MIGARO. TECHNICAL REPORT

No.4 2011年秋

[創立20周年記念号]













株式会社ミガロ.



ごあいさつ	01
Migaro.Technical Award 2011 お客様受賞論文/ミガロ.テクニカルアワード	
全社の経費処理業務を効率化した「e 総務システム」 最優秀賞 鈴木 英明様●阪和興業株式会社	04
「Web 進捗管理システム」でリアルタイム性を実現 ゴールド賞 堀内 一弘様●エスケーロジ株式会社	10
ONLY DESIGNATION OF THE PROPERTY OF THE PROPER	
「営業奨励金申請書」をたった 2 日間で開発 シルバー賞 簑島 宏明様●株式会社ケーユーホールディングス	16
A AM A MACHANIA IN THE STATE OF	
液体輸送における「配車支援システム」の構築 シルバー賞 桂 哲様●ライオン流通サービス株式会社	20
日日の水・フトリンルルルフ ことが水がられば	
Migaro.Technical Report 2011 ミガロ.SE 論文/ミガロ.テクニカルレポート	
Delphi/400 グラフ活用リファレンス 「初級者向け」 中嶋 祥子● RAD 事業部 技術支援課	26
[初級者间切] 中鳴 件寸● RAD 事業部 技術文族誌	
Delphi/400 Web サービスを利用して機能 UP!	40
[中級者向け] 福井 和彦 / 畑中 侑●システム事業部 システム 2 課	
Delphi/400 OpenOffice 実践活用	48
[上級者向け] 吉原 泰介● RAD 事業部 技術支援課 顧客サポート	
VCL for the Web 活用 TIPS 紹介	56
尾崎 浩司●システム事業部 プロジェクト推進室	
JC/400 で JavaScript 活用	65
JC/400 清水 孝将●システム事業部 システム 1 課	
jQuery 連携で機能拡張	71
Business4Mobile 國元 祐二● RAD 事業部 技術支援課 顧客サポート	

MIGARO. TECHNICAL REPORT

ごあいさつ

いつもミガロ.製品をご愛用いただき誠にありがとうございます。

弊社は1991年に設立以来、今年で20周年という節目の年を迎えることができました。ひとえに多くのお客様のご支援の賜物と感謝しております。

また、本年3月には未曾有の東日本大震災が発生しました。被災された方々には心よりお見舞いを申し上げます。大震災は、ライフスタイルやビジネススタイルにも大きな変化を与え、日本全体も大きな節目を迎えております。

このような節目の年にあたり、弊社もまた変革に向けた挑戦をしております。具体的には、ますます主流となりつつあるスマートフォンやタブレット端末に対応して Business4Mobile をはじめとする製品をリリースし、新たな業務活用のご提案をしてまいりました。

また、変革推進の一方、今まで積み上げてきた取り組みを「地道に継続していくこと」も大切にしたい、と考えております。その取り組みの1つが、技術論文集『Migaro.Technical Report』です。2008年10月の創刊以来、毎年発刊を続け、昨年は第3号まで号数を重ねることができました。お陰さまで、論文集は多くのお客様に高いご評価をいただいており、たいへん感謝をしております。

そして、今年も『Migaro.Technical Report』の第4号を無事に発刊する運びとなりました。本レポートは従来と同じく2部構成になっています。第1部は「Migaro.Technical Award 2011 お客様受賞論文」を掲載し、第2部は「ミガロ.SE論文」を掲載しています。

「Migaro.Technical Award」とは、日々アプリケーション開発・保守に携わるエンジニアの方々の努力と 創意工夫の成果を顕彰することを目的とし、「Delphi/400」「JC/400」「MKS Integrity」などの弊社製品を ご利用中のユーザー様に向けて実践レポート(論文)を公募し、厳正な審査・選考のうえ表彰する制度です。

今回の論文は、「駅すぱあと」連携で交通費精算の省力化を達成された総務システムや、上海現地法人の現地スタッフによって物流業務の作業進捗をお客様に発信できる Web システムなど、Delphi/400 の特徴を活かして業務課題を解決された事例が揃っております。

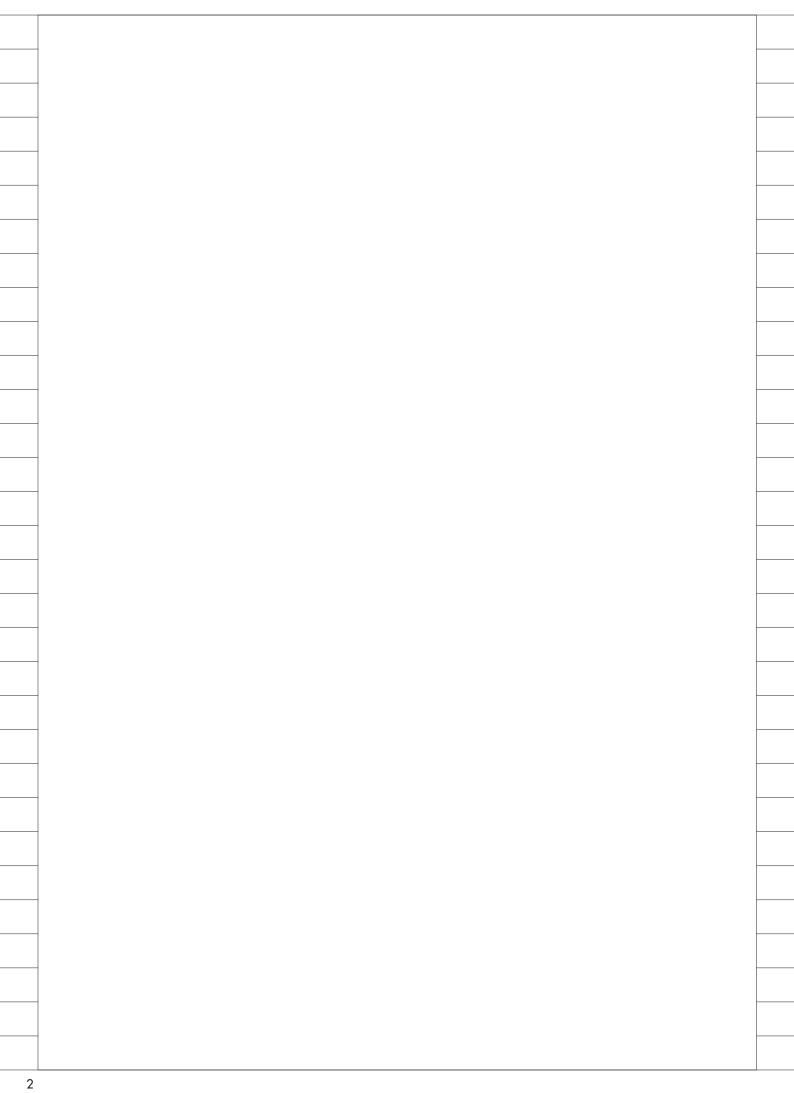
また「ミガロ. SE 論文」は、弊社 SE による技術論文を掲載しております。今回は、Delphi/400から OpenOffice や Web サービスへの連携、JC/400 および Business4Mobile の機能拡張手法などさまざまな応用テクニックを取り上げ、今後の開発に活かしていただくための技術情報として紹介しております。

本レポートが少しでも皆様の開発・保守のお役に立てれば幸いです。

最後に『Migaro.Technical Report』第4号を発刊するにあたりまして、多くのお客様・パートナー様にご支援、ご協力いただきましたことをこの場をお借りして、あらためて厚く御礼申し上げます。

2011 年秋

株式会社ミガロ. 代表取締役社長 上甲 將隆





全社の経費処理業務を効率化した「e総務システム」 一初のDelphi/400案件。IBM i 資産のGUI化モデルケースに

鈴木 英明 様

阪和興業株式会社 情報システム部 システム開発第一課



阪和興業株式会社 http://www.hanwa.co.ip/

独立系の老舗商社として、主力の鉄鋼や非鉄、水産物など各分野でシェアを伸ばしている。創業以来「流通のプロ」を基本理念に、高度な専門性と豊富なネットワークを活かし、グローバルに活動を展開している。

「総務システム」

阪和興業株式会社は、主力の鉄鋼をは じめ、非鉄、機械、石油、化成品、水産 物、木材など多くの品目を取り扱う商社 であり、1,000 名を超える社員が、日本 全国および世界で日々活動を行ってい る。

本稿では、IBM i の 5250 画面で稼働していた「(旧) 総務システム」を、GUI 画面の「e 総務システム」にリプレースして、全社の業務効率を改善した事例について紹介する。

最初に、総務業務と「総務システム」 の取扱範囲について簡単に説明したい。

当社の総務業務の中心は、社内の各営業所・事業所で日々発生する経費の処理業務である。経費は大別すると、出張費のように社員自身が使用する費用と、接待費・贈答費のように、当社のお客様に向けた費用に分類することができ、いずれの経費処理も総務業務として「総務システム」での管理を行っている。

なお、社内総務を所管する「総務部」

には多種多様な役割があり、それらの業務も総務業務と位置づけられるが、今回ご紹介する「総務システム」には含んでいない。「総務システム」は、当社の全社員の経費管理を主目的とするシステムと理解していただきたい。

社内からの声: 5250 画面の GUI 化と 「駅すぱあと」 連携

当社では、S/38の時代から、AS400、System iを経て現在のIBM iに至るまで、20年以上、一貫してIBMのミッドレンジサーバーをメインで使用している。経費関連業務についても、RPGで開発して5250画面で運用する「(旧)総務システム」が稼働していた。

一般の業務管理システム、例えば、輸入・輸出業務等の取引を管理するシステムでは、業務入力を行う人間は特定の事務担当者に限定されている。対して、総務システムの場合、営業担当者や社内従業員などすべての社員が、経費申請・報

告等のシステム画面に直接入力する。このため、5250画面のシステムの入力に普段からなじんでいない(Windows画面には慣れている)営業担当者などの間で、「(旧)総務システム」の使いにくさを指摘する声が多かった。

また、出張報告時の交通費にかかわる 煩雑さについては、営業担当者などの経 費申請者からも、チェックする側の総務 担当者からも、大きな問題として指摘さ れていた。この問題の解決手段として、 「駅すぱあと」と「総務システム」との 連携は、最重要の要件となった。なお「駅 すぱあと」とは、入力した出発地と目的 地をもとに、所要時間、所要金額を含む 最適な経路情報を応答してくれるソフト ウェアである。

以上を踏まえ「総務システム」の改善検討チームにより、「総務システム」をGUI化して「駅すぱあと」とも連携する計画を立案した。特に「駅すぱあと」との連携は、これができなければシステム再構築の意味がない、というほどの最重要課題であった。



図2





さて、ちょうどその頃、ミガロ.の開発ツール「Delphi/400」の紹介を得た。

そこで、当社の要件を満たすかどうかを検討。その結果、Delphi/400なら、Windowsに慣れたエンドユーザーを満足させる画面が開発できるうえ、既存のRPGプログラムやデータベース等のIBMi資産を有効に活用でき、将来的に自社で保守していくことが可能と判断できたので、導入を決定した。また、Delphi/400から「駅すぱあと」を呼び出すことにより、「総務システム」と「駅すぱあと」の連携を実現できることも、事前に確認した。

上記の経緯により、2006年から Delphi/400で開発に着手し、2007年に リリースしたのが「e 総務システム」で ある。

ユーザー要件と 「e 総務システム」

e 総務システムで実現したユーザー要件としては、以下が挙げられる。

- 5250 画面になじみのないユーザーへの対応
- 2. ワークフロー機能の維持・改善
- 3. 関連画像の表示
- 4. 関連 Web サイトの表示
- 5. 出張旅費精算の「駅すぱあと」連携
- 6. 出張報告での Excel データの取り込み

いずれも、従来の 5250 画面では実現できなかったニーズを、Delphi/400 でGUI 化したことにより実現した。以降より1つずつ説明する。

1. 5250 画面になじみのないユーザー への対応

5250 のグリーン画面から Windows ライクな GUI 画面へ移行したことは、コンピュータ入力を主業務にしていない 営業担当者等にとって、大きな改善ポイントとなった。

さらに、画面設計に関して以下のよう なルールを定めて各画面の使用感を統一 し、使い勝手を向上させた。

(1) 使用フォント:

一定以上のフォントサイズに統一し、

見やすさを確保する。基本は MS ゴシック 12 ポイント、タイトルエリアは 14 ポイントにした。

(2) コンポーネント規則:

ラベル、コンボボックス、ラジオボタン等の使用方法を統一する。

(3) 画面基本レイアウト:

タイトル、使用モード、時刻、メッセー ジ等の出力方法を統一する。

2. ワークフロー機能の維持・改善

出張費や贈答費などの経費処理は一般に、[担当者から決裁者への事前申請] → [決裁者の承認] → [経費の使用] → [報告] という流れをとる。「e 総務システム」においても「(旧) 総務システム」と同様、このワークフローに従って、①申請②承認③報告の登録が可能な仕組みを作成した。

「e 総務システム」ではさらに、以下のポイントに配慮し、使い勝手を向上させた。

- (1)「検索内容一覧」画面では、承認待 ちデータの検索が効率よく行え、一覧性 のある画面で表示できる。【図 1】
- (2)「申請承認画面」では、多数の承認者とその承認レベルを、1画面で表示できる。【図 2】

3. 関連画像の表示

顧客への贈答品を選択・手配する「贈答申請画面」では、各手配品の欄に実際の品物の写真を表示できるようにした。 【図3】

これにより、具体的な画像を見ながらの操作となるため、贈答品の選択が効率 化し、また、誤った品目を申請してしま うなどの手配ミスの発生も防止できた。

4. 関連 Web サイトの表示

国内出張・海外出張の「申請入力画面」では、飛行機、電車、レンタカー等の関連 Web サイトを参照して、費用の確認が行えるようにした。【図 4】

従来は、出張申請登録の途中で、自分

で別画面に切り替え、各会社のWebサイトで確認する必要があった。しかし「e総務システム」では画面上から直接、Webサイトを立ち上げることが可能になり、業務効率の改善につながった。

5. 出張旅費精算の「駅すぱあと」連携

「e 総務システム」の導入効果として 最も期待されたのが、「駅すぱあと」と の連携である。

従来までは交通費の報告をする際「(旧)総務システム」とは別に、Web サイト等で金額を再確認し、結果を入力する手間が必要であった。これが「e総務システム」では「駅すぱあとSDK」で連携できたことにより、出発地と目的地を入力するだけで経路別の運賃を自動計算し、表示された計算結果から選択すると、金額欄に自動入力されるということが可能になった。【図5】

●「駅すぱあと SDK」連携の仕組み

Delphi/400から「駅すぱあと SDK」を用いて連携する仕組みについて、簡単に説明する。

「駅すぱあと SDK」とは、交通費精算システムなどと組み合わせて、企業独自のアプリケーションを構築するための開発ツールである。マイクロソフトのCOM に準拠しており、Delphi/400から呼び出すことができる。

Delphi/400での開発方法は簡単で、最初に「駅すぱあと SDK」コンポーネントを Delphi/400 にインポートさえしておけば、そのまま Delphi/400 開発画面のツールパレットに組み込まれる。以後は、通常のコンポーネントとして画面設計に利用することが可能になる。【図6】

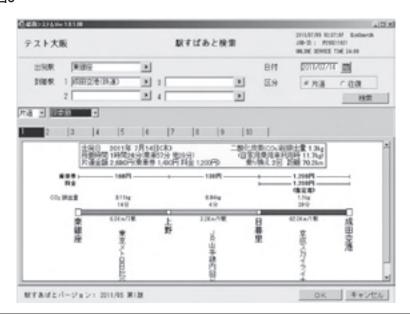
利用方法としては「出張報告画面」において「駅すぱあと選択」ボタンを選ぶと、「駅すぱあと検索」画面に遷移する仕組みとした。【図7】

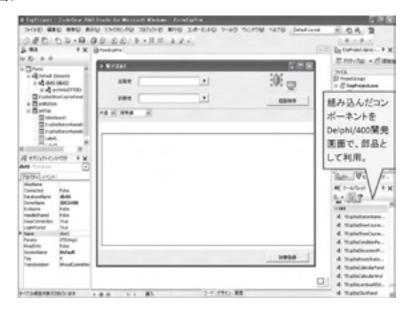
「駅すぱあと検索」画面では、入力した出発駅と到着駅情報をもとに、経路別の金額・所要時間等の情報を取得する。取得した情報の中から実際の経路を選択し、報告画面に戻ると、「駅すぱあと」の計算金額が自動入力されるという仕組みである。

さらに「(旧) 総務システム」で保有 していた社員の定期券データと「駅すぱ



図5





あと」を組み合わせることで、定期券区間分の費用を精算金額から除外するというロジックも組み込んだ。これは、交通費精算の社内規則に準拠したものだが、これにより申請者の手間を一段と削減するとともに、申請金額をチェックする総務担当者の作業も大きく省力化することができた。【図8】

なお「駅すぱあと」のデータは、実際 の運賃の改定等を反映してマスターファ イルを定期的にバージョンアップしてい る。利用者がそれぞれ「e 総務システム」 にログインしたときに、「駅すぱあと」 の最新データへの更新を促す注意メッ セージを出すという仕組みとした。【図9】

6. 出張報告での Excel データの取り 込み

海外出張の経費精算の場合、出張期間 が長期になる場合も多く、帰国後に一から精算入力を行うのは非常に大変であ る。出張中に、経費の発生都度データを 記録しておき、報告時の負荷を削減した いという要望があった。

解決方法として、海外出張中は出張者が固定フォーマットのExcelに経費データを登録できるようにしておき、帰国後、海外出張の「報告入力画面」で、「Excel取込」ボタンでExcelデータを取り込めるようにした。これにより、経費精算の手間も省力化できた。【図 10】

「e 総務システム」の評価

「e 総務システム」では、1000名を超える当社の社員全員がエンドユーザーとなる。日常的に活用されている利用度の高いシステムであるため、上述したさまざまなシステム改善により、全社の業務効率を大きく向上させ、業務支援に貢献することができた。

エンドユーザーからは、"新システムは慣れ親しんだ Windows 画面と同様の感覚で利用することができる""入力コードを記憶していなくても、リスト選択で入力できる"など、使い勝手が向上した点が評価された。また、最大の課題であった「駅すぱあと」連携による交通費精算は、特に全社員の作業の省力化につながり、非常に喜ばれている。

システム開発面では、「e 総務システ

ム」は当社での最初の Delphi/400 開発 案件であったが、これが無事に完成でき たことで、その後の IBM i の GUI 化案 件を Delphi/400 で行っていくためのよ いモデルケースとなった。

今後の展望

「e 総務システム」は、出張先や社外からのアクセスの実現など、さらなる利便性の向上を視野に入れて、今後はWeb 化なども含めたさまざまな改善を検討していきたい。

また「e 総務システム」の開発に続き、その他の総務・経理システムにおいても Delphi/400 による改善を推進中だ。今後も Delphi/400 を活用して、エンドユーザーの要望に応えるシステム開発を行っていきたいと考えている。

M



図8

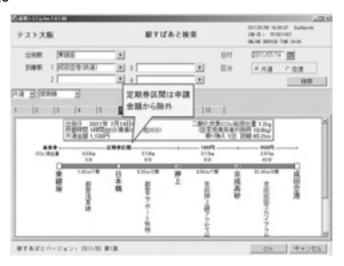
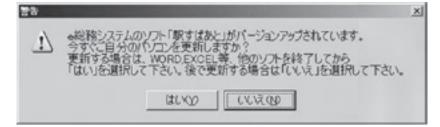
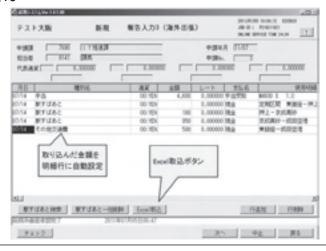


図9





「Web進捗管理システム」でリアルタイム性を実現ーリアルな物流データの提供により顧客満足度が向上

堀内 一弘 様

エスケーロジ株式会社 情報システム部 次長



エスケーロジ株式会社 http://www.sklg.co.jp/

2011 年 10 月、大阪を基盤とする物流会社 2 社(共栄倉庫株式会社・ 三和倉庫株式会社)が合併、設立。 繊維・アパレル商品の取り扱いを主体として、それぞれが保有する専門 技術やノウハウを最大限に活かして、付加価値の高い物流サービスを 提供している。

自社開発・運用の実績

※共栄倉庫株式会社と三和倉庫株式会社 は、2011年10月1日をもって合併し、「エ スケーロジ株式会社」となった。本稿は、 旧「共栄倉庫株式会社」での導入事例に ついて記述したものである。

(「当社」=「共栄倉庫株式会社」)

物流会社である当社は1947年に創業し、データ管理をコンピュータ化しはじめた時から、自社独自の倉庫管理を行うとともに、加えて荷主様からのさまざまな要望をシステムに取り入れることができるよう、すべてのシステムを自社で開発・運用してきた。

業務のシステム化については、それら 長年の経験を活かし、充実した体制を整 えながら、荷主である顧客のあらゆる要 望に応えてお客様の経営戦略を強力にサ ポートしている。

バックヤードとフロント ヤードのデータ連携

倉庫業務をコンピュータ化するにあたり、当社では、営業所別のデータはフロントヤードの処理として管理し、全社データの管理にはバックヤードとして汎用機を使用してきた。

時代の変化とともに、フロントヤードでの処理がオフィスコンピュータからWindows系のC/Sシステムに変わってきたが、汎用機のほうは過去データとの互換性を優先したために徐々にデータの連携がとれなくなってきた。

