業務運用・管理システムの開発から運用

中村　勝（工学部　技術部　情報システムグループ）

1. はじめに

昨年の技術発表会において業務運用・管理システムに関して,「業務運用・管理システムの紹介」,「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」の報告を行った.「業務運用・管理システムの紹介」では本システムの紹介や本システムを用いた依頼業務の流れを,「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」ではデータベース, Webプログラミング, セキュリティの技術を中心に報告した. しかし, 報告の時点ではシステムが完成しておらず, 本運用は行われていなかった.

そこで5月よりシステムが本稼働したので, その後の開発から運用までについて報告する.

2. 業務運用・管理システム構築の手順

本システム構築の手順を以下に示す.

1. システムに求められる要件と実現するサービスの明確化
2. データベース設計, 作成
3. サーバのハードウェア, ソフトウェアの選択と設定
4. 操作メニュー, 入力フォーム, Ｗｅｂ画面設計
5. フローチャート作成, プログラミング, セキュリティ
6. 試験運用, バグ・トラブル出し, 調整・最適化
7. 本運用開始（平成１６年５月）
8. バグ対処, 改良, 調整・最適化

構築の手順1で考慮した要件を以下に示す.

- 依頼者, 技術部メンバー双方の情報交換・共有をスムーズに実現すること
- 業務依頼から完了, 決裁までの全工程の手続きと情報を電子化すること
- 業務に関わる全ての情報は, データベース化し部内で共有すること
- ユーザの使用機器や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操作出来ること
- サービスの利用は, 利便性を損なわずに情報を保護すること
- 三重大学情報セキュリティポリシーを遵守すること
- システムの運用と監視, メンテナンス性に優れること
- 現在運用中の技術部 Webサーバで安定に稼動すること

以上をふまえた上で,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. 昨年はシステム構築の手順にて1から6までの報告を行ったが, 本システムの構築の手順5におけるプログラミングにおいて文書として印刷を行う行程(表1で示す「業務依頼の流れ」のうち⑤実施条件の確認および⑦業務完了の承認, 業務完了報告,承認)については作成されていなかった.

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，承認

2.1　実施証明書・報告書の作成

PHP では作成用ライブラリ PDFlib の機能をサポートしており，PDF ファイルを動的に生成し，データベースのクエリ結果を用いて PDF 文書として出力するといった処理が可能となる．この機能を用いることにより，文書を定型フォーマットでの印刷が可能となる．

業務運用・管理システムでは業務実施確認書及び業務報告書はこの機能を利用することにより Web からの印刷を可能とした．表 1 における⑤実施条件の確認では業務実施確認書を，⑦業務完了の承認・業務完了報告・決済では報告書の印刷を行っている(図 1)．

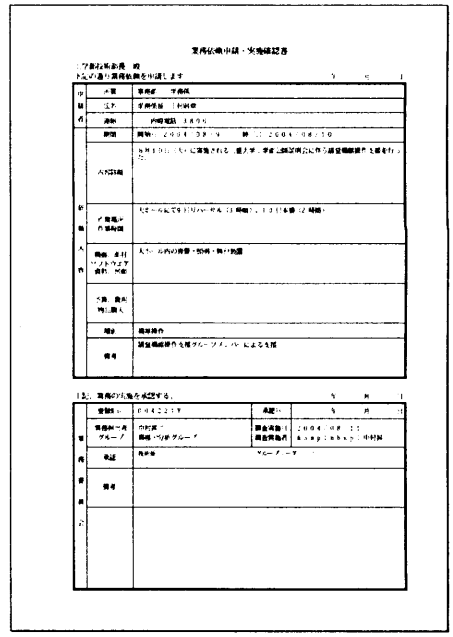

図 1　業務実施確認書

3.　業務運用・管理システムの運用

本システムを連続運転させるため以下の保守・管理を行っている．

- ログによる監視

  定期的にアクセスログをチェックすることで，サーバの状態・不正アクセス等の監視を行っている．また PHP 全般の動作設定を記述している php.ini にてエラー記録をログに出力(log_errors=On)されるようにも設定している．ここで()は設定を表す．

- セキュリティホール・バグへの対応

  ソフトウェアに潜むバグ・セキュリティホールに関する情報を定期的に収集し，問題の発生したソフトウェアについては，パッチ及びアップグレード等の作業を速やかに行っている．

