

法人運営システム導入と所内システム連携

Introduction of Independent Administrative Institutions System and Cooperation with Existing Systems

新田 仁* 石島 悌* 平松 初珠*
Hitoshi Nitta Dai Ishijima Hatsumi Hiramatsu
中西 隆*
Takashi Nakanishi

(2012年7月2日 受理)

キーワード：情報，システム，連携，オープンソース，Java，PHP，Ajax，JSONP，出退勤管理

1. はじめに

2012年4月1日，大阪府立産業技術総合研究所（以下，産技研）は地方独立行政法人に移行し，これに伴い，大阪府総務事務システムが担っていた出張休暇管理や財務管理などの総務関係の業務が，新規導入された法人運営システムに移行した。同時に，これまで大阪府総務事務システムは大阪府庁内イントラネット，研究管理等を担う産技研所内システム¹⁾は所内イントラネット，と別ネットワークに分断されていたが，法人運営システムは，所内イントラネットに配置したため，職員は，大阪府庁内イントラネット用端末と所内イントラネット用端末の間を移動する必要がなくなり，さらには，顧客情報と財務情報の連携など新たなメリットも生まれた。本稿では，法人運営システム導入の経緯や，所内システムとの連携に関連して，新たに開発したプログラムについて報告する。

2. 法人運営システムの導入

本章では，新規導入した法人運営システムの概要とその導入や運用について述べる。

2.1 法人運営システムの概要

1. 「はじめに」で説明した総務関係の業務を行う場

合，一般の企業では豊富に用意されている市販の業務パッケージシステムを導入するケースが多い。多くの市販のパッケージシステムが用意されている理由は，一つのパッケージシステムで多くの企業の業務を管理できるからである。

一方，自治体や独立行政法人などは，その数が一般的な企業に比べて圧倒的に少なく，業務システムの種類も少ない。さらに独立行政法人は，自治体と企業との中間ともいえるべき存在であり，その業務に関連する法令や規則も多岐にわたっている。そのため，既存の自治体向けの業務システムでは，我々の業務をカバーできるものが存在しなかった。そこで，自治体向けのシステムを開発している複数のベンダーに，不足している部分を開発してもらい，既存のもので足りる部分についてもカスタマイズが可能かを問い合わせた。その中で市町村などへの導入実績があり，開発期間が間に合うものを導入することとなった。

2.2 法人運営システムの導入と運用

法人運営システムは大きく以下の3つのシステムに分類することができる。

1. 人事給与・庶務事務システム
2. 文書管理システム
3. 財務会計システム

これらのうち，1. および2. は，自治体向けシステムをカスタマイズすることによって既存のシステムが利用可能であった。一方，財務会計については，会計に

* 顧客サービス室 業務推進課

関係する法令や規則が自治体に適用されるものとは異なっており、この部分の開発をベンダーに依頼した。

開発期間は平成23年度の単年度しかなく、産技研の財務会計担当者からのヒアリング、ベンダーによるコーディング、両者によるテストなどを、約8ヶ月間で行う非常に厳しいスケジュールが組まれた。また、後述のとおり、財務会計システムは、既存の所内システムとの連携が必要であり、この部分については、独立行政法人化まで2ヶ月しかない中での対応が求められ、より厳しいスケジュールとなった。

また、法人運営システムを動作させるためのサーバならびにネットワーク環境の構築と導入後の運営担当として、インフラ基盤チームが組織され、前述した3つの業務システムの担当者もそれぞれチームを組織し、計4チームでシステムの導入にあたった。

2.3 インフラ基盤チームへの参画

法人運営システムを動作させるサーバとネットワークの導入ならびに運用を担当するインフラ基盤チームには、本稿の執筆者である情報システム管理者があたることになった。このチームの最初の仕事は、ベンダーから提示された資料にもとづき、物理的・論理的にサーバをどこに配置するかを決定すること、ならびにそれらの仕様を作成することであった。

次章で説明する所内システムと異なり、法人運営システムはプロプライエタリな(オープンソースではない)システムで動作する。これまでは、このような比較的規模の大きいプロプライエタリなシステムの導入実績がなく、ベンダーとの交渉経験がほとんどなかったため、導入にあたっては十分な打ち合わせが必要であった。

