岩田 真和

株式会社ミガロ.

RAD事業部 技術支援課

JACi400環境でマッシュアップ!
――本格的なWeb2.0をJACi400で実現しよう!

これからのWebユーザビリティを考えると
どうしてもWeb2.0の技術が外せない。その際、JACi400により
どの程度要望に応えることができるか
その回答を考察する。

略歴
1984年7月7日生れ
2007年京都学園大学経営学部卒
2007年04月株式会社ミガロ. 入社
2007年04月システム事業部配属
2009年09月RAD事業部配属

現在の仕事内容
JACi400やDelphi/400などの開発経験を経て、現在はJACi400のサポート業務を担当。

はじめに

ミガロ.のWebアプリケーション開発ツールの1つである「JACi400」。これはRPG/COBOLといった現有プログラムスキルやDBなどの既存資産を生かして、スムーズなWebアプリケーション開発を可能にしてくれる製品である。

2008年のテクニカルレポートをご覧いただければ、比較的少ない工数で開発が可能なことや、開発自体の容易性など、そのメリットをご理解いただけると思う。

しかしながら、実際JACi400を手にしてWeb開発をスタートされたお客様の要望は、その“容易性”という視点にとどまらない場合が多い。BtoC、BtoBのやり取りに使用されるWebアプリケーションの場合はもちろんだが、社内・事業所間などで使用される場合にも、“Webとしてのユーザビリティ”というものが追求される必要がある。それはお客様が求められている「Web化」の中に、このユーザビリティが前提となっているという意味である。

さらに今後は、一般的に「Web2.0」と呼ばれる、新しいWeb技術を取り入れたいという要望も増えるであろう。それは、これからのWebユーザビリティを考えると、どうしてもWeb2.0の技術が外せなくなってきているからである。その際、JACi400により、どの程度その要望に応えることができるのか、というのは気になるところかと思う。

本稿は上記のような、現実にある要望と、これから多くなるであろう要望に対する回答になればと考えており、考察を行っていく。

内容は、参考ソースやサンプル画面の紹介も交えて「JavaScriptの活用」「Web APIの活用」「JACi400とのマッシュアップ」「JACi400アプリケーションの可能性」という4つの段階に分けて解説する。

結論的には、JACi400というIBM i専用のツールを利用しても、一般的なWeb技術がへだたりなく利用可能ということをこのレポートで強くお伝えしたい。すでにご使用の方々は、あらためてJACi400の可能性にご興味を持っていただければ幸いである。もちろんWebに精通されている方々なら、JACi400環境でも自由なWebアプリケーション開発が実現可能だということをさらに確信されるだろう。

※本レポートの内容はどちらかというと、JACi400の応用的な使い方が中心になっている。JACi400の基本的な使用方法(Web画面とIBM iとの連携)については、ここでは詳細に紹介していない。基本的な使用方法は、2008年発行のテクニカルレポートや、ミガロ.のホームページをぜひご覧いただきたい。

※本稿で紹介しているコードは、実行確認をしているものであり、できるだけそのまま使ってもらえるように心がけた。環境によっては、カスタマイズが必要な場合があることをご了承いただきたい。

ソース1

```
<body>↓
<form method="POST" name="frm">↓
<script type="text/javascript">↓
function migarodsply(){↓
var adhtml='http://www.migaro.co.jp/';↓
   var aWin=window.open(adhtml,null);↓
   aWin.focus();↓
}↓
</script>↓
```

ソース2

```
<!---// メッセージ --->↓
<INPUT type="text" name="MSG" id="MSG" style="display:none;">
```

図1

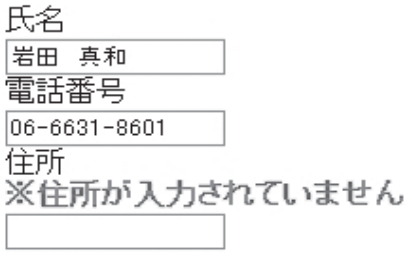
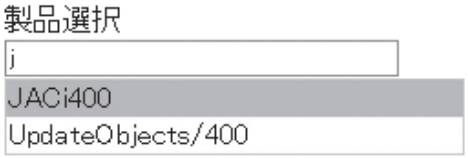


