のひとつであり、個人が保有する知識資産を、いかなる手段により組織として有効に活用するか、その知識管理手法が組織競争力³⁴⁾の源泉であると期待されたことに始まる。

ナレッジ・マネジメントは当初、「成果物(文書)」を知識資産と位置付け、これを組織で共有化し、活用するための情報共有³⁵⁾から出発することとなった。しかし、成果物は本来、「ひと」の頭の中に格納される膨大な情報の中のほんの一部のサンプルに過ぎず、情報資産として本来求められるべきは、普遍性のあるメタナレッジであると指摘されはじめた。つまり、技術革新が著しい昨今、共有資産化された知識・技術の陳腐化問題を解決する汎化知識をメタナレッジと呼び、これを有する「知的プロフェッショナル」と定義される人材を組織の情報資産ととらえる、「know who」³⁶⁾が説かれることとなった。

そして現在、ナレッジ・マネジメントとは、「成

果物(文書)」と「知的プロフェッショナル」との複合連携により、「新たな次元の価値創造を実現する経営手法」である³⁷⁾、と位置付けられるようになった。いわゆる「創発^{注2)}(emergence)」である。

これらの経緯を経て、ナレッジ・マネジメントは、図17に示すような組織レベルでのスパイラル状の知識資産の拡充³⁸⁾として説明され、これをITサポートにより実用化されるに至った。

図17 価値創造概念

ここでは、逐次蓄積されていくナレッジを有効活用し、優れた価値を創造し、そして創発していくためには、蓄積された膨大なナレッジの中から必要なナレッジをいかに適切に引き出すことができるかが重要なポイントとなる。つまり、これらの蓄積ナレッジを瞬時に理解し、活用していく手段が求められることとなる。特に、多次元的に分布し、かつ複雑に入り組んだ膨大な蓄積ナレッジ群の分布状況を瞬時に理解し、適切なナレッジへと誘導するための手段として、可視化が重要³⁹⁾であると筆者は考えた。そこで、分析情報として算出される膨大な数値情報群に対し、これらを視覚

注2) 局所的な相互作用を持ち、もしくは自律的な要素が多数集まる中から、その総和とは質的に異なる高度で複雑な秩序や概念が生じる現象のこと。所与の条件からの予測や意図の範ちゅうを超えた構造変化や創造が誘発されるという意味で「創発」と日本語訳される。

的にとらえることのできる可視化手段を用いることとし、これを「ナレッジマップ」と定義し、実用化することとした。

例えば、ある領域の地形形状が数値化された高さ情報を単にデータ群として提供されても、その地形の表面形状を理解することはできない。等高線表現、あるいは3次元表現などの手段を用いることで、初めて理解できるようになる。この観点から、数値化された知識情報を可視化情報に変換するユーザ・インタフェースを考案した。

一方、図18に示すように、人はさまざまな環境から新しい知識を獲得し、また人脈を広げている。これらの知識・経験の獲得過程を頭脳のニューロン・レベルで観測⁴⁰⁾した時、その蓄積状況はニューロンの発火度⁴¹⁾として表現することができる。

このシステムでは、この発火度に対応する情報を、ナレッジに関する数値情報として上述の可視化手段で表現する。

図18 習得知識の可視化

4.2. ナレッジ・マネジメントの要求機能

4.2.1. 多次元構造／ツリー階層化構造

形式知化された各知的プロフェッショナルのさまざまな暗黙知³⁾を、適切に保存管理し、これらを有効に活用していくには、適切な分類基準の設定が重要となる。

適切な分類とするためには、分類項目を増やす必要がある。その一方、分類数が多くなる程、登録される情報は分散され、活用されにくくなる。

この相反する事象を解決する手段として、情報分類の多次元構造化とツリー構造化が必要と考えた。例えば図19に示すように、例えば3次元構造を定義し、おのおのの次元の分類情報数を m, n, p とすると、