3. 所内システムの概要

ここで、法人運営システムと連携する所内システムについて解説する。所内システムは、産技研の研究や技術支援業務の効率化を目的に構築され、2001年に開発を始めて、10年に渡り産技研の中核システムとして利用され続けている。

所内システムの特徴の1つ目は、その大半を産技研の情報分野の職員が開発していることである。このことは、所内システムの仕様変更や機能追加に柔軟に対応できるというメリットをもたらしている。システムが組織に根付くためには、初期の仕様策定以上に、システム運用開始後の利用者ニーズへの対応が重要になる。システム開発を外注した場合、一定期間に限り不具合の修正をしてもらえることが多いが、仕様追加な

ど、それ以外の対応については、有償になることが大半である。有償となれば、利用者の細かいニーズがある場合でも、お金がかかるため、あきらめることが多くなる。しかし、ニーズの放置は、「このシステムは使いにくい」というイメージを利用者に徐々に植え付け、最後には、誰も利用しないシステムになってしまう。職員がこのニーズに迅速に対応してきたことで、所内システムが産技研に根付いたと言っても過言ではない。

また、職員がシステムを開発することは、職員のシステム構築におけるスキルやノウハウの蓄積というメリットにもつながった。参考書や論文からは得られない、実際にシステムを構築した経験は、産技研の使命である中小企業支援に大いに貢献した。

所内システムの2つ目の特徴は、システムをWebアプリケーションによって構築したことである。Webアプリケーションとは、ホームページの仕組みを使って実現したアプリケーションソフトウェアあるいはサービスのことである。Webアプリケーションを利用する際に、クライアント側で必要なソフトウェアはブラウザだけである。このため、OSやOSのバージョンに依存しにくく、多くのクライアントPCから利用できる。また、Webアプリケーションは、システムに関わるデータを全てWebサーバ側に保持しているため、システムの仕様変更や機能追加はWebサーバ上のデータ変更だけで完了し、クライアントPCごとの作業が不要である。産技研の所内システムは、利用者のニーズに応えるために高い頻度でバージョンアップし、また、クライアントPCの数が200台を超えていることを考慮すると、システム更新の負担が少ないWebアプリケーションは、産技研の所内システムに最適の仕組みであったと考えられる。なお、Webアプリケーションの開発において、確認メッセージの表示などサーバのデータベースの参照が不要な処理については、ブラウザ上で動作するJavaScriptを用いている。JavaScriptにはブラウザの種類やバージョンによって、実装上の差異が多く存在するが、この差異に依存しないプログラミングをしたり、jQuery²⁾を利用したりすることで、この問題を解決している。

所内システムの3つ目の特徴として、システムを構成するソフトウェアにオープンソースソフトウェアを採用したことが挙げられる。オープンソースソフトウェアは無償で利用できるため、システム構築コストの大幅な削減につながる。また、単にコスト削減だけでなく、バージョンアップによる最新技術の適用効果も考慮している。さらには、オープンソースソフトウェ

アは、プログラムソースが公開されているため、内容を自由に変更できる。プロプライエタリソフトウェアの場合、仕様の変更は開発したソフトウェアメーカに委ねられるため、仕様が壁となり、求める機能を実現できないことも起こりうる。オープンソースソフトウェアであれば、プログラムソースに対する十分な理解が前提ではあるが、自ら仕様を変更し、必要な機能を実現することが可能になる。

これらの特徴をもつ所内システムは、複数のサーバによって構成される。以下でその中核を担う2つのサーバについて紹介する。

3.1 所内ポータルサーバ

所内ポータルサーバ³⁾は、所内システムのポータル(玄関)機能や、各種研究管理、技術相談管理、情報交換用の掲示板などのサービスを提供している。サーバを構成するソフトウェアとしては、OSにLinux(Ubuntuサーバ)、データベースにPostgreSQL、ロジック部にJava、Javaアプリケーションサーバにresin⁴⁾を採用している。Webアプリケーション開発に際しては、MVCモデルに基づいたプログラム構造の採用が効率的と考え、その実現を支援するフレームワークとして主流となっているStruts⁵⁾を使用した。また、通常、Strutsでは表示ページとしてJSPを想定しているが、ソースコードの視認性の良さなどを考慮し、同じJakartaプロジェクト⁶⁾の成果物であるVelocity⁷⁾を採用した。

