

業務運用・管理システムと連携したグループウェアの活用方法

平山かほる(工学部 技術部第一系第一班)

山本 好弘(工学部 技術部第二系第四班)

1 はじめに

技術部再組織化後に発生する依頼業務を行うにあたり、業務運用・管理システムとグループウェア(「La!cooda WIZ」)をどのように連携させるか。また、依頼業務を迅速かつ円滑に行うために、グループウェアをどのように活用すればグループ内の意見交換、作業調整および情報共有を図ることができるかを提案する。

2 グループウェア

現在、技術部内で使用しているグループウェアの動作環境、環境設定などについて紹介する。

今回使用した「La!cooda WIZ」は大企業の部門や中小企業の全体的な活用のためのグループウェアであり、業務管理、コミュニケーション、情報共有、情報蓄積などが可能である。Linux や Windows 2000 などの OS において動作可能なマルチプラットフォーム対応であり、無償(ライセンスフリー)配布されている。また、業務上の必要な情報をデータベ

ースとして蓄積でき、スケジュール管理、行動計画表、会議室予約等の業務効率化のツールを備えている。

2.1 動作環境

「La!cooda WIZ」の提供元による動作環境と今回の動作環境を表 1 に示す。「動作報告されている環境」には報告されていないが、TurboLinux 8 Server において動作を確認している。また、動作前提条件のソフトウェアは全てソースからのインストールではなく、Red Hat 系で使用されている rpm パッケージを使用した。

2.2 設定

ここでは「La!cooda WIZ」を動作させるための環境設定について述べる。

表 1. 「La!cooda WIZ」の提供元による動作環境と今回の動作環境

サーバ環境		「La!cooda WIZ」提供元による動作環境		今回の動作環境
OS		Red Hat Linux 7.x/8.0	Windows NT4.0/2000	TurboLinux 8 Server
動作前提条件 ソフトウェア (推奨)	Webサーバ	Apache 1.3.x	Apache 1.3.x	Apache 1.3.27
	Script言語	PHP 4.2~4.3.1	PHP 4.2~4.3.1	PHP 4.2.3
	データベース	PostgreSQL 7.2.x、7.3.x	MySQL 3.23.x	PostgreSQL 7.2.2
ハードウェア スペック	CPU	PentiumII または Celeron (400MHz 以上推奨)		Pentium III 550 MHz
	Memory	64MB 以上 (256MB 以上推奨)		128 MB

動作報告されている環境

Red Hat Linux 8.0 Apache 1.3.27 PHP 4.2.2 PostgreSQL 7.1.3

Red Hat Linux 9 httpd-2.0.40-21.1 php-4.2.2-17 postgresql-7.3.2-3

Vine 2.6RC1 apache-1.3.27-0v11 php-4.2.3-0v11 postgresql-7.2.3-0v11

Plamo Linux 2.4.19-pc98 Apache 1.3.22 PHP 4.3.1 PostgreSQL 7.1.3

FreeBSD R4.8 Apache 1.3.27 PHP 4.3.1 MySQL 4.010

Debian GNU/Linux 3.0 (Woody) Apache 1.3.26 PHP 4.1.2 PostgreSQL 7.2.1

VineLinux2.6r1(FTP) Apache 1.3.29 + mod_ssl 2.8.16 PHP 4.3.3 PostgreSQL 7.3.4

Apache の設定 (httpd.conf)

PHP4 を使用する設定、「La!cooda WIZ」のトップ画面を認識させる設定

- PHP の MIME タイプを宣言
→ AddType application/x-httpd-php .php
- DirectoryIndex の設定
→ DirectoryIndex index.html index.php

PHP の設定 (php.ini)