また、荷主様への要望対応には、マイクロソフトの VisualBasic 6.0 で自社開発したソフトを使用してきたが、ERPで使用している Oracle DB や VB6.0 で使用している mdb のデータを、バックヤードである汎用機に取り込むことができない状況が続いていた。

そのような折、ある荷主様から「汎用 機なみの性能を持ちながらも、Windows 系やオープン系との連携にも優れている IBM の AS/400 を使ってみては?」と 薦められた。

それをきっかけに、それまで使用してきた ERP を廃止し、VB6.0 と ODBC 経由で AS/400 の DB2 を利用するという新しいフロントヤードを導入した。

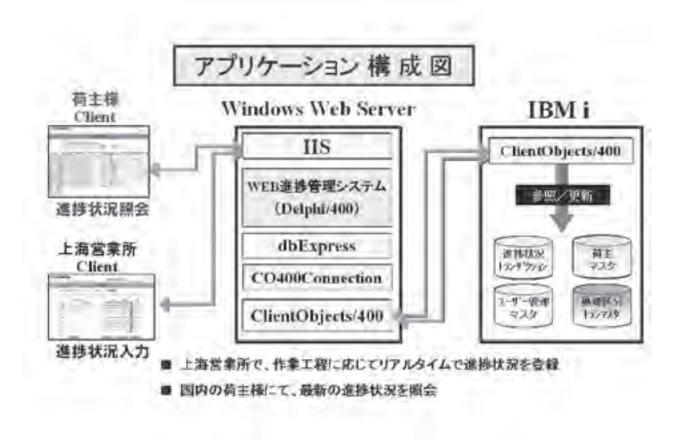
しかし、Windows 端末機の OS がどんどんバージョンアップされてゆき、端末機の OS が Windows 7 しか入手できなくなった状況の出現で、VB6.0 で開発したプログラムを使用し続けるには限界が見えてきた。

.NET vs Delphi/400

VB6.0 に代わる開発言語としては、マイクロソフトの VisualStadio.NET を採用するのが通常だろう。しかし、.NETではプログラムの記述方法が大きく変わり、Basic というよりは C 言語に近い記述になってしまっていたので、既存のプログラムをコンバートするよりは、一から書き直したほうが早いように思えた。

そのため、NET を導入すべきかどう

ネットワーク構成図 DMZ 社内LAN インターネット WEB Server Internet IBM i WEBサーバー Explorer (DB Server) IIS 7.0 データベース WEBアプリケーション DB2/400 Delphi/400 (VCL for the Web) Delphi/400 サプンステム データベースドライバ ClientObjects/400 ClientObjects/400 00411



か悩んでいたが、その時に出会ったのが Delphi/400という開発ソフトであった。

この Delphi/400 というソフトは Pascal という言語を使うという情報を ミガロ.より得て、また、Delphi/400 の開発販売はボーランドである (導入当時) と聞いた時、私が今まで探し求めて いた開発ツールかもしれないと思ったのである。

私自身はPascal という言語を使った 経験がなかったが、非常に効率のよい コードを書き出すという評判を聞いてい たし、ボーランドは優れた開発ツールを 数多く販売してきた会社と評価していた からだ。

また、Delphi/400 の 開発 方法 は、 VB6.0 の操作方法と似ており、しかも AS/400 とのデータ連携のためのコン ポーネントがいくつも用意されていて、 今まで VB6.0 で ODBC 経由のデータ操 作を記述していた苦労がかなり軽減でき ることもわかった。

いずれにせよ、荷主様個別対応のプログラムを一から書き直すなら、別々のmdbで管理しているデータを、AS/400のDB2で統一して管理できるようにプログラミングするほうがよいに決まっている。しかも、開発手法はVB6.0と似ているとなれば、習得期間も短くてすむだろうと思われた。

そして、なによりも Delphi/400 の導入を決定付けたのは「VCL for the Web」という Web 開発機能が、Delphi/400 で提供されていたことにあった。

リアルタイム・データの 要望

開発言語の老朽化を解消するという問題のほかにも、一部の荷主様より「現在の Web 情報サービスで閲覧できる情報だけでは、望んでいる処理ができない。もっとリアルタイムな情報も掲載してほしい」との依頼があり、荷主様個別対応の Web 用プログラム開発が必要となっていた。

しかし、当社でWeb用の荷主対応プログラムをVB6.0で開発するにはスキルが不足しており、また、開発できたとしても本稼働までに半年以上は必要であろうと判断した。なおかつ、Webサー

バーの OS バージョンは、Windows Sarver 2008となっていた。そのため、 VB6.0 で最新の Oracle DB を使用する 開発をしても、正常に動作するかどうかの検証にかなりの時間を割り当てなければならないだろうと推測した。

もし、Delphi/400を当社に導入すれば、開発言語やリアルタイム・データの提供といった上記の2つの問題が一気に解決される。残る問題は、Webプログラムの開発期間が3か月しかないということだけになった。

Delphi/400 による解決

Delphi/400 の導入を決めたものの、 初めて使う開発ツールで、しかもリアル タイムな「Web 進捗状況管理システム」 を3か月以内に完成させるというのはと うてい無理な話であった。

しかし、ミガロ.には開発支援のための仕組みが完備されていたので、今回のWebシステムはミガロ.に開発のサポートをしてもらうことにした。

開発前の打ち合わせでの注意点として、以下の3点について、慎重に検討してもらった。【図1】

要件

- (1) DMZ (非武装地帯) にある Web サーバーと、社内ネットワーク上にある AS/400 とを接続する。そのため、もしも Web サーバーがハッキングされてしまった場合は、AS/400 上の社内業務に影響を与えることはないのか。
- (2) Web ブラウザによる接続なので、ブラウザ画面の右上にある「×(閉じる)」ボタンで終了することが多くなると思われる。リアルタイムな更新処理を行っている時に閉じられてしまった場合、レコードロックが掛かったままの状態となり、他のユーザーからの処理に影響を与えることはないのか。
- (3) データを入力するのは上海にいる中国人スタッフだが、データを閲覧するのは日本にいる実務担当者である。基本データは日本語に限定し、現地スタッフが誤って中国語を入力しないような仕組みにしたい。

解決策

- (1) Delphi/400 のデータリンク層を使って AS/400 と 通 信 し て い る の で、 AS/400 上の社内業務に影響を与えることはない。ただし、念のためライブラリを分けて管理することにする。また、上 海の進捗状況データと社内業務データとの連携は、AS/400 側の RPG 等の別プログラムを作成して可能にする。これらにより、さらに安心なシステムとすることができる。【図 2】
- (2) AS/400 のレコードロック機能は使わずに、すべて SHARED で処理するようにする。更新処理が必要か否かを事前にチェックし、更新が必要な場合は瞬時にレコードを更新してしまう、というコーディングを行うことで対応する。
- (3) 現地スタッフが入力する項目については、コードを入力するだけのシンプルな操作画面とする。コードに対応する日本語項目などは、日本人の管理スタッフのみに操作可能な権限を与え、ログイン時に権限のチェックをするという仕組みにする。【図 3】

なお、日付入力項目等は、Delphi/400 に付属するカレンダー用コンポーネントを併用して、入力の手間を軽減するなどの処理も簡単に実装できることがわかった。【図 4】

「進捗状況照会」画面

データ登録だけでなく、データ参照方 法についても少々工夫をしてもらってい る。

今回の開発の大きな目的として、進捗 状況のリアルタイムな情報提供を可能に するのはもちろんだが、荷主様商品の荷 動きを Web 上で確認・分析できるシス テムを実現することが今回の開発の肝の 1つになっている。

そのため「進捗状況照会」画面では、 右側のグリッドには日々のリアルタイム な明細情報を、左側のグリッドには右側 に表示されている明細情報の処理区分 コード別の合計数量を表示するようにし た。【図 5】

日々の情報は増えていくばかりなの





で、明細をすべて見ようとすると限りなく上下にスクロールさせる必要がある。 そこで、左側のグリッドで知りたい情報 を選んでクリックすれば、その情報だけ を抽出して再表示するという方法を考え たのである。

通常の C/S システムのプログラミングでも、左側に表示している不特定なデータに一致するデータを、右側に再度表示するには少々テクニックを必要とする。Web ブラウザ上のこの動作をプログラムするとなると、少々のテクニックではおいつかないだろう。

とはいえ、ここでも Delphi/400 を用いた Web 開発ということで、Grid の Click イベントが使用できたため、簡単に実装することができたのである。【図6】

「CSV 出力」ボタンの 組み込み

当社では、荷主様へのサービスの一貫 として、情報を閲覧できるだけではなく、 右側の明細情報グリッドに表示している データを、CSV ファイルにしてダウン ロード可能にしている。

具体的には「CSV 出力」ボタンを設定した。荷主様がボタンをクリックすることで、顧客ごとの CSV ファイルをダウンロードでき、自社での集計や分析に利用いただけるようにした。ここにも、Delphi/400 と Windows との柔軟な連携機能を利用している。【図 7】

その他にも、ミガロ.による開発支援 では JavaScript や Cookie の利用、CSS 等も活用して、システム全体の見栄えや デザインにも工夫を加えてもらった。

ただし、プログラム開発の方法としては、コンポーネントを使用してコーディングしていくだけなので、現在の VB6.0 における開発と同等か、もしくはそれ以上に開発効率がよいかもしれない。

今後の IBM i 導入と 活用計画

- ●既存の「Web 在庫照会サービス」も、 Delphi/400 を用いて、Oracle DB から IBM i に集約していく。
- IBM i の入出庫管理システムと、各種の Web 照会サービスのマスター・

トランザクションデータについて連携を進め、リアルタイムな照会サービスへと展開させる。

- ●既存システムも含め、Delphi/400 で Excel データと連携させたり IBM i に mdb データを集約しながら、業務 効率化を推進させる。
- ●「Web 在庫照会システム」を発展させ、 Web オーダーシステムを構築したい。

メンテナンスと顧客の評価

従来のような C/S システムによる開発だと、拠点の端末機へのインストール作業が必須となる。今回「Web 進捗管理システム」の導入を実現できたことにより、上海で使用するプログラムの場合などは特にそうなのだが、現地への出張が不要となった。

加えて、プログラムのアップデート時にはサーバー側のプログラムだけを更新すればよく、メンテナンスが非常に楽になった。このアップデート時の対応スピードの向上などはメリットがかなり大きい。

また、新しく Web サーバーを立ち上げた時、今までは Oracle DB を使用するというのがほとんどであった。だが、IBM の AS/400 と Delphi/400 との連携機能を利用することで、Web サーバー側のデータベースが不要となった。さらに、Oracle のメンテナンスも不要になるため、金銭面・作業面でも Delphi/400 導入の効果は大きかった。

最後に、上海での「Web 進捗管理システム」の運用について、当初の予定ではデータ保存期間を3か月から半年ぐらいとし、それより古いデータは順次削除しようと考えていた。しかし、ご利用いただいている荷主様からは「1年以上のデータを見ることができれば、次年度の生産計画等にも利用できるのではないか」などの声をいただき、「Web 進捗管理システム」の実現とお客様ごとのデータ提供サービスを喜んでいただいている。

M



図6





シルバー賞

「営業奨励金申請書」をたった2日間で開発 一Delphi/400で、年120万円相当の営業工数を削減

簑島 宏明 様

株式会社ケーユーホールディングス 総合企画部 システムグループ 係長



株式会社ケーユー http://www.keiyu.co.jp/

ケーユーは東名高速横浜町田インターそばの本店を中心に東京、神奈川、千葉、埼玉、栃木へとネットワークを拡大している。トータルディーラーとしては、お客様の多彩なニーズに応えられることが強みである。株式会社ケーユーホールディングスは、株式会社ケーユー、他グルプ名社を管理・運営する持株会社。

2 つの目的を掲げて: ユーザー工数と開発 工数の削減

当社では14店舗で、約100人の営業担当者が中古車販売を行っている。2006年にDelphi/400を導入して、それを用いて使いにくかった5250画面の「顧客照会」および「車両照会」のプログラムを再構築し、大きな成果を上げた。

その後、業務の合間を縫って約20本のプログラムを開発してきた。中でも、特に業務改善効果が大きかったものとして「営業奨励金申請書」と「再査定書」のExcelへの出力システムが挙げられる。

これらは、開発担当者である筆者が以前、営業に所属していた際にデータの転記ばかりで無駄と思われていた事務作業を、当時の営業時代の経験を活かして、新たなソリューションとしてプログラム開発を行ったものである。

今回の開発では、開発ポイントとして、 1つ目はユーザーの事務作業工数の削 減、2つ目はできるだけ少ない開発工数 で実現する、ということを目指した。

当社では、システム部門の要員は管理職を含め3人で担当しており、システム開発から、関連会社を含めたシステム運用、J-SOX対応まで行っている。このため、できるだけ開発工数をかけずに開発を行うことは非常に重要である。

(1) 実績報告書の Excel 出力

「営業奨励金申請書」は、営業担当者がその月の販売実績(顧客名や売上金額、納品日など)と販売の詳細情報を記入し、会社へ奨励金申請をするものである。

導入前の業務

導入前の業務は、以下の2つのステップで作業を行っていた。【図 1-1】

①営業担当者は、IBM i から出力した5 つの帳票から、自分の実績を「実績 報告書」へそれぞれ転記し、奨励金 申請金額を記載して申請する。

(作業時間:約40分~1時間/人) ②事務担当者が、同じ帳票を用いて チェックを行う。

(作業時間:約1時間/人)

問題点

そもそも営業担当者による転記自体に 時間がかかり、申請作業が長時間になっ ていた。さらに、転記による記入ミス、 文字が読み取りにくいなどの問題があっ たり、計算ミスも生じていた。

そのため、事務担当者の確認作業にも 時間を要していた。

導入後の業務

導入後の業務は、以下の2つのステップで作業を行うことになった。【図1-2】

①事務担当者が「営業奨励金申請書.exe」を起動。数項目を指定し実行すると、 IBM i でデータを持っている従来の

図1-1 「営業奨励金申請書」導入前処理イメージ



図1-2 「営業奨励金申請書 | 導入後処理イメージ



<u>営業奨励金</u> 申請. Exe

IBM i のデータをExcelに重ね合わせプリンターから直接印刷



転記部分が1分程度で、すべて自動 でExcelへ出力される。事務担当者 は、出力されたExcelを確認する。

②営業担当者が、奨励金支払基準に基づき Excel のデータを調整し、最終金額を記入して、「営業奨励金申請書」の完成となる。

開発ポイント

「営業奨励金申請書」のプログラム開発は、開発開始からテストまで、たった2日で行うことができた。

開発のポイントとしては開発工数を削減できるよう、IBM i 側のデータを5つの Excel へ出力し、Excel のリンク機能で1つの Excel にまとめている。ユーザープログラム上では、最終のExcel だけを開くようにした。

導入効果:年間120万円相当の削減

まず、転記作業がなくなった。これにより、営業1人当たりの作業だけでも、 平均30分以上の工数削減となった。

これを時給 2000 円でコスト換算する と、以下のようになる。

100 人× 0.5 時間× 2000 円 / 時間

- = 10 万円 / 月
- = 120 万円 / 年

およそ月当たり10万円、年間120万円相当の大幅な削減であり、営業担当者の作業工数の削減へとつながった。

(2)「再査定書」の Excel 印刷

「再査定書」とは、半期末ごとの棚卸の際、長期在庫車両の評価を行う用紙である。中古車ビジネスは、その在庫評価額が、数か月経過しただけでも大きく変化するということが少なくない。そのため、税金が過剰に課金されないよう期末ごとに評価額を調査し、計上するということを行っている。

また中古車1台ごとに、JAAI (財団 法人日本自動車査定協会)の「カーチェックシート」という所定書式に対して、「査 定士」の資格を持つ拠点長による査定額の記載も必須となっている。

導入前の業務

①各拠点において「カーチェックシート」 に、IBMiから出力した車の情報と 車のカタログ情報を転記する。

自賠責保険や車検残日数、および残 日数に基づく残金などを、手作業で計 算して記載する。

(作業時間:約40分~1時間/人)

②「査定士」である拠点長が、実際の車 の評価(査定額)を記載する。

(作業時間:約2日/人)

- ③仕入部門でチェックを行う。
- ④経理部門でチェックを行う。

問題点

各拠点や本部では期末月(3月/9月)の繁忙期に、大量の在庫について、中古車1台ごとに車情報を転記するという無駄な業務が発生していた。

また、大量の転記データや細かい計算 のために、間違いが発生しやすいという 問題も抱えていた。

導入後の業務

導入後の業務は、以下のステップで作業を行うことになった。【図 2】

①「再査定書発行.exe」を起動し、 IBMiに登録されているデータを「再 査定書」に出力、印刷する。

自賠責保険残月数、車検残月数なども自動計算で出力、印刷される。カタログ情報のみ、手で記載をする。

- ②「査定士」の拠点長が、査定額を記載する。
- ③仕入部門でチェックを行う。
- ④経理部門でチェックを行う。

開発ポイント

所定様式「カーチェックシート」への 記載が必須のため、以下のような工夫を 行った。【図 3】

- ①スキャナで「カーチェックシート」の 所定様式を画像として取り込み、背景 を透明化した。
- ②「カーチェックシート」の画像を Excel に貼り付け、罫線に合わせてセ

ルの位置を調整した。さらに、必要な データを Excel に出力印刷した際に、 印刷位置が合うよう調整した。

③自賠責保険や車検残などのデータについては、細かい計算はプログラム側で行うことにした。これにより、計算間違いや転記ミスを防いだ。

この「再査定書発行」のシステムも、「営業奨励金申請書」同様、2日という短期間で開発を行うことができた。

導入効果: 大幅に作業時間を短縮

「再査定書」は税金の支払額に直結するため、従来は、記入作業もチェック作業も十二分に時間をかけ慎重に行っていた。

新システム導入後は、記入作業および チェック作業について、大幅に時間を短 縮することが可能になった。

まとめ: 2日間の短期開発

中古車の営業担当者が売り上げを上げるためには、1人でも多くのお客様に、少しでも多くの時間を費やして営業活動を行うことが重要である。今回の2つのシステム提供により、営業担当者の貴重な時間を無駄な作業に費やすことを防ぐことができた。

具体的には、例えば「営業奨励金申請書」のプログラム開発で、年120万円以上に相当する営業担当者の作業工数を削減することができた。

また、Delphi/400 は、IBM i との親和性が高く、IBM i のデータを利用するプログラムを効率よく開発できる。具体的には、今回取り上げた2つのプログラム開発は、どちらも2日間の短期開発で実現できた。

これからも、Delphi/400 をうまく活 用しながら、ユーザーニーズをとらえて 業務改善やサポートを行っていきたいと 思っている。

M

図2 「再査定書」導入後処理イメージ

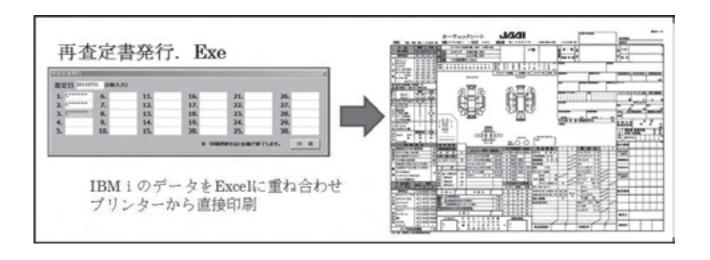
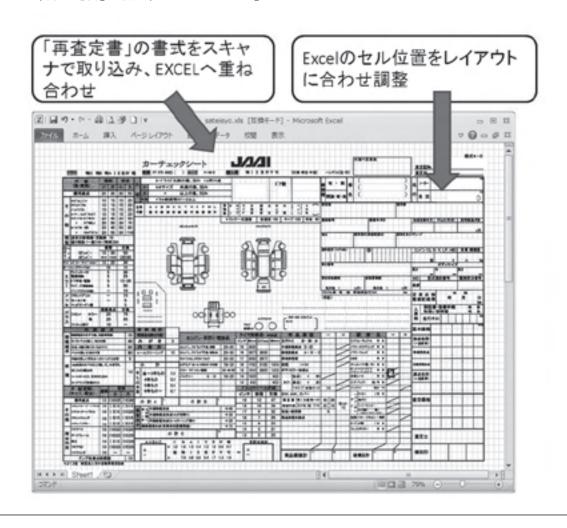


図3 「再査定書」の書式「カーチェックシート」にデータを出力



液体輸送における「配車支援システム」の構築一Delphi/400によるGUI化で、要員3名を削減

桂哲様

ライオン流通サービス株式会社 管理部 管理チーム



ライオン流通サービス株式会社 (ライオングループ) http://www.lion.co.jp/ja/ company/group/

1986年、ライオン株式会社100% 出資の物流子会社として設立。ライオングループにおいて、グループ製品の物流全般を担っており、効率化および物流品質の向上を目的に掲げ合理化を推進している。

1.「配車支援システム」 の 4 つの目的

当社は、グループ製品そのものの移送・配送業務のほかに、製品の元となる原料 (液体)の移送業務(原料工場→生産工場) を行っている。本稿では、全国 5 拠点 から構成される液体輸送業務に対して、 それを支援するために開発された「配車 支援システム」について述べる。