図2



ソース3

```
 (HTML部) ↓
<!---// メッセージ --->↓
<INPUT type="text" name="MSG" id="MSG" style="display:none;">↓
<!---// メッセージ位置 --->↓
<INPUT type="text" name="MSGR" id="MSGR" style="display:none;">↓
↓
 (JavaScript部) ↓
function error(){↓
        // message変数を定義する↓
        var message = '';↓
        // メッセージ内容をmessageにセット↓
        message = document.frm.MSG.value;↓
        // メッセージ位置にmessageを出力↓
        document.getElementById(document.frm.MSGR.value).innerHTML = message;↓
}↓
↓
 (表示先HTML) ↓
氏名<br>↓
<div id="NAMEM" name="NAMEM" style="color:red;font-weight:bold;"></div>↓
<input type="text" id="NAME" name="NAME"><br>↓
電話番号<br>↓
<div id="TELM" name="TELM" style="color:red;font-weight:bold;"></div>↓
<input type="text" id="TEL" name="TEL"><br>↓
住所<br>↓
<div id="ADDM" name="ADDM" style="color:red;font-weight:bold;"></div>↓
<input type="text" id="ADD" name="ADD"><br>↓
```

JavaScriptの活用

JavaScriptとは

まず取り上げたいのが「JavaScript」である。JavaScriptは、ブラウザ上で動く簡易言語と考えていただければよい。インターネットが誕生してまもなく登場し、以前から個人のWebサイトでも多く使用されているほど、敷居の低い言語として知られている。

なぜ、JavaScriptを最初に取り上げるのかというと、それだけWebの世界になくてはならない存在だからである。役目としては、画面定義するHTMLとデータベースに書き込むメインプログラム（RPG/COBOL）の仲介役と言えるだろう。今では「Ajax（エイジャックス）」という技術が話題になっているが、JavaScriptはその柱になっている。

さらに心強いことに、JACi400では入力文字数の制限・カンマ編集といったWebで必要とされている基本的なJavaScriptの機能が自動的に提供される。そのため、開発者が付け加えるのは、プラスアルファで組み込みたい機能のみとなる。

今回は、実際のWebアプリケーションで使えそうな、JavaScriptの活用例をいくつか紹介する。一般的なWebサイトに接続していることを想像しながらご確認いただきたい。

JACi400環境でのJavaScriptの使用

さて、例をご紹介する前に、JACi400上でJavaScriptを使用するにあたり、いくつかあるコツをお伝えしておきたい。

通常のWeb開発であれば、“HEADタグ”内にJavaScriptを記述するのが普通である。JACi400では、HEAD部分は、JACi400アプリケーションが起動する際に、自動的にJACi400の動作上必要なソースで上書きされる。そのため、JavaScriptを“BODYタグ”内に記述することにより、上書きを回避する必要がある。逆に考えれば、そこだけ気をつけていれば、JACi400アプリケーションでは、通常のWebと同様にJavaScriptを使用することができる。【ソース1】

また、よく使うテクニックとして、JACi400で使用するフィールドを「隠しフィールド」にしてしまう手法がある。テキストフィールドのstyle属性に"display:none;"を指定することで、そのフィールドを見えなくしてしまう。こうすると、画面からは見えなくても、内部的には値を持っていることになり、JavaScriptから自由に入出力が可能になる。【ソース2】

わかりやすいメッセージ出力

簡単な例として、JavaScriptを使用して、“わかりやすいメッセージ出力”というものについて考え、表現してみたい。

例えば、入力画面で入力不備があり、入力不備のエラーを発生させたいケースでは、通常は、あらかじめ設計者が決めたメッセージ出力用の場所にメッセージが出力される。その際、どんなメッセージを出力しても、常に同じ位置に表示される。この方法だと、ユーザーとしては、どの項目が原因でエラーとなっているのか把握しにくい。

そこで、JavaScriptを使用し、エラーの原因となっているフィールドの真上にメッセージが出てくるようにする。この場合は、隠しフィールドとしてメッセージ内容とメッセージ位置を用意し、エラー時にIBM iからセットするようにする。【図1】【ソース3】

【ソース解説】

JavaScriptにより、JACi400から取得した値を、表示先HTML部のSPANタグにセットさせる。ここでは、メッセージ位置がNAMEMの場合は氏名のSPANタグ、TELMの場合は電話番号のSPANタグにメッセージがセットされる。