分類定義数： $(m + n + p)$ 種類

にて、

分類数： $(m \times n \times p)$ 種類

の分類が可能となる。

図19 3次元分類構造

一方、上記の各次元について、図20に示すおのおの階層化ツリー構造を定義する。

図20 ツリー分類構造

ツリー構造第1次元の第1階層、第2階層、第3階層の分類項目数をおのおの m_1, m_2, m_3 とすると、第1次元は、

$$\sum_i m_i$$

の分類が可能となる。同様に、第2、第3次元に拡張すると、全体で

$$\sum_i m_i \times \sum_j n_j \times \sum_k p_k$$

もの情報分類が、下記分類定義数にて表現できる。

$$\sum_i m_i + \sum_j n_j + \sum_k p_k$$

例えば、m, n, p, i, j, kをすべて「3」とすると、「81」の分類定義にて、729とおりの分類が可能となる。

つまり、登録情報分類を簡素化しナレッジの登録方法の簡略化を図り、一方、登録済ナレッジの再利用の場面では、多岐にわたる抽出項目の中から、必要な条件を選定することで適切な切り出し手段を提供することができる。

4.2.2. ナレッジマップの定義

ナレッジマップの定義は、それぞれシステムごとに異なる。当社では、前節に述べた蓄積ナレッジの可視化機能をナレッジマップと定義し、図21に示す情報を提供している。

図21 ナレッジマップ表示例

同図は、MDKWH (MultiDimensional Knowledge WareHouse) と定義する多次元・多階層の分類構造上に、複雑に入り組んだナレッジ情報を視覚的に理解しやすく提供するため、2次元情報までブレイクダウンする手法を採用した。このフィルタリング処理により、4.1.1の課題を解決している。

さらに、提示された2次元情報について、概観し(マクロ解析)、あるいは詳細に分析し(ミクロ解析)、または同一条件(同一解像度)で他の事象を観察するなど、さまざまな切り口からの分析手段を提供している。一般にこれらの操作はおのの、「drill up/drill down/drill through」と呼ばれ、その分析概念を図22に示す。

図22 ナレッジマップ分析

同図にて、例えば「組織」次元と「専門分野」次元との2次元マトリックスをナレッジマップとして表現している。ある部署とある専門分野に着目し、これをドリルダウンすることで専門家を特定でき、一方、ドリルアップにより組織全体の専門性を分析することができる。

一方、ナレッジマップ提供の目的は、利用者に適切なナレッジを提供することにある。例えば、解決すべき課題に対して参考となる登録済ナレッジを提供し、あるいは解決すべき課題に対し適切なアドバイスを与える人脈を抽出し、提示する機能である。

前者の場合には、解決すべき課題分類をあらかじめ絞り込み、その中で評価の高いナレッジを見つけるという操作が、また後者は、解決すべき課題分類としての専門分野を指定し、その中で評価の高い人物を見つけ出すという操作が考えられる。

ナレッジマップ上に表示すべき情報は、例えば蓄積ナレッジの評価情報、また人物の評価情報である。このように、蓄積されたナレッジに対してさまざまな分析手段を提供する必要がある。

4.2.3. 評価関数の定義

登録済ナレッジへの評価は、その登録ナレッジ自体への評価とそれを登録した人物への評価とのふたとおりが考えられる。特に人物の専門評価では、組織異動などにより、専門を固定的に定義することは現実的ではなく、また多岐にわたる専門を有することも十分考えられ、これらトランジェントな遷移状況等を適切に表現する必要がある。

そこで本システムでは、評価対象を登録ナレッジとし、かつ人物評価は、当該人物が登録したナレッジの評価情報をすべて抽出し、分析した結果とすることとした。

注3)「暗黙知」とは、人の頭脳に記憶された知識・知恵、さらにはその人脈をも含めた知と定義される。「形式知」とは、暗黙知を第三者が理解し得る手段として提供された知と定義される。

図23 専門評価の表示例

これにより、特定の専門分野の専門家を抽出する機能に留まらず、図23に示す特定人物の専門性分析と、その専門レベルとを同時に分析することも可能となった。これは、自分自身の専門性が他人からどのように評価されているかを知る手段としても活用できることになる。

一方、登録ナレッジごとの評価結果の分析機能を搭載するほか、分野別・専門別・組織別の評価を分析することも可能とした。この機能により、例えば当社はどの技術分野に秀でているか（あるいは、手薄か）、またおのの組織は、それぞれどのような専門家集団で構成されているかなど、解析目的ごとにさまざまな分析手段を提供することができる。すなわち、組織経営上のさまざまな指標を提供する手段ともなり得る。

