工学部技術部ネットワークグループ活動報告2003

Webデータベースアプリケーションの開発と運用の事例紹介

伊藤 篤（工学部技術部 第一技術系第一班） 中村 勝（工学部技術部 第二技術系第四班）

1. はじめに

Webブラウザをインターフェースにデータベースを操作するアプリケーションを利用する機会が多くなってきている。身近にはインターネット上の会員登録やネットショッピング，ネットバンキング，企業内ネットワークで利用されているグループウェアなど幅広い分野で多くのサービスが利用されている。学内でも，図書館での文献・蔵書検索，シラバス，物品請求／予算執行検索，出張旅費のシステムが運用され，日常の業務に不可欠なサービスとなっている。技術部では，2年前よりグループウェア「Skyboard」と「Wiz」を運用し，グループ内の情報共有とデータベース化を行うとともに，その活用方法を探ってきた。また，Webとデータベースを連携させたシステム開発では，平成15年度機器・分析技術研究会参加登録システムの構築を行い，Webを用いて4ヶ月間に全国41の大学，高専，研究所から145名の参加登録を受け付けた。現在は，技術部再組織化に向けて検討中のWebを用いた技術業務の依頼受付と，それらの運用・管理を行うシステムを開発中である。本報告では，この技術部業務運用・管理システムの開発事例について，データベース，Webプログラミング，セキュリティの技術を中心に報告する。

2. 技術部業務運用・管理システム

本システムの開発は，①実現するサービスと手順の明確化 ②データベース設計，作成 ③サーバ設定 ④メニュー，Web画面作成 ⑤プログラミング ⑥試験運用，トラブル・セキュリティチェック，機能追加 ⑦運用開始の手順で進めており，現在は平成16年4月運用開始を目指して試験運用中にある。①の実現するサービスと手順の明確化では，次の②データベース設計以降の全てに決定的に関係してくるので，十分な検討が必要とされる。ここで考慮した5つの要件を以下にあげる。

1. システム上で実現するサービスは，依頼者，技術部メンバー双方の情報交換・共有をスムーズに行うとともに，業務依頼から完了，決裁までの全工程の手続きと情報の電子化を行うこと。
2. 依頼者，技術部メンバー双方のシステムを利用する全てのユーザには，使用機器やOSなどによらずWebブラウザをインターフェースに直感的にサービスを利用・操作出来ること。
3. サービスの利用は，利便性を損なわずにユーザの情報を保護するとともに，三重大学情報セキュリティポリシーを遵守するよう利便性とセキュリティのバランスを考慮すること。
4. 業務に関わる全ての情報をデータベース化するとともに，居室が分散している技術部メンバー間でもリアルタイムに情報を共有できること。
5. システムの開発と運用は技術部が行い，技術部Webサーバ上で安定に稼動すること。

以上の要件を踏まえた上で，技術部にとってシステム化が可能と思われる業務依頼の流れを表1に示す。

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	①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	②調査担当者選出
3. 調査打ち合わせ	③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	④業務担当者選出
5. 業務実施の確認	⑤実施条件の確認	実施条件	⑤実施条件の確認
6. 作業開始－終了	⑥作業進捗の確認	作業内容	⑥作業毎の内容報告
7. 業務完了	⑦業務完了の承認	業務完了報告	⑦業務完了報告，決裁

業務依頼は，依頼者がWebシステムに対して行う ①依頼内容の入力で始まり，技術部メンバーはその内容を確認後，②調査担当者選出を行う。調査担当者は，依頼者との ③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，④業務担当者を選出する。その後，⑤実施条件の確認を経て，作業を開始し ⑥作業毎の内容報告を行う。作業終了後，⑦業務完了報告，決裁を行う。以上の流れからなる。

