

【セッションNo. 2】

## Delphi/400開発ノウハウお教えします 現場で培った開発手法公開

株式会社ミガロ。  
システム事業部 システム3課 担当課長  
尾崎 浩司

## 【アジェンダ】

### Webアプリケーション 開発手法公開

- ① JAVAスクリプトとの連動
- ② ブラウザへのグラフ表示

### GUIアプリケーション 開発手法公開

- ③ WEBサービスの使用
- ④ ガントチャートの実装
- ⑤ オートフィルタ機能の拡張

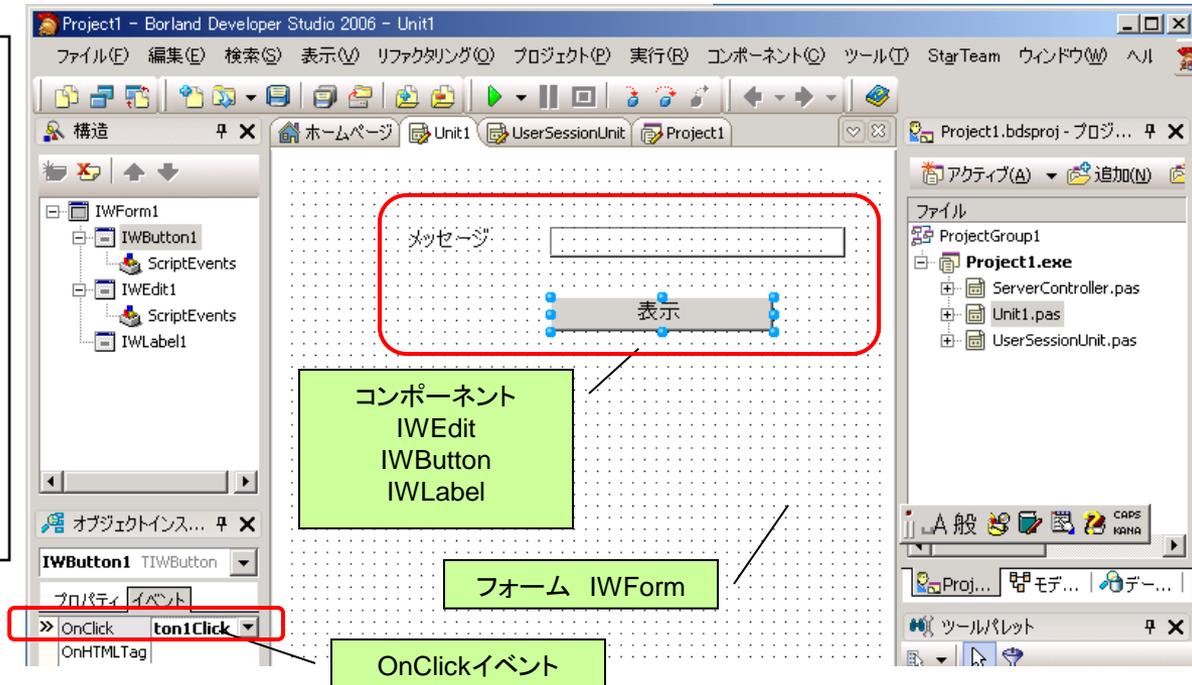
## Delphi/400によるWEBアプリケーション開発

### ■ VCL for the Web(IntraWeb)について

- GUIアプリケーションと同様の手順で、フォームやコンポーネントを使用しDelphi言語にてWEBアプリケーションを構築する手法
- HTML、CSS等の詳細を知らなくても、WEBアプリケーションを構築可能

#### <開発手順>

1. 新規プロジェクトの作成
  - スタンドアロンアプリケーション
  - ISAPIアプリケーション
2. フォームにVCL for the Web用コンポーネントを貼り付けてプロパティの設定
3. 必要に応じてイベントハンドラを作成
4. コンパイル
5. 実行(テスト)

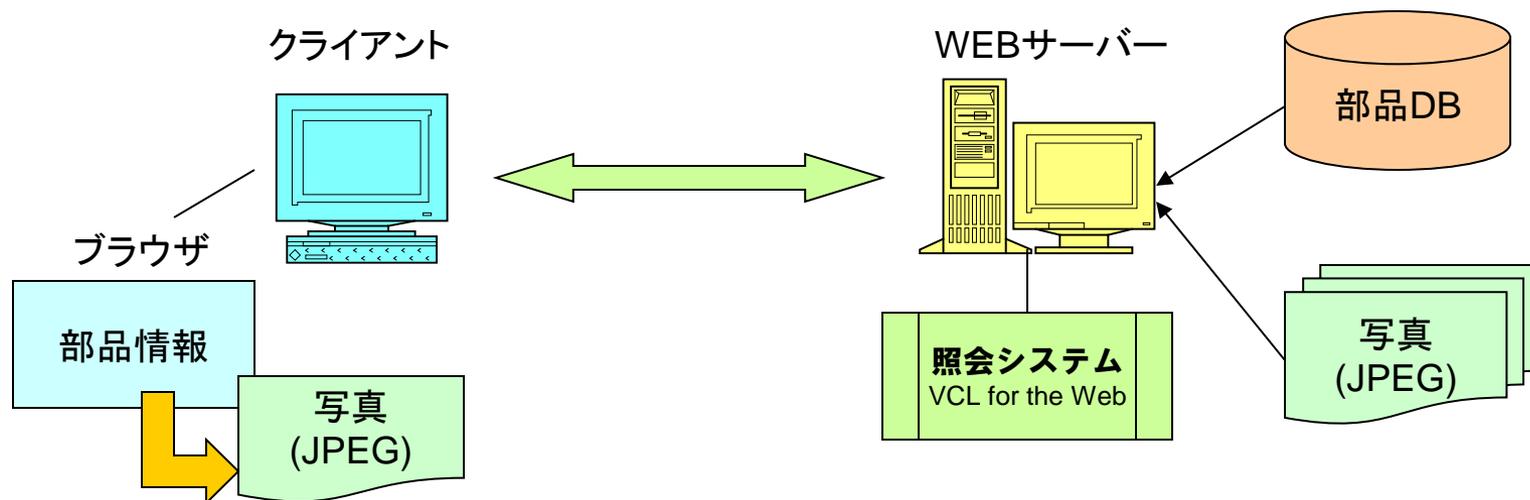


## Webアプリケーション開発手法公開 ① JAVAスクリプトとの連動

## ① JAVAスクリプトとの連動

### ■ 案件概要 (WEB部品照会システム)

- 写真情報を含む部品情報をスムーズに検索したい
  - 社内だけでなく、販売代理店にも部品情報を公開できるようにVCL for the Webを使用したWEBアプリケーションを開発
  - 検索した部品情報は、多数の写真も含むため、**JAVAスクリプト**を使用し原寸大の写真情報をポップアップ表示させることを実現



## ① JAVAスクリプトとの連動

### ■ 開発事例紹介

サムネイル画像をクリックすることにより、別ウインドウに原寸大の画像を表示

The screenshot shows a web browser window with a URL starting with 'http://127.0.0.1:8888/EXEC/2/1aqhx870ggdypw13xf7651hxx81k'. The page content includes a table with 'Packing details' and 'Size information'. Below this is a 'Parts Picture' section with three thumbnails of hardware parts: a cylindrical nut, a bolt, and a hexagonal nut. A yellow arrow points from the middle thumbnail to a larger window on the right.

Packing details	
重量単位	KG
タイプ	L * H * H cm
備考	

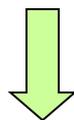
Size information	
情報	



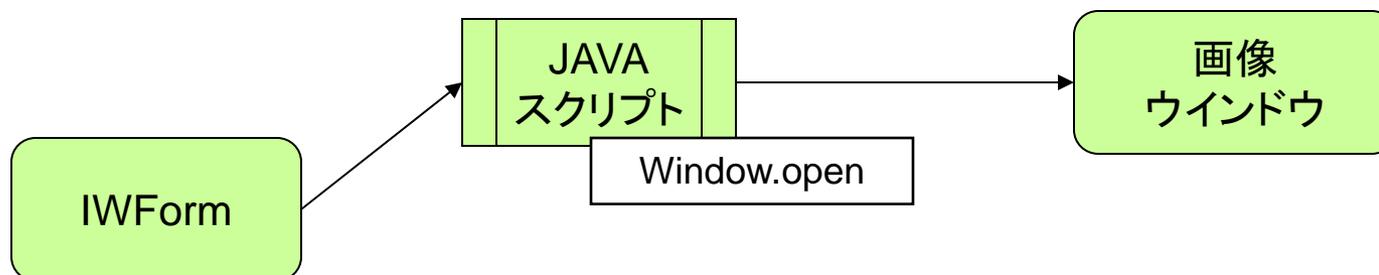
## ① JAVAスクリプトとの連動

### ■ 開発手法

- VCL for the Webのフォームに表示したサムネイル画像をクリックすることで、別ウィンドウに原寸大表示できないか？



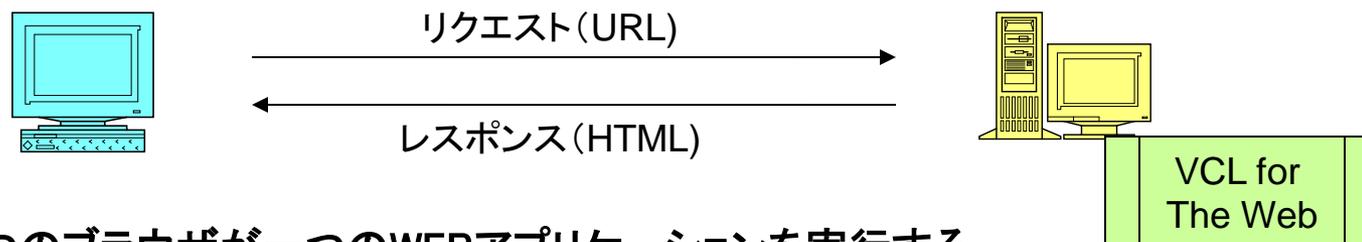
- **JAVAスクリプト**を使用することで、VCL for the WebのフォームをサブMITさせることなく、別ウィンドウを開くことを実現