評価のための分析方法はいくつか存在し、例えば人物の専門性評価を行う場合、ある人物の*i*番目の専門性評価値を*S_i*、また当該人物が登録した各ナレッジに対する評価を*E_{i,j}*とすると、「量」を重要視する時、

$$S_i = \sum_j E_{i,j}$$

で評価し、一方、「質」を重要視する場合には、

$$S_i = \frac{\sum_{j=1}^N E_{i,j}}{N}$$

で評価する。

一方、登録ナレッジ自体の評価を行う場合、例えばコミュニティ *k* の評価指標 *C_k* は、それに紐づくナレッジに対する評価値 *E_{k,j}* の総和として、以下にて算出する。

$$C_k = \sum_j E_{k,j}$$

このように、分析目的ごとに複数の評価関数を独立に定め、それぞれ評価値を算出することにより適切な分析結果を提示することができる。

当社独自開発のナレッジ・マネジメント・システム：K2B (Knowledge to Business)⁴²⁾は、以上の基本機能をベースに構築されている。

4.3. K2Bシステム

前節に述べた基本要件機能を満足するシステムとして開発したナレッジ・マネジメント・システム：K2B (商標登録済 Knowledge to Business) について紹介する。

K2Bは、図24に示す基本3次元構造を持ち、すべての登録済のナレッジは、これらすべての次元情報を有する。これにより、個々のナレッジを「コミュニティ（議論の場）別」、「専門分野（ナレッジの属性）別」、また「所属（目的）別」に切り出し、表示することができる。また、必要な情報をおのの次元からマイページにクリッピングする機能を搭載することで、ショートカットを自由に構成でき、各人が必要とする次元情報、ツリー情報、またナレッジを自由に構成・編集・管理することができる。

図24 基本3次元構造とクリッピング機能

また、メイン画面のひとつであるコミュニティの表示例を図25に示す。

図25 K2Bメイン画面構成例

同図に示すように、メイン画面は3つのサブフレームから構成される。ヘッダは、図24に示す3次元構造で規定されるメイン・メニュー、各メニューに従属する図26に示すサブ・メニュー、および図27に示す管理メニューより構成される。

図26 サブ・メニュー

図27 管理メニュー

また、図28に示すコントロール・フレームは、「コミュニティ」、「専門」（冗長性排除のための分類基準）、「所属」（目的別の分類基準）のおおのの次元ごとに、ツリー階層構造を任意に定義・運用できる。

図28 ツリー構造の定義例

そして、ナレッジ一覧を表示するメイン・フレームの以上3フレームから構成される。次にサブ・メニューに関し、同メニューの「記事」ボタンにより、図28にておのおの指定する条件に合致するナレッジの一覧を表示でき、その表示例を図29に示す。ここでは、ある専門を指定し、これに該当するナレッジ一覧を表示した例を示し、クリップ・マークは添付ファイルを、分数は「評価者数／閲覧者数」を、そしてタイトル、日付、登録者の各情報を表示している。また、既読/未読をリンク文字色の違いで表現している。

23/46	XMLデータのための検索技術	04/01/20	宮田 大介
20/20	データマイニングに基づいた文書分類手法	04/01/19	石川 統一
0/62	言語検索技術	04/01/19	中嶋 智
10/13	個人情報の分散強固保護機構	04/01/18	坂口 伸一
9/33	XMLの統計量を用いた関係データベース設計	04/01/18	田端 武
1/12	VRの医療応用	04/01/18	橋本 隆
3/35	An Event-Driven Approach for Human Crowd Simulation	04/01/15	神門 充
0/42	バイオメトリクス技術の国際標準化	04/01/15	江口 翠
0/12	eSikroad - perspective from research and education networks	04/01/14	森 茂
10/35	UMLによるビジネスモデリング	04/01/13	平尾 幸司
11/35	セマンティックウェブと自然言語処理	04/01/13	児玉 紀子
6/29	文字列の頻度分布による共通パターン	04/01/10	谷木 和明
5/54	多次元データの高次元特徴量アルゴリズム	04/01/10	増山 辰則
0/21	文書分類手法の比較評価	04/01/09	長沼 洋一
3/26	複数尺度の統計的統合	04/01/09	関口 弘幸
3/60	メタデータによるネットワーク制御	04/01/09	栗山 哲夫
0/12	Challenges and Opportunities for the Coming Decade	04/01/08	岸田 学
3/24	適合的汎化による情報検索システム	04/01/08	奥村 和子
10/28	モバイル・インターネット環境構築支援システム	04/01/05	関根 良治
12/35	複合語からの類義語抽出法	04/01/05	池田 智

