

# 技術部業務運用・管理システムの開発事例紹介

伊藤 篤，梅田 直明，中村 勝，新美治利，平山かほる，山本好弘

三重大学 工学部 技術部

## 1 はじめに

三重大学工学部では，技術部が担当する技術支援業務の範囲を，メンバーが所属する研究分野から工学部全体に拡大する検討が行われている。この業務には，新たに専門別に組織する5つのグループあたり，業務の運用と管理は委員会形式で行われる。今回報告する業務運用・管理システム<sup>[1]</sup>は，そのツールとして実務面からサポートすることを目的に開発，構築を進めているものである。本報告では，このシステムのデータベース，サーバ，Web，プログラミング，セキュリティの技術を中心に開発事例を報告する。

## 2 技術部業務運用・管理システム

本システムの開発は，実現する要件の明確化 データベース設計・作成 サーバ設定 メニュー，Web画面作成 プログラミング 試験運用，トラブル・セキュリティチェック，機能追加 運用開始の手順で進めており，現在は試験運用中にある。の実現する要件の明確化では，業務依頼から完了，決裁までの全工程の情報と手続きの電子化，データベース化，Webブラウザによる直感的な操作感，利便性・ユーザ情報保護・セキュリティのバランス，スムーズな情報共有及び技術部 Webサーバで稼動し，開発と運用は技術部が行う事があげられる。以上を踏まえた上で，実現可能と思われる業務依頼の流れを表1に示す。業務依頼は，依頼者が本システムに対して行う 依頼内容の入力で始まり，技術部メンバーは内容を確認後，調査担当者選出を行う。調査担当者は，依頼者との 依頼内容の打ち合わせを行い，その内容をシステム上にアップする。技術部メンバーは，その内容を確認後，業務担当者を選出し，実施条件の確認を経て，作業を開始し 作業毎の内容報告を行う。作業終了後，業務完了報告，決裁を行う流れからなる。<sup>[1]</sup>

表 1. 技術部業務運用・管理システムで実現する業務依頼の流れ

業務依頼の流れ	依頼者	Webシステム	技術部，業務委員会
1. 業務依頼	依頼内容の入力	依頼内容	確認
2. 調査担当者選出	確認	調査担当者	調査担当者選出
3. 調査打ち合わせ	依頼内容の打ち合わせ	調査内容	確認
4. 業務担当者選出	確認	業務担当者	業務担当者選出
5. 業務実施の確認	実施条件の確認	実施条件	実施条件の確認
6. 作業開始 - 終了	作業進捗の確認	作業内容	作業毎の内容報告
7. 業務完了	業務完了の承認	業務完了報告	業務完了報告，決裁

## 3 データベース

データベースの設計では，システム化する対象の業務運用・管理に関わる人と情報の流れがどのように関連し機能しているかを正確に把握することが重要である。本システムでもデータベース設計の一般的な開発手法である概念，論理，物理の工程に分けて作業を進めた。まず概念データベース設計では，データベース

の目的を明確化し、データベース上で管理する要件を出来るだけ理想に沿った形でモデル化を行う。ここで実際に使用するデータベースのリレーショナルモデルよりも高次のモデルである ER モデル (Entity-Relationship model)を用いて情報の管理対象となる実体と実体間相互の関連をモデル化した。図1に本システムの ER モデルを示す。図中では、実体を四角、属性を楕円、関連を菱形で示す。