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。データを共有資源としてシステムの中心におき，データの普遍性と安定性に着目したシステム開発の手法である DOA(Data Oriented Approach)に基づき，データベースの設計で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についてモデル化を行った。図1に業務運用・管理システムのERモデル(J.Martin 型と Chen 型の併用)を示す。図中では，実体を四角，弱実体を二重線四角，属性を楕円，関連を菱形で示す。

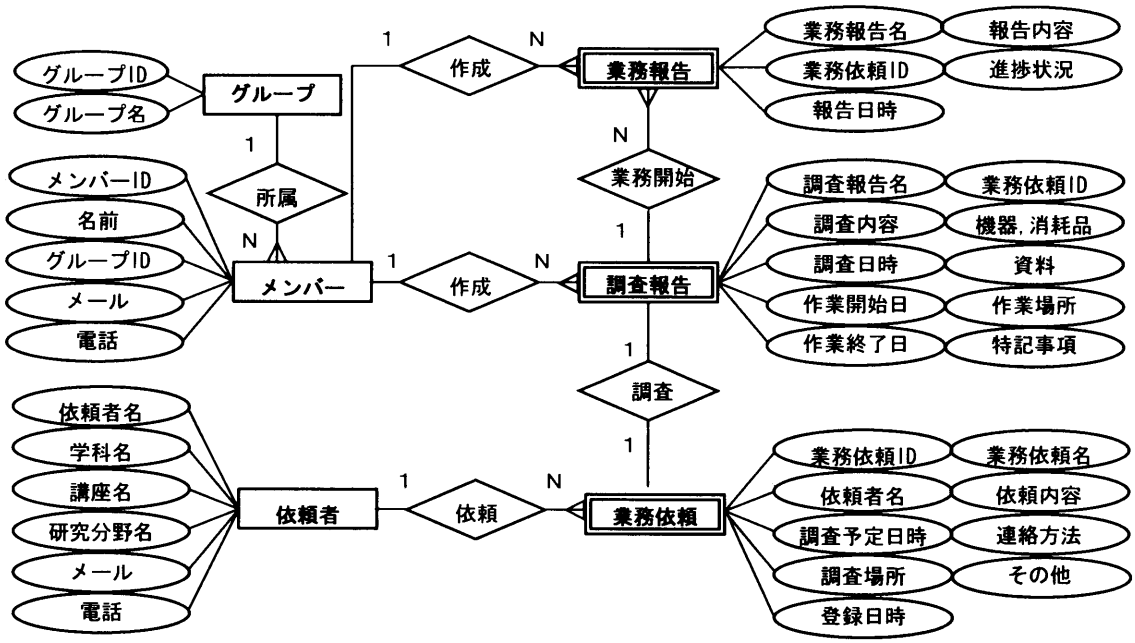

図1 業務運用・管理システムのERモデル(J.Martin 型と Chen 型の併用)

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。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。実体には，単独で存在できず他の実体と関係を持たないと成り得ないものを弱実体として区別した。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。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。

次に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，以下のルールに従って行った。

1. 実体, 弱実体 → テーブル , 属性→列, キー属性→ キーに変換する。
2. 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 しキーとする。
3. 弱実体は, 識別上のオーナに対応するテーブルの主キーを追加 し, このテーブルのキーとする。
4. N:Mの関連では, 関連自体をテーブルに変換しN, M双方の主キーを組み合わせて主キーとする。