「La!cooda WIZ」のプログラムで使用されている PHP スクリプトエンジンの動作とログに関する設定

- スーパーグローバル変数を有効
→ register_globals = On
- 略式表記(<? ?>)の PHP タグを有効
→ short_open_tag = On
- mbstring のコメント記号を外し、HTTP 入力変換を有効
→ mbstring.encoding_translation = On
- 透過的にセッション ID を付加する設定
→ session.use_trans_sid = 1
- php.ini によりエラーログを残す設定
notice 以外のログを出力
→ error_reporting = E_ALL & ~ E_NOTICE
PHP タートアップ時のエラーを非表示
→ display_startup_errors = Off
画面にエラーを非表示
→ display_errors = Off
エラーログ出力
→ log_errors = On
エラーログは /var/log/php
→ error_log = /var/log/php
syslog にログを残さない
→ ;error_log = syslog

PostgreSQL の設定 (pg_hba.conf)

データベースへの接続に関する設定

- クライアント認証
→ local all all trust
→ host all 127.0.0.1 255.255.255.255 trust

PostgreSQL の設定 (postgres.conf)

- デバッグ表示をしない設定
→ silent_mode = on

- TCP/IP 接続を許可する設定

→ tcpip_socket = on

- PostgreSQL への接続状況をモニタリング

→ Stats_command_string = true

→ Stats_row_level = true

→ Stats_block_level = true

「La!cooda WIZ」の画面による設定

- 基本情報の設定
新規ユーザ登録、管理者設定、使用機能の設定など
- 個人情報用データの設定
- 各画面用データの設定

2.3 動作の仕組み

クライアントから HTTP リクエストによる接続要求 (①) があると、サーバ側では Apache の中にロードされている PHP の PostgreSQL モジュールのうち、PostgreSQL への接続関数が呼び出される。libpq を通じて PostgreSQL のデータベースサーバ (postmaster) に接続要求 (②) が送られると、データベースエンジン (postgres) が起動 (③) し、PostgreSQL が利用可能となる。次に何らかの問い合わせ (④) を SQL 文の形で PHP から送信 (⑤) すると、postgres によって処理され、結果 (⑥, ⑦) が返される。この動作はデータベース処理が不要になるまで繰り返される。以上の動作の仕組みを図1に示す。

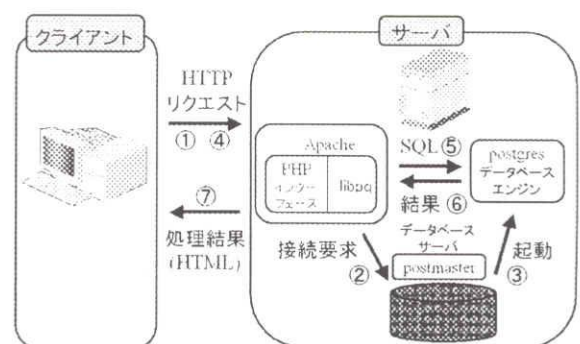
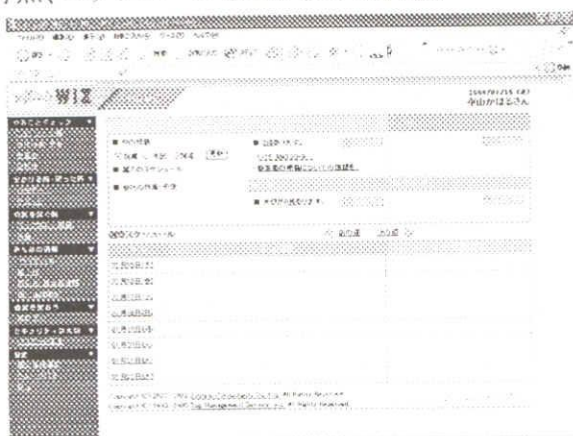