図29 指定条件下でのナレッジ一覧表示例

サブ・メニュー上の「ナレッジマップ」は、メイン・メニューで定義される3次元の内から2つの次元を指定し、マトリックス表示する機能であり、所属と専門のナレッジマップの表示例を図30に示す。ナレッジマップ上の輝度が高い程、例えば重要なナレッジが蓄積されていることを意味する。

図30 所属-専門ナレッジマップ表示例

さらに、所属をドリルダウンすることで、所属の最下層に位置する所属メンバ情報に行き着く。この操作の結果、人材ごとの専門性を分析するナレッジマップを得ることができ、その表示例を図31に示す。

図31 人-専門ナレッジマップ表示例

以下同様に、専門のドリルアップ/ダウン、コミュニティのドリルアップ/ダウン、また条件（解像度）を固定して、他の情報を分析するためのドリルスルー機能により、蓄積されているナレッジを一定条件の下で、さまざまな分析を可能にしている。

また、これらナレッジマップ上の輝度表現で表示された領域をマウスにてクリックすることで、ダイレクトに図29の例のようなナレッジ一覧を得

ることもできる。

例えば、図30に示すナレッジマップ上にある「生産技術」と「技術研究所」との交点に位置するナレッジ領域をクリックしたと仮定すると、表示されるナレッジ一覧は、

「生産技術」×「技術研究所」
の条件を満足するものとして抽出される。

すなわち、一般のシステムでは、欲しい情報を得るためには検索手段が用いられ、『「生産技術」×「技術研究所」』のAND検索を行う必要がある。これに対しK2Bでは、検索操作を明示的に行わせることなく、これを上記操作で代用することで、自由に必要な情報に絞り込む、あるいは範囲を広げるなどを実現していることが大きな特徴である。

また、図26サブ・メニューの「検索」は、上記機能に対し、従来機能としての明示的に検索を実行する手段を提供している。ここでは、簡易検索、詳細検索、特定分野の専門家検索、そして添付ファイル検索の以上4種類の検索機能を搭載するとともに、各検索条件を保持することで検索の試行操作を可能としている。その検索画面の例として、図32に添付ファイルの検索例を示す。

図32 詳細検索画面

図26サブ・メニューの「クリッピング」は、マイページにブックマークするための機能であり、個々のナレッジをはじめ、コミュニティ/専門/所属/人など、あらゆるK2Bで保持する情報を選択してクリッピングすることができる。この機能により、マイページからダイレクトに必要な情報へアクセスすることが可能となる。このマイページの表示画面例を図33に示す。

図33 マイページ表示例

同図は、専門に分類されるクリッピング情報がメイン・フレームに表示された例である。一方、同図中に示される履歴情報も管理されており、それぞれ「投稿・閲覧」履歴を閲覧できるほか、自分自身の専門性が他人からどう評価されているか、その専門性の評価結果を「専門評価」機能から分析することもできる。

図26サブ・メニューの「配信設定」は、各自が必要とするナレッジ領域に新規ナレッジの追加があった時点で、自動通知される電子メール配信機能を設定できる。

そして同サブ・メニューの「新規投稿」は、新しいナレッジを登録するための機能であり、その登録画面を図34に示す。

図34 新規登録画面

ここでは、タイトル・本文記入・種別選択・緊急度選択・ファイル添付機能のほか、専門分野の分類を定義することができる。この機能により、

それぞれのコミュニティに登録されるナレッジごとに、専門分類の指標でコミュニティを横断的に検索し、該当ナレッジを抽出することを可能としている。同様に、登録時点で投稿者が特定されることから、所属分類をキーとする横断検索も可能である。

図27に示す管理メニューに関し、「ログ」は、システム活用実績に関する管理情報を表示する機能である。また、アクセス・ユーザ別/日付別/コミュニティ別/専門別/所属別の各ランキング表示や、専門家ランキングを表示することができ、専門家のランキング表示例を図35に示す。ここでは、「基盤技術」分野の中の「ナノテクノロジー」に関する専門家をランキング表示した例である。