このリレーションモデルのテーブルを図2に示す。

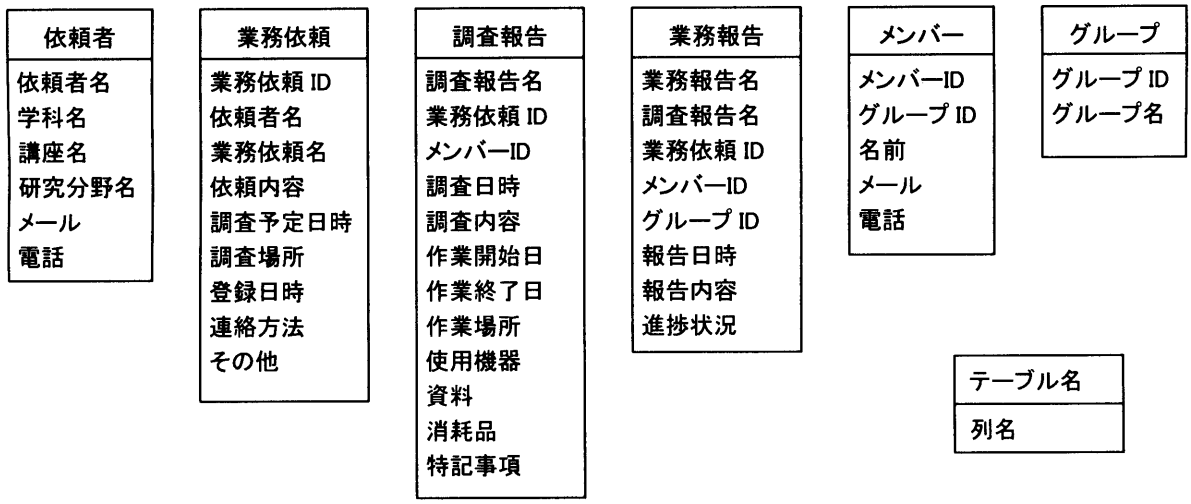

図2 業務運用・管理システムのテーブル定義

次にこ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。各正規形の定義と語句の意味を表2に示す。

表2 各正規形の定義と語句の意味

第一正規形	繰り返し項目の分離と導出属性の除去を行う。
第二正規形	主キーに非キー属性が完全関数従属するよう属性を整理する。
第三正規形	全ての非キー属性がどの候補キーにも推移関数従属性を持たないよう整理する。
導出属性	属性間でいずれかの2値が決まれば他のひとつの値が導き出される場合の属性。
候補キー	表の中で唯一のデータを特定出来る属性を指す。
主キー	候補キーの中でひとつだけ存在出来る取り扱い性に優れた代表的なキーを指す。
非キー属性	自身では候補キーになることが出来ず, かつ候補キーの一部にも属さない属性。
完全関数従属	X→Y の関数従属があるとき, X が複数の属性により形成されており, その属性集合のひとつの属性が欠けると関数従属性が成立しないこと。
推移関数従属性	X→Y, Y→Z の関数従属があるとき, Z は X に推移関数従属の関係にある。

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。例として技術部メンバーの名前は, ID に関数従属し, 技術部メンバー名前→ID と表される。正規化は, 第1形から第5形までと Boyce/Codd 形が知られており, リレーショナルデータベースでは第3正規化まで行うことが必須とされている。次に各属性のデータ型と制約の決定を行った。データ型と制約の意味を表3に示す。

表3 データ型と制約の意味

CHARACTER(n)	n 文字からなる文字列で文字数に満たない場合は，スペースが後ろに入る。
CHARACTER VARYING(n)	n 文字からなる文字列で文字数に満たない場合でもスペースは入らない。
TEXT	データベースで取り扱う上では文字数に制限のない文字列
TIMESTAMP WITH TIMEZONE	タイムゾーンを含めた年月日時分秒
SERIAL	32 ビット整数の順序数を自動的に生成
BOOL	論理値
Not Null 制約	列の値に未定を表す Null 値を許可しない。主キーは宣言が不要である。
Unique 制約	同一テーブル内で列内に重複値を許可しない。
Check 制約	列の値に任意の論理式を設定，評価する。
参照整合性制約	テーブル間の基数が 1 対 N にある時，表と表の関係に制約を持たせる。

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。主キーのインデックスは，自動的に設定される仕組みとなっている。

4. サーバ

本システムに求められる要件を実現するためのサーバ選択条件は，一般のネットワークサーバとそのサービスに求められる条件と同一ではあるが，具体的にどのツールを組み合わせて構築するかは多くの選択肢がある。今回はサーバ機器の制約上，そのOSであるLinux上で稼動するものを選択することが求められた。そこでこれまでにも部内グループウェアで運用・開発経験のあった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 を組み合わせて構築することとした。