なお、SPANタグにはあらかじめ赤色・太字のフォントが指定してあるので、通常よりも目立ってわかりやすいと思う。

suggest.jsによる自動補完（オートコンプリート）

最近、検索サイトなどでよく見かける、文字列の自動補完の機能を導入しよう。自作しようとすると高度な知識が必要になるが、「ライブラリ」と呼ばれる関数群をうまく使用することにより、シンプルなJavaScriptで機能の実現が可能になる。

多くのライブラリはオープンソースになっており、ライセンス規約を守れば無償での利用が可能である。suggest.jsはMITライセンスであり、再配布をする際にはライセンスの表記が必要になる。（※）【図2】【ソース4】

【ソース解説】

掲載したのは、ライブラリ部分から変更したところのみにした。今回はミガロ.の製品を選択する形である。

なお、このぐらいの項目であれば、コンボボックスのほうが使いやすいかもしれない。

このライブラリは、外部ファイルの読み込みも可能であるため、自動補完を有意義に使用できる場面で利用していただきたい。

※ライブラリ提供元：Enjoy*Study
http://www.enjoyxstudy.com/javascript/suggest/

Flashとの連携

「Flash」を用いると柔軟な表現が可能になるため、Webカタログなどで演出に最適である。

では、Flashで選択したデータをJACi400に送りたい場合、どのようにすればよいのだろうか。これも、JavaScriptによって可能になる。Flashの場合は、Flash内に「ActionScript」と呼ばれる（JavaScriptによく似ている）コードを記述することで、FlashとJavaScriptを連携させる必要がある。【図3】【ソース5】

【ソース解説】

ActionScriptは、Flash.externalパッケージのクラスを利用することで、JavaScriptの呼び出しを可能にしている。Flashで入力された文字列は、ActionScript内でtextと定義された変数として取得できる。これをパラメータとして、JavaScriptを呼び出すのである。JavaScriptに渡されたパラメータは、alert関数によりメッセージを出力している。

ちなみに、ここでは掲載しないが、JavaScriptからActionScriptを操作することもできる。そのため、IBM iから取得した初期値を、Flashに渡すとい

ソース4

```
（項目設定部分）↓
// 補完候補の配列作成↓
var list = [ 'JACi400', 'Delphi/400', 'UpdateObjects/400', '*noMAX', 'MKS Integrity' ];↓
↓
（表示先HTML）↓
<!-- 入力エリア --> ↓
<input id="text" type="text" name="text" value="" autocomplete="off" size="40" style="display: block"/> ↓
<!-- 補完候補を表示するエリア --> ↓
<div id="suggest"></div> ↓
←
```

図3

ソース5

```
（ActionScript部分）↓
import flash.external.*;↓
import mx.controls.*;↓
↓
btn.onRelease = function(){↓
        retuneText.text = ExternalInterface.call("FlashToJS",text);↓
}↓
↓
（JavaScript部分）↓
function FlashToJS(message){↓
        alert(message);↓
}↓
←
```

ソース6

```
|（HTML）↓
<form action="upload.php" method="post" enctype="multipart/form-data" name="upfrm">↓
  <input type="file" name="upfile" size="30"><br>↓
  <input type="submit" value="Webサーバーにファイルをアップロード"↓
         onmousedown="parent.document.frm.PATH.value = document.upfrm.upfile.value;">↓
</form>←
```

図4

表1

代表的なWeb API 一覧

サービス名	取得できる主な情報
Yahoo!デベロッパーネットワーク	ニュース、オークション
Amazon Web サービス	商品情報
価格.com	商品情報
お天気Webサービス	天気情報
できるじゃらんWebサービス	宿情報
ホットペッパーWebサービス	飲食店情報
Wikipedia API	Wikipedia情報
SOBA Web API	カメラ画像の共有、デスクトップ共有
Force.com WebService API	SalesForce情報
SimpleAPI:ウェブサイトサムネイル作成API	ウェブページのサムネイル作成

う仕組みも可能となる。

PHPプログラムとの連携（ファイルアップロード機能の付加）

画像ファイルをJACi400上で扱う場合、ポイントが2つある。まずは、画像ファイルをWebサーバーに置く。それから、IBM i 側でその表示したい画像のパスを指定する。このような連携を用いて、JACi400は画像を表示している。