プログラム開発は、WindowsPC上でVine Linux OSを仮想マシンとして実行し、その上でNetBeansをJava統合開発環境として利用している。OSを含めた開発環境を仮想マシンとすることで、作業を行うPCを選ばず、さらに開発環境そのものをイメージバックアップできるため、安心して開発を行うことができ

る。実行環境で使用しているJavaコンテナのresinは、NetBeansでサポートされていないため、開発環境におけるJavaコンテナはTomcatを使用している。

所内ポータルサーバのプログラム開発は、現在、複数名で行なっていることから、バージョン管理を適切に行い、デグレード(プログラム更新に伴う内容の退化や品質の低下)の発生を未然に防止する必要がある。これに対応するために、バージョン管理システムのCVSを導入した。NetBeansにはCVSを管理する機能が付属しており、別途、CVSサーバを準備すれば、比較的簡単にCVSを利用することができる。利用手順としては、図1のようにNetBeansの起動後に、チェックアウトの画面を開き、CVSサーバのIPアドレスやユーザ名などを入力した上で、“チェックアウト”を行う。これにより、CVSサーバに保存された最新のバージョンのデータを開発端末に反映することができる。ファイルの変更後は、“コミット”を実行することで、変更した内容をCVSサーバに書き戻すことができる。また、同一ファイルを同じタイミングで、複数の開発者が変更し、競合が発生した場合も、コミット時にNetBeansが相違箇所を指摘し、CVSサーバのデータとローカルのデータの不一致が解消されない限り、コミットができないように制御する仕組みとなっている。

3.2 試験・歳入管理サーバ

試験・歳入管理サーバ⁸⁾は、以下の業務を提供するものである。

1. 依頼試験
2. 機器開放・施設開放
3. 上二件の収入(歳入)

なお、収入管理とそれに付随する帳票の出力については外部のソフトウェアベンダーにその開発を依頼した。このサーバを構成するソフトウェアは以下のとおりである。

- OS: FreeBSD
- データベース: PostgreSQL
- アプリケーションロジック: PHP
- Webサーバ: Apache

Webアプリケーションの開発については、フレームワークなどの仕組みは用いていないが、個々の業務で共通する部分を部品化することによって開発の効率を高めている。利用者視点においても、負担の軽減を目的として、類似の処理を実施する場合は、過去に入力した内容を検索し、それをコピーできるようにするなどの処理を組み込んでいる。

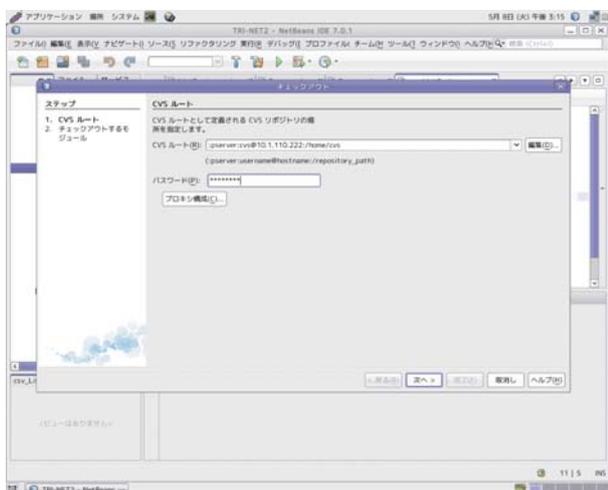


図1 NetBeansのCVSチェックアウト画面

4. 法人運営システムと所内システムの連携

前述のとおり，法人運営システムと所内システムは産技研の所内イントラネットを介してつながった．ここでは，双方のシステムの連携を支援する仕組みや，連携を活用して実現したサービスの開発事例を紹介する．