管理メニューにはその他、ユーザ個人ごとの設定情報（プロフィール）を変更する機能などを用意している。

図35 専門家ランキング表示例

一方、本システムを運用するうえでさまざまな管理機能が必要であり、これらを別途管理システムとして構築・運用している。

5. おわりに

2回にわたり、オープンソースの基本的な活用方法と、システム開発ツールを用いた独自システム構築方法との2つの視点からオープンソースに関する理解を深めてきた。

オープンソースはカスタマイズの容易性という特徴から、今後オープンソース活用ユーザに対し、さらに多くの恩恵をもたらすことが期待される。その一方、オープンソースの普及に伴い、同ベンダ各社は新しいビジネスモデルへの変革が求められる時代を迎える。それは、従来型ビジネスモデルの特徴であるクローズド・ソフトウェアの独占ライセンス販売による収益構造から、オープンソースによりもたらされる新しい付加価値概念ソリューション提供への脱却力学が働くからである。

この創発の可能性が、オープンソース活用の最大のメリットであると位置付け、今後とも真摯（しんし）な姿勢でオープンソース・ビジネスへの展開を模索・推進して行きたい。

参考文献

- 1) Samba Users Group Japan. “samba user group”. (オンライン), 入手先<<http://www.samba.gr.jp/>>, (参照2004-05-26).
- 2) Japan PHP users group. “PHP Japan”. (オンライン), 入手先<<http://www.php.gr.jp/>>, (参照2004-05-26).
- 3) Japan PostgreSQL users group. “PostgreSQL”. (オンライン), 入手先<<http://www.postgresql.jp/>>, (参照2004-05-26).
- 4) Cameron, Jamie. “Webmin”. (online), available from<<http://www.webmin.com/>>, (accessed 2004-05-26).
- 5) Japan apache users group. “Japanized apache server project”. (オンライン), 入手先<<http://www.apache.jp/>>, (参照2004-05-26).
- 6) NCSA. “The National Center for Supercomputing Applications”. (online), available from<<http://www.ncsa.uiuc.edu/>>, (accessed 2004-05-26).
- 7) NCSA httpd development team. “NCSA HTTPD”. (online), available from<<http://www.its.unimelb.edu.au/manuals/ncsa-httpd1.3/>>, (accessed 2004-05-26).
- 8) OpenSSL Project. “OpenSSL”. (online), available from<<http://www.openssl.org/>>, (accessed 2004-05-26).
- 9) JCS. “ApacheのSSL対応化と環境設定”. (オンライン), 入手先<<http://www.atmarkit.co.jp/flinux/rensai/apache04/apache04a.html>>, (参照2004-05-26).
- 10) IPA. “SSL (Secure Sockets Layer)”. (オンライン), 入手先<<http://www.ipa.go.jp/security/awareness/administrator/remote/capter7/9.html>>, (参照2004-05-26).
- 11) atmarkIT. “MAC Message Authentication Code”. (オンライン), 入手先<<http://www.atmarkit.co.jp/aig/02security/mac.html>>, (参照2004-05-26).
- 12) Japan PHP users group. “PHP japan”. (オンライン), 入手先<<http://www.php.gr.jp/>>, (参照2004-05-26).
- 13) Japan apache users group. “動的共有オブジェクト