今回の開発ではまず、以下の4つを目的として掲げ「配車支援システム」の開発に取り組んだ。

(1) 配車業務の合理化

全国 5 拠点において、担当それそれが 同様の配車業務を行っていた。「配車支 援システム」の導入により、全社一元管 理を可能とし、要員削減を図る。

(2) 配車の効率化

配車データの計画・実績把握が容易に なるため、それをもとに効率的な配車を 実施できるようにする。 また、実車率アップにもつなげる。

(3) ノウハウ・データベースの共有化

各拠点で独自に実施されてきた配車業 務や配車計画データが一元化できる。

これにより、情報の共有や累積が容易 に行える環境の整備・構築を図る。

(4) 危機管理

各拠点で各人が各々のPCでExcelを使用し、データ作成を行ってきた。しかし新しく開発される「配車支援システム」では、Delphi/400をインターフェイスとし、AS/400に配車業務のデータを蓄積する。

この新システムの導入により、データ の堅牢性を高める。

また災害時など、一部拠点で業務遂行 が困難になった場合は、ほかの拠点での 業務運用を可能にする。

2. システム導入前

輸送元である5拠点において、担当者

が荷主からの輸送指示に基づき「出荷計画表」を Excel で作成していた。また、作成した「出荷計画表」は運送会社へFAX し、物量に応じた車輌手配を依頼していた。【図1】

「出荷計画表」の作成作業には、各拠点の担当者が1か月分を作成するのに4~5日を要していた。作成後の変更については、その修正作業に大きな工数を要していた。【図2】

3.「配車支援システム」 の開発ポイント

(1) 各開発者の個人 PC 内に Excel で作成保存されていた配車計画データを、全社一元管理の「配車支援システム」として導入し、データベースを AS/400 上に構築する。

配車計画データの作成に必要となる "発着場所""品名""運送会社""車種" などはテーブル化し、AS/400 内に保持 している。

図1 業務フロー(システム導入前)

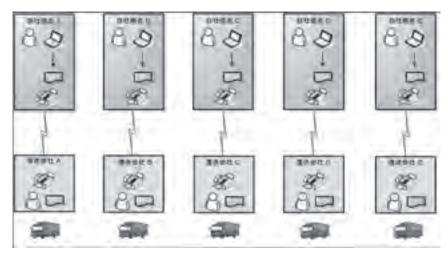


図2 出荷計画表(システム導入前)

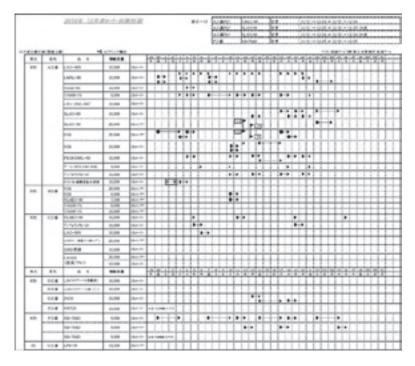


図3 配車計画入力画面



(2) 配車業務の担当者の操作性を重視 し、それを考慮した「配車計画入力」画 面を提供する。

Delphi/400 を使用することで、5250 画面の制約にとらわれずに、プルダウン 機能などを利用した画面構成になってい る。【図3】

(3) 運送会社への FAX 送信は、「FAX 送信」画面から「FAX 送信」ボタンを クリックするだけで可能となり、「出荷 計画表」が指定の運送会社へ自動で FAX 送信される。

FAX 送信のロジックには ActiveX を使用しており、PC にインストールされた FAX 送信ソフトのドライバーをコントロールして、FAX 送信を実現している。【図 4】

(4) FAX 送信される「出荷計画表」は、 「配車計画入力」画面の入力データをも とに作成されている。

この FAX 送信されるファイルの作り 込みには、AS/400 自身の機能や Delphi/400にバンドルされている RaveReports などさまざまな選択肢が あったが、レイアウトの自由度を勘案し て、見出し部を Excel で作成し、明細 部を Delphi/400 で書き出す手法を選択 している。【図 5】

4. 導入後: 要員3名の 削減

(1) 配車計画のデータ入力については Excel ベースの入力から大きく変わった。Delphi/400を介したことにより、「配車計画入力」画面において、必要項目のプルダウン選択方式を使用した入力方法に変更された。

また、日付の入力はカレンダー表示から選択するだけですむようになり、入力ミスが減った。

今回の「配車支援システム」の導入の結果、業務担当者の入力精度の向上と「出荷計画表」の作成工数の軽減に結びついた。

【効果】

「出荷計画表」作成工数:3.5日削減/月

(2) FAX 送信においては、「出荷計画表」 を送付先別(運送会社単位)に選別し、 担当者が都度送信していた。これが、「FAX 送信」画面から「FAX 送信」ボタンをワンクリックするだけで、運送会社へ自動 FAX 送信される仕組みになった。

「配車支援システム」の導入により、 FAX 送信作業の工数が軽減した。

- (3)「配車支援システム」の導入により、 配車計画の入力データは AS/400 でデー タベース化され、一元管理が可能となっ た。また、セキュリティ面の向上と実績 データの検証が行えるようになり、デー タベース管理が容易になった。
- (4)「配車支援システム」の導入により、 データベース一元化および作業工数削減 ができた。その結果、当該業務にかかわ る要員の削減も図れた。【図 6】

【効果】

液体輸送の配車業務の要員:3名削減

5. 今後の課題

Delphi/400 の導入で、配車業務の担当者に依存しない共通のデータ入力環境とデータベースの一元化が実現できた。しかし現時点では、FAX ソフトがインストールされた PC での運用しかできない。そのため、同一ネットワーク上にある FAX ソフトを介して運用できるような環境を構築していきたい。

M

ライオン流通サービス株式会社(詳細) (ライオングループ)

http://www.lion.co.jp/ja/company/group/1986年、ライオン株式会社 100%出資の物流子会社として設立。ライオングループにおいて、グループ製品の物流全般を担っており、効率化および物流品質の向上を目的に掲げ合理化を推進している。事業内容は貨物運送の取り扱いおよび倉庫業で、2010年の実績は年間売上高 146 億円、従業員数113名。

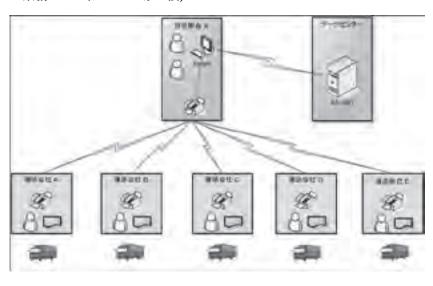
図4 FAX送信画面

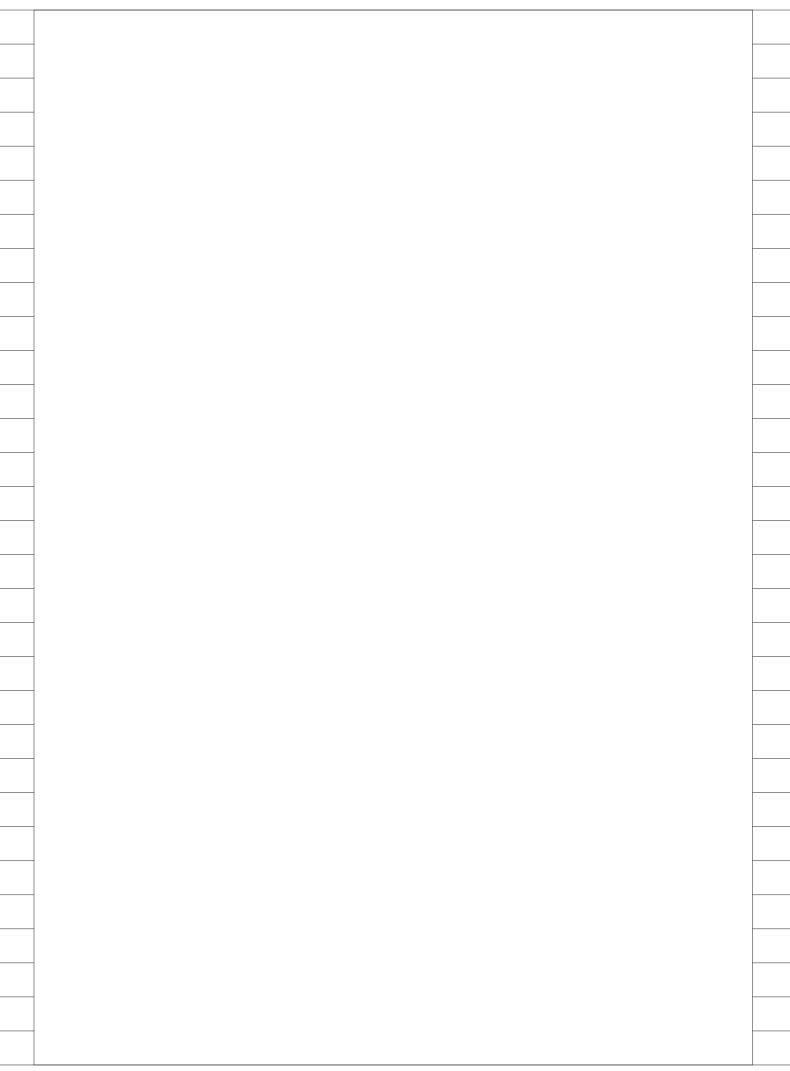


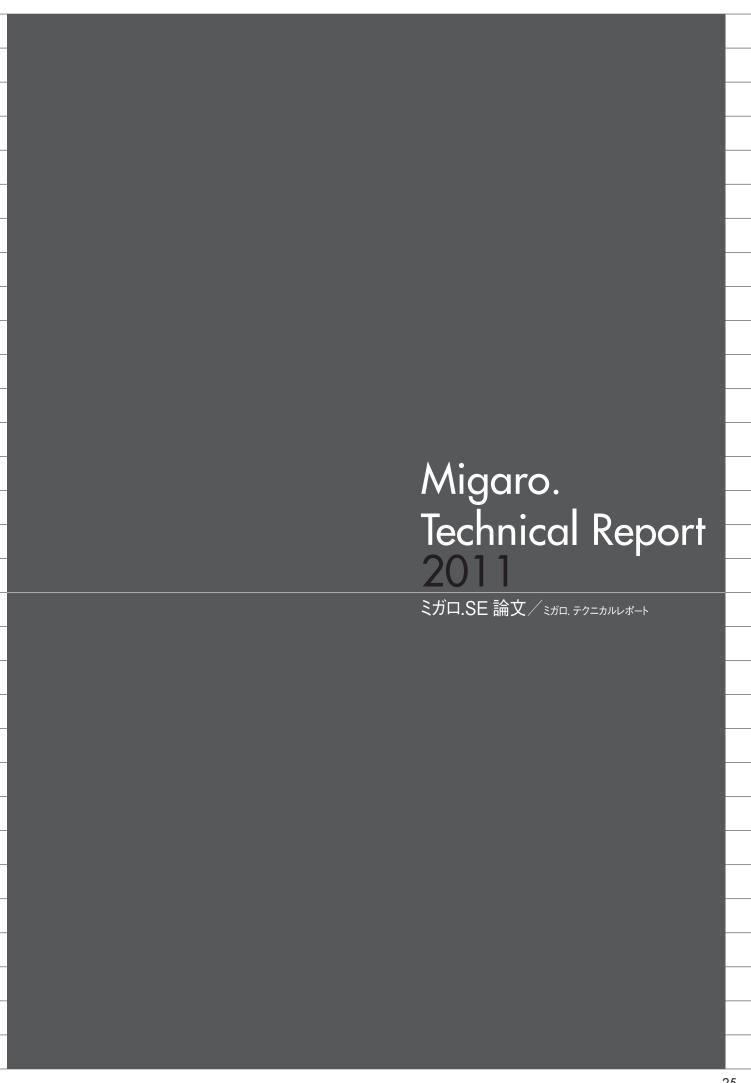
図5 出荷計画表レイアウト



図6 業務フロー(システム導入後)







中嶋 祥子

株式会社ミガロ.

RAD事業部 技術支援課

Delphi/400:グラフ活用リファレンス

Delphi/400 には、データをグラフ化するためのコンポーネント「TDBChart」がある。 その設定と活用について、基本的な操作をベースに説明する。

- ●はじめに
- ■TDBChartの基本操作
- ●TDBChartの応用
- TChart
- ●補足
- ●注意点
- ●まとめ



略歴 1968 年 02 月 23 日生 1990 年奈良女子大学家政学部卒 2002 年株式会社ミガロ. 入社 2002 年 11 月 RAD 事業部配属

現在の仕事内容

お客様からのDelphi/400 に関する 技術的な質問や問い合わせに対応し ている。また、メルマガ「Migaro News」やホームページのTips など、 開発に役立つ情報も担当。

1.はじめに

アプリケーションでは多様なデータを 処理するが、その際、処理や操作だけで なく、得られるデータをどう活用するか も重要なポイントとなってくる。 照会画 面などでデータを並べただけでは、デー タの特徴を直感的につかむことも難しい し、情報を活かしきることもできない。

それに対してグラフであれば、一目で 傾向や流れ、パターンを把握することが 可能である。

Delphi/400 には、データをグラフ化するためのコンポーネントとして「TDBChart」がある。よく使用されるTDBGridと同様に、簡単な設定で利用できる。その方法について、基本的な操作をベースにいくつかのポイントを説明する。また、「TChart」についても併せて紹介する。

2.TDBChartの基本操作

実際の操作手順を説明していこう。

フォームに TDBChart を配置する。な にも設定していない状態が図1である。 【図1】

TDBChartの設定

TDBChart をダブルクリックすると、 設定画面が表示される。ここで、実際の グラフである TSeries を追加・設定し ていく。

Add ボタンより表示されたダイアログから種類を選択すると、TSeries が追加され、図2の状態になる。ここでは、棒グラフである「Bar」を選択する。【図2】

追加された Series1 をクリックする と、TSeries の設定画面が表示される。 【図 3】

データとの関連付けは、図4のように、 Data Source タブ内で行う。【図4】

プルダウンリストより DataSet を選択し、次に DataSet: で使用するデータセットを指定する。このプルダウンリストには、そのフォームから参照できるデータセットが表示される。

次に、X 軸のラベル、X 軸のフィールド、Y 軸のフィールドを選択する。

X軸のフィールドは横軸の値になるフィールドで、それとは別のフィールドをラベルに指定できる。例えば、X軸のフィールドに店舗コードを指定すると、店舗コード順にグラフが作成される。そのときに、ラベルを店舗コード順に並び、ラベルを店舗名とすることができる。

今回のデータでは、X軸・ラベルとも 年月のフィールドとする。最後に、設定 が終了すれば、設定画面を閉じる。

以上が、データをグラフとして表示する設定である。

TDBChartの例

実際に TDBChart に表示するには、 データセットを Open する。

グラフの各座標が、データから(年月、 売上金額) = (201001,11834588)、 (201002,18078025) …と設定され、それ に応じて描画や設定が行われる。

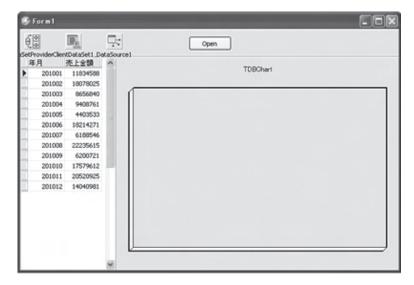
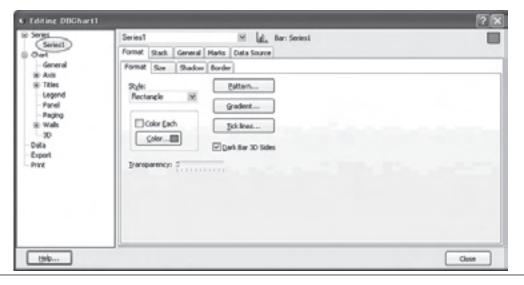


図2





また、設計画面でデータセットを Open しても、図5のようにグラフが表 示されるので、確認しながら操作するこ とができ、非常に便利である。【図5】

グラフの外観

このように、簡単な設定でコーディングなしでもデータをグラフ化できる。さらに外観を変える方法もいくつか説明する。設定と反映結果は、図6のようになる。【図6】

・タイトル

Chart の Titles から設定する。

• グラフの色

Series の Color を変更する。

・マーカー

マーカーとは、グラフから伸びた軸の 値を示すボックスである。表示値を、X 軸値か Y 軸値かを指定できる。基本操 作としてまず、表示 / 非表示の切り替え を行う。

凡例

グラフの右横にある、凡例の表示 / 非表示の切り替えを行う。

X軸のラベル

ラベルの操作は Chart の Axis (軸) に対して行うが、 Axis は 6 種類ある。 X 軸 はここでは下の軸 になるため、 Bottom Axis である。

現在は図6の通り、X軸のラベルである年月が2か月おきに表示されている。これはラベルの間隔が、描画されるラベル幅と TDBChart の幅によって自動的に調整されるためである。【図6】

TDBChart の幅を大きくしたり、ラベルのフォントサイズを小さくするなどで対応できることもあるが、設定によりすべてのラベルを表示するように変更できる。BottomAxis の Show all labelsにチェックを入れると、すべてのラベルを表示される。

だが、今回はラベルの文字列長がその幅よりも長いため、すべて表示されてしまうと、図7のように文字が重なってしまう。【図7】

すべてのラベルが表示され、文字も重ならないようにするには、Alternate にチェックを入れてラベルをずらして2段

に表示する方法か(図8)、またはラベルの Angle で回転角度を 270 度に指定することで、縦書きに見せる方法がある(図9)。これも設定画面のみで行える。 【図8】【図9】

このように、簡単な基本設定と数か所 の設定変更だけで、見栄えのよいグラフ が作成できる。

3.TDBChartの応用

TDBChartでは、複数のグラフを表示することもできる。その例として、次に支店ごとの月別売上データを用意し、設定を行う方法を説明していこう。

今回のグラフは、前述のように売上年 月別売上金額を表示し、支店ごとに比較 できるようにする。また基本操作と同様 に、グラフは売上年月を X 軸に、売上金 額を Y 軸とする。

まず支店の数の TSeries を追加し、基本操作と同様に Data Source でデータセットを設定する。 TSeries ごとに、対応する支店別の売上データを関連付ける。

データ以外に、X 軸のラベル表示など も基本操作で述べた方法で変更した状態 が図 10 である。【図 10】

マーカー

基本操作ではマーカーは非表示にした。 今回はマーカーの操作を説明するために、 表示したままにする。現在、マーカーに は X 軸の値が表示されているが、これを Y 軸の売上金額に変更する。

Style の選択値が、デフォルトの X value だと、X 軸の値が表示される。これを Value とすると、図 11 のように Y 軸の値つまり売上金額に変更され、またデフォルトでカンマ区切りの編集表示がなされている。【図 11】

X軸のラベル

次に X 軸のラベルを見ると、データ値 そのままの "201001" "201002" …となっ ている。

ラベルはフォーマット設定ができるので、見やすくするために"0000年00月"と年月のフォーマットを指定すると、図12のように反映される。【図12】

凡例

ここで凡例を見ると、TSeries 名が "Series1" "Series2" "Series3" と作成した時の名前のままで表示されている。

変更は各 TSeries に対して行うので、 設定画面で Series を選び、一覧から Seriesl を選択後、Title ボタンをクリック し、凡例に表示したい文字列を入力する。

それぞれのグラフが各支店に対応して いるので、支店名に変更しておく。

設定と反映結果は図13となる。【図13】

コードによる記述

TDBChartや TSeries に対する操作を、設定画面から行う方法で説明してきたが、 実はすべてプロパティの値として設定されている。そのため、通常のコンポーネントと同様に、オブジェクトインスペクタから設定することもできる。

図 14 は、TDBChart の BottomAxis の プロパティの一部である。【図 14】

通常のコンポーネントのプロパティと同じであるため、設計画面だけでなく、コードによってプロパティを制御することが可能である。

マーカー切替のコード例

実際にどのようなコードになるのかを、マーカーの表示を切り替えるコードを例として説明する。

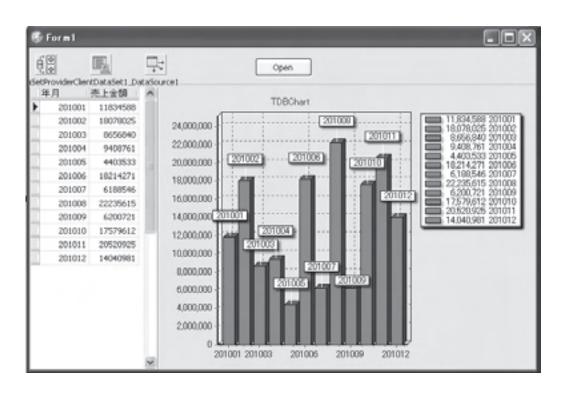
プログラムの動作として、各 TSeries に対応するチェックボックスを用意し、ここにチェックが入っていればその TSeries のマーカーを表示することとする。

チェックボックスの操作時に表示と非表示を切り替えるため、チェックボックスの OnClick イベントでソース1のように記述する。【ソース1】

これは、マーカーの Visible プロパティに、チェックボックスがチェックされれば True、されなければ False をセットしている。

TSeries のマーカーは、初期設定では表示状態になっている。起動時に非表示としたい場合、前述したように設計画面からでも行えるが、今回は Form のOnCreate イベントで行うことにする。また、このとき個々の TSeries を指定する方法とは別に、TDBChart にある TSeries





に順にアクセスしてプロパティを設定することもできるので、後者の方法を説明する。

最初の TSeries は DBChart1.Series[0]、次に Series[1] となる。また TDBChart にある TSeries の数は SeriesCount である。これらを使ってループさせることで、順に TSeries にアクセスし、そのプロパティを操作することができる。【ソース 2】