4.1 顧客情報の共有支援

依頼試験，機器開放などの有料サービスを利用した顧客への請求書発行は，法人運営システムが担う．しかし，請求書に記載する郵便番号，住所，氏名などの顧客情報は，所内ポータルサーバのデータベースに登録されている．このような場合，所内ポータルサーバに顧客情報を提供するインターフェースを用意し，法人運営システムが所内ポータルサーバの顧客情報を参照するのが一般的である．しかし，法人運営システムには外部サーバへ接続する機能がなかった．このままでは，法人運営システムと他のサーバの連携が全くできないため，法人運営システムの開発者と交渉し，CSVファイルのインポート機能が追加されることになった．この機能は，法人運営システムが提供するファイル共有フォルダに顧客情報を記録したCSVファイルを保存し，取り込み処理を行うことで，法人運営システム内のデータベースに顧客情報を流し込むものである．取り込み処理を手動で行う必要はあるが，手入力での顧客情報を入力することに比べれば，大幅な省力化につながる．この機能を用いるにおいて，注意しなければならないのは，CSVファイルのインポート機能は，顧客情報の新規登録にしか対応できない仕様になっていたことである．つまり，顧客情報に変更があった際には，法人運営システムの顧客情報編集画面において手作業で変更しなければならない．当初の計画では，法人運営システムの運用開始前に，所内ポータルサーバの全顧客情報をCSVファイルに出力し，上記

インポート機能を用いて法人運営システムに顧客情報を一括登録する予定だった．しかし，この場合，顧客情報に変更があった場合，所内ポータルサーバでの顧客情報変更処理に加えて，法人運営システムでの顧客情報変更処理も行う必要があり，事務作業の負担が大きくなってしまった．そこで，法人運営システムの運用開始時は，システムに顧客情報を登録せず，請求対象となった顧客情報だけをその都度，法人運営システムに登録する方法を採用することにした．この方法であれば，変更がある顧客が未だ有料サービスを利用したことがなければ，法人運営システムには対象の顧客情報が登録されていないため，所内ポータルサーバの顧客情報を変更するだけで済む．

この方法を実現するためには，請求が発生した顧客を日々把握する必要があるが，この情報は，所内システムの試験・歳入管理サーバに保存されている．そこで，図2に示すように，試験・歳入管理サーバ，所内ポータルサーバ，管理端末のそれぞれに処理プログラムを用意し，法人運営システムに請求対象の顧客情報を送り込む仕組みを構築することとした．以下に，各サーバおよび，管理端末で動作する処理プログラムについて説明する．

4.1.1 試験・歳入管理サーバ

依頼試験や機器開放などの有料サービスを利用した場合，請求書発行のために，その内訳や合計金額などの情報も法人運営システムに伝達する必要がある．これについても，法人運営システムに付属するCSVファイルのインポート機能を活用することとした．請求書の発行は，前日までに処理の終了した依頼試験および機器開放などが対象となる．そこで，夜間にバッチ処理として，必要なCSVファイルを生成する機能を追加した．

CSVファイルの形式は，法人運営システムの仕様で厳密に定められており，文字数が異なる，あるいは情報の形式が違っているといった小さな不具合があっ

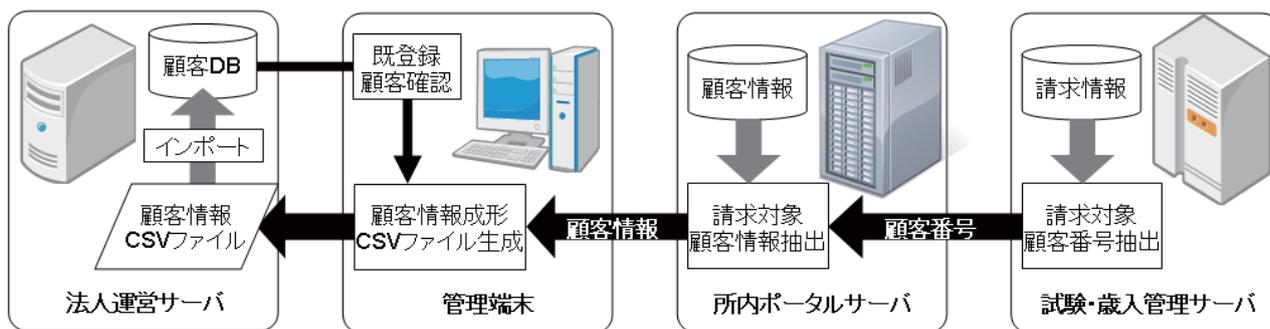


図2 法人運営サーバへの顧客情報登録の流れ

た場合、その CSV ファイルは法人運営システムでは全く処理できない。そのため、CSV による連携プログラムの開発には細心の注意を要した。

また、法人運営システムが出力する帳票のレイアウト上の問題から、一度決定した仕様から項目名の文字数に変更になるなど、ヒアリング → 開発 → テストのサイクルは年度末ぎりぎりまで続くこととなった。

