

【セッションNo.1】

直感的に理解できるシステムを目指して ～情報の”見える化”の取り組み～

豊鋼材工業株式会社
製造総括部
石井裕昭

アジェンダ

1. 会社紹介
2. 開発の経緯
 - ・開発のねらい
 - ・System iの利用方法の変遷
3. 新生産管理システム(Y-VIS(MP))の特徴
4. demonstration
5. 開発のポイント紹介
 - ・DBGrid の着色方法
 - ・DBChart の 機能を活用
 - ・WebBrowserコンポーネントの有効利用
6. 今後の計画と課題

Delphi/400

Technical Seminar

1. 会社紹介

創造 ファクトリー
豊鋼材工業株式会社
since 1958

商号

豊鋼材工業株式会社
(本社 福岡県)

設立

昭和33(1958)年6月4日

事業内容

鉄鋼および各種金属の加工および販売

資本金

4億5,000万円

代表者

木村 昭夫

取扱量

36万トン (2007年度連結)

従業員数

273名

Delphi/400

Technical Seminar

1-1. 事業所所在地

本社:

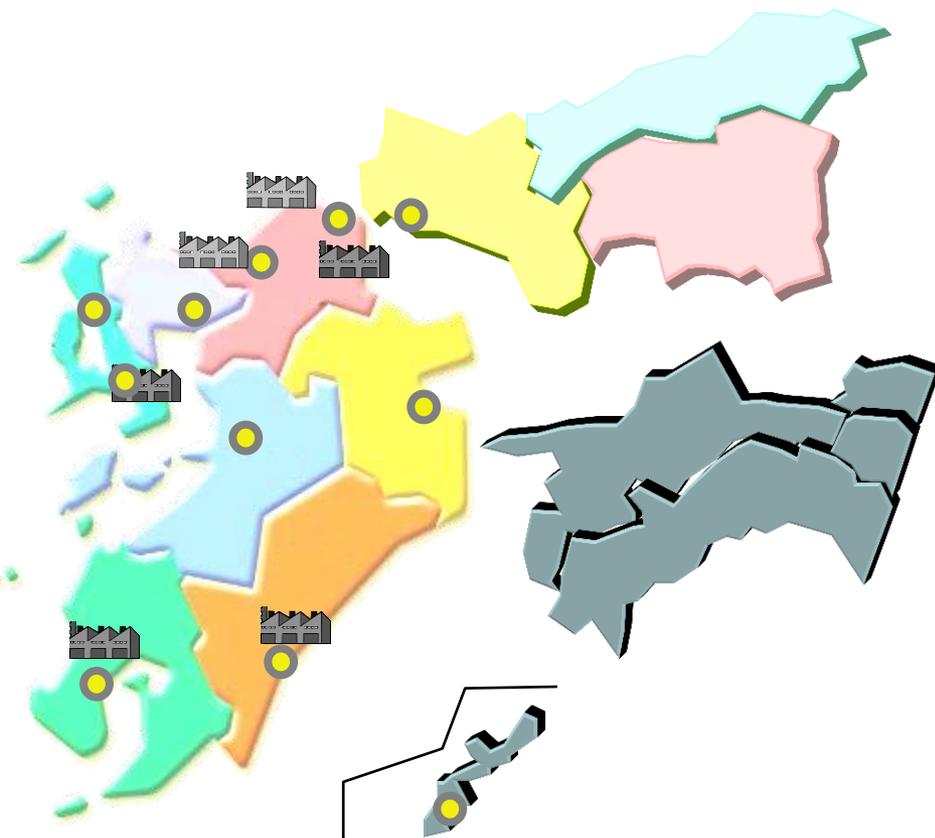
福岡県糟屋郡篠栗町

工場:

福岡、苅田、若松、長崎、
宮崎、鹿児島

支店・営業所:

北九州、長崎、熊本、宮崎、
中国、佐賀、大分、佐世保、
鹿児島、沖縄



Delphi/400

Technical Seminar

1-2. 事業内容

鉄鋼およびその他金属の加工、販売

総販売量

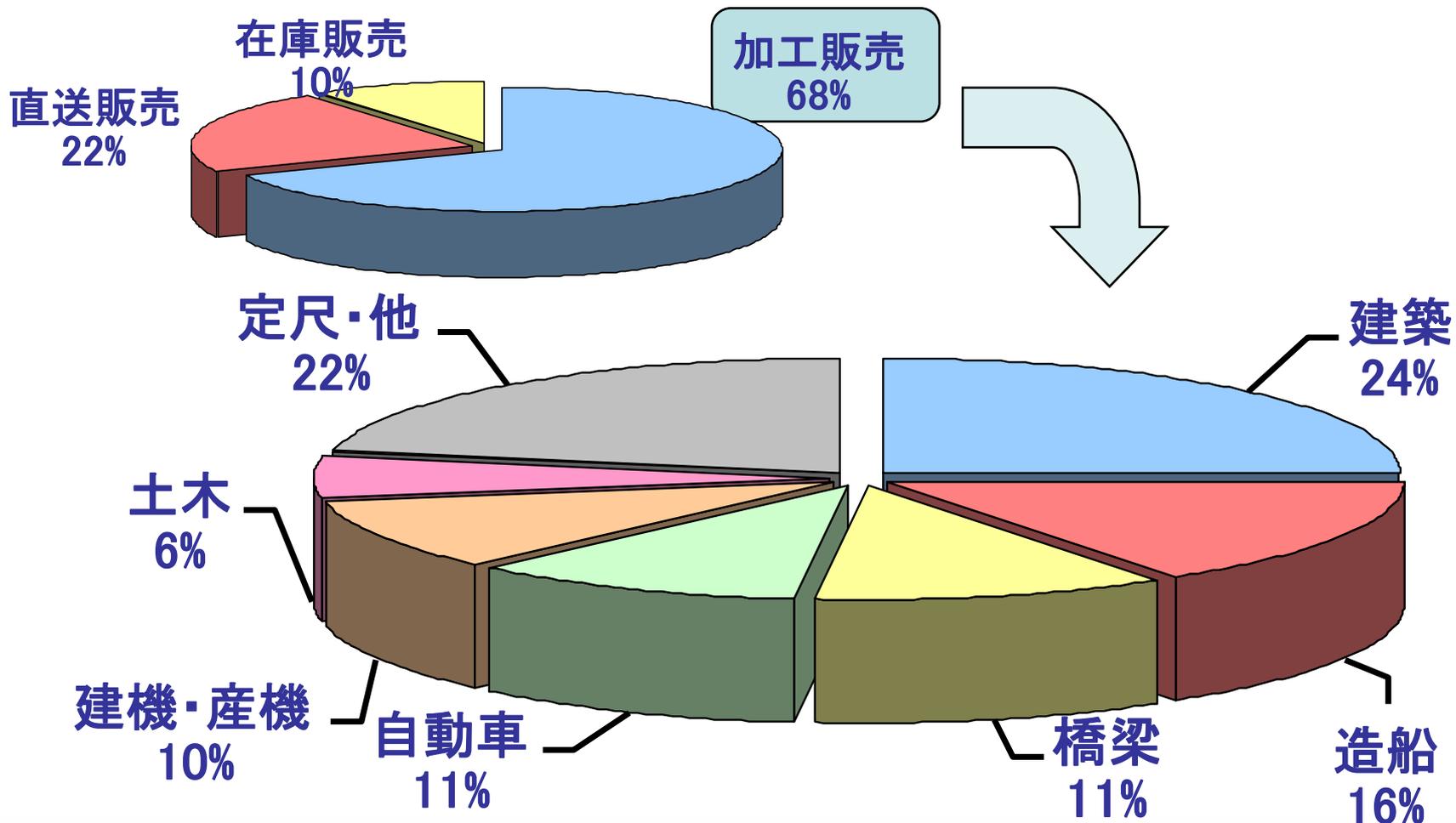
≒ 年間36万トン（連結）



鉄鋼およびその他金属の二次加工、販売



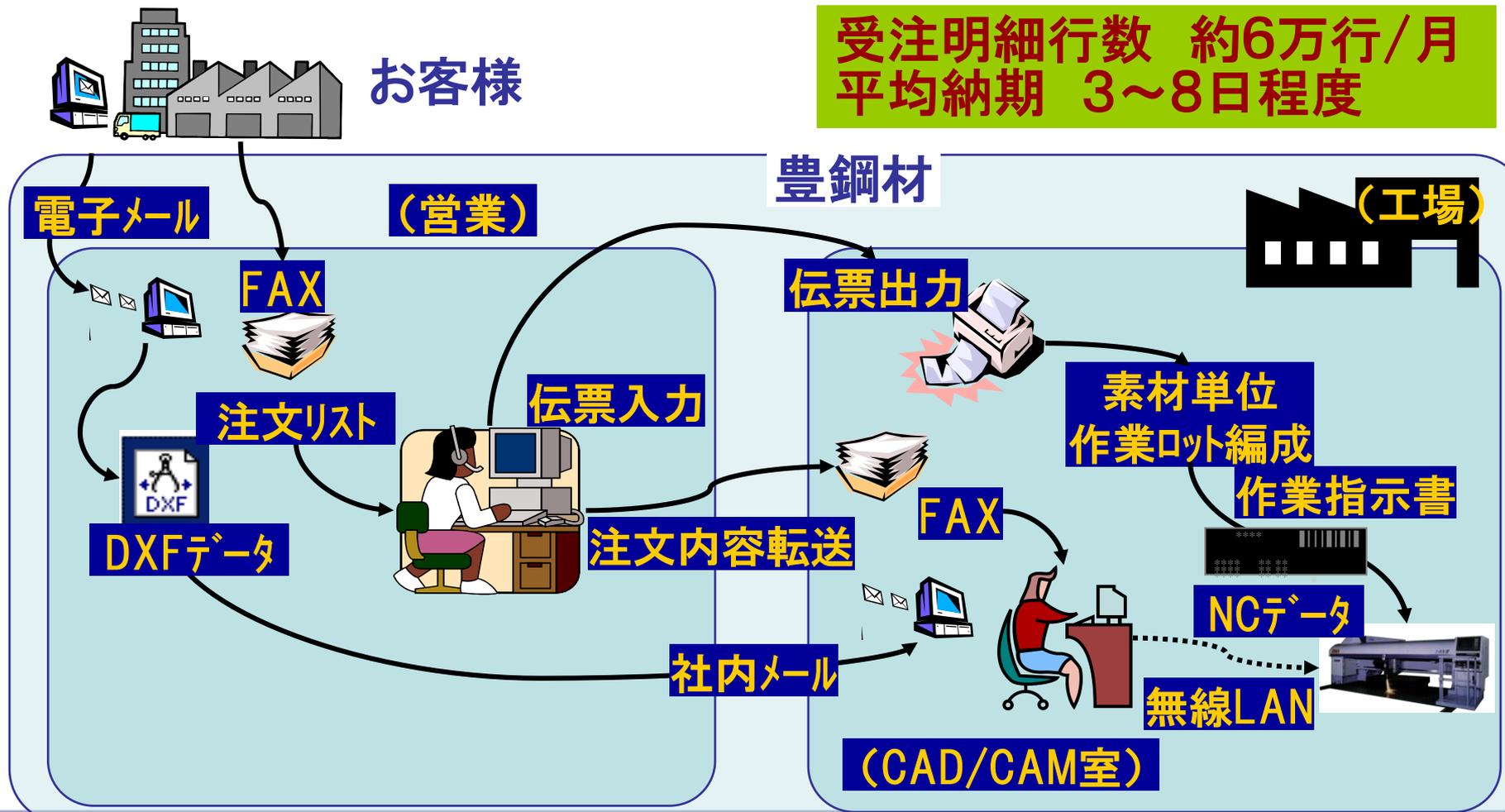
1-3. 用途、販売先



Delphi/400

Technical Seminar

1-4. 業務・システムフロー概要(受注～加工)



2. 開発の経緯

2-1. 開発のねらい

- ・生産情報のリアルタイム収集と共有化
- ・計画系情報の管理可能化
- ・在庫情報更新のリアルタイム化
- ・素材(端材)用ラベル発行の即時化
- ・基幹(ホスト)情報の有効活用 など