OSに用いたLinuxは，UnixライクなOSとしてネットワークサーバをはじめ，最近では基幹系サーバにも導入され豊富な情報と開発環境に特徴を持つ。Webサービスを提供する Apache は最も普及しているWebサーバであり，多くの機能がモジュール化され用途に応じて選択して組み込むことが出来る特徴を持つ。

プログラミングに用いた PHP は HTML 埋め込み型のサーバサイド・スクリプト言語でWebサーバ Apache のモジュール化や豊富なデータベースインターフェースと優れたエラーハンドリング，セッション管理などに特徴を持つ。本システムでは運用と管理の利便性から PHP モジュールを Apache の起動時に共有ライブラリとして動的に組み込む Dynamic Shared Module 方式をとった。

登録データの保存，管理はデータベースを用いて行った。そのメリットは，複数アクセス時の排他制御とトランザクション管理，データ保全の信頼性，データ量の増加と検索性に優れたメリットを持つ。PostgreSQL は RDBMS(Relational Data Base Management System)で多くのプラットフォーム上で使われており Structured Query Language に準拠しマルチバイト文字対応やトランザクション処理，バックアップなど多くの機能で商用データベースにも匹敵するなど優れた特徴を持つ。PHP とはインターフェースのモジュール化により安定した動作が実現されている。業務運用・管理システムの機能を分担する PHP, PostgreSQL, Apache は，相互に連携して動作し，PHP の PostgreSQL インターフェースは，PostgreSQL に付属の API(Aplication Programing Interface)である libpq を呼び出して機能する。参加登録時の各サービスの連携と動作の仕組みを図3に示す。最初に①ユーザがWebブラウザより参加登録システムにアクセス、つまりWebサーバに HTTP 接続要求を行うと、②Apache が接続要求のあった HTML コンテンツを呼び出し、③内部に記述され

たスクリプトが PHP により解析・実行される。データベースへのアクセスが必要な場合は、④PHP の PostgreSQL インターフェース libpq を通じて PostgreSQL のデータベースサーバ postmaster に対して PHP の PostgreSQL 関数により SQL 文によるクエリー実行問い合わせが行われ⑤postmaster はデータベースエンジン postgres を起動しデータベースへのアクセス処理後結果を返す。⑥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⑦PHP で結果を HTML 化し Apache がWebブラウザに返送する。Apache は②, ⑦, PHP とプログラムは③, ④, ⑥, ⑦, PostgreSQL は⑤の動作を分担, 連携して実行されている。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。