4.1.2 所内ポータルサーバ

所内ポータルサーバでは、試験・歳入管理サーバが提供する未請求の顧客番号に紐付く顧客情報を生成する。顧客情報の生成プログラムは、Java サーブレットとして開発した。生成プログラムは毎日定時に自動実行するが、何らかのトラブルで顧客情報の生成に失敗した場合に、任意のタイミングでシステム管理者が生成作業を行えるように、ブラウザから簡単に実行できる Java サーブレットを採用した。このサーブレットは実行すると、試験・歳入管理サーバから HTTP 経由で顧客番号が記載された CSV ファイルを取り込み、顧客番号に紐付く顧客情報を所内ポータルサーバのデータベースから抽出し、結果を管理端末からの ODBC 接続を許可したデータベースに保存する。

顧客情報の処理においては、所内システムで顧客情報を管理するテーブルの構造と、法人運営システムの要求するデータ構造に大きな違いがあるため、これを吸収するプログラムが必要になった。特に、住所について、所内システムでは「住所」という1つのフィールドに保存しているのに対して、法人運営システムでは、「都道府県」「市区町村」「丁目」「番地」「方書」の5つのフィールドに分ける必要があった。この課題に対応するために、図3のような正規表現による住所分割を行った。

分割のパターンマッチについて、当初は、「都道府県」は「都」「道」「府」「県」のいずれかで終わる文字列、「市区町村」は「市」「区」「町」「村」のいずれかで終わる文字列、という単純なパターンで分割していた。

しかし、例えば「三重県四日市市」を上記のパターンで分割すると、「三重県」+「四日市」+「市」となってしまう、市区町村フィールドに正しい住所が入らない。この問題には、パターンで正しく分割できない住所は、先に例外住所として分割することで対応した。

政令指定都市名をパターンに含めているのは、政令指定都市については市区町村フィールドに「〇〇市〇〇区」というパターンで分割するよう指定するためである。この指定をすることで、政令指定都市である「大阪府堺市北区」は、「大阪府」+「堺市北区」と正しく分割できる。一方、政令指定都市ではない「山口県周南市公園区」の場合では、「〇〇市」というパターンで分割され、「山口県」+「周南市」+「公園区」と正しく分割できる。また、前述の「四日市市」のように、単純に分割できない市区町村名については例外指定し、個別パターンとすることで正しい分割ができるようになった。

所内ポータルサーバでは、未請求の顧客情報生成 Java サーブレット以外にも、顧客番号を指定し当該顧客の顧客情報を生成する Java サーブレットも用意した。事務担当者が法人運営システムに顧客情報の登録漏れを発見した際には、このサーブレットで顧客情報を生成し、後述の管理端末での処理を行うことで対処できるようにした。

4.1.3 管理端末

管理端末では、未請求の顧客番号に紐付く顧客情報から法人運営サーバのデータベースに登録されていない新規顧客のデータを抽出し、新規の顧客情報 CSV ファイルを生成するための処理を行う。顧客情報 CSV ファイルの生成プログラムは、VB.NET で開発した。プログラムは、請求書発行手続きの一環として、担当者が毎朝実行する。このプログラムの実行により、所内ポータルサーバのデータベースに格納されている顧客情報が取り込まれ、法人運営サーバのデータベースに既に登録されているかチェックされる。そ

```
//政令指定都市群
static final String scity="札幌市|仙台市|さいたま市|千葉市|横浜市|川崎市|相模原市|新潟市|静岡市|浜松市|名古屋市|京都市|"
+"大阪市|堺市|神戸市|岡山市|広島市|北九州市|福岡市";

//パターン例外群
static final String reigai1="市原市|市川市|野々市市|四日市市|廿日市市|"; //「市」を含む市の例外
static final String reigai2="大和郡山市|小郡市|蒲郡市|"; //「郡」を含む市の例外
static final String reigai3="杵島郡大町町|"; //「町」を含む町の例外
static final String reigai4="柴田郡村田町|佐波郡玉村町|"; //「村」を含む町の例外
static final String reigai5="大島町|八丈島八丈町|"; //郡がない町の例外
static final String reigai6="御蔵島村|神津島村|青ヶ島村|小笠原村|新島村|三宅島三宅村|利島村|"; //郡がない村の例外
//例外の結合文字列
static final String reigai=reigai1+reigai2+reigai3+reigai4+reigai5+reigai6;
//正規表現マッチパターン
static final String pattern="(.{2,3}(?<[市])[都道府県])?¥¥s?(?:(.+(?<"+scity+"")[^区]+[区])|"+reigai
+"|(?:[1,3][区])|(?:[1,3][区])|(?:[^郡]+郡+村|町]+[村])|(?:[^郡]+[郡][^町]+[町])|"
+"(?:[^市][1,6][市])|(?:[1,15])?¥¥s?(.[1,20])?¥¥s?(.[1,15])?¥¥s?(.*)";
```