## ① JAVAスクリプトとの連動

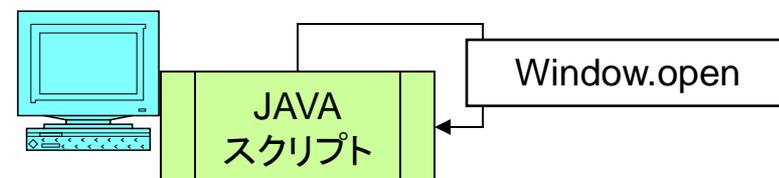
### ■ JAVAスクリプトについて

- WEBアプリケーションは、通常クライアントのリクエストとWEBサーバーのレスポンスを繰り返して稼働する



- 通常一つのブラウザが一つのWEBアプリケーションを実行する
  - WEBサーバーからのレスポンスとして別のウィンドウを開く事は不可能である
- JAVAスクリプトは、**ブラウザ(クライアント)上で実行可能**なスクリプト言語の為、クライアントの操作をトリガーに別ウィンドウをオープンするといった処理が可能である

⇒ VCL for the Webのようなサーバーサイドプログラムでは実現できない操作等は、**JAVAスクリプト**の使用を検討する



## ① JAVAスクリプトとの連動

### ■ JAVAスクリプト埋め込み手順

The screenshot shows the Delphi IDE with a form containing a button. The 'ScriptEvents' property of the button is being edited. A callout box explains that the JavaScript code should be entered into the 'ScriptEvent' property of the component. The 'VCL for the Web Event Scripts Editor' dialog is open, showing a list of events and a text area for the script code. A yellow arrow points from the 'ScriptEvents' property in the Object Inspector to the 'onClick' event in the dialog. A red box highlights the 'onClick' event in the list, and another red box highlights the code area where the JavaScript code is entered. A third red box highlights the 'ScriptEvents' property in the Object Inspector.

処理を記述したいコンポーネントの ScriptEventプロパティにJAVAスクリプトを記述

ScriptEvents プロパティ

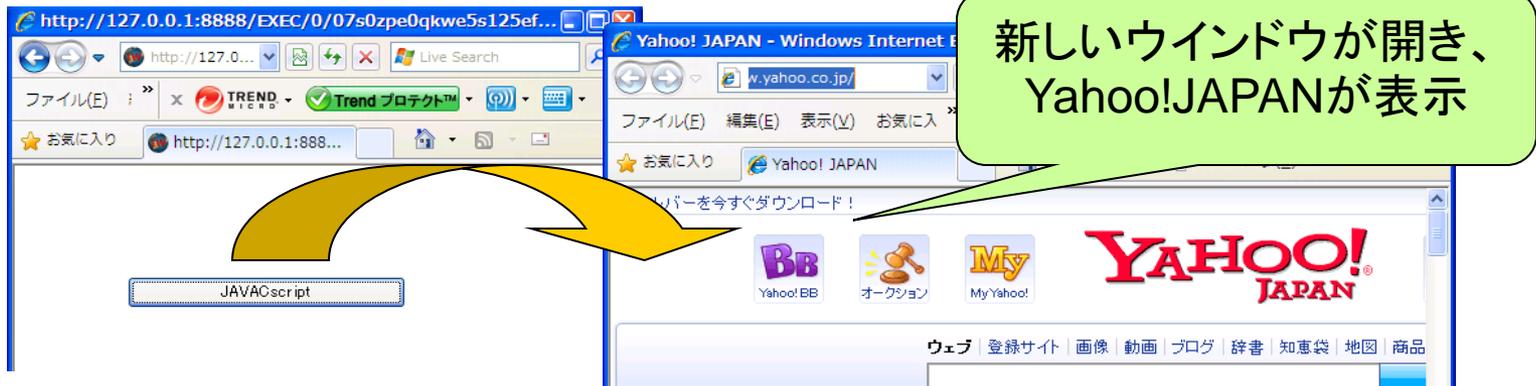
イベント種類

JAVAスクリプトソース

```
newWin = window.open("", "new", "");
newWin.location = "http://www.yahoo.co.jp";
```

## ① JAVAスクリプトとの連動

### ■ JAVAスクリプト埋め込みプログラムの実行



### ■ JAVAスクリプトをコードで記述する方法

- HTMLが生成されるタイミング(Renderイベント)で、JAVAスクリプトを追加する

```
procedure TIWForm1. IAppFormRender (Sender : TObject);  
begin  
    // ボタンonClicにJAVAスクリプト指定  
    IWebButton1. ScriptEvents. Values['onClic'] :=  
        'newWin = window.open("", "new", "");' +  
        'newWin. location = "http://www.yahoo.co.jp";';  
end;
```

JAVAスクリプトを  
文字列で記述

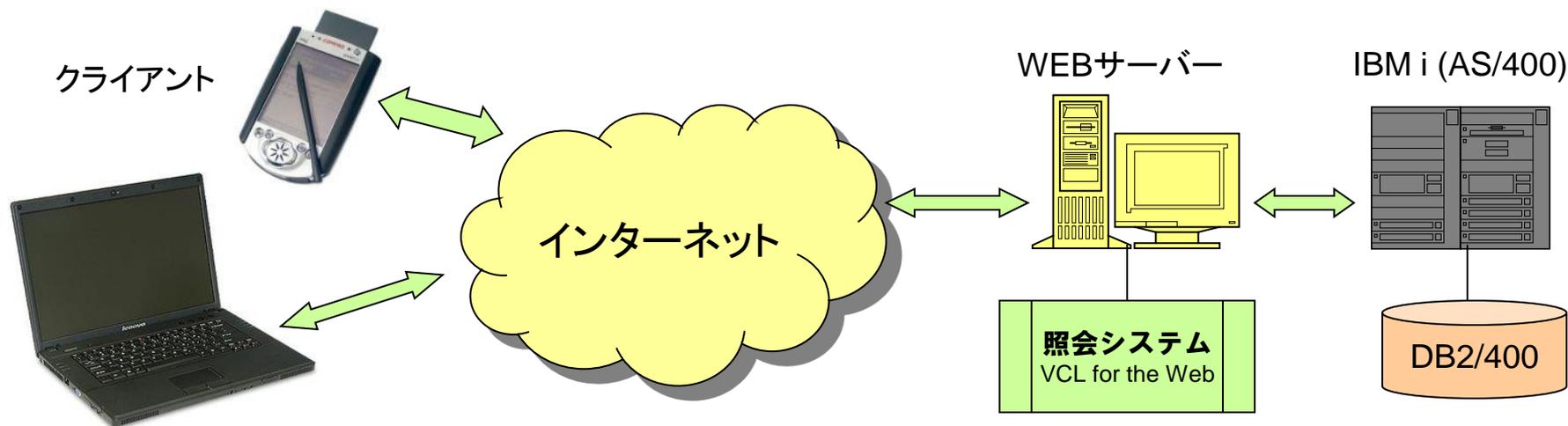
コードで記述の為、  
変数にする事も可能

## Webアプリケーション開発手法公開 ② ブラウザへのグラフ表示

## ② ブラウザへのグラフ表示

### ■ 案件概要 (WEB情報照会システム)

- 社内にある情報をいつでも、よりビジュアルな形でアクセスしたい
  - PCだけでなく、WindowsMobileやiPhone等の携帯端末からも情報にアクセスする為、VCL for the Webを使用したWEBアプリケーションを開発
  - 照会結果は、単なる数字の羅列だけでなく、概要をすばやく把握するために、**ブラウザ上にグラフを表示**することを実現



## ② ブラウザへのグラフ表示

### ■ 開発事例紹介

年度 2008 拠点 200 大阪

抽出条件を指定

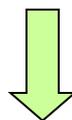
ブラウザ上にグラフを表示

	01月度	02月度	03月度	04月度	05月度	06月度	07月度	08月度	09月度	10月度	11月度	12月度
Delphi/400	500.0	700.0	500.0	200.0	600.0	1,000.0	300.0	200.0	1,000.0	1,200.0	1,500.0	2,000.0
JACi400				500.0	500.0	700.0	800.0	1,000.0	500.0	1,000.0	1,500.0	2,000.0
教育・研修				100.0	300.0	200.0	400.0	300.0	200.0	500.0	100.0	750.0