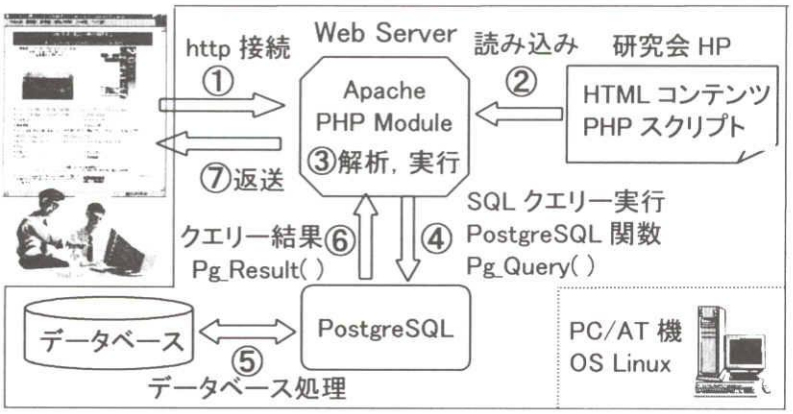

図3 各サービスの連携と動作の仕組み

5. 操作メニュー, Web画面, プログラミング

技術部業務運用・管理システムの依頼者による業務依頼のメニュー構成を図4に, 登録のWeb画面を図5に示す。全ての操作は出来るだけシンプルな構成とし, 依頼内容の登録では, 依頼－入力－確認－登録の最小限の操作で登録が完了する。このようなサービスでよく用いられるユーザ登録と個人認証は, シンプルで簡単な操作が必要とされるこのシステムでは取り入れていない。その代わりの手段として, アクセスは工学部内のみに制限したうえで, 業務依頼ごとにユニークな ID を発行することとした。依頼者自身が依頼内容を確認または作業状況にコメントを書き込む場合は, 依頼時に発行された ID で検索後, 直接Web上で操作が可能である。

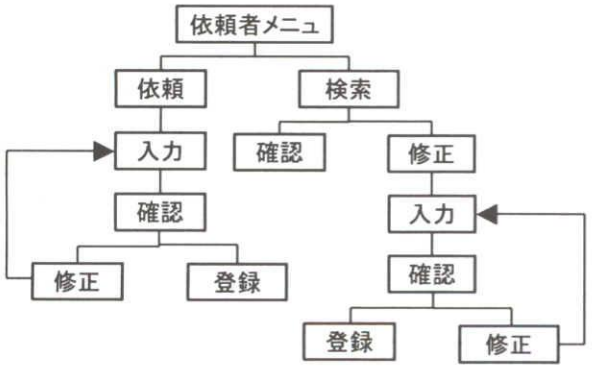

図4 依頼者用メニュー構成

図5 業務依頼登録画面

技術部メンバー用操作メニューの構成を図6に, 業務担当者選出画面を図7に示す。次に依頼を受けた業務の運用管理は, 技術部メンバーが業務依頼の一覧を確認し, 調査担当者選出－調査報告－業務担当者選出－業務報告の順に進行する。

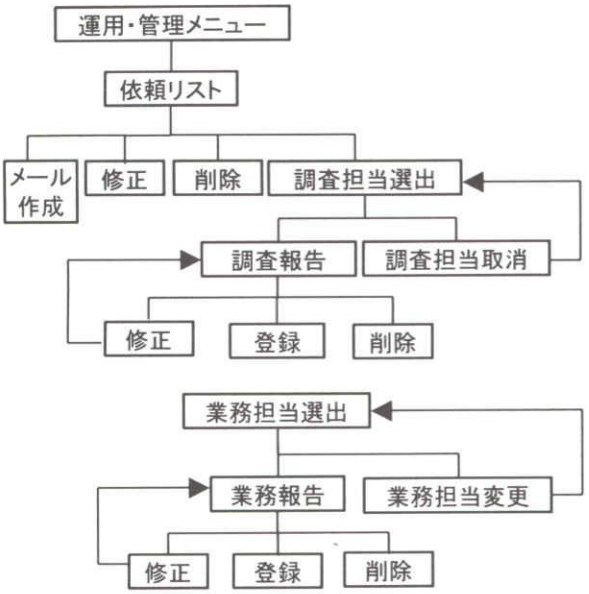

図６　技術部メンバー用メニュー構成

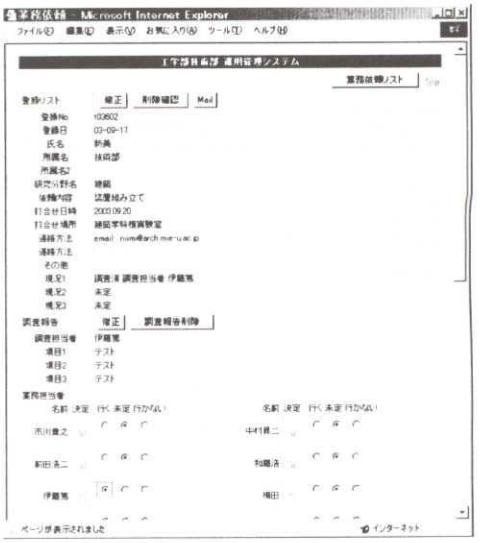

図７　業務担当者選出画面

Webブラウザをインターフェースにユーザと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。これはユーザとサーバ間でひとつの画面生成またはFORMタグのGET，POSTによる値の受け渡しを行った後に接続要求を行ったユーザを見分けられないことを意味する。今回のプログラムでは，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。業務依頼の登録を行うプログラムのフローチャートを図8に示す。また，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コードに統一した。