照会のみであれば、上記で問題はない。だが、ユーザーサイドから画像を送りたい場合は、別の手段を用いて、画像ファイルをWebサーバーにアップロードする必要がある。このような仕組みは、サーバーサイドで動作するプログラムによって実現できる。

いろいろな手段があるが、今回は一般的なWebにおいて一番敷居が低いと思われる、オープンソースのPHPを使用した実現方法を解説する。

ここでは、PHP Laboが提供するファイルのアップロード機能を参考に、画像ファイルをWebサーバーに転送する。JACi400上で使用する際のコツは、この画面をインラインフレームに置くという点である。そうすれば、制御がIBM i 側に戻らずにファイルをアップロードできる。【図4】【ソース6】

なお、PHPやPHP Laboに関しては、以下のURLを参照のこと。（※）

【ソース解説】

まず、Webサーバー上にPHPをインストールし、使用できる環境を用意する。

ここでは、JACi400との連携に向けて、アップロードしたファイル名を取得するためにJavaScriptを組み込み、親フレームに配置してあるJACi400のフィールドへ値を渡している。

※ PHP Labo
http://www.php-labo.net/tutorial/php/upload.html

以上、JACi400環境において、Webアプリケーション作成時に使えそうなJavaScriptの活用例を4点挙げた。

ここでご紹介した内容は応用的ではあるが、特別複雑なことを行っているわけではなく、一般的なWebサイトでは多く導入されている技術である。しかし、このようなことでも使用感が全く違ってくる。JACi400アプリケーションを作成する際にぜひ参考にしてほしい。

Web APIの活用

Web APIとは

次に取り上げたいのは「Web API」である。

「API」とは、Application programming interfaceの略であり、特定の機能を持ったプログラムの部品と解説される場合が多い。利用者はその都度プログラミングすることなく、その機能を利用できる。APIというとOSの機能をプログラムから利用するための窓口というイメージがあると思うが、Web APIはOS上ではなくWeb上に散在しているAPIである。一般的に「Webサービス」とも呼ばれる。

OSやフレームワークが提供するサービスではなく、Web上のサービスとして提供されており、その機能を取り込める。つまり、今までローカルのパソコンではできなかったさまざまな機能を簡単に利用できるようになる。

Web APIの代表例としてよく出される「Google Maps API」を題材に、簡単な活用例を説明する。

Google Mapsは、言わずと知れたGoogleの地図検索サービスであり、衛星写真やストリートビューといったサービスで話題になっている。しかし最近は、他のWebサイト上でもGoogle Mapsを見かける機会がないだろうか？

具体的には例えば、グルメサイトから、出張先でランチに食べるラーメン屋を探すとする。そこで表示された地図を見て、縮尺などを操作して印刷する。いつもの使用感と変わらないなと思っていたら、地図の左下に「POWERED BY Google」のロゴが入っていたことに気づく――実はこれが、Google Maps APIを使用したサイトの例だ。Google以外のサイト上でも、多くのユーザーが使い慣れているGoogle Mapsを利用できるのである。

このようにWeb APIを利用することで、いままで地図データを自社で用意しなければいけなかったものが不要になるだけでなく、その構築工数を省くことが可能になる。しかも地図の更新等は、Webサービスの提供元により自動的に行われる。

このグルメサイトの地図サービスの例は、既存の「飲食店検索システム」とのマッシュアップ例とも言える。“マッシュアップ（Mashup）”とはマッシュポテトをイメージしていただければわかりやすい。Web APIなどを掛け合わせて新しい価値を生み出す、というWeb2.0の用語である。

なお、マッシュアップ、JACi400でのマッシュアップの実践については、後半にくわしく説明する。

Web APIが提供する機能と必要な知識

JACi400との関係や連携方法を解説する前に、Web APIについて、どのような機能が提供されているのかを確認しておこう。また、それを使用する際に必要な知識を解説する。