## ② ブラウザへのグラフ表示

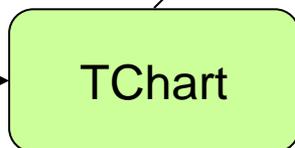
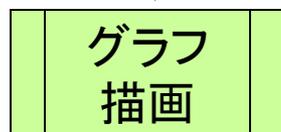
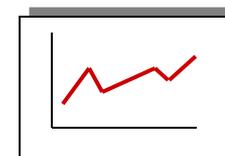
### ■ 開発手法

- グラフを描画するTChartをWEBアプリケーションで応用できないか？

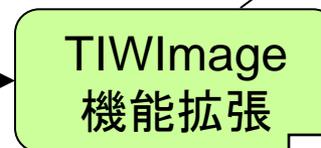


- TChartコンポーネントで作成した**グラフを画像化**し、ブラウザ上に画像として表示することで対応

PCCDKS	PCNMKK	PCNMKS	POKR01	POKR02
01	北海道	稚内	1088	1111
02	北海道	旭川	1054	1275
03	北海道	根室	1357	1211
06	北海道	帯広	1091	765
06	北海道	函館	1511	1228
07	東北区	青森	1290	1028
08	東北区	秋田	1893	1594
09	東北区	盛岡	1702	1044
10	東北区	仙台	1337	973



関連付け



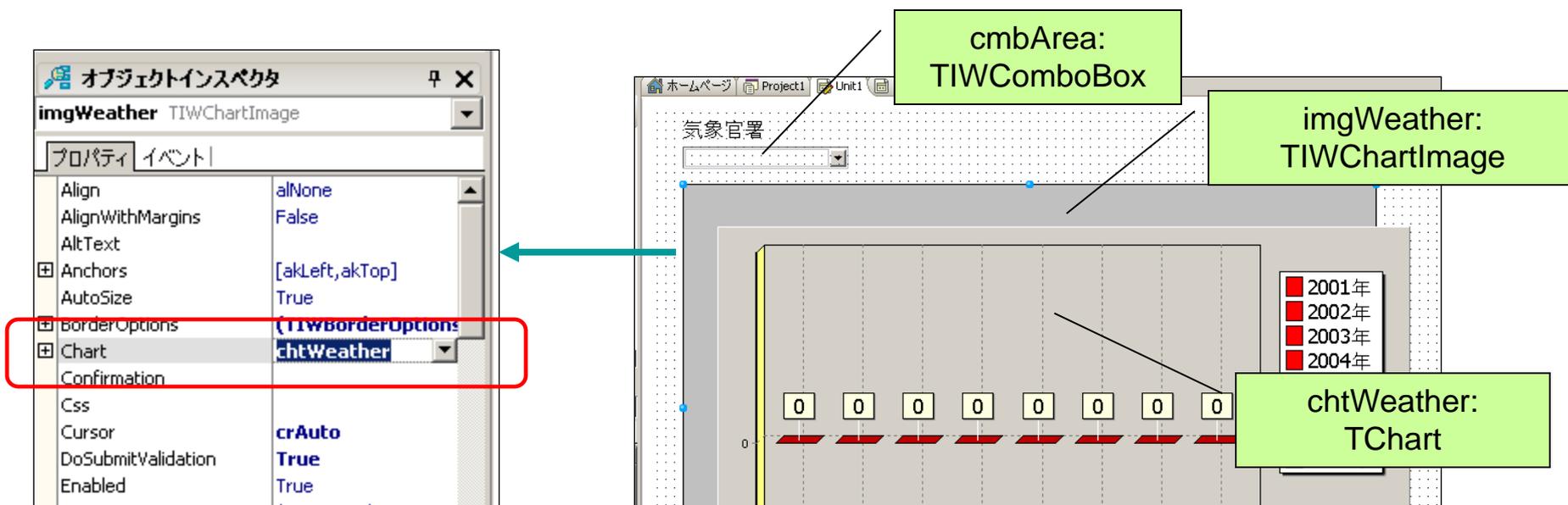
画像化



## ② ブラウザへのグラフ表示

### ■ サンプルプログラムの作成

- VCL for the Webに用意されている画像表示コンポーネント(TIWImage)に下記機能を拡張したTIWChartImageコンポーネントを使用する
  - TChartコンポーネントとの関連付けを可能にする (Chartプロパティ)
  - HTML作成時に、TChartにより作成されたグラフをTeeCreateBitmapメソッドを使用し、画像化したものをImageとして出力する



## ② ブラウザへのグラフ表示

### ■ サンプルプログラムソース

```
procedure TIWForm1.cmbAreaChange(Sender: TObject);
const
  cSQL = 'SELECT PCKR01, PCKR02, PCKR03, PCKR04, '
        + 'PCKR05, PCKR06, PCKR07, PCKR08 '
        + 'FROM D400PC00 WHERE PCNMKS = :PCNMKS';
var
  i: Integer;
begin
  //コンボボックスが未選択の場合
  if cmbArea.ItemIndex = -1 then
    //画像を非表示にする
    imgWeather.Visible := False
  else
    begin
      // SQLを実行し、グラフ（チャート）を描画する
      with UserSession.qry do
        begin
          Active := False;
          // SQL文指定
          SQL.Text := cSQL;
```

```
        // パラメータ指定（コンボボックスの値をセット）
        ParamByName('PCNMKS').AsString
          := cmbArea.Items[cmbArea.ItemIndex];
        // データセットをオープンする
        Active := True;
        try
          // チャート初期化
          chtWeather.Series[0].Clear;
          // チャート描画
          for i := 0 to Fields.Count - 1 do
            chtWeather.Series[0].Add(Fields[i].AsFloat,
              FormatFloat('0000年', i + 2001));
          finally
            // データセットを閉じる
            Active := False;
          end;
        end;
        // 画像の表示
        imgWeather.Visible := True;
      end;
    end;
```

データにもとづき  
チャートを描画

TChartに紐づく画像コン  
ポーネントを有効にする

## ② ブラウザへのグラフ表示

### ■ サンプルプログラムの実行

抽出条件を指定

指定条件に合致するデータより  
グラフを表示

年	値
2001年	1,740
2002年	1,221
2003年	1,635
2004年	744
2005年	1,379
2006年	1,282
2007年	1,338
2008年	1,605

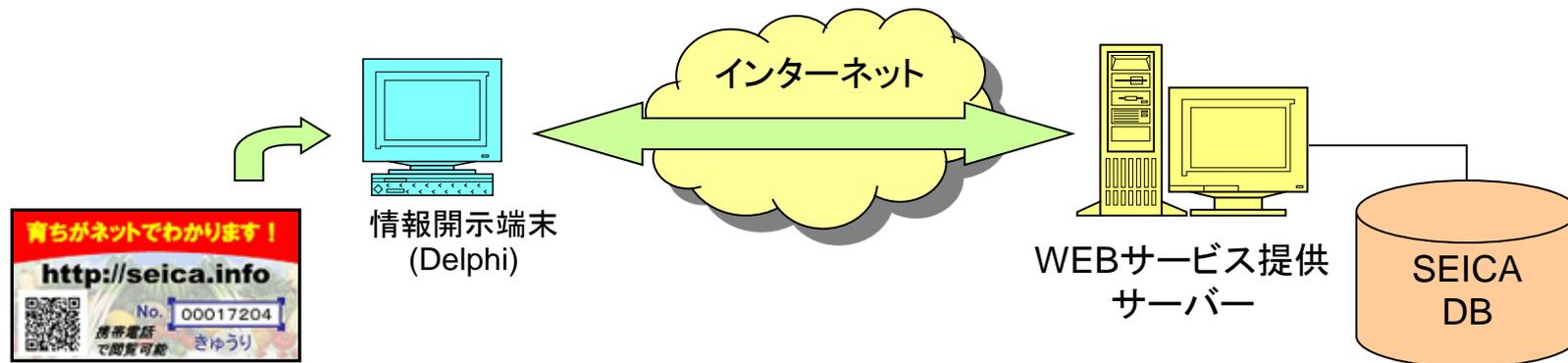
## GUIアプリケーション 開発手法公開 ③ WEBサービスの使用

## ③ WEBサービスの使用

### ■ 案件概要（情報開示システム）

- WEBサービスを使用した情報開示端末を開発したい
  - 野菜の包装に貼り付けたシールに記載されたカタログNoをタッチパネル入力し、生産者情報を照会するシステムを開発
  - 生産者情報取得に青果ネットカタログ（農産物データベース）が提供している**WEBサービスを使用**し、リアルタイムな情報取得を実現

参考：青果ネットカタログ <http://seica.info>



## ③ WEBサービスの使用

### ■ 開発事例紹介

カタログNoをタッチして検索すると、SEICAより情報を取得し、生産者情報を画面に出力

SEICA情報開示端末へようこそ！  
 カタログNoを入力して、『検索』をタッチしてください。

カタログNo.  
**65535**

7 8 9  
 4 5 6  
 1 2 3  
 0 ← CLR

**検索**

SEICA  
 http://seica.info

(財)食品流通構造改善促進機構が所有し、(独)農業・食品産業技術総合研究機構・食品総合研究所及び農林水産研究計算センター（農水省）の協力により運用している公的データベース「SEICA」のデータを活用しています。

ver 1.0.0.1

カタログNo	商品名	原産地	栽培区分
00065535	人参	茨城県	化学肥料不使用栽培





殺虫剤	0回
防除 除草剤	0回
殺菌剤	0回

認証  
 茨城県エコファーマー

生産者名  
 ひとみ人参組合

使用資材

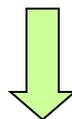
コメント  
 芯まで赤く、生で食べるとまるで硬めの柿をかじっているようなほんのりした甘さと柔らかな食感が広がります。また人参臭さが少ないのもこの人参の大きな特徴です。小さなお子様や人参苦手という人まで手にしてもらいことができると思います。一度食べたら忘れられない味になること間違いなしです。