実行すると、起動時にはマーカーは表示されていない。そしてチェックボックスを操作すると、それに対応して、マーカーの表示・非表示が切り替えられていることが図15のように確認できる。【図15】

4.TChart

TDBChart は、グラフの対象となるデータセットとフィールドを設定するだけで、自動でグラフが作成される。内部的にデータを解析して行われるため、コーディングを一切行わずに作成でき、非常に便利なコンポーネントである。だが、データから自動で行われるために、想定通りのグラフとならないことがある。

そのようなケースには、「TChart」を使用するとよい。これは、TDBChart と異なり、データ連動はしない。 TChartでは、グラフのポイント(座標)はすべてコードで記述するので、任意の位置にグラフを描画することができ、 TDBChartで対応できないときには有効である。

さて、その TChart の設定は、 TDBChart と基本的に同じで、TSeries を追加する。

初期状態では、ダミーのデータがセットされている。ダミーのデータを削除するか、FormCreate 時にソース 3 のコードでクリアすることで、なにもセットされていない図 16 のように初期状態となる。【図 16】【ソース 3】

グラフの座標設定

グラフの座標を設定する方法には、い くつかの関数がある。

① AddY (ラベルなし)

ソース4のように、Y座標の値のみ を関数に引き渡す。すると、X座標値は 0から自動で連番となることが図17で わかる。【図17】 【ソース4】

② AddY (ラベルあり)

ソース5のように、Y座標の値を関数に引き渡し、また同じAddYからX座標のラベルを指定することもできる。結果は図18となり、値でなく指定した文字が表示されている。【図18】【ソース5】

③ AddXY

ソース6のように、X座標とY座標を指定すると、X座標の位置を指定できる。これにより、図19のように、連続していないグラフが作成できる。【図19】【ソース6】

また座標の追加以外、TChart の場合も TDBChart と同じく、追加した座標から自動で外観の設定が行われる。このため、タイトルやラベルの表示等を変更したい箇所については、TDBChart のようにプロパティから設定することができる。

TChartの使用例

ここで、TChart を使用する方がよい 場合を説明する。

例えば、座標が(得意先コード、売上金額)であり、データ値が(1001, 100) (1002, 200) (2001, 300) (2002, 400) となるデータがあったとする。

TDBChartでは、図20のように"1001" と "1002" は連番となるが、"1002" と "2001" の間は連番とならず、間が空いてしまう。また X 座標の差が、X 軸全体の幅に対して小さすぎるため、"1001" と "1002"、"2001" と "2002" で グラフが重なってしまっている。【図20】

これは、X 軸の値が、データ値である 得意先コードの"1001""1002""2001" "2002"であり、値が連続していないた めである。TDBChartでは、あくまで データ値からグラフの座標が決定される のであって、レコード順で連番には設定 されない。

これに対して TChart では、座標の位置を指定できるので、ファイルを読み込んで順にグラフ表示するときに、ソース7のように AddY を使って Y値とラベルを指定する。

その実行結果が図 21 である。得意先 コードが X 軸にラベル表示されながら、 レコード順に連続してグラフが作成されている。【図 21】【ソース 7】

5. 補足

今回の説明ではすべて棒グラフを使用 した。棒グラフ以外にも、多くの種類の 形状が用意されている。

TSeries を追加する場合に指定できるが、グラフを作成した後からでも図 22 のように、Change ボタンから種類を変更できる。図 23 は変更前 (Bar = 棒グラフ)と変更後 (Line = 折れ線グラフ)の画面である。【図 22】【図 23】

また立体表示されているが、平面表示にすることもできる。これも、設定画面からでも行えるし、また、TDBChartのプロパティ View3Dの True/Falseで切り替えることも可能である。切り替えた結果は図 24 となる。【図 24】

6.注意点

グラフであるため、TDBChart の TSeries でX 軸とY 軸に指定するフィールドは、数値型あるいは日付型時刻型でなければならない。

もし文字型フィールドであれば、SQL 文の INT でキャストしたり、データセットの計算項目で数値型に変換したり、連番をセットするなどの対応が必要になる。

TChart であれば、ソースコード内から設定するので、StrToInt 関数などを使用して値を変換する方法などがある。

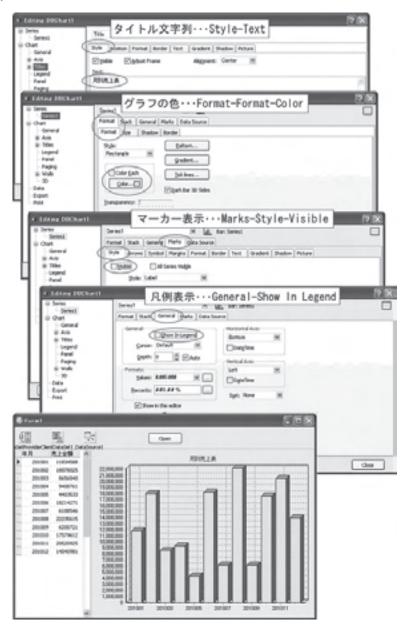
7.まとめ

TDBChart と TChart の大まかな内容を説明したが、非常に簡単な設定だけでグラフ表示できることが理解いただけたかと思う。

プロパティの数が非常に多く、慣れないうちは戸惑うかもしれないが、プロパティ構成は各軸やタイトル、凡例などの要素ごとに階層になっているので、どの要素かわかれば操作しやすい。

またプロパティが多くあることは、逆に考えると、コーディングなしでもグラフを多彩に変更することが可能ということになる。さらにコードを併用することで、利用の幅が広がることが期待できる。

M



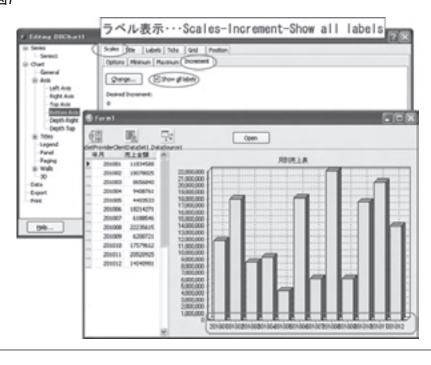
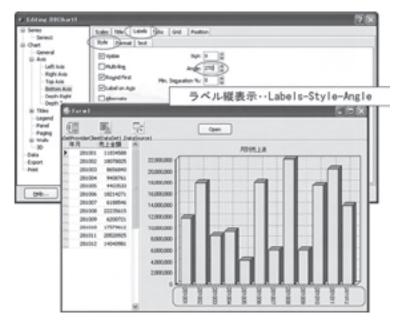
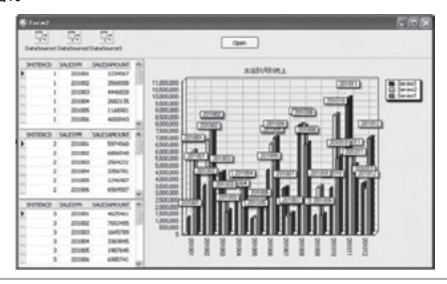




図9





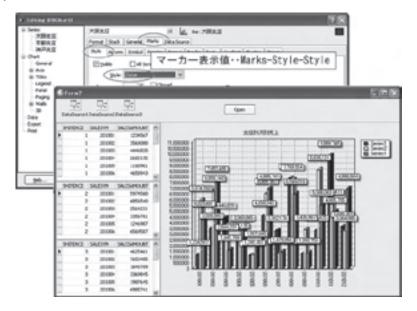
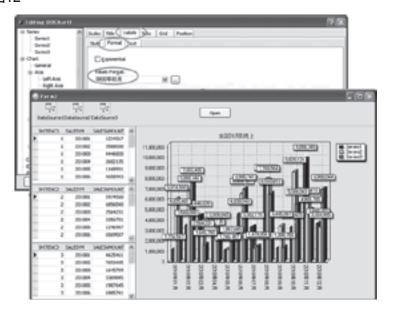
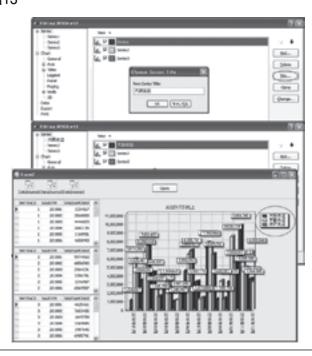
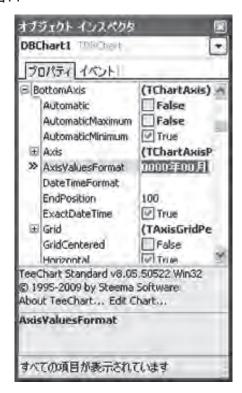
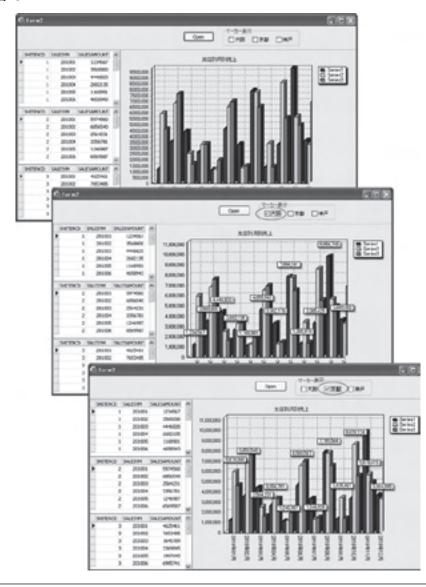


図12









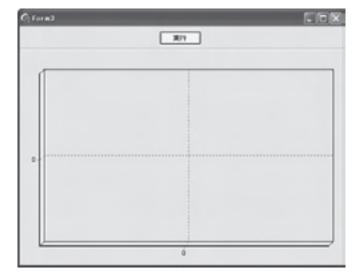


図17



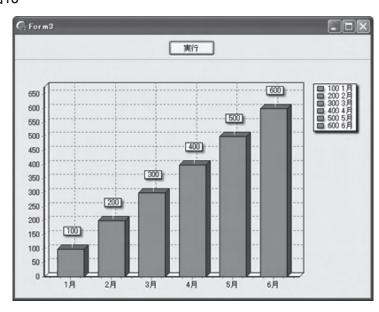
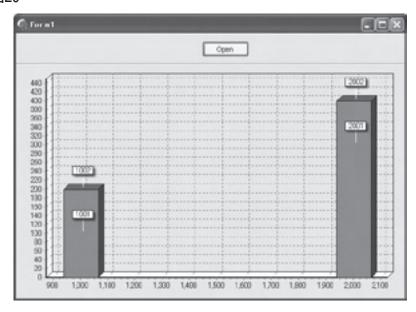




図20



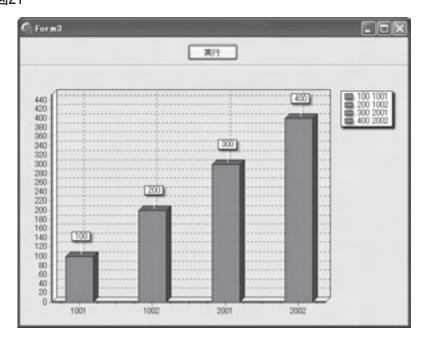
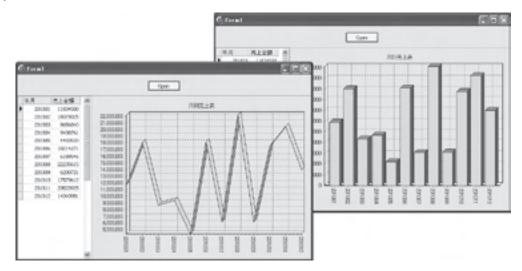
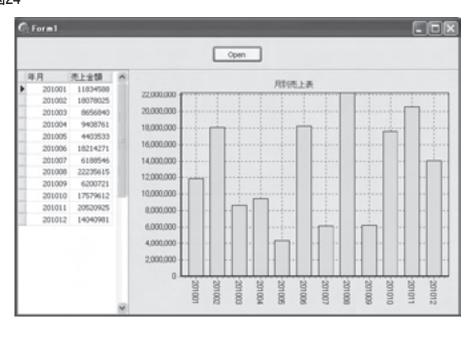




図23





```
ソース1
   procedure TForm2.ChbOsakaClick(Sender: TObject);
   begin
     Series1.Marks.Visible := ChbOsaka.Checked;
   end:
ソース2
  procedure TForm2.FormCreate(Sender: TObject):
    i : integer;
  begin
   for i := 0 to DBChart1.SeriesCount - 1 do
    DBChartl.Series[1].Marks.Visible := False;
  end;
ソース3
  procedure TForm3.FormCreate(Sender: TObject);
  begin
     Series1.Clear:
  end;
ソース4
  procedure TForm3.Button1Click(Sender: TObject);
  begin
     Series1.AddY(100);
     Series1.AddY(200);
     Series1.AddY(300);
    Series1.AddY(400);
    Series1.AddY(500);
    Series1.AddY(600);
  end:
```

```
ソース5
   procedure TForm3.Button1Click(Sender: TObject);
   begin
      Series1.AddY(100,'1月');
Series1.AddY(200,'2月');
Series1.AddY(300,'3月');
Series1.AddY(400,'4月');
      Series1.AddY(400,'4月');
Series1.AddY(500,'5月');
Series1.AddY(600,'6月');
   end:
ソース6
   procedure TForm3.Button1Click(Sender: TObject);
   begin
      Series1.AddXY( 1, 100);
      Series1.AddXY( 2, 200);
      Series1.AddXY( 4, 400);
      Series1.AddXY( 5, 500);
      Series1.AddXY( 7, 700);
      Series1.AddXY(10,1000);
   end;
ソース7
   procedure TForm3.Button1Click(Sender: TObject);
   begin
      with ClientDataSet1
     begin
        Open;
        while not eof do
        begin
          Series1.AddY(FieldByName('KINGAKU').AsInteger,
IntToStr(FieldByName('TOKUCD').AsInteger));
          Next;
        end;
     end;
   end:
```

福井和彦/畑中侑

株式会社ミガロ.

システム事業部 システム2課

Webサービスを利用して機能UP! 一既存機能に住所検索を追加

Delphi/400 で Web サービスを活用する。 この仕組みと方法を、住所検索の実装を通して紹介する。

- ●はじめに
- ●住所検索実装の課題
- ●Webサービスを利用する
- ●仕組みについて
- ●画面イメージと実装機能
- ●実装方法
- ●既存機能への組み込み
- ●最後に



略歴 福井 和彦 1972年03月20日生 1994年大阪電気通信大学工学部卒 2001年04月株式会社ミガロ. 入社 2001年04月システム事業部配属

現在の仕事内容

主に Delphi/400 を使用した受託 開発で、要件確認から納品・フォロー に至るまでのシステム開発全般に携 わる。また、Delphi/400 の導入 援やセミナーの講師なども担当。



略歴 畑中 侑 1983 年 07 月 06 日生 2006 年京都産業大学法学部卒 2006 年 04 月株式会社ミガロ・入社 2006 年 04 月システム事業部配属

現在の仕事内容

システム受託開発に携わって5年目。ミガロ・に入社し初めてプログラムを作成するも、現在は担当顧客を持ち、小規模なら中規模案件のリーダーや大規模案件のサブリーダーを務めるに至る。

1.はじめに

基幹システムにおいて、郵便番号や住所を入力するケースは少なくない。例えば、取引先マスターを見てみると、そこには郵便番号や住所といった情報を持っており、請求書や納品書への印字に利用することが多い。また発注業務においては、納入場所の指定など、郵便番号を含めた住所情報などの入力が多かれ少なかれ求められる。

ただし、これらの情報を入力していくのはけっこうな手間ではある。特にクライアント/サーバー型の基幹システムの場合、すべて手入力をしていることの方が多いのではないだろうか。もし、郵便番号から住所を検索できる機能を実装することができれば、入力の手間を大きく軽減することが実現できる。

2.住所検索実装の課題

クライアント/サーバー型のシステム で住所検索を実装しようとした場合、郵 便番号データのマスターファイルが必要となる。全国の郵便番号と住所情報は膨大な量になるが、日本郵政(郵便事業株式会社)のホームページよりダウンロード可能で、それを IBM i へ取り込むプログラムを作成することで、マスターファイルは作成できる。

だが、市町村合併等による郵便番号情報の変更に対応して、日本郵政ホームページのデータも月1回のペースで更新されている。このため、住所検索をシステムに実装したとしても、住所情報を保持し常に最新に保つことが課題となる。

しかし、これまで住所情報をすべて手入力で対応してきたのであれば、この"あると便利な機能"を、さらに手間を掛けて新たに実装するのは難しい選択であろう

3.Webサービスを 利用する

そこで本稿では、Web サービスを利用 した住所検索の実装方法についてご紹介 したい。「Web サービス」とは、インターネット技術を利用した、さまざまなサイトで提供されているサービスである。

今回題材としている住所検索についても、Web サービスを提供しているサイトがある。このサービスを利用することで、前述の郵便番号のファイルや仕組みの追加等を必要とせず、また最新データに保つためにメンテナンスに時間を割くこともなく、住所検索を実現することができるのである。

4. 仕組みについて

今回は例として、「郵便番号検索 API (※)」という Web サービスを利用して、住所検索の実装方法の説明を進めていく。

まず「郵便番号検索 API」を利用して住所検索を行う仕組みだが、Web サービスを提供している Web サーバーに対して、既定の書式の URL をリクエストする。そして、検索結果を XML 形式で取得し、検索結果を画面へ反映する流れ

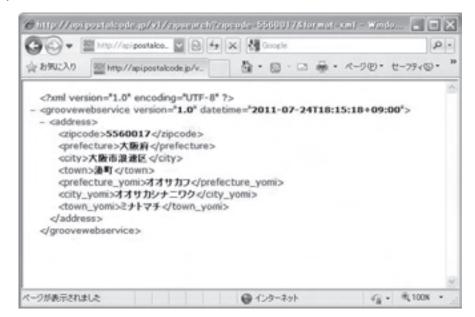


図2





<使用コンボーネント一覧>			
	コンボーネント	JOHF4 Name	
0	TPanel	Panell	
b	TRadb@roup	RadioGroupf	
ō	T5dt	5811	
ď	TBitBin	StBtrt	
	TStringOrld	StringOrlot	
f	TPanel	Panel2	
8	TDitBtn	DitDtri2	
h	TBitBtn	BitBtrd	
ŧ	TXMLDocument	YMLDocument!	
1	TIMETTE	EdHTP1	

となる。

次の例は、郵便番号 556-0017 で検索 する場合の URL である。これをブラウ ザで実行すると、【図 1】のような XML が取得できる。

この仕組みを利用した住所検索の作成 手順について、具体的に説明していきた い。

【例】郵便番号"556-0017"で検索 http://api.postalcode.jp/v1/zipsearch ?zipcode=5560017&format=xml

※利用規約について

最近のWebサービスは、今回紹介する郵便番号検索をはじめ、地図表示から、ショッピング、オークション、商品検索までその種類も範囲も多種多様である。利用するにあたっては、Webサービスを提供している提供元がその利用方法を定めている。それが利用規約である。提供元の利用規約をよく読み、理解したうえで使用していただきたい。

本稿で取り上げている、グルーブテク ノロジー提供の「郵便番号検索 API」 についても同様で、クレジット表記およ び利用規約の同意が必要である。

【郵便番号検索 API】

http://groovetechnology.co.jp/index.