図3 住所分割用正規表現マッチパターン

して、法人運営サーバのデータベースに格納されていない顧客情報があれば、その顧客情報を顧客情報 CSV ファイルに書き込む。その際、法人運営サーバの顧客 DB に登録できるよう指定されたフォーマットで書き込む。また、新規に登録するデータの抽出だけでなく、すでに法人運営サーバに登録されているデータがある場合、住所や部署に差異がないか比較する処理も行っている。差異がある場合は、後で手作業で修正する必要があるため、そのデータを別ファイルに保存する。作成された顧客情報 CSV ファイルのデータは、法人運営サーバ上で稼働するプログラムを介して、顧客 DB に登録される。

4.2 住所変更警告機能

所内システムでは、顧客の来所時に図 4 に示す来所受付画面を用いて来所受付処理を行っている。新規顧客の場合は、会社名、氏名、住所などの情報を入力し新規登録する。そして、顧客番号の情報が含まれたバーコードを印刷した顧客カードを渡す。次回以降の来所時には、顧客カードを提示してもらい、バーコードリーダーで顧客番号を読み取り、来所受付処理を行う。その際、住所などの顧客情報の変更依頼があった場合は、図 4 の画面上で当該情報を書き換えて登録することで顧客情報が変更される。また、法人運営システムに登録済の顧客の場合は、前述のとおり、別途、法人運営システムの顧客情報の変更作業が必要になる。

ここで注意しなければならないのは、有料サービスを利用後、まだ、請求書を発行していない顧客の住所を変更する場合である。すでに会社の全機能が移転している場合は新住所に請求書を送るべきだが、費用支出を担当する部署が移転していない場合には旧住所に送る必要がある。この確認を受付で行うために、来所受付画面で顧客情報を変更する際に、未請求のサービス実績のある顧客については、事前に警告を行う“住所変更警告機能”を構築することとした。

図 5 に、住所変更警告機能を実現する構成を示す。来



図 4 来所受付画面

所受付処理は所内ポータルサーバが担っているが、サービスの利用実績は試験・歳入管理サーバのデータベースに保存されているため、試験・歳入管理サーバに顧客ごとの未請求のサービス利用実績の有無を確認できるインターフェースを PHP プログラムで用意し、それを所内ポータルサーバから参照する構成とした。所内ポータルサーバから上記インターフェースを参照するための方法として、jQuery による Ajax を採用した。これは、来所受付画面で登録ボタンを押した際に、画面遷移することなく、住所変更警告を表示するための措置である (図 6)。一般に、JavaScript においては、通常、セキュリティ対策のため、別ドメインからのデータ取得ができない。所内ポータルサーバと試験・歳入管理サーバの間の通信もこの制約を受ける。この制約を回避する方法として、JSONP を採用することとした。JSONP は、HTML における Script タグの src 属性のデータについては、上記制約を受けないことを利用し、別ドメインのサーバとの JavaScript による通信を