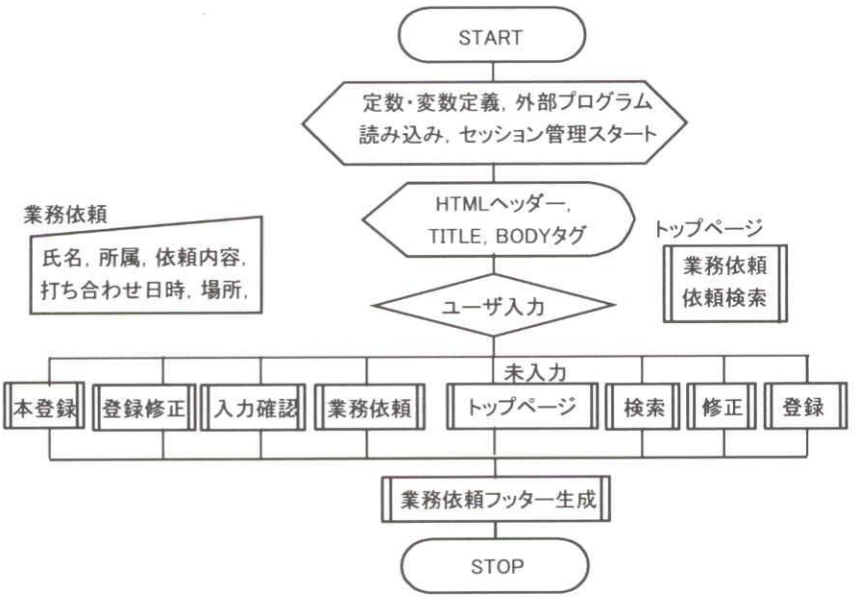

図8　業務依頼登録プログラムフローチャート

6. セキュリティ

一般にネットワークサービスのセキュリティ確保と対策は，OSと各デーモン Apache，PHP，PostgreSQL および PHP で作成したプログラムの全てで行う必要がある。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に従った速やかな実施が不可欠である。2003年の1年間でのセキュリティアナウンス数は67件であった。本システムで実施している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。本システムの PHP では，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 で運用していた。ここで下線文字は設定を表す。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，①グローバル変数制限の register_globals=Off，②外部ホストへのアクセスを制限する allow_url_fopen=Off，③システム実行制限 safe_mode=On，④Webサーバ変数の優先 variables_order="EGPCS"，⑤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，⑥エラー非表示 display_errors=Off，⑦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たが，実用上問題ないことが確認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 の syslog=2，debug_print_query=true，logconnection=true として稼働状況の監視を行った。

7. 今後の課題

業務依頼登録時にブラウザの Reload 機能誤用による重複登録の防止策として，セッション管理を用いてプログラミングを行った。これはユーザにより改変される可能性のある変数の引き渡しでも有効である。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したケースでは異なるセッションとなり，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と確定，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。

業務依頼に関わる全工程の情報は，定型フォーマットで印刷できるように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。

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，2年前より運用してきたグループウェアと連携して活用することで，業務運用・管理に必要とされる情報共有の最適化を見いだしたい。また，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，公開・閲覧，ファイル共有，ポータルサービス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] 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

[2] 北川博之，“データベースシステム”,昭晃堂, ISBN 4785620463

[3] 日本 PHP ユーザ会 Web サイト, http://www.php.gr.jp/

[4] 日本 PostgreSQL ユーザ会 Web サイト, http://www.postgresql.jp/