5. 画面イメージと実装機能

今回作成する住所検索のサンプル画面 のイメージは、【図 2】となる。また、 使用するコンポーネントは、【図 3】を 参照していただきたい。

明細表示用コンポーネントとして、 TStringGridを使用している。このため、 明細にデータを表示させるための項目転 送ロジックを記述する必要がある。

そして、実装する検索機能としては、「郵便番号検索」と「キーワード検索」の2種類の方法を実装する。これらの機能は、Webサービス「郵便番号検索API」で提供されている機能である。

●郵便番号検索

指定する郵便番号が3桁または7桁の 場合のみ、検索結果が返ってくる。例え ば郵便番号を3桁で指定した場合は、最初の3桁がその指定と一致する郵便番号が、検索結果として返ってくる。

●キーワード検索

指定したキーワードが、"都道府県" "市 区" "町村" のいずれかに含まれている 検索結果が返ってくる。

また、検索結果を呼び出し元画面へ反映する方法としては、OK ボタンの押下時に明細で選択されている郵便番号と住所を、フォームの property 経由で呼び出し元画面へ渡す仕組みとしている。

フォームの property 設定については、 【ソース 1】を参照していただきたい。

6. 実装方法

ここから、各機能の実装方法について、 順番に紹介していく。

今回利用する Web サービスは、URL を指定すると、XML で情報を返してくれるサービスである。

XML情報とXMLスキーマ

Web サービスで提供される XML には、定義されたタグが設定されている。 【図 4】

これをスキーマ定義というが、XML 情報 (図1) を見るとわかるように、ど んな情報が存在するかは容易に確認する ことができる。

ただ、プログラムから利用するにあたり、この文字列を解析して、プログラムを作成するとなると大変である。Delphi/400ではこの解析を自動で行い、ユニットとして作成する機能がある。それが「XML データバインディング」である。

この機能はウィザード形式になっており、XMLファイルを指定することで、XML情報を簡単に扱うことのできるユニットが自動生成される仕組みである。開発環境の1機能として、Delphi/400に備わっている。

この機能を利用すれば、文字列の XML情報をプログラムで解析する必要 がなくなり、自動生成されたユニットを 利用することで、プロパティとして簡単 にデータにアクセスできる。 XML データバインディングの手順は、【図 5】【図 6】を参照いただきたい。

また今回、XML データバインディングで使用する XML ファイルについては、グルーブテクノロジーの Web サイトより取得することが可能である。取得方法は【図7】を参照いただきたい。

実際に作成されたユニットを見てほしい。XML情報で確認したタグごとに設定された項目が、自動でプロパティとしてアクセスできるようになっていることがわかるだろう。【ソース 2】

また、グローバル関数として XML 情報を引数に持ち、プロパティに格納するメソッドも自動で生成される。今回はこれらを利用することで簡単に実現することができる。【ソース 3】

コンポーネント

Delphi/400で利用するにあたり、ポイントとなるコンポーネントは2つある。 URLを指定して結果を取得する TIdHTTPコンポーネントと、XMLを 扱うためのTXMLDocumentコンポーネントである。

コーディング解説

ここから、コーディングの解説を行っていく。【ソース 4】

最初に、XML データバインディング で自動生成されたユニットを、ユニット 参照する。

①サンプル画面に表示されているように、画面上に"検索方法"として設けたRadioGroupのItemIndexにより、Webサービスで設定されているURL定型文に検索キーを埋め込む。(ソースサンプルでは、URL定型文をconst定義として記述)

●検索方法=郵便番号の場合

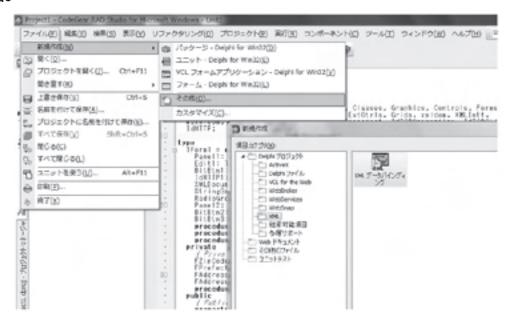
URL 定型文と郵便番号とで、リクエスト URL を作成し、変数 "URL" に格納する。

【例】

http://api.postalcode.jp/v1/zipsearch ?zipcode=556&format=xml&ie=Shift_ JIS&oe=Shift_JIS

95	(基明)	
address	性所の郵便番号の情報です。apode, presisture, city, town, prefecture yormi, city, yormi, town, yormi/ はづされます。	
nffice	事業所の信息的便事者の情報です。zigoode, prefecture, dty, town street, office name, office name, yom/か設定されます。	
ztoqude	郵便番号が、「88999981」の形式で設定されます。	
prefecture	修道府員名が設定されます。	
dey.	市区町村名が設定されます。	
6a/wr	町場名が設定されます。	
office_name	大口事業所等名が設定されます。	
street	大口事業所等名の小学名。丁目、最地等が設定されます。	
prefecture yomi	都道府県名のよみ(カタカナ)が設定されます。	
aty,yomi	市区町村名のよみ(カタカナ)が設定されます。	
town_yomi	町場名のよみくわダカナトが、原定されます。	
office frame youni	大口事業所等名のよみ(カタカテ)が設定3れます。	

図5







●検索方法=キーワードの場合

URL 定型文とキーワードとで、リクエスト URL を作成し、変数"URL"に 格納する。

【例】

http://api.postalcode.jp/v1/zipsearch? zipcode= 湊町 &format=xml&ie=Shift_ JIS&oe=Shift_JIS

- ② TIdHTTP コンポーネントの Get メソッドを、変数 URL で呼び出し、戻り値として得られる XML 情報の文字列を、TXMLDocument コンポーネントの XML プロパティにセットする。
- ③ XML データバインディングにより、 自動生成されたユニットの関数 GetgrooveWebservice の 引 数 に、 TXMLDocument を指定する。
- ④関数 GetgrooveWebservice は、郵便番号などを配列のプロパティとして持つ。そのプロパティの値を取得し、取得件数分を画面に表示する。

またここで、前述の TStringGrid へ 表示するための項目転送ロジックを記述 することになる。

7. 既存機能への組み込み

既存機能に郵便番号や住所を入力する 画面があれば、完成した住所検索を組み 込み、動作を確認していただきたい。

今回は住所検索を呼び出し、検索結果 を既存画面へ反映させる機能であった。 さらに、既存機能の郵便番号入力時に、 住所を直接取得できるようにすることも 可能である。

後者は、既存機能の郵便番号入力時に チェックとして、【ソース 5】の関数 GetAddress を組み込むことで実装が可 能である。

なお、別ユニットのため、呼び出し元 画面では uses 節への追加、関数を記述 する前のフォームの Create、および呼 び出し語のフォームの Release といっ た別途記述が必要である。

コーディング解説

①戻り値の初期化を行う。(それ以降の

記述は、前章の検索画面で解説したもの と重複するので割愛する)

②引数となる郵便番号で検索した結果、 取得できない、あるいは結果が単一でな い場合は、そのまま処理を終了する。

③検索した結果が単一である場合は、検索結果をプロパティに格納し、戻り値に True をセットする。

呼び出し元画面では、戻り値に応じて、 True の場合はプロパティから必要な情報を画面にセットし、False の場合は、 エラーとして処理するか検索画面を表示 させる。こういった制御を行えば、ユーザーにもわかりやすいと思われる。

8.最後に

Delphi/400 で Web サービスを利用 することができるということは、ご存知 の方も多いだろう。しかし、具体的にど のサービスをどのように利用すればよい のか、わからない方も少なくないと思う。

今回"既存機能に住所検索を追加する" と利用目的を明確にし、具体的な事例の 1つとして、Web サービスを利用した 機能追加を紹介してきた。本稿を通して、 Web サービスの利用方法について、少 しでもイメージしていただけたのであれ ば幸いである。

なお、Web サービスを利用する場合、サービスを提供する側の都合により仕様変更やサービスの停止が起きる可能性を認識しておく必要がある。したがって、Web サービスを利用した機能を実装した場合、定期的な Web サービス提供サイトの状況確認が必要となる。

このようなリスクはあるものの、昨今のWebサービスでは、他にもさまざまなサービスが提供されている。これらサービスを上手に利用していくことは、利便性の高いシステムを構築していくうえで手助けの1つになるのではないかと思う。

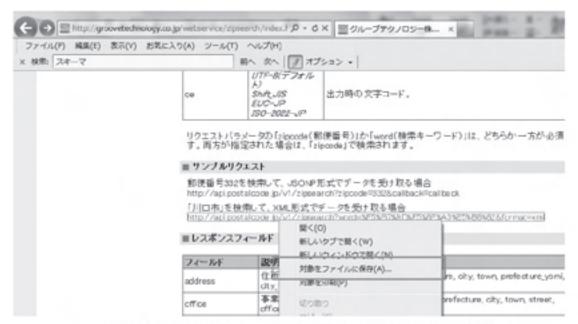
M

畑中 侑

現在の仕事内容 (詳細)

システム受託開発に携わって5年目。ミガロ.に入社し初めてプログラムを作成するも、現在は担当顧客を持ち、小規模から中規模案件のリーダーや大規模案件のサブリーダーを務めるに至る。

Delphi/400 や RPG などのプログラム開発経験を積みながらスキルを磨きつつ、お客様のご要望に耳を傾け、一歩一歩提案力をつけるための修行中の毎日。



参考URL(グループアクノロジー):http://groovetechnology.co.jp/webservice/zigsearch/index.html

```
備ホームページ 回 Unit (最 speearch)
                                         numit zipsearch;
sinterface
                                                  wses xmldom. XMLDoc. XMLIntf;
                                                   (前方宣言)
                                                                   IXMLGroovewebserviceType = interface;
                                                                   IXMLAddressType = interface;
IXMLAddressTypeList = interface;
IXMLOfficeType = interface;
IXMLOfficeTypeList = interface;
                     20
                                                        [ IXMLGroovewebserviceType ]
                                     IXMLGroovewebserviceType = imterface(IXMLNode)
['(099F2E98-49A2-4FA3-858-E0A7203F3AE2)']
| プロパティ節細胞 |
| function Get Version: WideString:
| function Get Datetime: WideString:
| function Get Datetime: WideString:
| function Get Office: IXMLOfficeTypeList:
| function Get Office: IXMLOfficeTypeList:
| procedure Set Version(Value: WideString):
| procedure Set Datetime(Value: WideString):
| / メッドとプロパティ|
| property Version: WideString read Get Version write Set Version:
| property Datetime: WideString read Get Datetime write Set Datetime:
| property Address: IXMLAddressTypeList read Get Address;
| property Office: IXMLOfficeTypeList read Get_Office:
| end:
                       30
                       40
                                                      [ IXMLAddressType ]
                                                      | IXMLAddressType = imterface(IXMLNode)
['(0785088E-2051-43AE-A040-793950830E20)']
| JOIC + # ##### |
| function Get_Zipcode: Integer:
| function Get_Prefecture: WideString:
| function Get_City: WideString:
| function Get_City: WideString:
| function Get_Prefecture_you!: WideString:
| function Get_City: WideString:
| function Get_City: WideString:
| function Get_City: WideString:
| function Get_City: Your: WideString:
| procedure Get_City: Your: WideString:
| procedure Set_Prefecture(Value: WideString):
| procedure Set_City(Value: WideString):
| procedure Set_City: Your(Value: WideString):
| procepty: Incode: Integer read Get_City: Write Set_City: Prefecture:
| property City: WideString read Get_City: Write Set_City: Prefecture:
| property Prefecture: WideString read Get_City: Write Set_City: Property Prefecture: WideString read Get_City: Write Set_City: Property City: WideString: WideString: Property City: Write Set_City: Property City: WideString: WideString: WideString: Write Set_City: Property: Write Set_City: Proper
                     50
                     64
                                                        [ IXWLAddressTypeList ]
```

```
ソース4
```

```
備ホームペーク 目 UNEI
         Form1: TForm1;
                                          「XMLデータバインディング」で自動生成された
                                              ユニットを忘れず、ユニット参照する。
        implementation
        wses zipsearch:
           enst

cSearch_UPL = 'hitp://api.postalcode-jp/vl/zipsearch?';

cSearch_Type1 = 'zipcode=';

cSearch_Type2 = 'word=';

cSearch_P1 = '&format=xml&ie=Ehift_JIS&oe=Shift_JIS';
        {$R *.dfs}
                                                                                 URL定型文をconstとして記述
             S 15
         procedure Iformi.BitStnIClick(Sender: IObject);
war
i: Integer;
UBL: 8927
           I: Integer.
URL: String:
XMLResuitList: IXMLGroovewebserviceType:
        begin
//StringGridE 9 9 7 7 6
ListClear;
           //人力テェック
if (Editi.Text = '') then
begin
/エフーメッセージ
Mossagelig('検索操作を入力してください。', atError, [abYes], 0);
Exit;
   80
           //郵便番号での検索
if RadioGroup1.1temIndex = 0 them
begin
/郵便番号をセント
URL := cSearch_URL + cSearch_Type1 + (Edit1.Text) + cSearch_F1;
   90
            end:
  O.
            end:
//キーワードでの被索
if RadioGroup1.ltemindex = 1 them
            begin

//チーフードをセット

URL := cSearch_URL + cSearch_Type2 + (Edit1.Text) + cSearch_P1;
           //迷惑数程を行い、レスポンスのXMLをXMLDocument1に特納
XMLDocument1.XML.Text := [dHT1P1.Get(URL);
            // XMLDocament1のX肌から結果セットを取得
XMLResultList := Getgroovewebservice(XMLDocument1);
   0-
             //受け取ったAMLデータから住所リストきStringGridへセット
or i := 0 to XMLResultList.Address.Count - 1 do
           for | := [ to XM.
begin
//行遊加
| if | 〇 0 then
               begin
StringGrid1.RowCount := StringGrid1.RowCount + 1;
end:
              //リストへセット
StringGrid1.Cells[0, i + 1] := IntfoStr(i + 1);
  or.
              StringGrid1.Cells[1, i + 1] := Foreatfloat("000-0000",
XMLResultList.Address.Items[i].Zipcode);
              StringGrid1.Cells[2, i + 1] := XMLResultList.Address.ltems[i].Prefecture:
  120
              StringGrid1.Cells[3, i + 1] := XMLResultList.Address.Items[i].City;
              StringGrid1.Cells[4, i + 1] := XMLResultList.Address.Items[i].Town;
       end;
```

吉原 泰介

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

Delphi/400:OpenOffice実践活用

オープンソース OpenOffice.Org の「Calc」。これを題材にして、Delphi/400 において Excel 同様「Calc」もプログラムから利用できることを紹介する。

- OpenOfficeとは
- ●Delphi/400からの活用
- ■OpenOfficeのプログラム操作
- ●応用開発
- ●まとめ



略歴 1978年03月26日生 2001年龍谷大学法学部卒 2005年07月株式会社ミガロ. 入社 2005年07月システム事業部配属 2007年04月RAD事業部配属

現在の仕事内容 Delphi/400 と JACi400 の製品試験、および月 100 件に及ぶ問い合わせやサポート、セミナー講師などを担当している。

1.OpenOfficeとは

近年はWeb上から無償でダウンロードして、利用することができるソフトウェアが増えている。OpenOffice はそうした無償利用できるソフトウェアの1つである。

OpenOffice は、正式には「OpenOffice. Org(オープンオフィスオルグ)」というソフトウェア名である。名前の通り、オープンソースで開発されたオフィス統合環境ソフトとなっている。(※)【図1】

【OpenOffice.org 日本語プロジェクト】 http://ja.openoffice.org/

一般に使われているマイクロソフトの Office 製品とも非常に互換性が高く、品 質も非常によい。もちろん Excel や Word などの主要な機能(アプリケー ション)はほとんど揃っている。

例えば、Excel に対応する表計算ソフトとしては「Calc」、Word に対応するワープロソフトとしては「Writer」、

PowerPoint に対応するプレゼンテーションソフトとしては「Impress」等があり、そのほかペイント、HP 作成などの機能が用意されている。

サンプル画面は Excel と思われるかもしれないが、OpenOffice の Calc という機能である。外観も使い方もほぼ Excel と同じである。【図 2】

この OpenOffice は誰でも無償で利用できることもあり、個人だけではなく、企業や官公庁・教育機関など幅広く利用されている。マイクロソフトの Office 製品を購入して使用している場合でも、2 台目以降の PC には OpenOffice を導入することも少なくない。

また、OpenOffice はマイクロソフト 製品ではないので、Linux や Solaris な ど Windows 以外のプラットフォームで 使用できる。これも大きな特徴だと言え るだろう。

特にここ数年では、Windows2000のメーカーサポートが終了したことにより、古くなった Windows2000の PC をLinux に移行するなどのケースでも

OpenOffice が活用されている。

※本稿においては、バージョン 3.2.1 の OpenOffice.org を題材としている。

2.Delphi/400からの活用

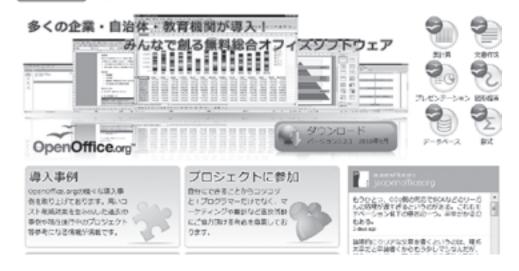
Delphi/400では、IBM i から抽出したデータを Excel にアウトプットするプログラムを作ることが多い。本稿では、Excel と互換性が高い OpenOffice のCalc を題材にしている。 Delphi/400 において、Excel 同様に Calc もプログラムから利用できることを紹介したい。

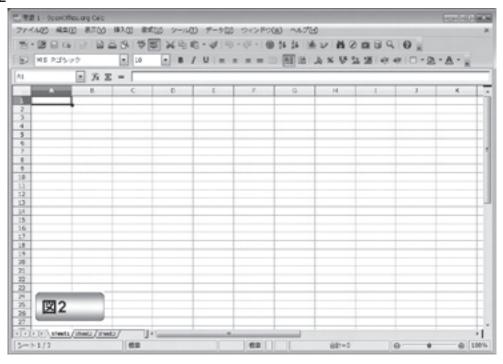
Delphi/400で Excel を利用する場合には、通常 OLE (Object Linking and Embedding) という技術を使う。OLE とは簡単に説明すると、「①別のアプリケーションソフト (Excel) の機能を、あたかも②自分の機能 (Delphi/400)であるかのように提供することができる技術」である。

つまり、Excel 側(①) には OLE で 利用できる機能が予め用意されているの









ソース1

ソース1:Calc操作を行うための共通変数・関数

```
private
vOpenOffice: Variant; //サービスマネージャ
vStarDesktop: Variant; //サービス
vOcument: Variant; //チキュメント
vSheet: Variant; //シート
function dummyArray: Variant; //空プロパティ配列設定用

//空のプロパティ配列を返却
function TForml.dummyArray: Variant;
begin
Result:= VarArrayCreate([0, -1], varVariant);
end;
```

で、Delphi/400 側 (②) はそれを呼び出して活用しているわけである。

Calc にもこの OLE の機能が用意されているので、①部分が Calc に変わるだけと考えてほしい。このようにイメージすると、Calc が Excel と同じ方法で、Delphi/400 から利用できることがわかる。

なお、本稿では、Delphi/400で Excel 操作をプログラム作成したことがある開発者向けに、OLEでの Excel 操作を類似例として取り上げ説明を加えているので、理解の一助としていただきたい。

3. OpenOfficeの プログラム操作

前章では、Delphi/400 での OpenOffice の扱い方について触れた。ここからは具体的なプログラミングについて説明していきたい。

3-1. 共通変数/関数の定義

まず、OpenOffice の Calc をプログラム上で操作するために、共通変数や共通関数を用意する。これは必ずしも必要なわけではないが、用意しておくと各プログラムで便利なので参考としてほしい。【ソース1】

共通変数については、4つ用意している。この共通変数はExcelで考えるとそれぞれ、次のような役割となる。

Calc	Excel
サービスマネージャ	Office
サービス	Excel
ドキュメント	Book
シート	Sheet

dummyArray

また1つだけ、dummyArrayという 共通関数を用意している。この関数は、 単純に空の配列(ダミー配列)を返却す るだけの関数である。この関数を用意す ると、Calcをプログラムで操作する際 に非常に便利になる。その理由を以下に 説明する。

Calc では、用意されている OLE の機能のパラメータに、Property Value と

いう構造体がよく使用されている。この パラメータは、オプション名(Name) と設定値(Value)を配列で扱う構造体 である。

例えば、ソース 2 を見てもらいたい。 【ソース 2】

このソースはファイルを保存する操作になる。この最後の行で、Store To URLという関数に、Property Valueのパラメータを渡している。この Store To URL関数を使用している行の前部分は、Property Valueの配列を作成する内容である。

Property Value は便利なパラメータではあるのだが、パラメータを必要としない場合でも、このような受け渡しのためにわざわざ配列を作成しなければならない。

そこで、前述のソース1で作成した dummyArray 関数を使ってみると、 ソース3のように簡略化することができ るのである。【ソース3】

ここまでが、プログラムの下準備となる。

●プログラム操作

続いて、具体的な Calc のプログラム 操作を説明していく。大きく分けて次の 3点の操作をおさえれば、Excel と同様 に、Delphi/400 から自由に Calc を利用 することができるだろう。

- ・起動と終了
- ・セルの編集
- ・ファイルの保存

例として、IBMiからデータを抽出 して Calc でファイル保存する、という 工程を想定して説明していきたい。

3-2. 起動と終了

Calc を起動するには、ソース4のようなプログラムになる。【ソース4】

Calc を起動する場合、まず、前述した サービスマネージャ、サービス、ドキュ メントを作成する。これは、Excel で Office、Excel、Book を作成する操作 と同じである。

プログラムでは、新しいドキュメント (Book) を作成している。もし既に作成しているファイルを読み込む場合は、ソース4でコメントをしているように、

「既存ドキュメントを読み込む場合」の プログラムコードを使うことになる。こ の時、パラメータに既存のファイルを設 定すれば、ファイルを読み込んで開くこ とができる。

次に Calc を終了する場合のプログラムは、ソース 5 を参考してもらいたい。 【ソース 5】

終了のプログラムは、起動の際に作成 したサービスやサービスマネージャを終 了して、破棄するだけである。

以上が、基本となる起動と終了のプログラム操作である。

3-3. セルの編集

次に、ドキュメントのセルを編集する プログラム操作の説明である。セルの値 を編集することができれば、IBM i か ら取得したデータを書き込むことができ る。つまり、これでデータの出力などに 利用することもできるようになる。

セルを編集するプログラムは単純で、シート上のセル(Excel と同じ考え方)を指定して、値を代入するだけである。
【ソース 6】

IBM i からデータを出力する場合は、1 レコードずつ読み込んで、このセル書き込みを応用すれば実現できる。【ソース 7】【図 3】

3-4. ファイルの保存

最後に、編集したファイルを保存する ことになる。プログラム操作は、ソース 8のようになる。【ソース8】。

ここでは、SaveDialog コンポーネントを使い、ダイアログを開いて保存するファイル名を指定させている。

Calc で保存や読み込むファイルは、標準では ".ods" という拡張子のファイルになる。もちろん Excel との互換性が高いので ".xls" のファイルを保存したり、読み込んだりすることもできる。(これについては、次章で説明する)。