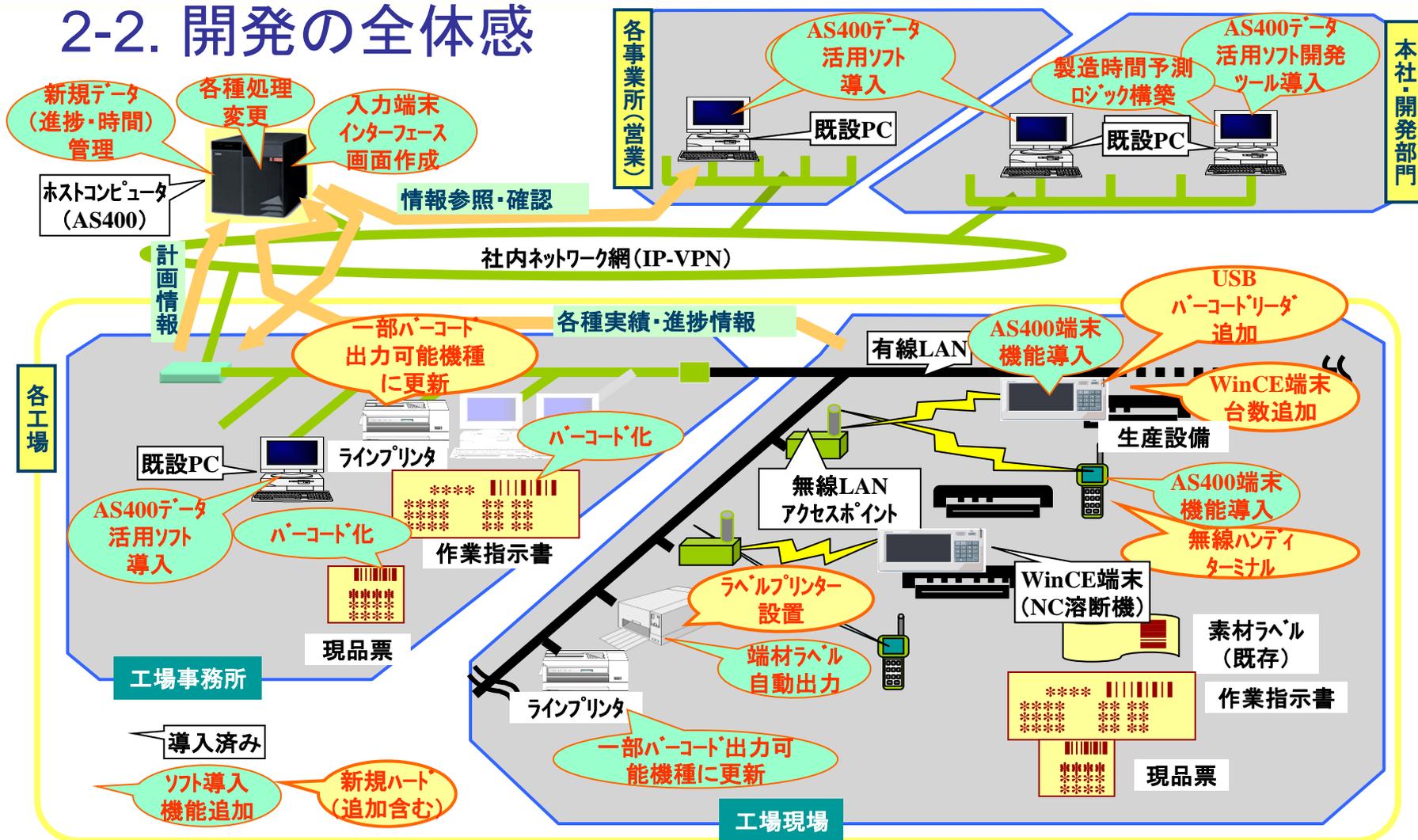


- ・お客様へのサービス向上
- ・戦略的情報活用
- ・業務効率化、正確化
- ・生産性向上

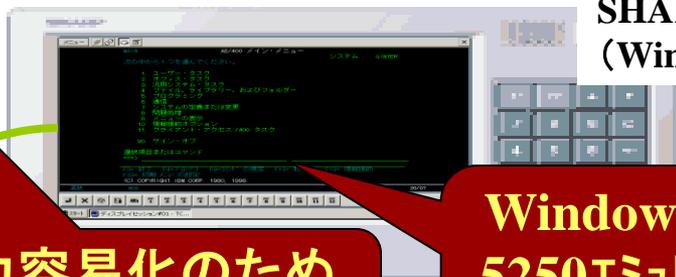
Delphi/400

Technical Seminar

2-2. 開発の全体感



2-3.現場導入ハードのイメージ



SHARP HC-T100
(Windows-CE機)

入力容易化のため
バーコードリーダー追加

Windows-CE端末に
5250エミュレータ(業務画面表示ツール)を導入



機種、用途によっては
無線対応ハンディターミナル使用
・現品票処理、在庫処理等

同様に5250
エミュレータ導入



ガス切断機に搭載した端末

端材用ラベルプリンタ
(ホストより直接印刷
→PC不要)