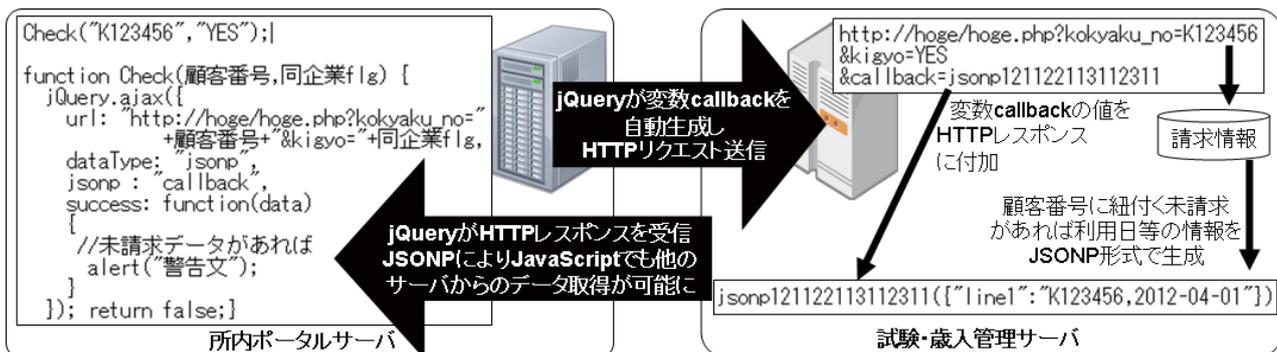


図 5 住所変更警告機能の構成

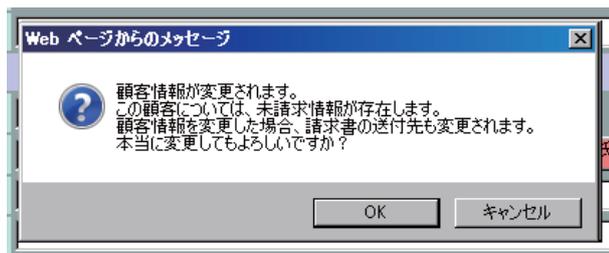


図6 住所変更警告表示

可能にする手法である。

JSONP を利用するためには、コールバック用の変数の準備が必要になる。例えば、所内ポータルサーバの jQuery において、コールバック用変数の名前を "callback" と指定した場合、jQuery は変数 callback にランダムな値を与えた上で、試験・歳入管理サーバに変数 callback を GET パラメータとして送信する。次に受け側となる試験・歳入管理サーバでは、インターフェースプログラムが、未請求のサービス利用実績の有無に関する応答値と変数 callback の値を JSONP 形式に成形した上で、所内ポータルサーバに送信する。このデータを受けた所内ポータルサーバの jQuery は、送信した変数 callback の値と、受信した JSONP 形式のデータに付与された値が一致することを確認した上で、受信データに含まれる未請求のサービス利用実績の有無に関する応答値を取り出す。その結果、未請求のサービス利用実績が存在する場合は、図6に示すような住所変更警告を表示し、作業者にその対応を促すようにする。

なお、住所変更警告機能の構築においては、参照先の試験・歳入管理サーバがイントラネット上にあるため JSONP を採用したが、インターネットに公開しているサーバについては、クロスサイトリクエストフォージェリの脆弱性による情報流出の可能性があるため、JSONP の採用においては、十分な配慮が必要である。

4.3 出退勤管理システム構築

今回導入した法人管理システムでは、職員が出勤あるいは退勤した時間をネットワーク接続型のタイムレコーダー (Online Time Recorder: OTR) を使って処理するシステムがオプションで用意されていたが、このオプションは諸般の事情で付加することができなかった。

しかし、これまでのシステムでは、職員証に磁気ストライプが入っており、これを磁気カードリーダーで読み込むことにより、出退勤の管理ができるようになっていたため、押印やオンラインでないタイムカードな

どにより出退勤を管理することには抵抗感があった。そこで、比較的低廉なシステムで新たに退勤システムを構築することとなった。

ここまでで説明したとおり、法人運営システムには、必要な情報を外部システムから CSV 形式のファイルで取り込むことが可能となっている。そこで、市販されているカードリーダーと、法人運営システムが必要とするデータ形式を変換するシステムを構築した。