注意点としては、通常 Windows のファイルパスは "¥" 形式になるが、OpenOffice でのファイルパスは "/" 形式で指定する必要がある。このソース 8 においては、SaveDialog コンポーネントで取得したファイルパスを、StringReplace という置換関数を使って

ソース2: PropertyValueのパラメータ

```
ファイル保存のAPIを利用する例
//バラメータ用プロバティ配列生成
vaProperties:= VarArrayCreate([0, 0], varVariant);
//プロパティ用オプション生成
vProperty:= vOpenOffice.Bridge_GetStruct('com.sum.star.beans.PropertyValue');
//プロパティ用オプション値設定
vProperty.Name := 'FilterName':
vProperty.Value := 'MS Excel 97';
//オプション内容をプロパティ配列に設定
vaProperties[0]:= vProperty;
//ファイルを保存
vDocument.StoreToURL('file:///' + sFileName, vaProperties);
```

ソース3

ソース3: PropertyValueのパラメータ(dummyArray)

```
ファイル保存のAPIを利用する例
//ファイルを保存
vDocument.StoreToURL("file:///" + sFileName, dummyArray);
```

オプション設定が必要ない場合、 dummyArrayで代用する

ソース4

ソース4: Calcを起動する

```
procedure TForm! .btmCalcStartClick(Sender: TObject);
 //サービスマネージャ生成
 yOpenOffice := CreateOleCbject("com.sun.star.ServiceManager");
 //サービス生成
                                                                新規ドキュメント
 vStarDesktop := vDpenDffice.oreateInstance('com.sun.star.frame.Desktop');
 //新規ドキュメントを生成する場合
                                                                 'private:factor
 vBocument := vStarDesktop.loadComponentFromURL('private:factory/scale'
                                                                y/scalc'を指定
                                            blank'.
                                         . 0
                                         , dumyArray);
                                                                 _blank:
 //既存ドキュメントを読み込む場合
  vDocument := StarDesktop.loadComponentFrontRL('file:///C:/Temp/Test.ods'
                                                                 すでにファイル
                                         . _blank
                                                                 が開かれていて
11
                                                                 も新しいウイン
                                          . dummyArray);
                                                                 ドウを使って指
 VSheet := vDocument.Sheets.getByIndex(0);
//Sheet名変更
 //アクティブなSheetを設定[1器目のSheet]
                                                                 定のファイルを
                                               既存ドキュメントを
                                                                 開く
                                               開く場合はこちら
 vSheet.Name := '新しいシート';
```

ソース5

ソース5:Calcを終了する

```
procedure TForm1.btnCalcEndClick(Sender: TObject);
begin
//サービス終了
vStarDusktop.terminate;
//サービス破棄
vStarDusktop := unassigned;
//サービスマネージャ破棄
vOpenOffice := unassigned;
end;
```

"¥"を "/"に変換している。通常は、 このプログラムコードを真似すれば問題 ない。

また Calc を終了する際に、ファイル変更の警告ダイアログを出したくない場合は、Modified := False というプログラムコードで、ファイル変更なしに設定しておくとよい。

以上で、Delphi/400 から OpenOffice の Calc を操作して、抽出データをファイルに出力することができた。

4. 応用開発

前章では、Delphi/400からの基本的な Calc の操作方法を説明してきた。この章では、Calc を業務アプリケーションでさらに活用するためのテクニックを何点か紹介したい。

4-1. xls形式での保存

Calc での標準ファイル形式は、前述 したように ".ods" という拡張子になる。 もちろん Excel のファイルを開いたり、 保存したりすることもできる。

ソースを見てもらうとわかるが、基本のプログラムは".ods"での保存と同じである。異なる部分に注釈を入れているが、PropertyValueを作成して、パラメータにファイル形式がExcelであることを設定している。このパラメータによって、本来".ods"形式の保存を".xls"形式で保存することができるのである。【ソース9】

4-2. pdf形式での保存

Calc では、Excel と同じように ".xls" 形式で保存できることを説明したが、実は ".xls" 形式だけでなく、".pdf" 形式での保存も可能である。

つまり、作成した出力データを Excel 用にも、PDF 用にも加工することがで きる。プログラムはソース 10 を見ても らいたい。【ソース 10】

pdfファイルを作成するとなると難しいプログラムを想像するかもしれないが、ソース9とソース10を比べると、実は1行しか差異がない。具体的には、PropertyValueでExcel形式を指定していた内容が、PDF形式に変わっただ

けである。これだけ指定すれば、PDF の変換処理は Calc がやってくれるのである。

簡単かつ PDF が作成できる実践的な テクニックなので、ぜひ活用いただきたい。

4-3. プリンタへの出力

続いて、ファイルとして保存するだけ でなく、プリンタへ直接印刷するプログ ラムを説明する。

ソース 11 は、プリンタへの印刷プログラムになる。【ソース 11】

実はこのプログラムコードのうち、印刷を行っているのは、最後の print という 1 行だけである。デフォルトプリンタへの印刷であれば、この 1 行で実行できる。

では、その前のプログラムは何を行っているかというと、印刷するプリンタの指定である。通常、指定なしで印刷すると、そのPCでデフォルトに設定されているプリンタへ印刷が行われる。

プリンタを指定したい場合は、このソースのように、PropertyValue でプリンタの名前をセットして、setPrinterという関数を実行するとプリンタを切り替えることができる。

以上で、作成したファイルをプリンタ へ直接印刷することが実現できる。

4-4. その他応用操作

ここまでいろいろな Calc の操作方法 を具体的に説明してきたが、実際にアプリケーションを開発する際には、これ以 外の操作が必要になってくる場合もある だろう。そのような場合は、プログラム 操作をマクロで調べることができる。

図4と図5に、Calc でのマクロ操作 方法を載せている。【図4】【図5】

操作としては、Excel でのマクロ操作と同じである。マクロを保存すると、操作した内容をプログラムコードとして確認することができるので、それを参考にすれば Delphi/400 プログラムに活用することが可能になる。

5.まとめ

以上本稿では、Delphi/400で、 OpenOfficeのCalcを利用するための プログラミングを説明してきた。Excel を OLE で利用したことがある開発者であれば、多少プログラムの書き方が違うだけで、意外に簡単に思われたのではないだろうか。

また、初めてここで OLE の操作プログラムを読んだ開発者でも、掲載しているプログラムコードをコピーして真似すれば、簡単にアプリケーションに組み込むことができるだろう。

OpenOffice は無償でダウンロードしてすぐに試すこともできるソフトウェアなので、ぜひチャレンジして開発の幅を広げていただきたい。

M

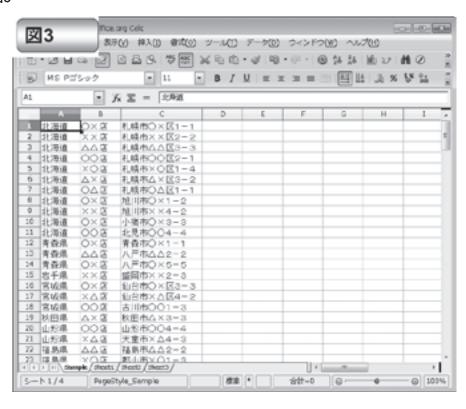
ソース6:Calcでセル値を編集する

```
procedure TFormI.btnCalcCellSetClick(Sender: TObject);
begin
//セルに値を代入
vSheet.getCellRangeOyName( 'C2' ).String := 'オープンオフィスに書き込み';
end;
セルを指定してアクセス
```

ソース7

ソース7:Calcで読み込んだデータをセル値に出力する

```
procedure Tfrm01.btnCalcCellSetClick(Sender: T0bject);
  i: Integer; //行数カウント
begin
  i := 1;
  with Table1 do
  begin
     Open;
     while not(EOF) do
     begin
//ファイルを1レコードずつ読み込んでせんに値を代入
       vSheet.getCellRangeByName('A' + IntToStr(i)).String := FieldByName('Addr1').AsString: vSheet.getCellRangeByName('B' + IntToStr(i)).String := FieldByName('Addr2').AsString; vSheet.getCellRangeByName('C' + IntToStr(i)).String := FieldByName('Addr2').AsString;
        lno(i):
       Next;
     end:
     Close:
  end:
end;
```



ソース8:Calcでファイルを保存する

```
procedure TForm1.btnCalcSaveClick(Sender: TObject):
                                                  保存パスはURL形式で
 sFileName
          : String: //保存ファイル名編集用
begin
 //保存ダイアログ実行
                                                  ×...¥...¥Test.ods
 if SaveDialogl.Execute then
                                                  O.../.../ Test.ods
 begin
  //ダイアログの保存パスを/形式に置換
   sFiloMano := StringReplace(SaveDialog1.FiloName, 'Y', '/', [rfReplaceAll]);
   //ファイルを保存
  vDocument.StoreToURL['File:///' + sFileName, dummyArray); =
                                                     保存パスを指定して
   //変更保存等ダイアログを表示しない
   vDocument.Modified := False;
                                                     保存APIを実行
 end;
end:
                           Calcを終了する際に
                           変更保存ダイアログを
                           表示させない設定
```

ソース9

ソース9:Calcでxls保存する

```
procedure TForm1.btnCalcKIsCliidk(Sender: T0b)lect);
  vaProperties: variant: //保存パラメータ用(配列)
  Property : variant: //保存オプション用・配列値>
sFileName : String: //保存ファイル名編集用
begin
  //保存タイアログ実行
if SaveDialog1.Booute then
  begin
     //ダイフログの保存/(スを/形式に置換
     #FileName: StringReplace(SaveDialog).FileName, 'Y', '/', [rfReplaceAll]);
//パラメータ用プロパティ配列件成
     //パフメータ用プロバティ的が軽級

vsProperties: 1 VarArrayCreate(ID, 0], vsrVariant);

//プロバティ用オプション生成

vProperty: VpenOffice.Bridge.GetStruct("com.sun.star.beans.PropertyValue");

//プロバティ用オプション値認定

vProperty.Name: "FilterName";

vProperty.Value: "WS Excel 97";

//オージャンは対象をデーサロンディルのデータを
                                                                                                                       基本はodsと同じ
                                                                                                                       xlsで保存する場合は
                                                                                                                       別形式なので
                                                                                                                        フィルタープロパティで
     //オブション内容をプロバティ配列に設定
                                                                                                                       指定が必要
     voProperties[0] := vProperty;
//ファイルを保存
     vicousent.StorelotR.('file:///' * sFileName, val'reperties);
//変更保存等ダイアログを表示しない
vicousent.Nodified := Falses
```

ソース10

ソース10:CalcでPDF保存する

```
procedure TForm1.btnCaldPOFC1idx(Sender: T0bject);
var
 vaProperties: variant; //保存パラメータ用(配別)

Property : variant; //保存オプション用(配別値)

sFileName : Strins: //保存ファイル名篠集用
begin
 //保存ダイアログ実行
 if SaveDialog1.Execute then
 begin
   //ダイアログの保存パスを/形式に置換
    sFileName: "StringReplace(SaveDialos).FileName, "Y", "/", [rfReplaceAll]);
/パラメータ用プロパティ配列生成
    vaProperties := VanArrayCreate([0, 0], varVariant);
    //プロパティ用オプション生成

vProperty:= v0pen0ffice.Bridse.GetStruct("com.cun.star.beans.PropertyValue"):
    //プロパティ用オプション追続な
vProperty.Nane := "FilterNane";
vProperty.Value := "cale_pdf_Export";
                                                                                           基本はodsと同じ、
                                                                                           pdfで保存する
    //オブション内容をプロバティ配列に設定
vaProperties[1] := vProperty;
                                                                                           場合は別形式なので
                                                                                           フィルタープロパティで
    //ファイルを保存
    Occurrent, StoreToLRL("file:///" + sFileName, vaProperties);
                                                                                           指定が必要。
 end:
```

ソース11:Calcで印刷する procedure TForm1.btnCaloPrintClick(Sender: T0bject); vaProperties : Variant: //ブリンタ指定パラメータ用(配列) vProperty : Variant: //ブリンタ指定オブション用(配列値) bogin //パラメータ用プロパティ配列生成 プリンタを vaProperties :=VarArrayCreate([0, 0], varYariant); //プロパティ用オプション生成 vProperty := vOpenOffice.Bridge_GetStruct('com.sun.star.beans.PropertyValue'); 指定 がプロパティ用オプション値投電 パプロパティ用オプション値投電 *Property.Name := 'Name': *Property.Yalue := 'FimePrint': //設定したいブリンタ名 //オプション内容をプロパティ数列に設定 vaProperties[0] := vProperty; //ブリンター指定 vDocument.setPrinter(vaProperties); デフォルトプリンタへの印刷で vDocument.print(dummyArray); あればこれだけ

図4

end;

図4:Calc上での操作をマクロで保存

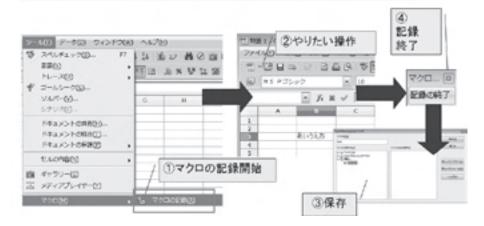
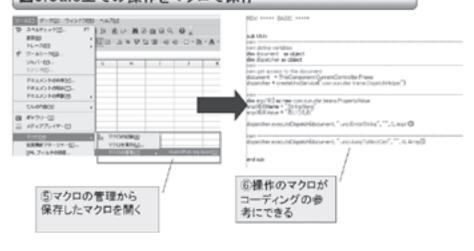


図5

図5:Calc上での操作をマクロで保存



尾崎 浩司

株式会社ミガロ.

システム事業部 プロジェクト推進室

Delphi/400:VCL for the Web活用TIPS紹介

Web アプリケーション開発に役立つ 4 つの TIPS。「VCL for the Web」を用いることで GUI 開発と同様、生産性の高い開発が実現できる。

- ●VCL for the Webとは
- ●Ajaxを使用したアプリケーション構築方法
- ■動的に生成したファイルのダウンロード
- ●コネクションプーリングを使用した Delphi/400アプリ開発
- ●スマートフォン対応ページの作成方法
- ●まとめ



略歴 1973 年 08 月 16 日生 1996 年三重大学工学部卒 1999 年 10 月株式会社ミガロ. 入社 1999 年 10 月システム事業部配属

現在の仕事内容 ミガロ . 入社以来、主に Delphi/400 を利用した受託開発を担当している。

1.VCL for the Webとは

「VCL for the Web (旧 IntraWeb)」 とは、Delphi/400 に付属する Web ア プリケーション開発ツールのことであ る。

この VCL for the Web は、通常の VCL フォームアプリケーションと同様 にフォーム上へコンポーネントを貼り付け、必要に応じプロパティやイベントを 設定することで開発を行えることが特徴 である。

ミガロ. では、Web アプリケーション 開発において積極的にこの VCL for the Web を使用しており、Delphi/400 における Web アプリケーション開発のノウハウも蓄積されつつある。

本稿では、これらノウハウの中から役 に立つであろう 4 つの TIPS について、 具体例を交えながら紹介しようと思う。

※ VCL for the Web をはじめとする Delphi/400 における Web アプリケーションの基本的な開発手順については、

ミガロ.のホームページに詳しく紹介しているのでそちらを参照していただきたい。

【Delphi/400 の技術情報(インターネット)に関する Tips】

http://www.migaro.co.jp/contents/products/delphi400/tips/web/index.html

2.Ajaxを使用した アプリケーション 構築方法

VCL for the Web では「Ajax」を使用することが可能である。Ajax とは「Asynchronous JavaScript + XML」の略で、Webページのリロードを伴わずに、Webサーバーと XML 形式のデータのやり取りを行って処理を進めていく対話型 Web アプリケーションのことである。

従来のWebアプリケーションは、図1aのように入力後ボタン等を押下する

ことで、サブミットを行い、新しいページを要求するという構成であった。【図 1a】

Ajax を使用すると、図 1b のようにページを切り替えることなく、入力中にどんどん最新情報に更新していくことが可能になるのである。【図 1b】

この仕組みを VCL for the Web では 使用することができる。しかも、JavaScript を使用せず、Delphi 言語をそのまま使 用することが可能であるため、とても魅 力的である。

では、VCL for the Web での Ajax 使用方法を見てみよう。

図 2 は、入力用コンポーネントである IWEdit のイベント一覧である。【図 2】

この中にある「OnAsync~」という イベントが、Ajax に関連するものであ る。通常のコンポーネントと同様のイベ ントが用意されていることがわかるであ ろう。つまり、通常のイベントと同じ感 覚で処理を記述することができるのであ る。

具体例として、図3のような仕様を検

図1a



図1b





討する。【図3】

顧客コード欄(edtCustCd)にフォーカスインしたタイミングとフォーカスアウトしたタイミングとでイベント処理を行い、画面上の値が変更された際に顧客マスターより値を取得すればよいのである。【ソース1】

このプログラムを実行すると、図4のようになる。【図4】

このように Ajax を使用すると、都度 画面をリロードしなくても対話できるア プリケーションが開発できるので、とて も便利なのである。

3.動的に生成した ファイルのダウンロード

Web アプリケーションを作成している と、ファイルをダウンロードさせたい場 面があるだろう。

この際、あらかじめ特定のフォルダに 用意されたファイルをダウンロードする だけならば、WebApplication 変数にある SendFile メソッドを実行すればよいだけ である。

しかしながら、データベースから抽出した結果等をダウンロードさせるとなると、動的にファイルを生成しなければならない。Web アプリケーションは Webサーバー上で実行されるため、クライアント側の要求ごとに Web サーバー側に動的にファイルを生成するというのはファイル名の重複等の問題もあり、難しいのではないだろうか。

では、どうすれば抽出結果から、ファイルをクライアント側にダウンロードさせることができるのだろうか。このような場合、サーバーではファイルではなくストリームとして作成し、クライアントへデータ転送を行えばよいのである。

具体例を検討してみよう。図5を見ていただきたい。【図5】

これは、データベースから取得した結果を、IWGridに表示している例である。この画面にある CSV 出力リンクをクリックすると、画面表示されているグリッド内容をクライアントに CSV ファイルとして転送させるのである。

ソースコードを紹介しよう。【ソース 2】

このプログラムでは、CSV データを

作成するために文字列リスト (TStringList)を使用している。この文字列リストには、ストリームオブジェクトに書き込む SaveToStream メソッドがあるので、メモリーストリームに保存できる。そうすることで、CSV データを直接ストリーム化することができるのである。

さらに、作成したストリームは WebApplication 変数の SendStream メ ソッドを使用すれば、クライアントにダ ウンロードさせることができる。

この方式を使用すれば、Web サーバー 側にファイルを生成することが一時的に も生じることなく、クライアント側にダウンロードさせることが可能になる。

Web アプリケーションでは、CSV 形式のダウンロードが要求される場面が多いと思うので、本ロジックを開発時の参考にしていただきたい。

なお、このプログラムをそのまま実行してしまうと、VCL for the Web の画面ロックの制約により、ダウンロード後、画面上の動作が一切受け付けられなくなってしまう。これを解消するには、ダウンロード対象となる IWLink コンポーネントの OnScriptEvents イベントに、設定が必要となるので注意してほしい。【図 6】

4.コネクションプーリング を使用したDelphi/ 400アプリ開発

「コネクションプーリング」とは、データベースへアクセスの都度、接続(コネクション)を確立するのではなく、あらかじめ事前に一定数のコネクションを確立しておき、それを使い回す手法のことである。データベースアクセスの負荷を減らすために用いられる。

コネクションプーリングは Web アプリケーション開発でよく使用する手法だが、これを VCL for the Web でも使用することが可能である。

VCL for the Web で、コネクション プーリングを使用する手順を紹介しよう。

新規作成にて「VCL for the Web Application Wizard」を選択すると、ウィザードが起動する。この画面で、[Options] → [Pool Data Connections]

のチェックを ON にすればよい。【図7】 このオプションを設定してプロジェクトを新規作成すると、図8のようになる。 【図8】

ServerController.pas に、Pool と 命名された IWDataModulePool コンポーネントが設定されているのがわかる。この Pool コンポーネントには、PoolCount というプロパティが用意されており、ここに指定した数の接続が維持されるのである。(Delphi/400 の場合、PoolCount に指定した数だけジョブが生成されると考えればよい。)

では、このコネクションプーリングが 使用できる環境下で、実際にデータベー スにアクセスするにはどうすればよいの だろうか?