カタログ検索を開始します。 ⇒ **カタログ検索**

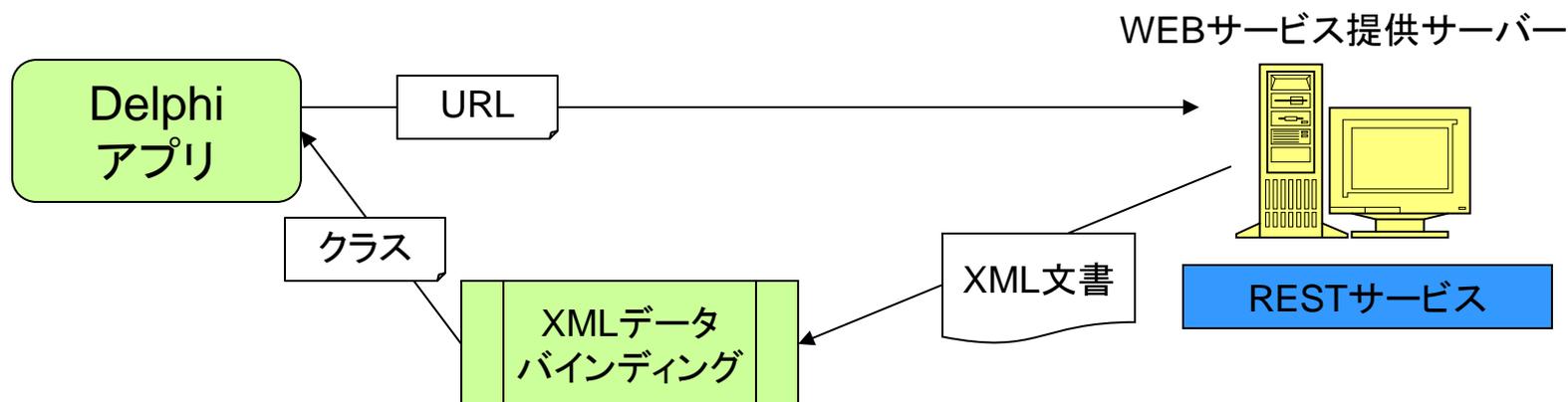
## ③ WEBサービスの使用

### ■ 開発手法

- SEICAのようなWEBサービスをDelphiから容易に扱えないか？



- REST型WEBサービス及びXMLデータバインディングを使用した効率的な開発手法を採用



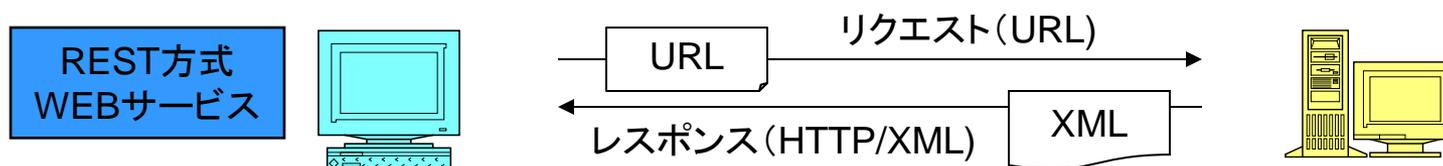
## ③ WEBサービスの使用

### ■ WEBサービスについて

- インターネットの技術を活用し、遠隔サイトにあるアプリケーションの機能をネットワークを通じて利用できるようにしたものをWEBサービスという
  - データのやり取りにXMLを使用する
  - WEBサービスのアクセス手段にはSOAPとRESTが存在する

### ■ SOAPとREST

- クライアントとWEBサービスサーバーとの間をSOAPメッセージというXMLでやり取りする方式を**SOAP**という
  - WEBサービスを使用するためのルールが記述されたXML(WSDL)を使用し、通信を行なう
- URLを指定しWEBサービスサーバーにアクセスすると結果がXML形式で返却される方式を**REST**という
  - WEBサービスへのアクセスが、URLのみで仕組みがシンプルである



## ③ WEBサービスの使用

### ■ RESTを提供するWEBサービス

- 当初WEBサービスといえば、SOAPが主流であったが、近年容易に扱えるRESTの利用が多くなっている
- Yahoo!やGoogle、アマゾンといったサイトがREST式のWEBサービスを提供している

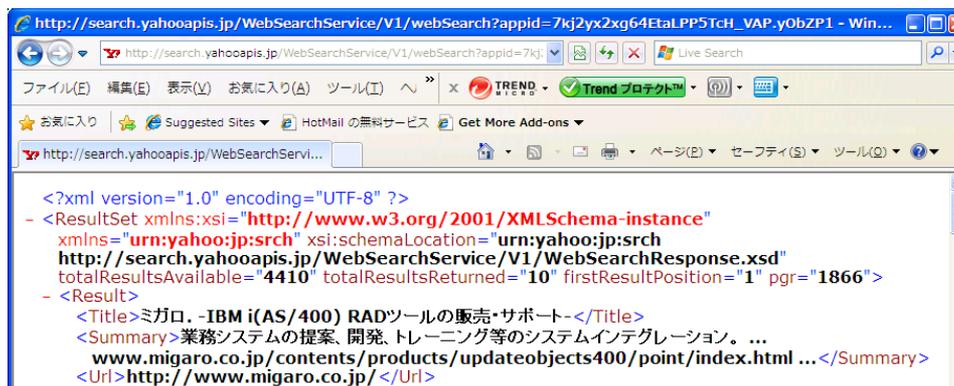
例) Yahoo! 検索 WEBサービス

キーワード “migaro” で問い合わせを実施

【リクエストURL】

<http://search.yahooapis.jp/WebSearchService/V1/webSearch?appid=<アプリケーションID>&query=migaro>

【レスポンスXML】



```
<?xml version="1.0" encoding="UTF-8" ?>
- <ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:yahoo:jp:srch" xsi:schemaLocation="urn:yahoo:jp:srch
  http://search.yahooapis.jp/WebSearchService/V1/WebSearchResponse.xsd"
  totalResultsAvailable="4410" totalResultsReturned="10" firstResultPosition="1" pgr="1866">
- <Result>
  <Title>ミガロ、-IBM i(AS/400) RADツールの販売・サポート-</Title>
  <Summary>業務システムの提案、開発、トレーニング等のシステムインテグレーション。...
  www.migaro.co.jp/contents/products/updateobjects400/point/index.html ...</Summary>
  <Url>http://www.migaro.co.jp/</Url>
```

参考: Yahoo!デベロッパーネットワーク  
<http://developer.yahoo.co.jp/>

## ③ WEBサービスの使用

### ■ サンプルプログラムの作成

- Yahoo!ニュースを提供するREST型WEBサービスにアクセスし情報を取得する
  - HTTPクライアントを実装する**TIdHTTPコンポーネント**を使用する。
  - URLを指定して、結果のリソースを取得するには、**Getメソッド**を使用する。

```
procedure TForm1.btnGetClick(Sender: TObject);
const
  cCMD = 'http://news.yahooapis.jp/NewsWebService/V1/Topics'
        + '?appid=<アプリケーションID>';
var
  sData: String;
begin
  //URLを指定して実行結果を取得
  sData := TIdHTTP1.Get(cCMD);
  //文字コードを変換 (UTF8->Ansi)
  sData := Utf8ToAnsi(sData);
  //取得結果をメモコンポーネントに表示
  Memo1.Lines.Text := sData;
end;
```

※アプリケーションIDは、Yahoo!デベロッパネットワークより取得する

Yahoo!ニュースリクエストURL

結果リソース (XML)取得

## ③ WEBサービスの使用

### ■ サンプルプログラムの実行

Form1

ニュース取得

結果セットが<ResultSet>タグに出力  
totalResultsReturned=取得件数

最新のYahoo!ニュース情報がXML文書として出力

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultSet xmlns="urn:yahoo:jp:news"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:yahoo:jp:news
http://api.news.yahoo.co.jp/NewsWebService/V1/NewsService.xsd"
totalResultsReturned="30"
>
  <Result>
    <datetime>2009-06-07T14:33:09</datetime>
    <title>記録途切れたイチローが3安打</title>
      <topicname>イチロー</topicname>
    <category>スポーツ</category>
    <subcategory>メジャーリーグ</subcategory>