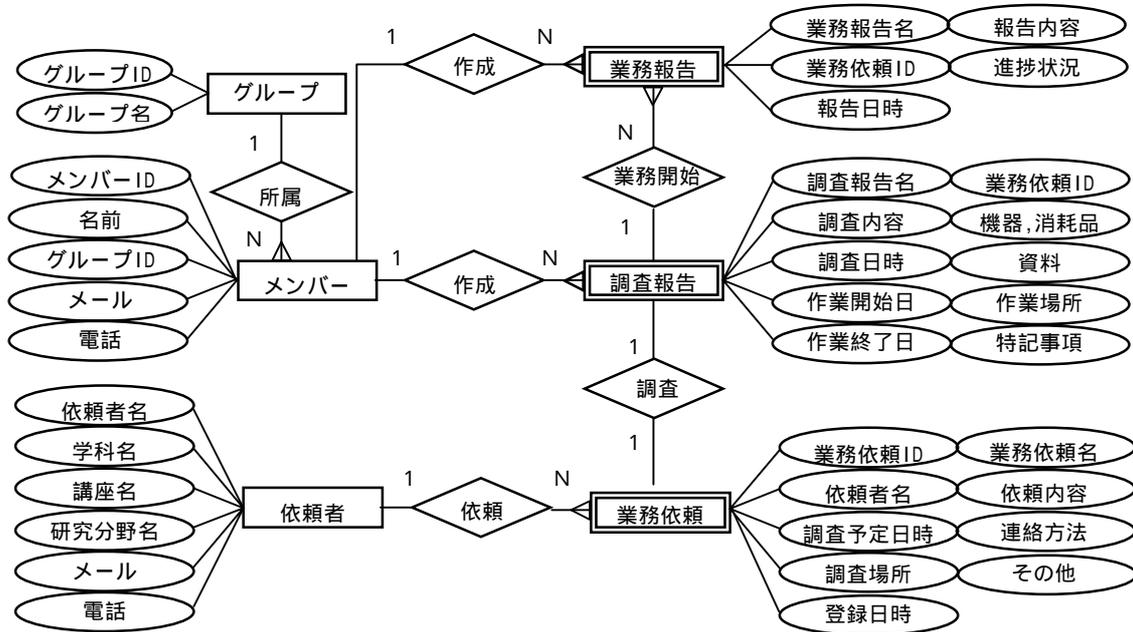


図1 業務運用・管理システムの ER モデル(J.Martin 型と Chen 型の併用)

情報の管理対象となる実体には、依頼者、技術部メンバー、グループ、業務依頼、調査報告、業務報告の6つを抽出した。各実体には、それぞれが保有する情報で実体の性質や特性を表現する要素として属性を設定し、その中でも実体を唯一識別可能とするものをキー属性とした。実体間の関連を正確に表す方法として数の関係に着目した基数(Cardinality)を ER モデル中に記した。例として、複数の技術部メンバーから構成され重複した所属を不可とするグループの場合、所属の関係にある基数は1対Nである。次にこの ER モデル上で業務運用・管理のシステム化に際して、人と情報の流れとその関係が十分に機能するのか検討を重ねた。

論理データベース設計では、概念データベース設計で作成した ER モデルから実際に使用するリレーショナルモデルの表と SQL(Structured Query Language)のテーブルへと変換を行った。この変換作業は、実体をテーブルに属性を列にキー属性をキーに変換する。1:Nの関連では、N側テーブルの主キーに1側テーブルのキーを追加しキーとした。

次にこのリレーションモデルのテーブルのデータ構造を整理する正規化を行う。正規化の目的は、データ更新時の異常発生を抑制するとともにデータの独立性を高めることにある。ひとつのテーブルに複数の事実が存在していると、データ登録、更新の誤りやタイミングの制限、データ整合性の喪失が発生しやすくなることから正規化は必須の作業である。正規化は、属性間の意味的な関係を関数従属性で関連付け、実体と構成する属性の独立性を高め単純化する。ここで関数従属性とは、属性間の値の決定関係を表したもので、ある値に対応してもうひとつの値が決定されることを意味する。

本システムでは、リレーショナルモデルの目安とされる第3正規化まで行い、各属性のデータ型と制約の決定を行った。最後に物理データベース設計では、アクセス方法を考慮したファイル群(データ、作業、ログ領域)の配置と容量の見積もり、CPU、HDD、I/O、ネットワークなどハードウェアの処理能力、トランザクション解析、インデックスの作成およびアクセス効率を考慮した非正規化を行う。インデックスは、問い合わせ処理を高速化するための付加的なデータ構造で索引の機能を持つ。<sup>[2][3]</sup>