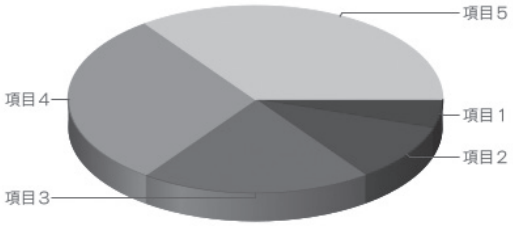
Web APIの実例の中で、有名でわかりやすい例として挙げたいのが「Google AJAX API」である。前述した「Google Maps API」もこの一部である。またGoogle AJAX APIは、簡単にグラフ化を行う「Google Chart API」や任意のサーバーからRSSフィードを取得する「Google AJAX Feed API」など、多くのAPIを提供している。

Google以外にもニュースや天気予報など、切りがないほどのWeb APIが公開されている。しかも、一部有償のものもあるが、ほとんどは無償で提供されている。【表1】

1つ1つの情報は、ネット検索でその都度調べれば不便ではない情報ばかりだが、これらWeb APIを組み合わせる（マッシュアップ）ことにより、新しい価値を生み出すことができる。つまり、JACi400を使用してマッシュアップすれば、IBM i の資産とつながることが可能になる。

これらを実装する際には多くの知識は必要なく、ハードルはそれほど高くない。前述したようなJavaScriptの基礎を理解していれば、ある程度は扱えるようになる。もちろん、使用方法は各デベロッパーサイトにも載っている。なにより、これらWebの世界は技術情報が多くWeb上に存在しているため、信用できるサンプルソースが入手できれば、それ

図5



ソース7

```
 (HTML) ↓
<input type="text" id="JS1A" name="JS1A" style="display:none;"><!--項目１-->↓
<input type="text" id="JS2A" name="JS2A" style="display:none;"><!--項目２-->↓
<input type="text" id="JS3A" name="JS3A" style="display:none;"><!--項目３-->↓
<input type="text" id="JS4A" name="JS4A" style="display:none;"><!--項目４-->↓
<input type="text" id="JS5A" name="JS5A" style="display:none;"><!--項目５-->↓
<input type="text" id="JS1B" name="JS1B" style="display:none;"><!--データ１-->↓
<input type="text" id="JS2B" name="JS2B" style="display:none;"><!--データ２-->↓
<input type="text" id="JS3B" name="JS3B" style="display:none;"><!--データ３-->↓
<input type="text" id="JS4B" name="JS4B" style="display:none;"><!--データ４-->↓
<input type="text" id="JS5B" name="JS5B" style="display:none;"><!--データ５-->↓
 (JavaScript) ↓
function dsp()[↓
        var str = '';↓
        str = str + '<img src="http://chart.apis.google.com/chart?';↓
        str = str + 'chs=500x200';↓
        str = str + '&amp;chd=t:';↓
        str = str + document.frm.JS1B.value;↓
        str = str + ',';↓
        str = str + document.frm.JS2B.value;↓
        str = str + ',';↓
        str = str + document.frm.JS3B.value;↓
        str = str + ',';↓
        str = str + document.frm.JS4B.value;↓
        str = str + ',';↓
        str = str + document.frm.JS5B.value;↓
        str = str + '&amp;cht=p3&chco=0000ff';↓
        str = str + '&amp;chl=';↓
        str = str + document.frm.JS1A.value;↓
        str = str + '|';↓
        str = str + document.frm.JS2A.value;↓
        str = str + '|';↓
        str = str + document.frm.JS3A.value;↓
        str = str + '|';↓
        str = str + document.frm.JS4A.value;↓
        str = str + '|';↓
        str = str + document.frm.JS5A.value;↓
        str = str + '" alt="売上種別" />';↓
↓
        document.getElementById('TEST1').innerHTML = str;↓
]↓
 (出力用HTML) ↓
<div id="TEST1"></div>↓
```

を参考にして組まれていってもいいだろう。一般的なWebの技術情報をそのままJACi400に応用できることも、HTMLを自由に作成できるというJACi400の魅力である。連携も非常にシームレスである。

JACi400環境でのWeb APIの使用 ―JACi400でグラフを表示させる

JACi400環境での、Web APIのシンプルな例を1つだけ紹介する。活用については、マッシュアップという形で説明したいと思うので、ここでは、Web APIとJACi400との連携方法を中心に述べる。

例えばIBM i上にある売上情報をビジュアルに把握したい場合、どうしてもグラフを活用したくなる。しかし、Web上でグラフを表現する場合、テーブルを駆使したり、複雑なJavaScriptを実装したり、そのためだけにFlashを導入したり、ハードルの高いイメージが少なからずある。