    <url>http://dailynews.yahoo.co.jp/fc/sports/ichiro</url>
    <topflg>1</topflg>
  </Result>
</ResultSet>
```

1件のニュースが<Result>タグに出力  
<datetime>: 更新日時  
<title>: トピックのタイトル  
<category>: カテゴリ  
<url>: 関連するURL

## ③ WEBサービスの使用

### ■ XMLについて

- XML文書はHTMLと同じマークアップ言語と呼ばれ、タグにより情報の意味や構造を定義したテキスト文書である
  - HTMLと異なり、XMLはタグ自体の意味を自由に定義できるのが特徴である
- XML文書が使用するタグや構造を定義したものを**XMLスキーマ**という
  - XMLスキーマ(.xsd)自体もXMLにて記述されている

参考: Yahoo!ニュースWEBサービス

<http://developer.yahoo.co.jp/webapi/news/news/v1/topics.html>

★ お気に入り Yahoo!デベロッパーネットワーク - ニュース - ...

### レスポンスフィールド

トピックスWeb APIのスキーマの記述は、以下のURLをご覧ください。

<http://news.yahooapis.jp/NewsWebService/V1/Topics.xsd>

フィールド	説明
ResultSet	クエリーレスポンスのすべてを含みます。
Result	各個別レスポンスを含みます。
datetime	最後にトピックを更新した日時です。
title	Yahoo! JAPANトップページに表示されているような15文字程度の見出し(テキスト)です。

### ■ DelphiによるXML文書の取り扱い

- XMLスキーマを取込みDelphiコードで読み書き可能なクラスを作成することが可能である
- **XMLデータバインディング**を使用すれば、クラスを自動生成できる

## ③ WEBサービスの使用

### ■ サンプルプログラムの改良

- XMLをDelphiコードで利用可能にする為に、XMLデータバインディングを使用する

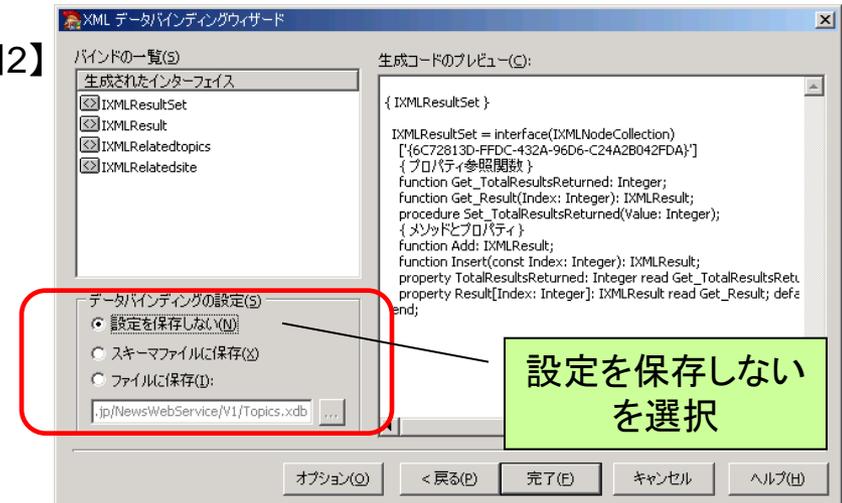
- 【設定手順】**
1. 新規作成⇒その他を選択
  2. XMLフォルダのXMLデータバインディングを選択
  3. ソースコード欄にXSDファイルを指定(図1)
  4. ウィザードを進み、データバインディングの設定を保存しないに設定し(図2)、完了

【図1】



Yahoo!ニュース XMLスキーマ  
http://news.yahooapis.jp/  
NewsWebService/V1/Topics.xsd

【図2】

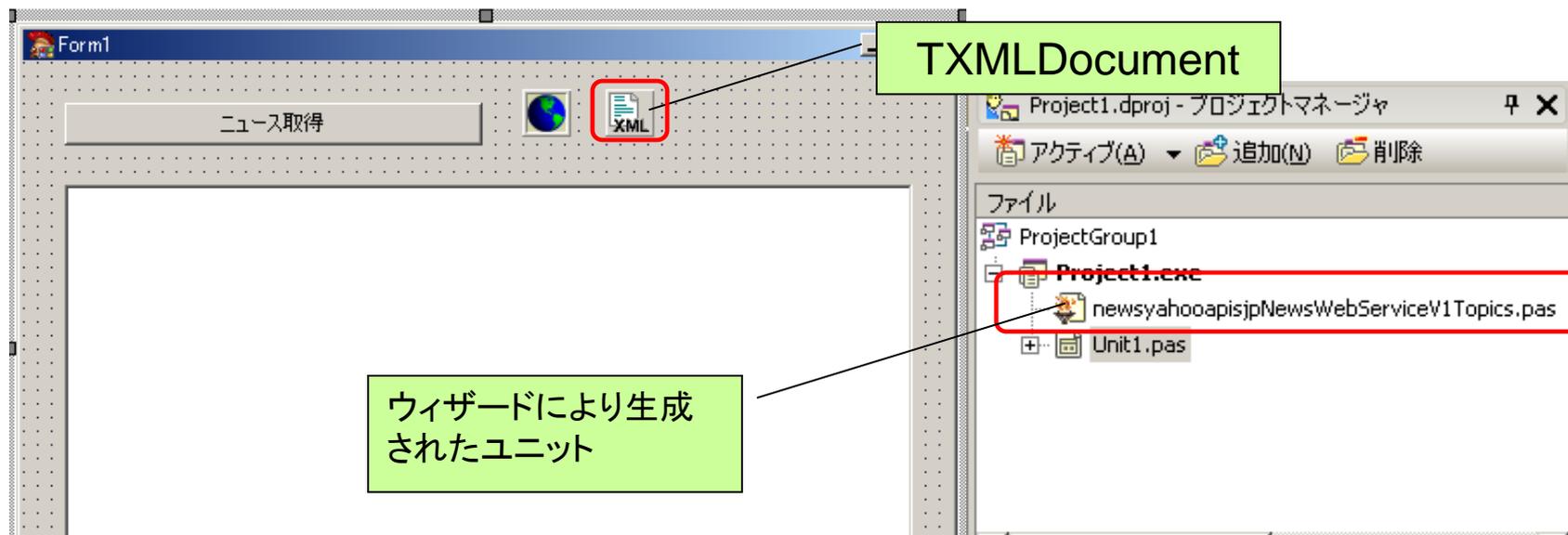


設定を保存しない  
を選択

## ③ WEBサービスの使用

### ■ サンプルプログラムの作成

- RESTにて取得したXML文書をデータバインディングにより作成したクラスを使用しDelphiソースで情報を取得する
  - XML文書をDelphiで取り扱うには、**TXMLDocument**コンポーネントを使用する
  - XMLデータバインディングで生成されたnewsyahoopisjpNewsWebServiceV1TopicsユニットにあるGetResultSet関数にXML文書をセットすると情報にアクセス可能になる



## ③ WEBサービスの使用

### ■ サンプルプログラムソース

XMLデータバインディングにて作成されたユニットを追加

```
uses newsyahooapis.jpNewsWebServiceV1Topics;
```

```
procedure TForm1.btnGetClick(Sender: TObject);  
const
```

```
  cCMD = 'http://news.yahooapis.jp/NewsWebService/'  
        + 'V1/Topics?appid=<アプリケーションID>';
```

```
var
```

```
  i: Integer;  
  sData: String;
```

```
  AXMLResultSet: IXMLResultSet; //結果セット  
  AXMLResult: IXMLResult; //1件の結果
```

```
begin
```

```
  //メモコンポーネントをクリアする  
  Memo1.Clear;
```

```
  //URLを指定して実行結果を取得  
  sData := IdHTTP1.Get(cCMD);
```

XMLDocumentにロードし、オブジェクトとして使用可能にする

```
//取得結果をXMLコンポーネントにセット  
XMLDocument1.LoadFromXML(sData);
```

```
//XML結果セットの取得
```

```
AXMLResultSet := GetResultSet(XMLDocument1);
```

XMLをセットし、結果セットとして操作できるようにする