## 4 サーバ

本システムは、これまでも部内グループウェア<sup>[4]</sup>で開発、運用経験のある Web サーバ：Apache，プログラミング言語：PHP，リレーショナルデータベース：PostgreSQL，OS：Linux で構築した。PHP の PostgreSQL インターフェイスは、PostgreSQL に付属の API(Application Programming Interface)である libpq を呼び出し機能する。各サービスの連携と動作の仕組みは、最初に ユーザが Web ブラウザより Web サーバに接続を行うと、Apache が接続要求のあった HTML コンテンツを呼び出し、スクリプトが PHP により解析・実行される。データベースへのアクセスは、PHP の PostgreSQL インターフェイス libpq を通じてデータベースサーバ postmaster に対して PostgreSQL 関数により SQL 文でクエリーの実行、問い合わせが行われ、postmaster はデータベースエンジン postgres を起動し、データベースアクセス処理後結果を返す。再度、PHP から SQL 文による問い合わせを行い、検索結果を得て、PHP で結果を HTML 化し Apache が Web ブラウザに返送する。Apache は、PHP とプログラムは、PostgreSQL は の処理を分担、連携して動作する。ここで PHP による Apache プロセスから PostgreSQL に接続要求を行うと、postgres プロセスがひとつ生成されることから、Web 接続数に応じた最大プロセス数の管理に注意が必要とされる。<sup>[2][5][6]</sup>

## 5 Web 画面，プログラム

業務依頼の登録では、登録者の個人認証を省き、代わりにユニークな ID を発行する方式とした。業務依頼登録の Web 画面を図 6 に示す。

Web ブラウザをインターフェイスにユーザと Web サーバ間で情報を交換するプロトコル http(hyper text transfer protocol)はステートレスな接続であり、Web サーバはブラウザからの接続要求の履歴を管理する機能を持たない。本プログラムでは、PHP のセッション管理機能を用いてこの問題に対応した。この機能は、ユーザが使用するブラウザの HTTP クッキーが利用可能か自動判定し、利用不可の場合も URL 引数や hidden タグを使うなどして推測が困難なセッション ID を生成しブラウザに引き渡すことで接続履歴を管理する。

プログラムは、PHP，PostgreSQL，Apache が相互に連携して動作している都合上、Web 画面に表示する文字化けを防ぐために文字コードに注意する必要がある。本システムでは、Web ページ、PHP の内部文字コード、データベースの文字コードを EUC-JP コードに統一してプログラムとデータベースを作成した。<sup>[2][5][6]</sup>

The screenshot shows a web browser window titled '業務依頼 - Microsoft Internet Explorer'. The page content is for '工学部技術部 業務依頼' (Faculty of Engineering, Department of Technology, Business Request). The form includes the following fields and options:

- 氏名 (Name): テスト
- 所属 (Department): 工学部 (Engineering Department)
- 研究分野名 (Research Field): テスト
- 依頼内容 (Request Content): テスト (100 characters or less)
- 打合せ日時 (Meeting Date/Time): 2003/12/11
- 打合せ場所 (Meeting Location): 電気電子棟3階1315室
- 連絡方法 (Contact Method):  e-mail (test@eng-mie-u.ac.jp),  内線電話 (Internal Phone: 3883)
- その他 (Other): テスト

At the bottom of the form, there is a button labeled '入力確認' (Confirm Input) with a red arrow pointing to it and the text 'ココをクリックすると確認画面になります。' (Clicking here will lead to the confirmation screen).