- (DSO)サポート”. (オンライン), 入手先<<http://httpd.apache.org/docs-2.1/ja/dso.html>>, (参照2004-05-26).
- 14) Japan PostgreSQL users group. “PostgreSQL”. (オンライン), 入手先<<http://www.postgresql.jp/>>, (参照2004-05-26).
 - 15) 鈴木麗子. “アクセスログ解析 / WebMasterへの道”. (オンライン), 入手先<http://www.scollabo.com/banban/wbm/wbm_16.html>, (参照2004-05-26).
 - 16) 日本Analogユーザ会. “Analog”. (オンライン), 入手先<<http://jp.analog.cx/index.html.ja>>, (参照2004-05-26).
 - 17) phpPgAdmin. “What is phpPgAdmin?”. (online), available from <<http://phppgadmin.sourceforge.net/>>, (accessed 2004-05-26).
 - 18) MyHomeWeb. “MyHomeWebでのPostgreSQLご利用方法”. (オンライン), 入手先<<http://www.my-home-web.com/phpPgAdmin.html>>, (参照 2004-05-26).
 - 19) 松原司牧. “CVSを使ってみよう”. (オンライン), 入手先<<http://shimaki-hp.hp.infoseek.co.jp/cvs/book1.html>>, (参照2004-05-26).
 - 20) Anchor systems. “peggy”. (オンライン), 入手先<<http://www.anchor-systems.co.jp/>>, (参照2004-05-26).
 - 21) Peggy. “VSS, CVS, RCSによるバージョン管理”. (オンライン), 入手先<http://www2.noritz.co.jp/anchor/ashp/peggy/cvs/cvs_intro.html>, (参照2004-05-26).
 - 22) FFA PHP Projects. “PHPコーディング規約”. (オンライン), 入手先<<http://www.s55.net/~paka/index.php?PHP%A5%B3%A1%BC%A5%C7%A5%A3%A5%F3%A5%B0%B5%AC%CC%F3>>, (参照 2004-05-26).
 - 23) Macromedia. “STUDIO MX 2004”. (オンライン), 入手先<<http://www.macromedia.com/jp/>>, (参照 2004-05-26).
 - 24) 沼田哲史; 秋吉綾都. “JavaTM言語のコーディング規約”. (オンライン), 入手先<http://www.tcct.zaq.ne.jp/ayato/programming/java/codeconv_jp/CodeConventions.doc.html#16712>, (参照2004-05-26).
 - 25) OpenSSL Project. “OpenSSL”. (online), available from <<http://www.openssl.org/>>, (accessed 2004-05-26).
 - 26) Japan apache users group. “Japanized apache server project”. (オンライン), 入手先<<http://www.apache.jp/>>, (参照2004-05-26).
 - 27) Japan PostgreSQL users group. “PostgreSQLダウンロード情報”. (オンライン), 入手先<<http://www.postgresql.jp/PostgreSQL/download.html>>, (参照2004-05-26).
 - 28) ITmedia Enterprise. “パスを追加したい〜bashrc編〜”. (オンライン), 入手先<<http://www.itmedia.co.jp/help/tips/linux/10284.html>>, (参照2004-05-26).
 - 29) IDG. “「/etc/ld.so.conf」 ファイルの内容”. (オンライン), 入手先<http://www.idg.co.jp/lw/weekly_2/030916/>, (参照2004-05-26).
 - 30) Japan PHP users group. “PHP Japan”. (オンライン), 入手先<<http://www.php.gr.jp/>>, (参照2004-05-26).
 - 31) phpPgAdmin. “What is phpPgAdmin?”. (オンライン), 入手先<<http://phppgadmin.sourceforge.net/>>, (参照2004-05-26).
 - 32) Analog. “Analog: Download”. (オンライン), 入手先<<http://sunsite.sut.ac.jp/pub/archives/WWW/analog/download.html>>, (参照2004-05-26).
 - 33) Analog. “Analog 5.91beta1: はじめに”. (オンライン), 入手先<<http://www.tokyo-kasei.ac.jp/~matsuki/analog/jp5.911/Readme.html>>, (参照2004-05-26).
 - 34) 犬塚 篤. 知識のフローとストック. 信学技報. Vol.103, No.182, 2003, p.11-16.
 - 35) 溝口理一郎. 学習の知的支援. 人工知能AI. Vol.16, 2001, p.29-34.
 - 36) Desouza, K. C. Facilitating tacit knowledge exchange. Commu ACM. Vol.46, No.6, 2003, p.85-88.
 - 37) 大崎紘一. 日本における IT を活用した経営戦略に関する研究. 長崎総科大紀要. Vol.43, No.2, 2003, p.333-338.
 - 38) 大崎紘一. ナレッジマネジメントにおける構造化知識構成法. 経工誌. Vol.53, No.6, 2003, p.437-447.
 - 39) 日本総研. 情報可視化によるドキュメント構造化の調査研究. 情報可視化によるドキュメント構造化の調査研究平成12年度. 2001, p.96.
 - 40) 段木亮一. 誤差逆伝播法による数値計算機能獲得に関する検討. IEICE. NC92-67, 1992, p.53-60.
 - 41) 甘利俊一. ニューラルネットワーク研究の過去・現在・未来. 人工知能学会. Vol.4, No.2, 1989, p.89-96.
 - 42) 段木亮一. “K2B: Knowledge to Business”. (オンライン), 入手先<<http://k2b.kmsys.info/>>, (参照2004-05-26).