```
//結果セットより取得した件数分情報を取得する  
for i := 0 to
```

```
  AXMLResultSet.TotalResultsReturned - 1 do  
begin
```

```
  //1件の情報を取得する
```

```
  AXMLResult := AXMLResultSet.Result[i];
```

```
  //メモコンポーネントに件名をセットする。
```

```
  Memo1.Lines.Add(AXMLResult.Title);
```

```
end;
```

```
end;
```

## ③ WEBサービスの使用

### ■ サンプルプログラムの実行

取得したXML文書を解析し、取得したニュース件数分のタイトルをリスト出力

ニュース取得

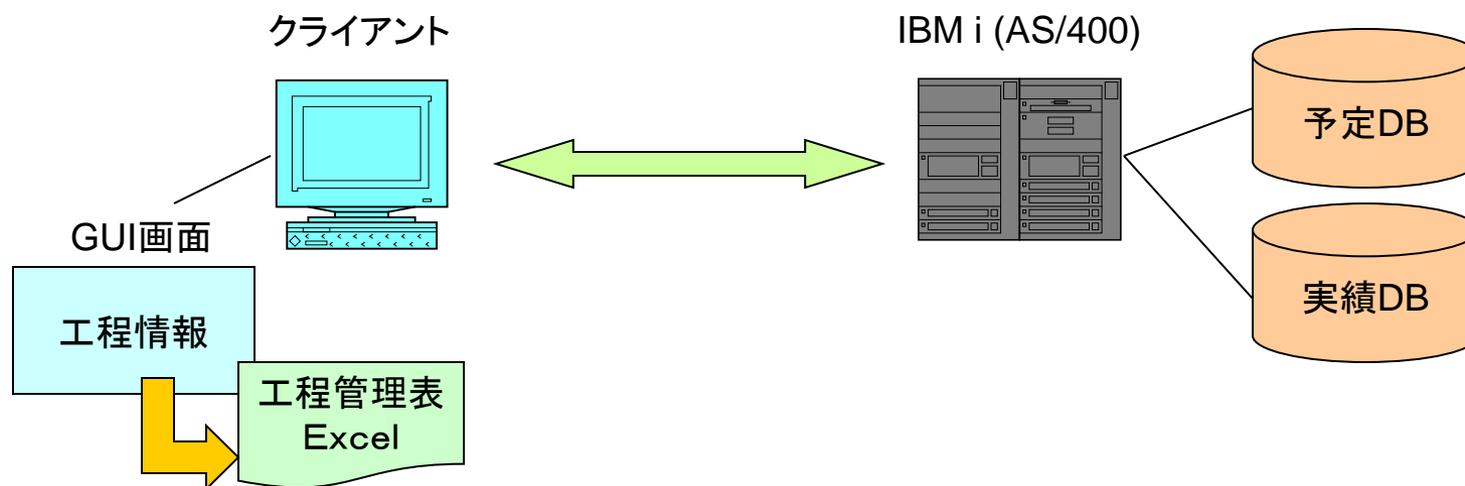
オープン価格で値が上がる理由  
行方市「爆弾女ブログ」が人気  
石橋の長女・穂のか TV生出演  
空気悪くない 橋下知事が反論  
印象的な告白1位は「101回目」  
直轄負担金 不適切なら拒否も  
文化祭準備の高1はねられ重体  
ウオッカが安田記念を連覇  
キムタク映画は「悪ガキ3人」  
土井宇宙飛行士が国連に転身  
記録途切れたイチローが3安打  
女子高生を刺して少年自殺か  
3戦連発 岡崎ヘッドでW杯決定

## GUIアプリケーション 開発手法公開 ④ ガントチャートの実装

## ④ ガントチャートの実装

### ■ 案件概要（工程管理システム）

- 工程ごとの作業進捗や負荷状況を一目で把握・管理したい
  - 工程ごとの納期や作業者の負荷状況を把握しながら、工程計画を作成する工程管理システムを開発
  - 工程の状況を一目で把握する為に**ガントチャート**を使用した画面ならびに帳票を実現



## ④ ガントチャートの実装

### ■ 開発事例紹介

プロジェクト、担当者ごとの作業工程状況がチャートとして出力

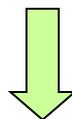
工程負荷状況一覧表として、Excelに出力

枝	品名	ME	担当者	区分	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
00	プロジェクト統括	M	藤井	予定							構想・仕様	提出	先行手配										
				実績																			
01	主材輸送配管材	M	松元	予定							構想・仕様	提出	先行手配										
				実績																			
02	空気側配管材	M	村谷	予定	先行手配	納入仕様書	顧客承認	確定図															
				実績																			

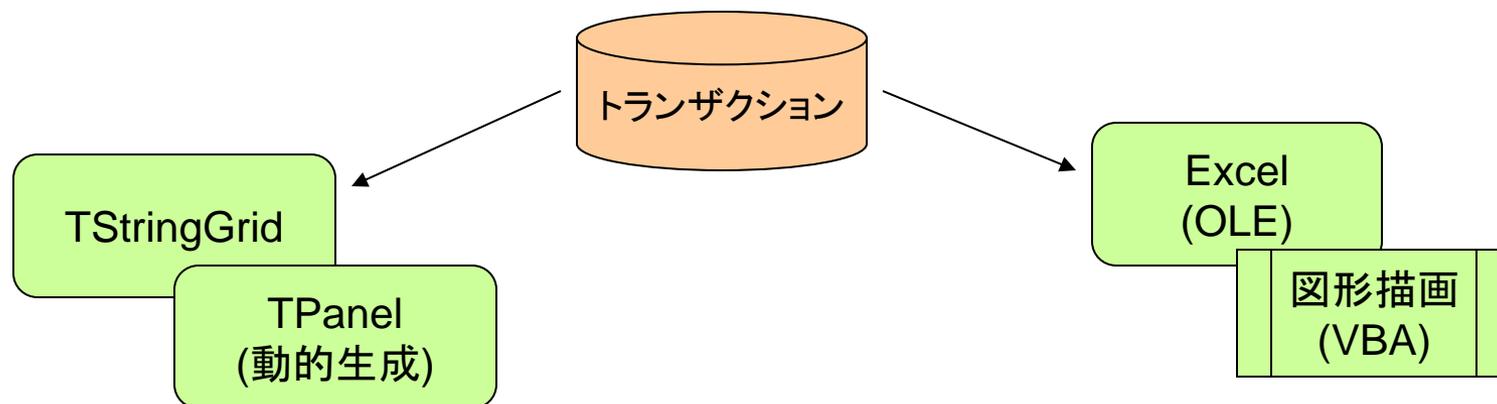
## ④ ガントチャートの実装

### ■ 開発手法

- 工程状況をチャート形式で照会したり印刷したりできないか？



- StringGridとPanelとの組み合わせによりチャートを実現。トランザクションデータから工程データを作成し、動的にPanelを生成することで工程表を作成
- Excelのマクロ(VBA)をDelphiから操作することによりExcel出力を実現



## ④ ガントチャートの実装

### ■ チャート実装サンプルプログラムの作成

- チャート出力位置、サイズを指定する入力欄とチャートを出力するTStringGridを画面に配置したフォームを作成する
  - StringGridのセル幅、セル高さ、行数、列数を設定する

edtX: TMaskEdit

edtY: TMaskEdit

edtLength: TMaskEdit

btnSetPanel: TBitBtn

StringGrid1: TStringGrid

- DefaultColWidth(幅) = 70
- DefaultRowHeight(高さ) = 40
- ColCount(列数) = 13
- RowCount(行数) = 10

## ④ ガントチャートの実装

### ■ チャート実装サンプルプログラムソース

```
private
  [ Private 宣言 ]
  FPanelCount: Integer;

procedure TfrmSample.BtnSetPanelClick(Sender: TObject);
var
  Panel: TPanel;
begin
  //パネル表示カウンターをインクリメントする
  Inc(FPanelCount);

  //パネル表示処理 (パネルを動的にCreateする)
  Panel := TPanel.Create(Application);
  with Panel do
  begin
    //パネル表示位置指定
    Left := (StringGrid1.DefaultColWidth + 1)
      * StrToInt(Trim(edtX.Text));
    Top := (StringGrid1.DefaultRowHeight + 1)
      * StrToInt(Trim(edtY.Text)) + 3;
    Height := StringGrid1.DefaultRowHeight - 6;
```

パネル数をカウントするグローバル変数

StringGridのセルサイズと画面指定した位置、サイズによりパネルの出力位置、幅、高さを算出

```
Width := (StringGrid1.DefaultColWidth + 1)
  * StrToInt(Trim(edtLength.Text));
```

```
//表示文字列指定
Caption := 'サンプル' + IntToStr(FPanelCount);
//コンポーネント名指定
```

```
Name := 'Panel' + IntToStr(FPanelCount);
```

```
//親コンポーネントの指定
```

```
Parent := StringGrid1;
```

```
//パネルの色指定
```

```
Color := clAqua;
```

```
//パネルの再描画
```

```
Invalidate;
```

```
end;
```

```
end;
```

PanelのNameにPanel1,Panel2...と名付ける

## ④ ガントチャートの実装

### ■ チャート実装サンプルプログラムの実行

The image shows two screenshots of a software application titled "チャート作成" (Chart Creation). The top screenshot shows the input fields: 横軸 (0-12) [03], 縦軸 (0-9) [2], 長さ (1-13) [05], and a button "パネルを表示". A green callout bubble points to these fields with the text: "チャートを描画する横軸、縦軸、長さを指定してボタンをクリック". A large yellow arrow points from this screenshot to the bottom screenshot. The bottom screenshot shows the same application with three cyan bars drawn on a grid. The bars are labeled "サンプル1", "サンプル2", and "サンプル3". A green callout bubble points to the bars with the text: "指定した位置と長さにもとづくチャートが描画". The bottom screenshot also has a "閉じる" (Close) button.

## ④ ガントチャートの実装

### ■ Excel出力サンプルプログラムの作成

- DelphiよりOLEオートメーションにてExcelを操作し、チャートを作成する
  - テンプレートとなるExcelを用意し、予めチャートを図形として描画するマクロ(VBA)を埋め込んでおく
  - Delphiよりパラメータを指定してマクロを実行することで、チャートを描画する

The image shows a Delphi application window titled 'frmSample' with a button labeled 'Excel 出力' (Excel Output). A yellow arrow points from this button to a green box containing the text 'OLEでマクロ実行 Excel.Application.Run'. Another yellow arrow points from this box to a Microsoft Excel window titled 'Microsoft Excel - Sample.xls', which displays a spreadsheet with columns A, B, and C, and rows 1 through 7. A third yellow arrow points from the Excel window to a Microsoft Visual Basic code editor window titled 'Microsoft Visual Basic - Sample.xls - [Module1 (コード)]'. The code editor shows the following VBA code:

```
(General) | ExcelOutChart
'列、行、期間、配色、コメント、文字色の設定
Sub ExcelOutChart (FromD As Variant, FRow As Variant, ToD As Variant, Iro As Variant, Comment As Variant)
Const 上余白率 = 20 '余白の割合(%)
Const 下余白率 = 20 '余白の割合(%)
Const 右余白率 = 0 '余白の割合(%)
Const 左余白率 = 0 '余白の割合(%)