図 6 業務依頼登録画面

## 6 セキュリティ

OS と各種デーモンのセキュリティ対策は、ディストリビュータによるセキュリティアナウンス (1 年間で 67 件) に従い実施した。本システムのサービスと直接関係するものでは、PHP の「透過セッション ID サポートにクロスサイトスクリプティングの脆弱性」が報告されたが、当初よりクッキー使用不可時の URL 自動書き換え設定である php.ini の session.use\_trans\_sid=Off で運用しており問題はなかった。またクロスサイトスクリプティングと SQL インジェクション対策としてフォーム入力には htmlspecialchars()関数による特殊文字の HTML エントリ変換と ereg\_replace()関数による特殊文字の全角変換を行っており問題は発生しなかった。

PHP 本体は、Apache のモジュールとして組み込まれ、同一プロセスの元に nobody 権限で実行され安全性に配慮した運用を行っている。PHP 全般の動作設定は php.ini に記述され、グローバル変数制限の register\_globals=Off、外部ホストへのアクセスを制限する allow\_url\_fopen=Off、システム実行制限 safe\_mode=On、Web サーバ変数の優先 variables\_order="EGPCS"、ユーザ入力の引用符等文字列のエスケープ設定 magic\_quotes\_gpc=On、エラー非表示 display\_errors=Off、エラー記録 log\_errors=On で運用した。

Apache の運用では PHP の遮蔽を行い、httpd.conf で AddType application/x-httpd-php.html とした。この設定では全てのコンテンツが PHP Engine を通じてパースされるためにレスポンスの低下が懸念されるが、他のシステムでの運用例から、本システムでは実用上問題ないと推察されたので、安全性を考慮し実施することとした。システムと各デーモンの監視は、通常のシステムログに加え PostgreSQL の設定ファイル postgresql.conf を syslog=2、debug\_print\_query=true、logconnection=true として稼働状況の監視を行った。<sup>[5][6]</sup>

## 7 今後の課題

業務依頼登録時にブラウザの Reload ボタン誤用による重複登録の防止策として、PHP のセッション管理機能を用いてプログラミングを行った。しかし、1台の PC 上で異なるブラウザを2つ起動してアクセスするケースでは、重複登録が防ぎきれないので対策が必要である。また、ブラウザがローカルに管理するキャッシュやネットワーク経路にあるプロキシサーバに対して、PHP が HTTP プロトコルに則した接続を行った場合でもブラウザの進む、戻るボタンの挙動はブラウザごとに異なることから、完全な対応は困難であり他の対策が必要とされる。データベースの運用では、テスト期間内にスキーマ定義と正規化の検証、確定、動作設定のチューニング、データベースを複製するレプリケーションによるバックアップ体制の整備が必要である。また、業務依頼に関わる全工程の情報を定型フォーマットで印刷するために PDF ファイルの生成機能を追加する必要がある。プログラムでは、データベース操作、画面生成および両者をコントロールするロジックに分けてクラス化しプログラム効率を高める MVC(Model View Controller)化を進める必要がある。<sup>[2]</sup>

## 8 むすび

業務運用・管理システムを構築し、ひとまず最低限の機能とセキュリティ対策を行った。今後、本運用までにシステムの完成度を高めるとともに、運用中のグループウェア<sup>[4]</sup>と連携して活用することで、グループ内の情報共有の最適な在り方を見出したい。Web をインターフェースにデータベースを利用するアプリケーションサービスは、今回構築した業務運用・管理システム以外にも参加登録、予約、情報記録、閲覧など多くの用途があり、サービスの質もより高機能なものが求められている。今後、データベースと Web 技術を連携させたシステム構築を通じて技術のスキルアップを図ることを課題とする。

## 参考文献

- [1] 新美治利,中村勝,et al, “Web 技術を用いた技術部業務運用・管理システムの紹介”,平成 14 年度東京大学総合技術研究会報告集,平成 15 年 3 月, P5-36 – P5-38 , P5-68 – P5-70
- [2] 石井達夫, “PHP × PostgreSQL で作る最強 Web システム”,技術評論社, ISBN 4-7741-1647-5
- [3] 北川博之, “データベースシステム”,昭晃堂, ISBN 4785620463
- [4] 平山かほる,et al, “Web データベースアプリケーションによるグループウェア開発”,平成 13 年度核融合科学研究所技術研究会報告集,平成 14 年 3 月, P459-462
- [5] 日本 PHP ユーザ会 Web サイト, <http://www.php.gr.jp/>
- [6] 日本 PostgreSQL ユーザ会 Web サイト, <http://www.postgresql.jp/>