データベース接続 (コネクション) を 複数の Web セッションで共有して使用 するため、使用開始する際に明示的にコ ネクションをロックし、使用が終了した らロックを解除する必要がある。

実際の使用例をソース3に示すので参 考にしてほしい。【ソース3】

コネクションプーリング制御下でデータベースにアクセスする場合、アクセス対象とするデータモジュールをロックするために、変数を定義する。(ソース3では、ADataModuleという変数を宣言。)

そして、データベースにアクセスする 直前で LockDataModule 関数を実行す ると、その時点で空きのあるコネクショ ンを探し、ロックを行う。

使用が終了したら、UnlockDataModule 手続きでアンロックする、という流れで ある。(これらの関数/手続きは、 ServerControler.pas に処理が記述され ているため、使用する際には、データモ ジュールユニットと共にユニット参照を 行うこと。)

このようなロジックを追加することで、Webアプリケーションに対し、コネクションプーリング制御ができるのでぜひ検討してみてほしい。

なお、CO400Connection ドライバ (dbExpress 接続) でコネクションプーリングを行う際、データモジュール生成時 (OnCreate) に、あらかじめTSQLConnection の接続を行うと正常に動作しない。初回のデータベースアク



図4





セス時(初めて TSQLQuery 等がオー プンする時)に、接続されるようにする のがポイントとなるので注意いただきた い。

5.スマートフォン対応ページの作成方法

近年、スマートフォンがブームとなっている。今後業務アプリケーションにもスマートフォン向けサイトの対応を迫られることが予想される。

スマートフォンは標準でブラウザ機能を持っているので、これまでのWebアプリケーションをそのまま動作させることもできるが、PCと違い、スマートフォンでは解像度の制約がある。そのため、通常のサイトをそのまま一覧表示すると、図9のようにとても細かな表示となってしまう。【図9】

解決策の1つとして、スマートフォンに最適な画面設計を行えばよいわけだが、スマートフォンは機種により解像度が異なるなど、そのままではなかなか調整が難しいものがある。

この場合役に立つのが、HTMLのメタタグの1つである「ViewPort」である。 このメタタグは、スマートフォンでの可 視領域やズームなどの設定するものであ る。記述例を図10に示す。【図10】

では、このメタタグを VCL for the Web のフォームにどのように組み込め ばよいのだろうか。それは、IWForm にある ExtraHeader プロパティに記述 すればよいのである。【図 11】

そして、メタタグを設定したフォームを実行すると、図 12 のようにスマートフォンに最適化され、不要な拡大縮小を伴わない画面が構築できるのである。【図 12】

実際の画面設計では項目数を制限するなど、スマートフォンならではの制約も出てくるであろうが、メタタグの可視領域を設定することで、よりスマートフォンに最適化された画面が構築できる。

6.まとめ

今回は、VCL for the Web にまつわる 4つの TIPS を紹介した。Web アプリ ケーション開発の一助としてほしい。

また、これまで Web アプリケーショ

ンを作成したことがない方も、一度 VCL for the Web による開発を実践い ただき、GUI 開発と同様に生産性の高 い開発が可能であることを確認いただけ ると幸いである。

M

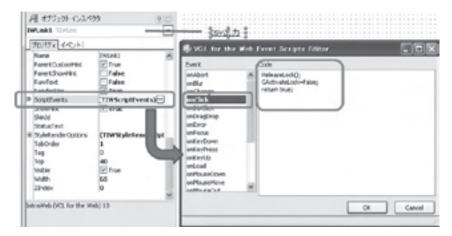


図7



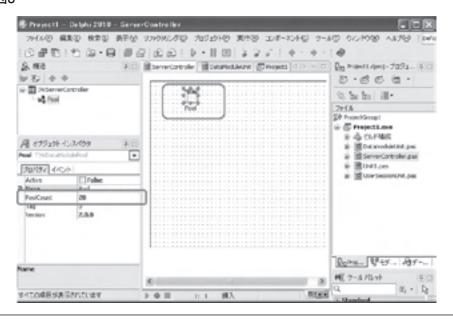




図12



図10





```
[ IINForm1 ]
    procedure TIMForm1.edtCustCDAsyncEnter(Sender: TObject;
   EventParams: TStringList);
    begin
      // フォーカス取得時、現時点の値(初期値)を変数に保持
FCustCode := edtCustCD.Text;
    end:
    procedure TIMForm1.edtCustCDAsyncExit(Sender: TObject;
50
       EventParams: TStringList);
    begin
        // フォーカス終ア時、初期値と値が変わった場合、処理を行う
if FCustCode ◇ edtCustCD.Text then
       begin
         // 顧客名欄を初開化 (クリア) する
edtCustCD.BGColor := cl#indow;
IblCustName.Caption := ';
          // コード値が入力されている場合、顧客マスタ検索を行う。
if edtCustCD.Text 〈〉 then
60
         begin
            try
              // GotCustName回数で顧客コードより顧客名を取得
|blCustName.Caption := GetCustName(edtCustCD.Text);
            except
              // エラー (側外生成) 時、エラーメッセージを表示する
on E: Exception do
              begin
                 edtCustCD.BGColor := clWebAQUA;
70
                 IblCustName.Caption := E.Message;
              end;
            end:
         end:
74
       end:
    end:
```

```
@procedure TIMForm2.IMLink1Click(Sender: TObject):
        var
          i, j: Integer;
sStr: String;
sCSVList: TStringList; // 各列毎の文字列を保持するリスト
sCSVRowList: TStringList; // 各刊毎の信仰を哲学するリスト
memCSVStream: TMemoryStream; // CSV出力用 メモリーストリーム
                j: Integer:
      besin
            //CSVデータを管理する文字列リストを生成
sCSVList := TStringList.Create:
           try
//CSV用文字列リストの初期化
eDSYList.Clear:
//告列毎にある値を管理する文字列リストを生成
sDSYRowList: TStringList.Create:
               try
//簡単グリッドよりデータ取得する
for i := 0 to IMGrid1.RowCount - 1 do
                  begin
                     egin
//利毎の値を保持するリストの初期化
sCSVRowList.Clear:
//各列の値をリストに追加
for j := 0 to [WGrid1.ColumnCount - 1 do
                      begin
                         sStr := IMGrid1.Cell[i, j].Text;
sCSVRowList.Add(sStr);
                      end;
//CSV用文字列リストに行情報を代入
sCSVList.Add(sCSVRowList.CommaText);
                  end:
               finally
sCSVRowList.Free;
               //メモリーストリームのた成
memCSVStream := TMemoryStream.Create;
memCSVStream.Clear;
//CSV用文字列リストをメモリーストリームに仮存
sCSVList.SaveloStream(memCSVStream);
140
               //メモリーストリームからのダウンロード処理
WebApplication.SendStream(memCSVStream, True, '', 'CustonerList.csv')
            Tinally
               sCSVList.Free;
            end:
        end:
```

```
ソース3
```

```
function TIMForm1.GetCustName(ACustCode: String): String;
yar

ADataModule: TDataModule]: //ロックを行うデータモジュール
begin
/ 初期化
Result:: ';
//データキジュールをロックする
ADataModule: LockDataModule:
try

| //データを輸出する
with ADataModule.grySQL do
hexin
| SQLConnection: ADataModule.connAS400; // ロックデータモジュールを使用
//データ取得
Active: False;
ParamBvName('CONOCO').AsAnsiString: ACustCode;
//データセットを耐く
Active: True;
try
//対象データが存在するしない場合エラーとする
if Eof and Bof then
    raise Exception.Create('顧客コードが正しくありません。');
//放卵した動産を全セットする
Result: FieldBvName('CONAME').AsString;

finally
//データセットを閉じる
Active: False;
end:
finally
//データキジュールの使用が全フしたらアンロックする
UnlockDataModule(ADataModule);
end;
end;
end;
end;
```

清水 孝将

株式会社ミガロ.

システム事業部 システム1課

JC/400でJavaScriptを活用

Web サーバーや IBM i とアクセスすることなく、動的な画面機能を実装しよう。 JC/400を用いて、Web アプリケーションへの JavaScript の実装方法と活用例を紹介。

- ●はじめに
- ●JavaScriptの参照方法
- ●入力項目のブランクチェックの実装
- ●エンターキーの制御
- ■エラー項目の色の反転
- ●最後に



略歴 1983 年 10 月 04 日生 2008 年甲南大学文学部卒 2008 年株式会社ミガロ. 入社 2008 年 04 月システム事業部配属

現在の仕事内容

入社4年目でDelphi/400 および JC/400 の開発業務を担当。Web に関する知識や技術を身につけ、 Web アプリケーションのスペシャ リストを目指している。

1.はじめに

JC/400の画面は、HTML だけで作成することができる。しかし、HTML 自体は静的なコンテンツなため、エラー項目の色の反転や項目のチェックなどの動的な Web 画面を作成しようとしても、HTML だけでは実現は不可能である。

JC/400 の機能を利用して、項目がブランクか否かをチェックすることも可能であるが、チェックのたびに、Web サーバーや IBM i とアクセスすることとなり、レスポンスが心配されるケースもあるだろう。

そこで JavaScript の出番である。 JavaScript を活用することで、簡単な 入力チェックや背景の反転など、ダイナ ミックな動作が実現できる。Web サー バー、IBM i とアクセスすることなく、 クライアントサイドだけで項目をチェッ クすることも可能なのである。

今回は、JC/400のWebアプリケーションへのJavaScriptの実装方法と活用例を紹介したい。

2. JavaScriptの参照方法

JavaScript は、外部ファイルに記述 する方法と、HTML内に記述する方法 の2つが存在する。

●外部ファイルに記述する方法

複数の HTML 文書から参照する関数は、外部ファイルに記述することで、各 HTML 内に記述せずに利用することが可能である。

アプリケーション内で共通化が可能な 関数は、外部ファイルに記述しておくと よいだろう。【ソース 1】

● HTML 内に記述する方法

HTML内に記述する場合は、同文書内のみで関数を使用することができる。内部的な変数を定義するのに適しているので、画面固有の処理はHTML内に記述するとよいだろう。【ソース2】

今回使用する JavaScript は、すべて HTML 内に記述する方法で進めていき たい。JavaScript は HEAD 要素、BODY 要素の任意の場所に記述可能であるが、 通常は HEAD 要素内に記述することが 主流である。【ソース 3】

3.入力項目のブランク チェックの実装

ここからは、実際に JavaScript のコー ディングを解説していきたい。

今回は、入力項目のブランクチェックを例として紹介する。図1のように、項目 A と項目 B が必須入力の画面を設計した。【図1】

要素と属性の概念

最初に、簡単ではあるが HTML や JavaScript における「要素」という用 語の定義について説明しておきたい。要素とは、HTML 内に使われている部品の1つ1つのことを指している。入力項目やボタン、コンボボックス、画像のそれぞれが1つの要素なのである。

図1の画面の要素をそれぞれ記述した ものが、ソース4である。【ソース4】

< INPUT > タグや< A > タグ、< IMG > タグが画面の部品、いわゆる要素となる。ここでの項目 A と項目 B は、< INPUT > 要素となっている。

また、要素の1つ1つはそれぞれ「属性」を持っており、実際にWebブラウザに表現される長さや形、内容は属性によって決定する。後述のID属性はその1つである。<INPUT>要素は他に、type属性やvalue属性も持っている。

type 属性は、< INPUT > の種類を定義する。例えば、> 回の"text"という設定値であれば単1 行入力項目となり、"button"と設定すればボタンの形となる。

value 属性は、type 属性がtextの場合、 入力値を保持している。この value を利 用して、値のチェックを行うのである。

要素の取得

まずは最初に、チェック対象とする項目の要素を取得する必要がある。

要素を取得する方法は何通りかあるが、今回は、JC/400と HTML との連携に利用される ID 属性から要素を取得する、getElementById (要素のID)メソッドを使用する。

項目 A には "INPTA"、項目 B には "INPTB" とそれぞれに ID を割り振っている。

ソース 5 は、関数 initpage () という処理の中で、JavaScript 内の変数 "ObjINPTA" "ObjINPTB" に対して、それぞれ項目 A と項目 B の要素を格納する記述である。予め変数に代入しておくことで以後の処理に活用できる。【ソース 5】

関数の名称は任意で定義できるが、initpage () という名称には大きな意味がある。JC/400ではWebページを生成するとき、JavaScript内にinitpage ()という名称の関数を定義しておくと、自動的にBODY要素のOnloadイベント時に処理を行う仕組みになっている。(イベントの説明は後述する)

これで、画面が表示されたタイミングで、ObjINPTA と ObjINPTB という変数内に項目 A と項目 B が格納されている状態になるのである。

エラーチェック関数の実装

エラーチェック関数の実装についての 処理は、すべて initpage () 関数で取 得した変数 ObjINPTA と ObjINPTB の要素を利用する処理となり、それらを 記述する。

ソース 6 は、関数 CheckBlank ()を 定義している。この中で ObjINPTA と ObjINPTB の Value 属性の値をチェッ クし、ブランクの場合は警告を表示後、 処理を中断するように記述している。 【ソース 6】

あとは、これを実行ボタンのクリック 時に呼び出す設定を行うだけである。

イベントハンドラへの設定

イベントハンドラとは、ユーザーが行った動作や操作に対して、特定の処理を与えるためのトリガーとなる命令である。

要素にはそれぞれ、JavaScript のイベントハンドラが準備されている。例えば onclick は文字通り、ボタンや画像がクリックされたときに実行される。

使用可能なイベントハンドラは要素によってさまざまであり、今回は画像リンクを設定している<A>要素のイベントハンドラを使用する。なお、<A>要素にもonclick のイベントハンドラは存在するが、onclick イベントは JC/400 が先にハンドリングしてしまう。よって、onclick の前に動作するイベントとして、onmousedown 使用することにする。

ソース 7 は、onmousedown イベント時に、前述した CheckBlank ()が実行されるように設定している状態である。これにより、実行ボタンがマウスでクリックされたタイミングで、エラーチェックの処理が走る仕組みが完成した。【ソース 7】

図2は、項目Aに値が入力されていない場合に、実行ボタンをクリックしたときのイメージである。【図2】

項目 A に値を入力した場合は、図 3 のようになる。【図 3】

4. エンターキーの制御

JC/400 アプリケーションではエンター押下時に、画面が Submit される。

エンター押下では、onmousedown のイベントは実行されないためエラーチェックも実行されない。

RPG 側の制御で、何も処理をせずに 画面をリフレッシュするだけの制御にす ることも可能だが、ユーザーによっては 更新してしまったと思われる方もいるで あろう。

以降に、エンターキーを無効にする制 御を紹介したい。

イベントキーコードの変換

ブラウザでは、キーボードの押下やマウス押下のタイミングで、イベントキーコードを取得する仕組みとなっている。(今回は、JC/400がサポート対象としている Internet Explorer のバージョン8を前提として話を進める)

Internet Explorer 8 では、エンターキー押下時に"13"のイベントキーコードを取得する。このイベントキーコードを内部的に置き換えることで、エンターキーを無効化する。

ソース8は、取得したイベントキーコードが"13"の場合、イベントキーコードを"99999"に置き換える処理である。ちなみに"99999"というイベントキーコードのキーは実際に存在しないため、結果何も起こらないという仕組みとなる。【ソース8】

ソース 9 のように、〈BODY 〉要素のイベントハンドラに設定する。これにより、フォーム上で発生する onkeydownイベントに対して、この関数が実行される。【ソース 9】

なお余談ではあるが、タブキーのイベントキーコードは"9"なので、この応用で簡単にエンター押下時の項目移動を 実現することも可能である。

5.エラー項目の色の反転

ここまでの処理で、必須項目のブラン クチェックが実現できた。

次は、エラー項目をより視覚的に分かりやすくするために、項目の背景色を反転させたいと思う。

前述のCheckBlank () 処理に、項目 の色を反転させる処理を追記した。 ObjINPTA.style.backgroundColor に、 色の値をセットすることで実現してい

```
ソース1
   <script type="text/javascript" src="/jaci400/js/JCTest.js">↓
   </script>↓
ソース2
   <script language="JavaScript">↓
   <!--↓
   //ここにJavaScriptを記述↓
   -->
   </script>↓
ソース3
   <HEAD>↓
   Kscript type="text/javascript" src="/jaci400/js/JCTest.js">\upsilon
   </script>↓
   <script language="JavaScript">↓
   <!--+
   //ここにJavaScriptを記述↓
   -->
   </script>↓
   </HFAD> \
図1

#655838 - テスト高音 - Windows Internet Explorer

    g http://
     TRIBAT
                   ※必須入力
     項目目
                    ※必須入力
     THILD
     項目の
     多実行
    ❸ インターネット | 保護モード: 無効
                              G - 6,100% -
ソース4
   <BODY onkeydown="return PreventEnterPost(event);">+
   <FORM method = "post">4
   項目A <INPUT type ="text" value = "" id = 'INPTA' >※必須入力<BR>+
   (BR) +
   項目B < INPUT type = "text" value = "" id = 'INPTB' > ※必須入力 < BR>+
   項目C <INPUT type ="text" value = "" id = 'INPTC' ><BRO+
   <BR>+
   項目D (INPUT type ="text" value = "" id = 'INPTD' ><BRD+
   <BRO+
   <A href="#" onmousedown ="CheckBlank();" > simg src="/jaci400/images/button2.gif" id="BTUNA" border="0"> </A>:
   <BR>+
   </FORMD+
   </B00Y>+
```

る。これは、スタイルの値を動的に置き 換えている処理である。【ソース 10】

図4のように、エラーとなった項目が 赤色に反転された状態となる。【図4】

6. 最後に

今回紹介した項目のブランクチェック は、おそらくほぼすべてのシステムで活 用できるのではないかと思い選んだ。

JC/400 を導入いただいている皆様の 中には、新たに HTML を学習された方 も多いのではないだろうか。そのような 中、JavaScript まではなかなか手が出 せないという言葉もよく聞く。

しかし、JavaScriptを活用することでより、よいシステムを実現することが可能なこともまた事実である。またJavaScriptは、HTMLと同様に書籍やインターネット上にノウハウやリファレンス情報が豊富に存在する。何か実装したい処理をキーワードでインターネット検索をしてみると、簡単に情報が見つかったりする。一見馴染みなくとっつきにくいと思っていても、他のプログラム言語に比べても学習しやすいものではないかと思う。

本稿が JavaScript の活用のきっかけ になれば幸いである。

M

ソース5 var ObjINPTA = null;↓ var ObjINPTB = null;↓ function initpage(){ ↓ ObjINPTA = document.getElementById("INPTA"); ↓ ObjINPTB = document.getElementById("INPTB"); ↓ }↓ ソース6 function CheckBlank(){↓ if (ObjINPTA.value == "") {↓ alert('項目Aは必須入力です。');↓ return false;↓ } \ if (ObjINPTB.value == '') {↓ alert('項目Bは必須入力です。');↓ return false;↓ }↓ }↓ ソース7 img src="/jaci400/images/button2.gif" 図2 図3 着 #655837 - テスト画面 - Windows Internet Explore 愛+655057 - アスト副第 - Windows Internet Exp (E) Mily / http://t AMA BIK 海田A 新設権入力 医必須入力 M出版入力 **MB0** ※必押入力 項目B 項目() 活用で 項目の 項目の 图实行 別與行 A MENSONATOR A MEGASANTOT. G = 9,100% = ⊕ インターネット | 何度モード: 無池 G = 8,100% + ⑥ インターネット | がまてード: 発力

```
ソース8
   function PreventEnterPost(e){ ↓
        var e = window.event; ↓
        if(e.keyCode == 13){↓
       window.event.keyCode = 99999;↓
   }↓
ソース9
   <BODY onkeydown="return PreventEnterPost(event);">↓
ソース10
   function CheckBlank(){↓
      ObjINPTA.style.backgroundColor = "white";↓
      ObjINPTB.style.backgroundColor = "white";↓
      if (ObjINPTA.value == "") {↓
      alert('項目Aは必須入力です。');↓
      ObjINPTA.style.backgroundColor = "red";↓
      return false;↓
      } \
      if (ObiINPTB.value == '') {↓
      alert('項目Bは必須入力です。');↓
      ObjINPTB.style.backgroundColor = "red";↓
       return false;↓
      }↓
   }↓
図4
    必 #555865 - デスト施園 - Windows Internet Explorer □ □ 回 回る
    @ http://b
    項目A
                ■※必須入力
    項目的
                 医必須入力
    項目の
    項目ロ
     多実行

← インターネット | 保護モード: 無効
```

國元 祐二

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

Business4Mobile:jQuery連携で機能拡張

スマートフォンから社内 DB にアクセスできる「Business4Mobile」。スマート デバイスを、さらにビジネス活用するためのテクニックを紹介する。

- ●はじめに
- Business4Mobile とは
- Business4Mobile の活用
- JavaScript (iQuery) 連携による機能拡張
- ■スマートフォン連携の機能拡張
- ●画像表示の機能拡張
- ●おわりに



1979年03月27日生 2002 年追手門学院大学文学部 アジア文化学科卒 2010年10月株式会社ミガロ. 入社 2010年10月RAD事業部配属

現在の仕事内容 Delphi/400 * JC/400. Business4Mobile の製品試験やサ ポート業務などを専門に担当している。

1. はじめに

ここ数年で、iPhone や iPad などのス マートフォンやタブレットが広く普及し てきた。もちろん、スマートデバイスの 活用は個人だけでなく、ビジネスにおい ても急速に注目されるようになってきて いる。

Web プラットフォームの業務システ ムも現在では一般に使用されているが、 これも Web 環境の普及の結果である。 同様に今、注目を浴びているスマートデ バイスも、これからのビジネス進展の中 で必要な環境の1つとなっていくのだろう。

ミガロ.では今年4月、スマートデバ イスを業務で有効に活用するためのソ リューションとして「Business4Mobile (ビジネスフォーモバイル)」という新製 品を発売した。【図1】

本稿では、この Business4Mobile と いう製品の魅力を伝えるとともに、すで に製品を使用している方々に、さらにビ ジネス活用するための具体的なテクニッ クをいくつか紹介したい。

2. Business4Mobile

Business4Mobile はスマートフォン やタブレットなどのスマートデバイスを 利用して、外出先から社内のデータベー スを参照することが可能なソフトであ る。この Business4Mobile には、大き く分けると次の3つの製品特徴がある。

(1) マルチスマートデバイス対応

1つ目の特徴は、すべてのスマートデ バイスから利用できるという点である。 【図 2】

Business4Mobile のアプリケーショ ンは、特定のスマートフォンへの対応で はなく、Web ブラウザを使って利用す る。そのため、iPhoneやiPad、Android などの機種を問わず、あらゆるスマート デバイスから活用できる。もちろん、PC の Web ブラウザでも利用可能である。