図1. 動作の仕組み

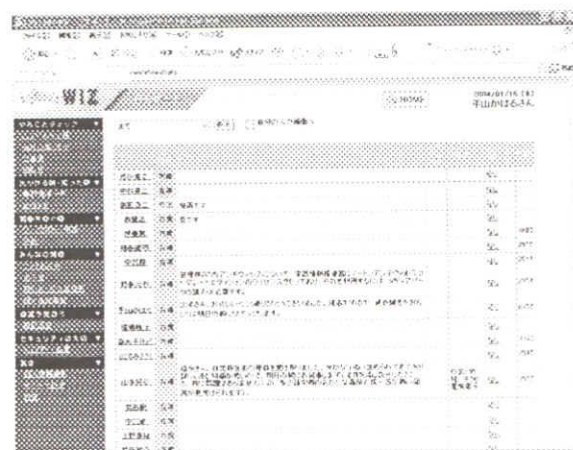
2.4 主な機能

「La!cooda WIZ」の主な機能とその画面を紹介する。

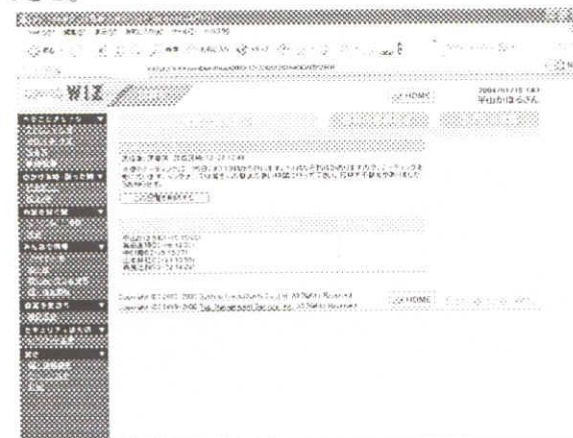
「HOME」－ログインユーザの予定、伝言・回覧の有無、スケジュールなどが表示される。



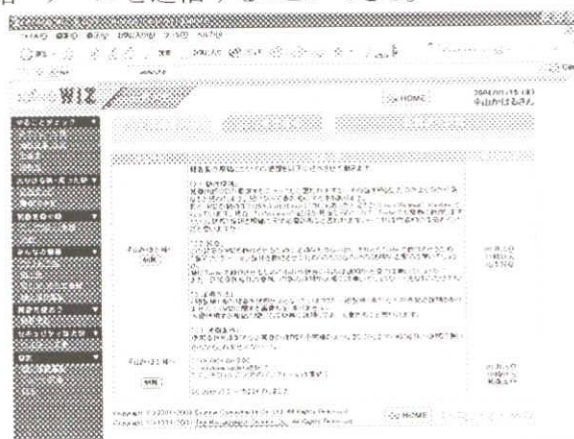
「行先ボード」－スケジュールと連動しているので、登録ユーザの所在が表示される。手動で随時情報を更新することができる。



「回覧板」－グループで情報を共有したいときに便利な機能である。着信一覧を見ること、着信回覧にコメントを付けること、また、新規に作成することができる。



「伝言板」－連絡事項を伝える機能であり、メッセージ内容は、個人情報と連動しているので携帯電話へメールを送信することができる。



「掲示板」－いろいろな情報を通知できる。フォルダを作成したり、メッセージを検索することができる。



「ファイル共有」－あらゆる形式のファイルを Web 上で共有できる。共有ファイルはフォルダへの分類ができ、公開レベルを設定することができる。



3 活用方法

グループウェアと業務運用・管理システムを連携させる場合、グループウェアのどの機能を使用するか。また、作業を行う場合、作業同士士の日程調整や連絡にどの機能を使用するか。図2は業務の依頼から終了までの流れである。主に依頼内容調査担当者の決定、依頼業務担当グループおよび作業担当者決定、作業開始後の連絡・調整において述べる。