そこで、前述のGoogle Chart APIを用いてグラフ化を導入してみる。Google Chart APIは非常にシンプルな構造になっていて、「データ」と「フィールド名」をURLのパラメータとして受け渡せば、グラフを画像として表示してくれるものである。

なお、この方式を一般的に「REST方式」と呼び、導入がシンプルでわかりやすいため、多くのWeb APIがこの方法を採用している。

呼び出しURLの例

```
http://chart.apis.google.com/chart?cht=p3&chd=t:60,40&chs=250x100&chl=Hello|World
```

例えば、このようなURLを実行すると、自動的に2つの項目があるグラフとして画像を表示させることができる。

このURLを分解すると以下のようになる。

```
① http://chart.apis.google.com/chart?
② cht=p3&
③ chd=t:60,40&
④ chs=250x100&
⑤ chl=Hello|World
```

まず、①とそれ以降に大きく分かれる。①がグラフを表示する関数（もしくはAPI名）で、それ以降はパラメータと思っていただくとわかりやすいかと思う。

ちなみに、②はグラフの種類、③は項目の値、④はグラフの大きさ、⑤は項目の表題にあたる。このようなシンプルな構造により、動的にURLを作成するだけで、動的なグラフ表示が可能になる。

図5は、JACi400と連携を行ったグラフサンプルだが、ここで記述しているソース7についてはそれほど多くないことに気づくであろう。このソースでは5項目固定になっており、動的に増やそうと思えば、それに応じたJavaScriptを記述してURLを完成させればいい。【図5】【ソース7】

また、ここで1つ気をつけていただきたいことがある。このソース7をそのまま実行すると、項目名に、例えば全角文字を指定した際にその文字が欠けてしまう。これは、多くのWeb APIがWeb上で主流になりつつあるUTF-8の文字コードを採用しているためであり、うまくパラメータの受け渡しができていないのが原因である。

Shift_JISで開発を行っている場合は、何らかの処置が必要である。一般には、JavaScriptであらかじめ用意されているencodeURIComponent関数を利用することによって、対処が可能だ。

```
encodeURIComponent(document.frm.JS1A.value);
```

セットする際に、このencodeURIComponent関数を挟んで受け渡すことで、文字コードの問題を回避できる。図5のような形で、JACi400上でのグラフ表示が可能になる。

Web APIを利用する場合の注意点

今回、非常に便利なサービスとして、Web APIの活用を取り上げた。ここで念のため、Web APIの利用において配慮が必要な点を2つ挙げておきたい。

●利用許諾条件

利用用途によっては、サービス対象外のケースとなる。例えば、Google Mapsは一般のWebユーザーが観覧できないサイトでなければ使ってはいけない、という規約が存在する。注意してほしい。

●サービスの停止リスク

第三者から提供されるサービスであるため、サービスの停止などには留意していただきたい。複数のWeb APIを使用する場合は1つのサービス停止が、システム全体の停止につながらないように工夫が必要だ。また、これらのサービスを使用する場合、事前に代替サービスを探しておくことも必要な対策と言える。

JACi400とのマッシュアップ

例：ミガロ. オリジナルTシャツ注文システム

今回は、JavaScriptやWeb APIなど、Webアプリケーションのさまざまな仕組みをご紹介した。せっかくなのでこれらの技術を使用し、JACi400でのマッシュアップを実践してみたい。

今回作成するのは、在庫照会注文システムだ。わかりやすいように「ミガロ. オリジナルTシャツ注文システム」と名称をつけ、ユーザーからオリジナルTシャツの注文をもらうシンプルな仕組みをWebアプリケーションに作り上げる。【図6】【図7】【図8】

これまで紹介した技術、JavaScript、Web APIなどをできるだけ多く使用するものとする。

①タブ「商品を選ぶ」

まず、Flashでビジュアルなカラー選択ボタンを作成した。もし、すでにWebカタログとしてFlashがある場合は、それをカスタマイズしてもよい。自社で作成しなくても、この部分が得意なWebデザイン会社等に依頼するなどと想定していただければと思う。