- データのバックアップ・トラブル時の早期対応

  業務運用・管理システムに挙げられたデータやサーバの設定ファイルの定期的な保存を行っている．また，何らかの原因によりシステムがダウンした場合を想定し，バックアップ機の作成を行うことで，サービスの早期回復を図っている．

図 2　システムログの例

```
Date: Tue, 22 Jun 2004 15:01:02 +0900
From: root <root@tgi.elec.mie-u.ac.jp>
To: root@tgi.elec.mie-u.ac.jp
Subject: tgi 06/22/04:15.01 system check

Security Violations
=-=-=-=-=-=-=-=-=-=-
Jun 22 14:25:57 tgi postgres[1764]: [1] DEBUG:  connection: host=[local] user=nobody database=tgi2004
Jun 22 14:25:57 tgi postgres[1764]: [2] DEBUG:  query: select getdatabaseencoding()
Jun 22 14:25:57 tgi postgres[1764]: [3] DEBUG:  query: update tantou set m9='iku' where dataid='O0415K'
Jun 22 14:25:57 tgi postgres[1764]: [4] DEBUG:  query: update irai set jti2='業務担当者 情報システムグループ:中村勝'
where dataid='O0415K'
Jun 22 14:25:57 tgi postgres[1764]: [5-1] DEBUG:  query: SELECT * from irai WHERE dataid='O0415K' and dataid not in  (select
dataid from gijti
Jun 22 14:25:57 tgi postgres[1764]: [6] DEBUG:  query: select * from gyoumu_houkoku where dataid='O0415K' ORDER BY nitiji
desc
Jun 22 14:25:57 tgi postgres[1764]: [7-1] DEBUG:  query: SELECT * from chousa_houkoku WHERE dataid='O0415K' and
Jun 22 14:25:57 tgi postgres[1764]: [8-1] DEBUG:  query: SELECT * from irai WHERE dataid='O0415K' and dataid not in  (select
dataid from gijti
Jun 22 14:25:57 tgi postgres[1764]: [9] DEBUG:  query: SELECT gijti1 from gijti WHERE dataid='O0415K'
Jun 22 14:26:02 tgi postgres[1771]: [1] DEBUG:  connection: host=[local] user=nobody database=tgi2004
Jun 22 14:26:02 tgi postgres[1771]: [2] DEBUG:  query: select getdatabaseencoding()
Jun 22 14:26:02 tgi postgres[1771]: [3] DEBUG:  query: select dataid from irai where nitiji>=current_date
Jun 22 14:26:02 tgi postgres[1771]: [4] DEBUG:  query: select dataid from irai where jti1='受付済'
Jun 22 14:26:02 tgi postgres[1771]: [5] DEBUG:  query: select dataid from irai where jti2='未定'
```

4. 今後の課題

業務運用・管理システムでは複数のユーザが同じ業務依頼に対し同時にその内容を修正する(同一のデータベースにアクセスしデータの修正を行う)可能性がある．つまり，2 つの異なる変更がデータベースに書き込まれる場合が発生し，2 つ目の更新によって 1 つ目の更新が失われてしまう．これを防ぐため，1 つ目の更新時に 2 つ目以上の更新(データベースにアクセス)が出来ないようにしている．そのため，1 つ目の更新が完了しなければ，業務依頼の修正(データベースにアクセス)が出来なくなってしまう．これを解決する方法を現在調査中である．

また，セッション管理を用いてプログラミングを行っているが，1 台の PC 上異なるブラウザを起動しアクセスすると，重複登録が防止できないので対策が必要である．データベースでは正規化の検証，動作設定のチューニングが，プログラムでは MCV(Model View Controller)化を進める必要である．

5. 終わりに

現在は業務運用・管理システムを構築し運用に至っており，大きなトラブルもなく稼働している．今後はセキュリティ対策も考慮しつつ，ユーザが利用しやすいシステムへの変更やプログラムの修正等を行っていく必要がある．またデータベースと 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っていきたい．

参考文献

[1] 廣川類・桑村潤・小山哲志，“PHP4 徹底攻略実践編”，ソフトバンクパブリッシング

[2] Richard Stones・Neil Matthew，“エキスパートから学ぶ PostgreSQL 活用テクニック”，インプレス