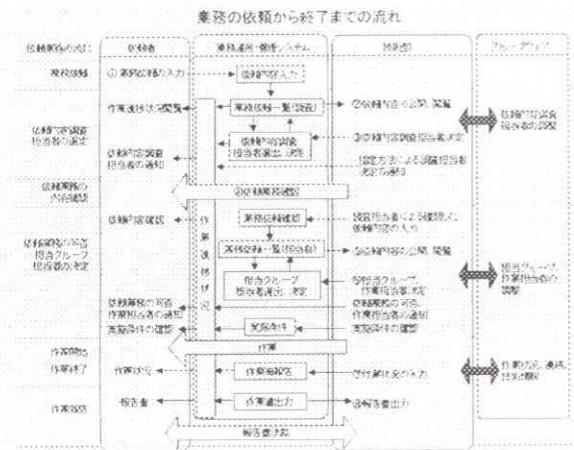


図2. 業務の依頼から終了までの流れ

3.1 使用条件

依頼業務を行うには、迅速かつ確実なグループ内の連絡が不可欠である。そこで、以下の条件を設定する。

- (1)「伝言メモ」、「回覧板」の有無が表示される「HOME」画面を1時間に1度は確認する。これにより迅速な対応が図られる。
- (2)「行先ボード」の変更を正確に行うとともに、「HOME」同様、1時間に1度は確認する。これによりグループメンバーの状態を把握し、確実な連絡が図られる。

3.2 業務運用・管理システムとの連携 依頼内容調査担当者決定

業務運用・管理システムに依頼業務が書き込まれると、全ての技術部メンバーにその書き込みが公開される。そこで、グループ内で最初に依頼業務を確認したメンバーは、グループ内メンバーにグループウェア「回覧板」を利用し、依頼業務内容調

査担当者を行うかどうかの通知を行う。この場合、必ず「コメント付きで受信確認」にチェックを入れる。この通知を受けたグループ内メンバーは、「回覧板」の「コメント」に意志表示を行う。意思表示がないメンバーに対しては、グループ内の業務委員が「行先ボード」により状態を確認し判断する。グループ内の意志が決定したら、グループの代表もしくはそれぞれが業務運用・管理システムに表示を行う。(図3参照)

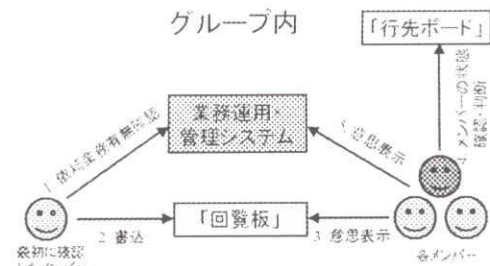


図3. グループ内の依頼内容調査担当の意志調整

タイムリミットまで依頼内容調査担当者が決定しない場合、最初に業務運用・管理システムの依頼内容調査担当者に担当意志表示者が無いことを確認した業務委員は、各業務委員に「回覧板」（「コメント付きで受信確認」にチェック）を利用し、依頼内容調査担当者の調整の書き込みを行う。これを確認した各業務委員は直ちに「回覧板」の「コメント」により調整を行う。また、業務委員の代表者が業務運用・管理システムに調整結果を表示する。(図4参照)

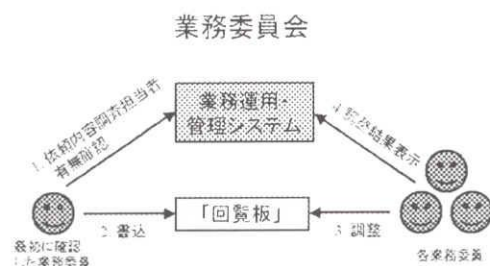


図4. 業務委員会内の依頼内容調査担当の調整

依頼業務担当グループ、作業担当者決定

依頼内容調査担当者が確認した依頼内容が、業務運用・管理システムにより公開されるので、各グループ内で最初にこれを確認したメンバーは、

グループ内メンバーに依頼業務を担当するかどうかの書き込みを「回覧板」（「コメント付きで受信確認」にチェック）を利用して行う。この連絡を受けたメンバーは依頼業務内容調査担当者の選出時同様、「回覧板」の「コメント」により意思表示を行う。意思表示がないメンバーに対しては、グループ内の業務委員が「行先ボード」により状態を確認し判断する。決定したグループ内の意思をグループの代表もしくはそれぞれが業務運用・管理システムに表示する。（図5参照）