Flashを使用しているので滑らかな表現が可能になる。ここでカラー選択、個数指定なども可能になる。人気カラーの内訳はGoogle Chart APIによるものである。【図6】

②タブ「画像を合成する」

Flash指定が終われば、ページ自体は変わらずタブを移動する。この機能はJavaScriptによって実現している。

ユーザーは、PHPの機能によってファ

図6


MOTOS
Migaro. Original Tshirts Order System
ミガロ.オリジナルTシャツ注文システム
1. 商品を選ぶ → 2. 画像を合成する → 3. ご注文内容の確認をする
お好きなカラーと個数を選択してください。
Blackを選択しています。
人気カラー
20枚/月
40枚/月
30枚/月
Black
Blue
Yellow
現在の在庫：5個
Black
Blue
Yellow
3
個
商品確定

図7


MOTOS
Migaro. Original Tshirts Order System
ミガロ.オリジナルTシャツ注文システム
1. 商品を選ぶ → 2. 画像を合成する → 3. ご注文内容の確認をする
イメージをアップロードしてください。
C:¥Documents and Settings¥m
参照...
UPLOAD
ミガロマラソンクラブ
ななうまさがえ
画像確定

図8


MOTOS
Migaro. Original Tshirts Order System
ミガロ.オリジナルTシャツ注文システム
1. 商品を選ぶ → 2. 画像を合成する → 3. ご注文内容の確認をする
ご注文内容にお間違いがなければ、注文確定ボタンを押してください。
Color:1
数量:3
画像:logo.jpg
Web ページからのメッセージ
注文を確定してよろしいですか？
OK
キャンセル
注文確定

イル(ロゴデザインイメージなど)をアップロードし、JavaScript により T シャツとアップロードした画像を組み合わせたイメージ（T シャツの完成イメージ）を確認することができる。【図 7】

③タブ「ご注文内容の確認をする」

最後に、確認画面に移る。この時点ではまだ IBM i には情報を渡していない。今まで入力された情報はすべて隠しフィールドに存在しているので、その値を確認情報として出力している。注文確定の OK ボタンを押して注文を完了させる。

応用したいツールが Flash であっても Web API であっても Ajax であっても、結局はその隠しフィールドに値が入力される。そこからの処理は JACi400 の通常機能であり、JACi400 だから複雑なロジックが必要になる、といったことは発生しない。

なお、ブラウザとしての画面遷移は最後の 1 回だけになる。そのほかは、タブ形式で徐々に右に移っていくという視覚的にわかりやすい流れになっている。【図 8】

実際のアプリケーションでは、このように各種の技術を必要以上に盛り込むことは実用的とは言えないだろう。とはいえ、今回のプログラムにより、JACi400 環境において、Web2.0 と呼ばれるような先進の技術との連携の可能性が確かめられた。

JACi400アプリケーションの可能性

今後は、今まで以上に基幹システムと Web システムが統合していく社会が訪れる。しかも、それは金融 / 銀行業やネット販売業、ネットサービス業といった、Web システム自体が基幹システムとなっている業界だけの話ではない、と言われている。

これは、多くの企業がこれまで見てきた「Web ＝情報発信」の常識が変わろうとしているということである。情報発信を超える可能性が Web にはある。

そんな中、JACi400 というツールに注目していただくメリットとしては、次のようなことが考えられる。

- **IBM i のメリットを最大限に生かした Web アプリケーションの実現**
- **基幹システムとのシームレスな連動**
- **開発コスト、運用コストの削減**

そして、本稿で取り上げ、ここまで考察を加えてきた以下のポイントにより、JACi400 の利用方法も今後、大幅に広がっていくと思われる。

- **あらゆる Web ツールとの連携による可能性**

さいごに、本レポートを執筆して、あらためて JACi400 が初級者から上級者まで扱えるツールだということがわかった。さらに Web 初心者だからこそ、JACi400 によって、IBM i 上から直接データを配信し、今までにない新しい価値を創出することもできる。そういったユーザーの方々が増え続けることを願っている。

M

現在の仕事内容（詳細）
JACi400 や Delphi/400 などの開発経験を経て、現在は JACi400 のサポート業務を担当。
幅広いバックボーンを備えた SE を目指し、日々業務に邁進中である。今後はさらに CSS や JavaScript、Ajax、その他 Web に関する知識を高めたい。