用いるカードの種類は、これまで所内で鍵を管理するために用いていたカードや、交通系カード、電子決済 (電子マネー) カードとして広く利用されている Felica (フェリカ) と、職員証などの ID カードとしての利用が期待されている MIFARE (マイフェア) とした。これらのカードはいずれも非接触型の IC カードである。非接触 IC カードについては、これまでの IC カード (RFID) による研究実績⁹⁾があり、そこで得られた知識が開発に大いに役立つこととなった。また、ここで述べた IC カードは広く普及しており、それとともに、IC カードリーダーも多くの機種が発売されている。これによって、カードリーダーの選択肢が広がることとなり、結果として安価なカードリーダーを用いることができるようになった。

残念なことに、カードリーダーに IC カードをかざした際に出力される情報は、法人運営システムが要求するものと形式が違っていた。そこで、試験・歳入管理サーバにおいて、その形式を変換することにした。

形式を変換するシステムは Perl により記述した。IC カードリーダーは、カードがかざされた日時と、かざしたカードの ID を出力する。一方、法人運営システムでは、誰が何時に出勤 (あるいは退勤) 処理を行ったかを要求する。そこで、このシステムでは、IC カードの ID から職員を検索することによって、カードリーダーが出力する情報を、法人運営システムが必要とする情報に変換した。検索にはハッシュ形式のデータベースをあらかじめ用意しておくこととし、登録されている IC カードの枚数が増えても処理時間が伸びないようにした。

5. 効果と今後の課題

今回の取り組みは、職員端末での総務関係の手続きを可能にただけでなく、総務事務作業の軽減にもつながった。特に、法人運営システムへの顧客情報の登録作業の自動化は、産技研が独立行政法人化し、他の事務作業が増大する中で、総務事務担当職員の負荷軽減に大きく寄与した。また、出退勤管理システムを独

自開発したことで、所内システムの画面上で、その日の出勤打刻時間が確認できるようになるなど、独立行政法人化前にはなかったサービスを提供できるようになった。その一方で、法人運営システムの仕様の制限から、手作業でしか顧客情報の変更が行えないことなど、いくつかの課題が残っている。今後、法人運営システムの開発業者への仕様変更依頼や、仕様が変更されない場合の次善策の検討を行い、早期の課題解決に取り組む予定である。

6. おわりに

本稿では、産技研の独立行政法人化に伴い導入された法人運営システムや、その連携に関するトピックスをまとめたが、法人化に際しては、本稿の内容以外にも、産技研の大幅な組織改編や業務内容の変更に対応するために、所内システムやその他システムにおいて多くの改修作業を実施した。法人化の期限が迫る非常にタイトな日程の中での対応となったが、情報分野の職員が総出で作業を行い、情報システムの面では大きなトラブルなく法人化を乗り切ることができた。

今回の経験から再認識したのは、職員間の連携や情報共有の重要性である。本稿でも、CVSによる開発内容の共有や、サーバ間での連携などを紹介したが、これらの取り組みも、日頃、所内システム掲示板やミー

ティングを通して、情報共有や意思疎通ができていたからこそ、効率的に実現できたものである。その一方で、情報分野の職員と、総務・企画部門の職員の情報共有ができていれば、法人運営システムの導入や法人化対応スケジュールの面で、さらに効率的に取り組めた、と思返される点も多くある。

今後は、この経験を糧として、産技研の全職員が緊密な連携を行い、さらなる顧客サービスの向上につながるよう努めていきたい。

参考文献

- 1) 石島 悌, 平松初珠, 中西 隆, 袖岡孝好, 中辻秀和, 森田 均: 電子情報通信学会 技術研究報告, **106**, 34 (2006) 73.
- 2) jQuery, The jQuery Foundation: URL <http://jquery.com/>
- 3) 中西 隆: 大阪府立産業技術総合研究所報告, No.19 (2006) 9.
- 4) resin, Caucho Technology, Inc: URL <http://www.caucho.com/>
- 5) Struts プロジェクト: URL <http://struts.apache.org/>
- 6) Apache Jakarta プロジェクト: URL <http://jakarta.apache.org/>
- 7) Apache Velocity プロジェクト: URL <http://velocity.apache.org/>
- 8) 石島 悌: 大阪府立産業技術総合研究所報告, No.19 (2006) 19.
- 9) 平松初珠, 石島 悌, 得竹 浩, 砂田 茂, 石田良平, 内橋義人, 井田康人: 第8回 情報科学技術フォーラム 講演論文集, 第4分冊 (2009) 305.