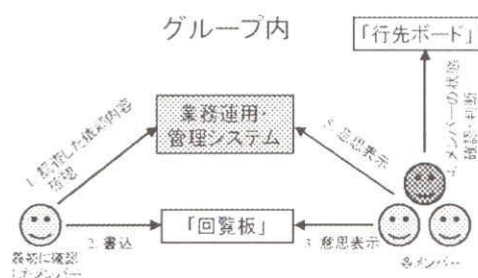


図5. グループ内の依頼業務
担当意志の調整

業務委員会は依頼業務開始までに依頼業務の可否を決定しなければならない。業務運用・管理システムの作業担当者がない場合、これを最初に確認した業務委員は、各業務委員に「回覧板」（「コメント付きで受信確認」にチェック）を利用し、調整を行う。また、調整結果を業務運用・管理システムに表示する。（図6参照）

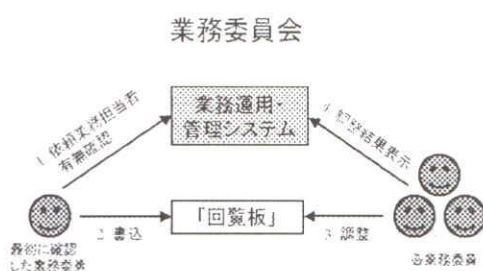


図6. 依頼業務に対する
担当グループの調整

作業開始後

（作業日程の連絡・調整）

作業開始後、作業グループ内メンバーへの日程調整や連絡は「伝言メモ」を利用する。この「伝言メモ」を受け取ったメンバーは、「伝言メモ」により確認通知または再調整依頼を行う。（図7参照）

作業担当グループ内

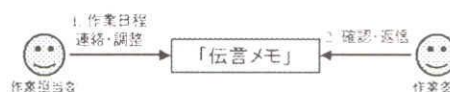


図7. 作業担当者間の連絡・調整

作業開始後

（技術共有）

技術共有という観点から、作業担当者は一日の作業内容の詳細な記録を「掲示板」に記録する。その際、各項目毎に整理して行う。これにより、同様の依頼業務が発生した場合、他の技術部メンバーがスムーズに対応できる。また、記録された作業内容に対し、技術情報の提供ができる。また、内容量が多い場合は、「掲示板」には要点のみを書き込み、詳細は「ファイル共有」にアップロードする。（図8参照）

技術部内

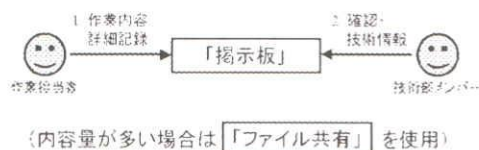


図8. 技術情報の共有

3.3 その他

「行先ボード」の「メモ」は「回覧板」や「伝言メモ」などを利用するまでもない”独り言”などの書き込みを利用すると便利である。

また、会議資料や技術資料は「ファイル共有」にアップロードすると保存・配布が容易である。

4 運用・管理

このグループウェアのサービスを行っているサーバのセキュリティ確保のため、1時間に1度メールにより配信されるログの監視を行っている。また、データベースのバックアップ（月 2～3 回）を行い、万が一データベースに障害が生じた場合のリストアに備えている。その他、「La!cooda WIZ」の ver1.1.1 から ver 1.1.2 へ、ver 1.1.2 から ver 1.1.3 へのアップデートを行っている。

5 おわりに

以上、業務運用・管理システムと連携したグループウェアの活用方法の一例を提案した。今後、依頼業務を行っていく中で、さらに技術部に適した活用方法の研鑽を行う。また、WIZ のプログラムを解析し開発を行うことで、技術部に特化したカスタマイズを行う。