2-4. System iのデータ利用方法の変遷

・帳票印刷 → 手入力

ホストデータの
有効活用の始まり

利用者は限定的
タスク設定等必要
操作性に難あり

1998頃～

・データ転送 (P-Comm) スポット → CSVファイル

→ MS-ACCESSにリンクし、加工

処理の汎用化
利用者拡大

2002頃～

・データ転送 (P-Comm) 定時 (ACCESS-VBAで起動)

→ CSVファイル → MS-ACCESSにリンクし、加工

リアルタイム性
転送トラフィックの影響
ACCESSの処理時間

2006頃～

・Delphi/400のアプリケーションでリアルタイムに利用

3. 新生産管理システム(Y-VIS(MP))の特徴

Y-VIS(MP)とは **Yutaka's-Visual Information System
(for Material and Production)**

【機能上の特徴】

- ・1つのアプリケーションに多くの業務機能を実装
- ・照会情報のEXCEL(ピボットTB含)連携、印刷可能
- ・汎用的な抽出条件の自動絞込み
- ・ユーザーの使用条件に応じたカスタマイズ
- ・着色、グラフ等により直感的判断をサポート
- ・瞬時に表示データのフィルタリング、並べ替え
- ・任意の条件による集計、分析可能
- ・WEBブラウザー機能によるシステム連携

4. Demonstration

5. 開発のポイント紹介-概要1

- ・TqueryのSQL→ClientDataSetパターン
- ・Query、ClientDataSet、DBGrid等のセット配列宣言、Tag使用
- ・各種カスタマイズ内容のINIファイル化
- ・変更頻度少のマスター情報のローカル保有
→ 定期的な取込み (BatchMove使用)
- ・iナビの活用 (TB、索引作成、SQL適正化、CSV取込み 等)
- ・Timer、ApplicationEventsコンポーネントでアプリケーション使用
有無を判定 → 表示情報の定期再表示

瞬時の
フィルタ、並べ替え

類似処理の汎用化

ユーザー操作の簡略化、使
用条件適正化

抽出条件の選択肢限定、
レスポンス

System I 連携

リアルタイム
情報

5. 開発のポイント紹介-概要2

- DBGridのOnDrawColumnCellイベントで着色ロジック関数呼出
- DBCrossTabSourceによる多目的グラフ、系列操作でのパラメータ取得とイベント活用
- WebBrowserコンポーネントとデータベース連携
 - GoogleMap連携と表示コントロールの使用
 - ネットワークカメラのアクセス管理
 - 文書管理ソフト ⇔ 複合機 ⇔ OCR、QRコード 連携
- 動的なマニュアル(DHTML形式)呼出し

5. 開発のポイント紹介 - DBGridへの着色方法(1/3)

```
procedure TForm1.AnyColorGridDrawColumnCell(Sender: TObject;
  const Rect: TRect; DataCol: Integer; Column: TColumn; State: TColumnState);
var
  n:integer;
begin
  //呼び出し元のGridを特定
  n:=(Sender as TDbGrid).Tag;

  //着色のロジックに応じて設定
  with (Sender as TDbGrid).Canvas do
  begin
    if GetGridColorForOP(n,DataCol)<>0 then
      Brush.Color:=GetGridColorForOP(n,DataCol);
      FillRect(Rect);
    end;
    mainDBGrd[n].DefaultDrawColumnCell(Rect,DataCol,Column,State);
  end;
end;
```

OnDrawColumnCell
イベントを利用

呼出し元ID (Tag)、
列番号より
着色用関数で色取得



5. 開発のポイント紹介 - DBGridへの着色方法(2/3)

```
function TForm1.GetGridColorForOP(n:integer;j:integer): integer;
```

```
var
```

```
  selcolor:Tcolor;
```

```
  colornum:integer;
```

```
begin
```

```
  colornum:=0; //初期値
```

```
  selcolor:=CIWhite; //初期値
```

```
case n of
```

```
  88:
```

```
  begin
```

```
    if (mainDBGrd[88].columns[j].FieldName='JU2KF4') then
```

```
      if Trim(mainCDS[88].FieldByName('JU2KF4').Value)='3' then
```

```
        selcolor:=$000D42FB;
```

```
      end; //n=88の場合ここまで
```

```
        //(中略)
```

```
    if selcolor<>CIWhite then colornum:=selcolor
```

```
    else colornum:=0;
```

```
    result:=colornum ;
```

```
end;
```