Dim 座標1 As Double, 座標2 As Double, 座標3 As Double, 座標4 As Double
Dim FCol As Integer, TCol As Integer, TRow As Integer, TRow As Integer

'変数の初期化
FCol = FromD
TCol = ToD
TRow = FRow

'位置の初期設定
Range("A1").Select
'開始位置の指定
ActiveCell(TRow, FCol).Select
```

A red box highlights the subprocedure signature in the code editor. A green box on the right contains the following text:

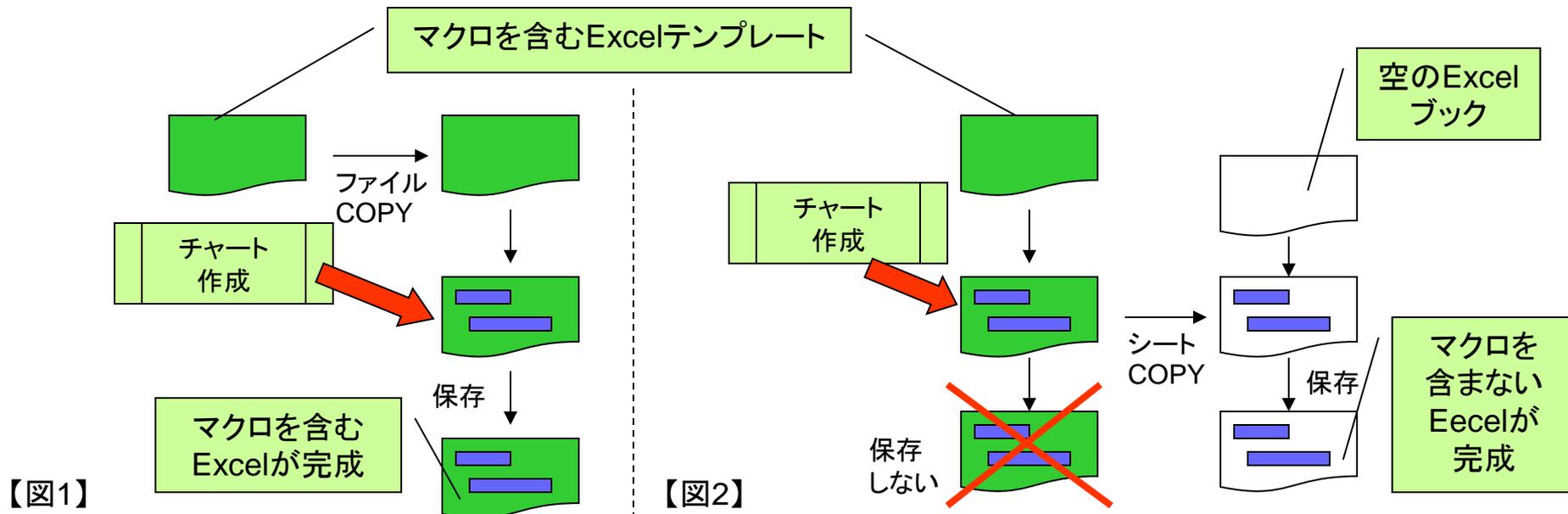
**【図形を描画するVBA】**  
メソッド名: ExcelOutChart

引数にて指定したセルの開始位置、終了位置にもとづき、図形(チャート)を描画

## ④ ガントチャートの実装

### ■ Excel出力サンプルプログラムの作成

- マクロを含むExcelテンプレートに直接データ作成後、新しいブックに作成したシートをコピーして保存し、テンプレートのExcelは変更を保存せずに終了する
  - マクロを含むExcelをテンプレートとする場合、ファイルコピーしたExcelを使って作成するとマクロを含むExcelが完成し、Excelをユーザーが開く際にマクロ警告が発生してしまう(図1)
  - 新しいブックに作成したシートのみをコピーすると、マクロはコピーされない為、警告が発生しなくなる(図2)



【図1】

【図2】

## ④ ガントチャートの実装

### ■ Excel出力サンプルプログラムソース

```
procedure TfrmSample.btnExcelOutClick(Sender: TObject);
const
  cFormatFNM = 'Sample.xls';
var
  i: Integer;
  sSourceFile, sCopyFile: String;
  Excel, WorkBook1, WorkBook2, WorkSheet1
    , WorkSheet2: Variant;
begin
  //コピー元ファイルの指定
  sSourceFile := ExtractFilePath(Application.ExeName)
    + cFormatFNM;
  //ファイル名の決定
  sCopyFile := ExtractFilePath(Application.ExeName) +
    FormatDateTime('YYYYMMDDHHMMSS', Now) + '.xls';
  //エクセルの起動
  Excel := CreateOleObject('Excel.Application');

  //コピー元のExcelをオープンする
  WorkBook1 := Excel.WorkBooks.Open(FileName :=
    sSourceFile, ReadOnly := False);
```

OLEでExcel起動

```
//表示の有無
Excel.Visible := False;
//アラートを非表示にする
Excel.Application.DisplayAlerts := False;
// Sheet1 を選択
WorkSheet1 := WorkBook1.Worksheets['Sheet1'];
// シートをアクティブにする
WorkSheet1.Activate;
//データの書き出し(マクロの実行)
Excel.Application.Run('ExcelOutChart',
  cLeft,
  cTop,
  cKikan,
  55, //色指定(灰色)
  'サンプル');
```

ExcelOutChart  
プロシーチャーを実行

<次ページに続く>

## ④ ガントチャートの実装

### ■ Excel出力サンプルプログラムソース

//コピー先のファイルを開く

```
WorkBook2 := Excel.WorkBooks.Add;
```

新しいブックを作成

//Sheetを1つだけ残して他を削除

```
for i := 0 to WorkBook2.Sheets.Count - 2 do  
begin  
  WorkSheet2 := WorkBook2.WorkSheets[1];  
  WorkSheet2.Delete;  
end;
```

新しいブックの最終シート  
の後ろに、テンプレートの  
シートをコピー

//コピーの実行

```
WorkSheet2 := WorkBook2.WorkSheets[Workbook2  
.Sheets.Count];  
WorkBook1.WorkSheets[1].Copy(, WorkSheet2);
```

//残っているSheetを削除

```
WorkSheet2 := WorkBook2.WorkSheets[1];  
WorkSheet2.Delete;
```

コピー先のブックを保存し、  
テンプレートのブックは  
保存せずに終了

//コピー先ファイルの保存