(2) マルチスデータベース対応

2つ目の特徴は、さまざまなデータベー

スで利用できるという点である。【図3】

Business4Mobile で使用できるデータ ベースは、IBMのDB2、DB2/400、 Informix をはじめ、Oracle、SQLServer、 PostgreSQL など多種多様なデータベー スに対応している。

最近では、企業内で複数のデータベー スを運用している場合も多い。つまり、 Business4Mobile は、上記のデータベー スであれば複数のデータベースに接続す ることができ、スマートデバイス上から 利用することが可能になる。

(3) プログラミング不要

3つ目の特徴は、プログラミングなし で、簡単に利用できるという点である。

Business4Mobile では「クエリー」 と呼ばれる条件設定を作成しておくだけ で、スマートデバイスからデータベース を参照することができる。

なお、Business4Mobileのクエリーは、 IBM iの Query (キューリー) や Microsoft Access のクエリをご存知の 方であれば、同じようなものだと考えて かまわない。【図4】

具体的には、クエリーの設定はBusiness4Mobileに付属するツール上で、参照したいファイルや項目を選択するだけで設定することができる。プログラミングが必要ないため、プログラム開発者でなくても簡単に使えることが魅力である。

また本稿後半で、プログラム開発者向けに「JavaScript(jQuery)」を使った機能拡張のテクニックを紹介するが、Business4Mobileの標準機能を使う場合には、プログラミングは一切必要としないので安心してほしい。

3. Business4Mobile の活用

Business4Mobile を使えば、スマートフォンやタブレットから、社内データベースを利用できるという製品概要を説明してきた。

ここからは、Business4Mobile を使えば、どんな画面や用途に活用できるのかを例を挙げて紹介していきたい。

(1) 客先での出荷状況確認

例えば、客先で商品出荷状況を確認したい、といった場合に活用することができる。

図5のように、スマートフォンから条件を指定して製品を検索し、発注情報(出荷番号)から詳しい出荷状況などを入手することで、リアルタイムの情報をお客様に案内することが可能になる。【図5】

(2) 急なアポイント情報の取得

例えば、営業が社外に出る場合、今までは担当者自身が顧客の連絡先情報などを詳細に、しかも事前に把握しておく必要があった。

しかし Business4Mobile を使えば、外出先から社内のデータベースを参照することが可能になるので、得意先担当者のアポイント情報などが急に必要になっても社内の人間に依頼したり、いったん帰社するといったことをせずとも情報を確認できるようになる。【図 6】

スマートフォンでこうした活用ができ れば、外出先でのビジネスをさらに効率 よく行えるようになるだろう。

(3) メール送信やグラフ活用

Business4Mobile では、前述したように、必要なデータを簡単に抽出することができる。さらに、クエリー結果画面の右下には、メールボタンも用意されている。つまり、このボタンを使用することで、結果を PDF の添付ファイルとしてメール送信することができる。【図7】

またデータを抽出して、表形式で参照 するだけでなく、グラフで表示できる機 能もついている。そのため、ビジュアル 形式のわかりやすい分析に展開して、さ らなる業務活用に利用することができ る。【図 8】

本稿で、Business4Mobile の詳細を すべて説明することはできないが、製品 のイメージを多少はつかんでいただけた だろうか。Business4Mobile では、上 記のような画面や充実した機能を、だれ もが(プログラム開発者でなくとも)簡 単に扱うことができる。

そして、プログラム開発者であれば、 Business4Mobile を便利なソフトとして使うだけでなく、開発ツールのようには扱えないのだろうか、自分のほしい機能を追加していきたいと考える方がおられるかもしれない。

以降からは、そのようなさらなる Business4Mobile 活用を希望されるプログラム開発者に向けて、「jQuery」という JavaScript 連携による機能拡張の可能性について紹介していきたい。

4. JavaScript (jQuery) 連携による機能拡張

今回の機能拡張では「jQuery(ジェイクエリー)」という JavaScript ライブラリを使用する。jQuery は、HTMLを拡張するライブラリである。これを利用することで、さまざまな機能が少しのコードで手軽に実装可能になる。

JavaScript はブラウザで利用できる 簡易な言語だが、ブラウザの種類ごとに プログラムの実装方法が異なるといった 場合がある。しかし、jQuery ではブラ ウザの種類による違いを吸収する仕組み を持つため、プログラマーはロジックの 実装に専念できるという利点がある。

加えて、JavaScript のライブラリは 各種存在するが、jQuery はほかのライ ブラリと比べると利用者が多く、Web 上での情報取得も容易であり、比較的簡 単に利用できる。

また、jQueryのライセンスはフリーのオープンソースのため、ライブラリ内の著作権表示さえきちんと残せば、商用・非商用を問わず自由に利用することが可能である。

さらに、世界中のさまざまな有志が jQueryを利用したサンプルを公開して おり、難しい機能でもそれらを参考にす ることで、短期間に実装可能になる。

jQuery の利用方法

jQuery の利用方法は、最初にライブ ラリファイルを jQuery 公式サイトにア クセスして、画面右側中央付近にある Download (jQuery) ボタンからダウン ロードしておこう。【図 9】

ダウンロードのボタンを選択すると、 現時点の最新版ファイルが取得できる。 ダウンロードしたファイルを HTML で読 み込むと jQuery の機能が利用可能となる。

【jQuery 公式サイト】 http://jQuery.com

jQuery の記述方法

jQueryを利用した JavaScript の記述 方法は、セレクター API により、操作 する要素(HTML 中のタグ)を選択して、 その要素に対して「コマンド」を実行す る。また、コマンドは互いを「連鎖」さ せることができるため、コマンドの結果 を、別のコマンドに送ることが可能にな る。これらの組み合わせにより、少ない コード記述でさまざまな効果を生み出す ことができる点が特徴である。

少し詳しく述べると、通常の JavaScript で要素を取得する場合は、document. getElementById メソッドで要素を取得するが、これはブラウザにより挙動が異なる場合がある。しかし、jQuery では、\$() 関数を利用して\$("# 要素の ID 名")で要素を取得可能である。

また、それだけでなく\$() 関数では、 クラス名やタグ名での要素取得やさまざ まな「フィルター」を利用することで、 簡単に HTML の要素を取得できる仕組 みになっている。

Business4Mobile

スマートフォン活用 データ検索ツール

Business 4 Mobile

(ビジネスフォーモバイル) BUSINESS Lymobile

簡単な設定で

スマートフォンから社内データベースにアクセス



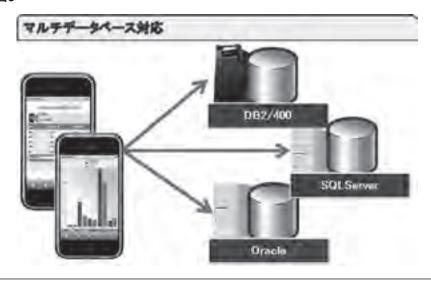


図2

マルチスマートデバイス対応



図3



さらに前述したように、jQuery のコマンドの「連鎖 (メソッドチェーン)」もある。メソッドチェーンは名前のとおり、メソッドを実行して、その結果に対してさらにメソッドを実行する機能である。

今回の Business4Mobile の拡張でも、 上記のさまざまな機能を利用しながら実 装を行っていこう。

5. スマートフォン連携の機能拡張

ここから、Business4Mobileを用いて、 スマートフォンのブラウザ機能をさらに 連携活用するための機能拡張について説 明していきたい。

具体的には、スマートフォン上に表示される実行結果の画面に、リンクを追加し、図 10 に示すように、以下の 3 つの機能を呼び出すこととする。【図 10】

(1) GoogleMap 表示の機能

表示された住所を選択すると、その位置を示す GoogleMap ページを開く。

(2) 電話発信の機能

表示された電話番号を選択することで 直接、電話発信する。

(3) メール送信の機能

メールアドレスを選択すると、メール アドレスを送信先にした状態で、メール ソフトを起動する。

処理のフローは図 11 のようになる。【図 11】

- ①クエリーの実行結果画面の項目タイト ル名を確認
- ②キーワードが存在する場合は機能追加 処理
- ③キーワードの項目タイトル名を消去
- ④画面表示

そして、実行結果画面は図12のようになる。列項目のタイトルに含まれるキーワード(@GMAP)などが削除されており、表示データには、キーワードに対応したリンク機能が追加されている。【図12】

事前設定

JavaScript による実装の事前設定と して、以下を行う。

●外部 JavaScript ファイルの読込

Business4Mobile の実行結果画面は、 約 10 種類の HTML テンプレートファ イルから作成されている。

最初に、Business4Mobile の実行結果画面のHTMLテンプレートファイルを変更する。実行結果画面のHTMLテンプレートファイルは、resultlist_template.htmlである。このファイルに、jQueryへの参照と今回追加する外部JavaScriptファイルの参照を記述する。

まず、jQueryファイルをリネームする。

jQuery-1.6.2.min.js → jQuery.js

その後、Business4Mobile がインストール された Apache サーバーに、jQuery のディレクトリを作成して配置する。

インストール先ディレクトリ:

/jQuery/jQuery.js

続けて、今回追加する JavaScript は、ext ディレクトリを作成して配置する。

インストール先ディレクトリ: /ext/MigaroExtension.js

最後に、追加した2つのファイルを、 外部 JavaScript として HTML から読 み込むように設定を行う。【ソース1】

● div タグの追加

Business4Mobile で処理された結果 は\$result に設定されるため、id 属性に Extension と設定した div タグで囲む。 JavaScript 内で、この id の要素を取得 して処理を実行する。【ソース 2】

●キーワード定義

実行結果画面に表示される列項目タイトルの見出しを判別して、さまざまな機能を付加する。そのためには、Business4Mobileのツールからクエリーを設定する時点で、キーワードとなる項目名の設定が行われている必要がある。それらのキーワードは JavaScript で

それらのキーワードは JavaScript で 定義している。【ソース 3】

●キーワード設定

Business4Mobile のツール上でクエ

リーを設定する際には、図13のように、 列の項目タイトル名に @GMAIL や @ TEL、@EMAIL などのキーワードをつ けておく。そして、キーワードの有無で 表示される項目に対して、機能の追加処 理を行う。【図13】

JavaScript の実装

いよいよ JavaScript の実装に入る。

●「一覧形式」or「詳細形式」の判別

Business4Mobile の実行結果画面は、図 14 のように「一覧形式」と「詳細形式」の 2 種類が用意されている。

一覧形式は、表形式のデータ表示で上部に項目のタイトル表示があり、その下に複数のレコードを表示することができる。一方、詳細形式は、1 レコードのデータを縦に並べ、たくさんの項目を表示することができる。【図 14】

一覧形式と詳細形式では、出力される Table タグの構造が異なるため、実行結果画面をプログラムで判別する必要がある。識別には、出力される HTML 内の特定のタグの存在有無で判断する。実行結果は一覧形式、詳細形式ともに、HTMLの Table 要素として出力されている。

一覧形式の実行結果内の 要素に は、Business4Mobile が出力する html に titleWhite クラスが存在する。その ため、要素に titleWhite のクラス が存在する場合は一覧形式であり、存在 しない場合には詳細形式と判断する。 【ソース4】

● jQuery を利用した要素取得

詳細形式を表示する場合には、左側に列の項目、右側にデータが表示されるため、左側の列の項目の 要素をiQueryを利用して取得する。

このような場合、jQuery の\$()のフィルター機能を利用することで、簡単に要素を取得することができる。

今回、判別のために取得したい内容は、 列の項目タイトル名のため、左側の 要素のみとなる。今回は、evenフィ ルター (偶数の要素取得)を利用して取 得する。

\$("#Extension td:even")

Business4Mobile クエリー設定



図5

Business4Mobileの活用「お客様先での製品出荷状況確認」



図6

Business4Mobileの活用「外出先で得意先へのアポイント」



このセレクターの内容は "id 属性が Extension の要素内に含まれる 要素で、偶数の タグのみを取得する" という内容となっている。【ソース 5】

なお詳細を後述するが、取得した 要素内の列項目を取得後に、項目 のタイトルの見出しの文字を走査して特 定のキーワードが設定されている場合に は、タグを書き換える操作を行う。

また前述したように、@GMAP などのキーワード文字に、タイトル項目名での設定内容が一致すると、そのキーワードに対応する機能を追加する処理が行われる仕組みとなっている。

それらタグ要素への操作も、jQuery を利用することで容易に行える。

以降から、データを表示している 要素に対して、アンカー機能をそれぞれ追加して、GoogleMapへのリンクや電話番号のリンク、メーラー起動のリンクなどを実現していこう。各リンク機能についておのおの説明する。

GoogleMap へのリンク

GoogleMapへのリンク機能は、データタグ要素を、GoogleMapへのリンクが設定されたjQueryのwrapInnerメソッド(各要素の子要素を、HTMLで作成した要素で囲む)を利用して <a>タグではさみこみ、要素が選択された際に GoogleMap のページを表示する仕組みになっている。

注意点としては、GoogleMapへのURLの引数に住所を渡す際に、URLエンコーディングが必要となる点が挙げられる。URLエンコーディングには、JavaScriptのencodeURIComponent関数を利用する。【ソース6】

処理前

大阪府浪速区湊町 2-1-57 処理後

<a href='http://maps.google.co.jp/maps?q="

(エンコードされた住所)" target ='_blank'> 大阪府浪速区湊町 2-1-57</ a>

jQuery を用いることで、少しの実装で GoogleMap へのリンク機能を追加す

ることが可能となっている。また、ほか のリンク機能もこの応用で作成できる。

電話発信へのリンク

スマートフォンからの電話発信のリンク機能は、電話番号のデータ要素に <a> タグを追加後、href 属性に「tel:電話番号」と設定することで実現できる。 【ソース7】

処理前

06-6631-8601

処理後

06-6 631-8601

このように記述しなければ、電話発信 はスマートフォンのブラウザ機能に依存 してしまう。端末の種類によっては利用 できない可能性が生じるが、機能を実装 することにより、スマートフォンの機種 を問わず電話発信が可能となる。

メール送信のリンク

スマートフォンからのメール送信のリンク機能も、電話発信と同様に、メールアドレスのデータ要素に <a> タグを追加する。メーラーの起動は、href属性に「mailto:メールアドレス」と設定することで利用可能となる。【ソース8】

処理前

info@test.co.jp 処理後

 info@test.co.jp

以上で、3つのリンク機能が追加された。

列タイトル項目名置換

最後に、キーワードが項目名に表示されてしまうため、キーワード項目の文字の置換を行い、実行結果画面へ表示しないように処理を行っておこう。【ソース9】

処理前

@GMAP 地図

処理後

地図

以上の手順により、Business4Mobile に GoogleMap 連携、電話発信、メール 送信とリンク機能を簡単に追加すること ができた。

そのほかにもこのリンク機能を応用することで、Web サーバーに配置されたWord や Excel、PDF ファイルの参照なども設定次第で容易に実現可能である。

6. 画像表示の機能拡張

Business4Mobile のデータベースの データの出力結果を加工して、図 15 の ように画像を表示する機能を追加する。 【図 15】

img タグの追加

Business4Mobile の実行結果画面に画像を表示するには、まず、クエリー作成時に列項目見出しにキーワード「@IMG」を設定しておき、また、ファイルの列のデータにあらかじめ、Webサーバー上に配置してある画像のパス(URL)を入力しておく。

設定した列項目タイトルの見出しをプログラムが判別した場合には、img タグを追加して、img タグの src 属性にファイルの URL を指定することで画像を表示する。

これら要素の追加も、jQuery を利用 すれば簡単に行える。

\$('').attr('src',' 画像パス '). appendTo(\$(argDataCol));

このたった1行のコードで実装ができる。 これは、jQueryのメソッドチェーン の機能を利用しているためで、img タ グを追加後、src 属性に画像パスの URL を設定して、img タグを現在の列要素(td 要素)へ追加している。

このように複雑な内容の処理も、少ない行数で記述できるのである。【ソース10】

処理前

http://exampleServer.jp/example.jpg

処理後

Business4Mobileの活用「メール送信」



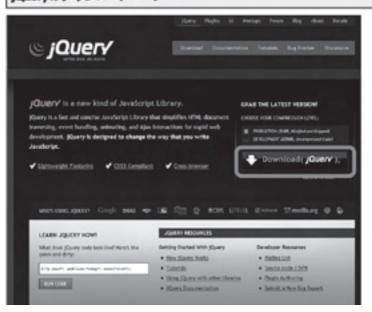
図8

Business4Mobile活用例「グラフ表示」



図9

JQueryのダウンロードページ



今回の例のように、動的に HTML の 要素が追加できるため、実装によっては、 Business4Mobile で Flash や動画など を埋め込むという拡張も考えられる。

7. おわりに

Business4Mobile は、もともとパッケージソフトなのでプログラミングを必要とせず、製品の機能だけで、スマートフォンやタブレットから簡単に活用することができる。本稿では最初に、このBusiness4Mobile の特徴や機能を述べた。

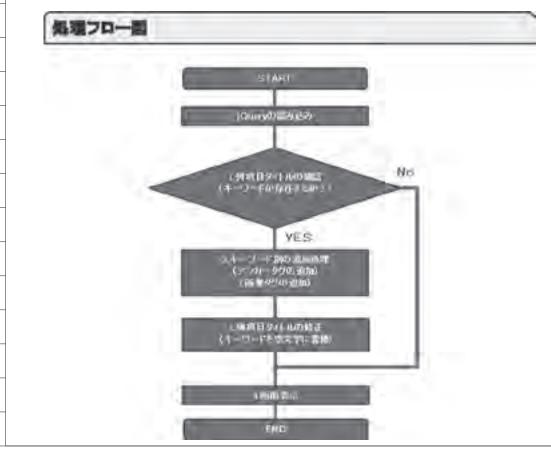
続いて、Business4Mobile に対して、 jQuery を利用した JavaScript のプログ ラミングについていくつか紹介したが、 プログラミングによって、こうした標準 以上の機能を独自に追加できる拡張性を 知っていただければ幸いである。

JavaScript を使われたことがある方であれば、そのような Business4Mobile の機能拡張も簡単に実現できるであろう。また、JavaScript は習得するのも比較的容易なプログラム言語なので、初めての方もぜひ、本稿を参考にして機能拡張にチャレンジしていただきたい。

М



図11





ソース1

HTMLから外部JavaScriptファイルの 読み込み設定

ソース2

divタグの追加

ソース3

◆MigaroExtension.js //キーワード記例 var J(EYWORD_ARR: ~ J(EYWOR

図13

クエリーの列項目名にキーワードを設定

 Business4Mobile Administration Module Cased NO CE JO DAAYev Detaine No Decemb Query Drill Down (") Tables (2.84:S) E 1 Field BOW #(I) 旦皇 ⊕ Parameters on \$266 周章 Sun Fields 回春 95'93.0nd Constaints Sun Contacts (i) Order Column/Itributes Mode Internation

図14



「一覧形式」or「辞報形式」の判別

```
#MigaroExtension.js

/##

/ 一覧形式 or 詳細形式の判別(**

* enince 2011/08/09**

* enince 2011/08/151 List形式 or DISPLAY RECURD Record形式**

* enince 2011/08/151 List形式 or DISPLAY RECURD Record形式**

* enince 2011/08/151 (**)

* yar selector=$(**-!|t|e#hite*);**

//一覧形式の場合は、t|t|e#hite*);**

//一覧形式の場合は、t|t|e#hite*);**

//一覧形式の間合は、t|t|e#hite*);**

/ 一覧形式の間合は、t|t|e#hite*);**

/ 一覧形式の間合は、t|t|e#hite*);**

/ 一覧形式の間合は、t|t|e#hite*);**

/ 一覧形式の間を見るが存在する。

* enince 2011/08/09**

* enince 2011/08/09
```

ソース5

JQueryを利用した要素取得

```
♦MigarnExtension.Is
```

ソース6

GoogleMapのリンク追加

```
♦MigarnExtension,|s
```

電話発信のリンク追加

```
◆MigaroExtension.js
```

ソース8

メールのリンク追加

```
    MigaroExtension is
```

ソース9

例タイトル項目名置換

```
    MigaroExtension.js
```

画像表示機能



ソース10

MIGARO. TECHNICAL REPORT

Migaro.Technical Report

No.4 2011 年秋

ミガロ.テクニカルレポート

[創立 20 周年記念号]

2011年11月1日初版発行

◆発行

株式会社ミガロ.

〒 556-0017

大阪府大阪市浪速区湊町 2-1-57 難波サンケイビル 13F

TEL: 06(6631)8601 FAX: 06(6631)8603

http://www.migaro.co.jp/

◆発行人

上甲 將隆

◆編集協力

アイマガジン株式会社

◆デザインフォーマット

近江デザイン事務所

©Migaro.Technical Report2011

本誌コンテンツの無断転載を禁じます

本誌に記載されている会社名、製品名、サービスなどは一般に各社の商標または

登録商標です。本誌では、TM、®マークは明記していません。



MIGARO. TECHNICAL REPORT

ミガロ.テクニカルレポート











株式会社一プロ

http://www.migaro.co.jp/

本社

〒556-0017

大阪市浪速区湊町2-1-57 難波サンケイビル 13F

TEL:06(6631)8601 FAX:06(6631)8603

東京営業所

〒106-0041

東京都港区麻布台1-4-3 エグゼクティブタワー麻布台 11F

TEL:03(5573)8601 FAX:03(5573)8602