```
WorkBook2.SaveAS(sCopyFile);
```

//コピー元Excelを保存せずに終了する

```
WorkBook1.Close;
```

//エクセルを表示する

```
Excel.Application.DisplayAlerts := True;  
Excel.Visible := True;
```

```
end;
```

## ④ ガントチャートの実装

### ■ Excel出力サンプルプログラムの実行

Excel出力

Microsoft Excel - 20090602141814.xls

	A	B	C	D	E	F	G	H
1								
2								
3		サンプル						
7								

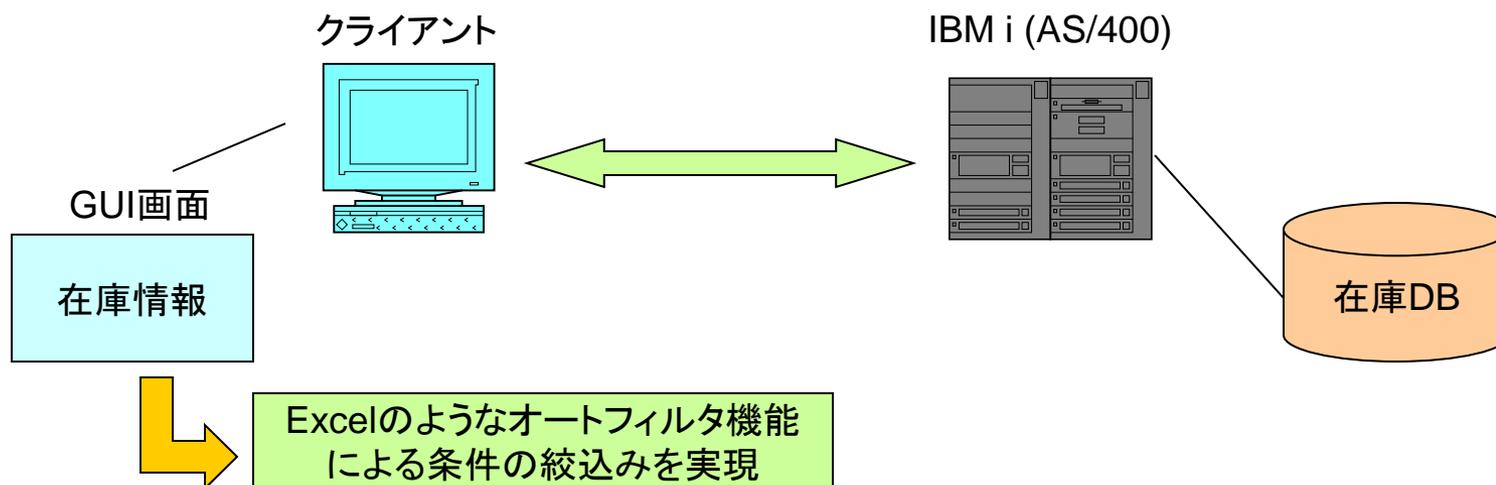
チャート(図形)が描画されたマクロを含まないExcelが開く

## GUIアプリケーション 開発手法公開 ⑤ オートフィルタ機能の拡張

## ⑤ オートフィルタ機能の拡張

### ■ 案件概要（在庫照会システム）

- 多数の品目在庫を効率よく照会したい
  - 在庫状況をユーザーが自由に条件指定して照会できる在庫照会システムを開発
  - 自由な条件指定を行う為、照会画面のグリッドに**オートフィルタ機能**を実現



## ⑤ オートフィルタ機能の拡張

### ■ 開発事例紹介

タイトルクリックにて、項目値リストが表示  
値を選択することで絞込みが可能

	規格	サイズ	長さ	在庫残数量	在庫残重量	品番	チャージNo.
<input checked="" type="checkbox"/>	A	13	50.000	70	500	SNO001A	CNO1001
<input type="checkbox"/>	A	13	100.000	100	1000	SNO001A	CNO1000
<input checked="" type="checkbox"/>	A	13	150.000	150	1500	SNO001A	CNO1000
<input checked="" type="checkbox"/>	B	13	50.000	500	500	SNO001A	CNO1001
<input checked="" type="checkbox"/>	B	13	100.000	1000	1000	SNO001A	CNO1000
<input checked="" type="checkbox"/>	B	13	150.000	1500	1500	SNO001A	CNO1000
<input checked="" type="checkbox"/>	C	15	50.000	350	350	SNO001A	CNO1001
<input checked="" type="checkbox"/>	C	15	100.000	700	700	SNO001A	CNO1000
<input checked="" type="checkbox"/>	C	15	150.000	1050	1050	SNO001A	CNO1000

フィルタオプション

抽出条件の指定:

在庫残数量

70

AND  OR

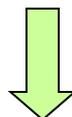
と等しい  
と等しくない  
より大きい  
以上  
より小さい  
以下

式の指定ならびに複合条件  
指定も可能

## ⑤ オートフィルタ機能の拡張

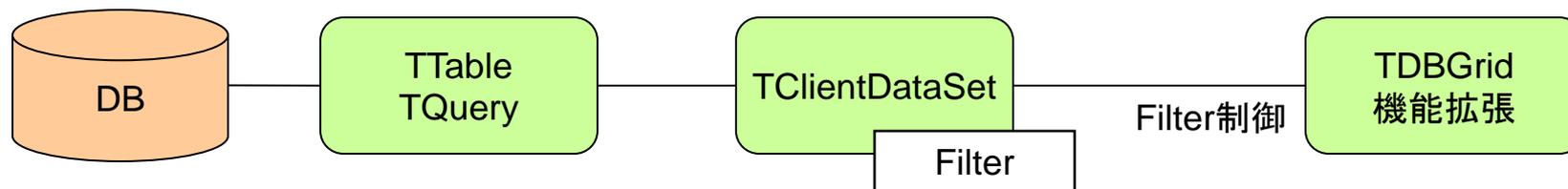
### ■ 開発手法

- DBGridで表示している一覧照会画面に汎用的なオートフィルタ機能を組み込みたい



- ClitentDataSetのフィルタ機能を使用するDBGrid拡張コンポーネントを作成することで対応

製品No.	製品名	板厚	幅
▶ SNO01	ミガロ・製品01	(すべて)	
SNO02	ミガロ・製品02	10	
SNO03	ミガロ・製品03	20	
SNO04	ミガロ・製品04	30	
SNO05	ミガロ・製品05		10
SNO06	ミガロ・製品06		10



## ⑤ オートフィルタ機能の拡張

### ■ コンポーネント開発

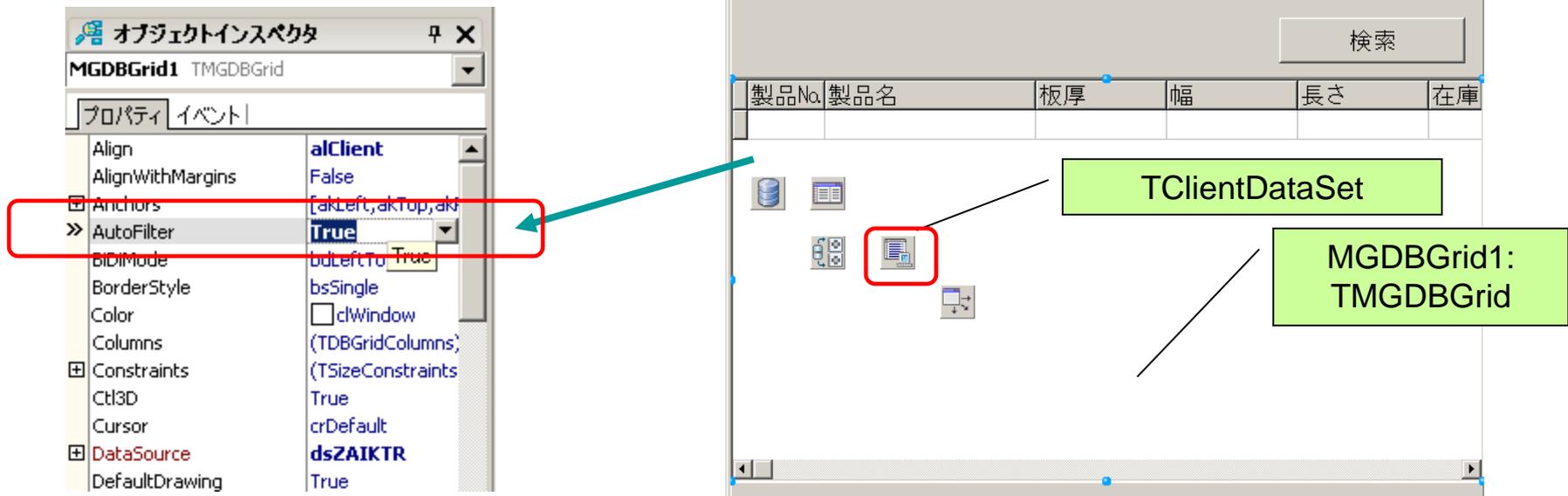
- Delphiは、標準コンポーネントを使用するだけでなく、独自の追加コンポーネント開発が可能である
- コンポーネントの開発には、次の2つが検討できる
  - 新規コンポーネントの設計  
自由度は高いが、全てのプロパティ、イベント、メソッドの設計・開発が必要となる
  - 既存コンポーネントの機能追加  
標準コンポーネントに対し、必要な機能のみ開発すれば、元の機能はそのまま継承される為最低限の工数でコンポーネント追加が可能である
- オートフィルタ機能のような汎用的な処理は、コンポーネント化に向いている
  - オートフィルタ機能は、アプリケーションの仕様に関わらず開発できる為、プログラムと分離しコンポーネント化することが可能である
  - 機能のさらなる拡張(例えば、フィルタオプション機能の追加)は、アプリケーションプログラムの開発とは別に行なうことできる

⇒ オートフィルタ機能サンプルコンポーネントとしてTMGDBGGridを作成

## ⑤ オートフィルタ機能の拡張

### ■ サンプルプログラムの作成

- DataSetに紐づくグリッド表示コンポーネント(TDBGrid)に下記機能を拡張したTMGDBGridコンポーネントを使用する
  - オートフィルタ利用可否を指定するプロパティを追加する(AutoFilterプロパティ)
  - AutoFilter=Trueの場合、実行時に列タイトル欄をクリックすると、フィルタ対象となる値のリストが表示され、リストから値を選択することでFilter機能によるデータの絞込みを可能にする



## ⑤ オートフィルタ機能の拡張

### ■ サンプルプログラムの実行

列タイトルクリックにて項目の一覧が出力

製品No.	製品名	板厚	幅	長さ
▶SNO01	ミガロ.製品01	(すべて)		100
SNO02	ミガロ.製品02	10		100
SNO03	ミガロ.製品03	20		100
SNO04	ミガロ.製品04	30		120
SNO05	ミガロ.製品05	10	10	120
SNO06	ミガロ.製品06	10	150	
SNO07	ミガロ.製品07	10	150	1
SNO08	ミガロ.製品08	20		
SNO09	ミガロ.製品09	20		1
SNO10	ミガロ.製品10	20	100	1
SNO11	ミガロ.製品11	20	120	
SNO12	ミガロ.製品12	20		1

値を選択

指定した条件のFilterが有効となりデータが絞り込まれる

製品No.	製品名	板厚	幅	長さ	在庫数量
▶SNO08	ミガロ.製品08	20	100	5	1,150
SNO09	ミガロ.製品09	20	100	10	1,200
SNO10	ミガロ.製品10	20	100	15	1,100
SNO11	ミガロ.製品11	20	120	5	700
SNO12	ミガロ.製品12	20	120	10	700
SNO13	ミガロ.製品13	20	150	5	500
SNO14	ミガロ.製品14	20	150	10	500

## まとめ

## Webアプリケーション 開発手法公開

### ① JAVAスクリプトとの連動

VCL for the WebへのJAVAスクリプト埋込手順

### ② ブラウザへのグラフ表示

TIWImageの機能拡張によるTChartの画像化

## GUIアプリケーション 開発手法公開

### ③ WEBサービスの使用

RESTを使用したWEBサービスの利用とXMLの扱い

### ④ ガントチャートの実装

TPanelの動的生成ならびにExcelとの連動

### ⑤ オートフィルタ機能の拡張

TDBGridの機能拡張による処理の汎用化