DBGrid着色用関数

呼出し元ID → n

列番号 → j

呼出し元に応じた
ロジックを記述
(これはID=88のケース)

該当列名 (columns[j].FieldName)、
該当レコードの内容 (Value)
により条件一致の場合に着色

DBGridの着色例

Y-VIS(MP) -- 新生産・在庫管理メニュー

素材在庫関係(M) 生産・出荷状況(P)

担当、物件等より抽出 | 受発注より抽出 | 工場単位で抽出

15:08 TIME DATE

受注毎の製造・出荷状況

納期範囲 2008/07/22 ~ 2008/08/05

赤の項目は指定・入力必須

部門名 鋼材課

担当者 相川 一樹

受注種別 リスト

納期経過後日数 10

自動更新 更新時間 15:04

一覧表示 計画情報 地図表示

生産、出荷明細不要

出荷(過状態発行)済除く

優先 種別先

受発注別生産・出荷実績

表示リスト (46 / 508)

別ウィンドウに一覧表示

生産指示未 生産指示済

生産一部済 生産済

配車済 出荷済

配車対象に追加

地区GR	受	発注	行	部門	担当	客先	持込先	工場	売区	受区	納期	入力日	日付	生産予定	完成時間	最終完成日	積込予定	配車最終日	出荷
福岡近郊	E	23575	13	鋼材課	相川 一樹	福岡	区福岡5-16	330	11	1	80715	80708	80709	0	0		710	0	80723
福岡近郊	E	23577	2	鋼材課	相川 一樹	福岡	区福岡5-16	430	11	1	80715	80708							80723
福岡近郊	E	23577	2	鋼材課	相川 一樹	福岡	区福岡5-16	430	11	1	80715	80708							80723
筑後・鳥栖	E	23576	1	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80804	805	1508		0	0	
筑後・鳥栖	E	23576	2	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80804	805	1508		0	0	
筑後・鳥栖	E	23576	3	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80804	805	1426				
筑後・鳥栖	E	23576	4	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80804	805	1426				
筑後・鳥栖	E	23576	5	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80804	805	1426				
筑後・鳥栖	E	23576	6	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80805						
筑後・鳥栖	E	23576	7	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80805						
筑後・鳥栖	E	23576	8	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80805						
筑後・鳥栖	E	23576	9	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80805						
筑後・鳥栖	E	23576	10	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80805						
筑後・鳥栖	E	23576	11	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80709	0	0		710	0	
筑後・鳥栖	E	23576	12	鋼材課	相川 一樹	大牟田	場	330	11	1	80725	80708	80709	0	0		710	0	
筑後・鳥栖	E	23578	1	鋼材課	相川 一樹	大牟田	場	430	11	1	80725	80708							
筑後・鳥栖	E	23578	2	鋼材課	相川 一樹	大牟田	場	430	11	1	80725	80708							
熊本市近郊	E	24568	1	鋼材課	相川 一樹	熊本		389	10	4	80722	80714							80722
熊本市近郊	E	23591	2	鋼材課	相川 一樹			430	11	1	80724	80715							
熊本市近郊	E	25801	1	鋼材課	相川 一樹			430	11	1	80725	80717	80722				722		80724
北九州近郊	F	43444	1	鋼材課	相川 一樹	東邦	若松工場内	330	13	1	80724	80718	80722	723	1006		723	0	80724
北九州近郊	F	43444	2	鋼材課	相川 一樹	東邦	若松工場内	330	13	1	80724	80718	80722	723	903		723	0	80724

生産状況・履歴明細 | 配車・出荷履歴明細

生産状況明細 表示10件

選択分のみ 全明細

Excel出力

明細データ取込み設定されていません

受発注の明細単位での
工程進捗状況を色分け

5. 開発のポイント紹介 - DBGridへの着色方法(3/3)

選択した行全体をわかりやすくしたい

```
type  
  TAccessDBGrid = class(TCustomDBGrid);
```

TAccessDBGridを宣言

```
n:=(Sender as TDbGrid).Tag;
```

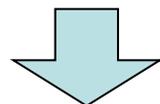
DBGridの
OnDrawColumnCell
イベント中に記述

```
//選択行を着色 描画行と現在行が一致するとき色を設定  
with TAccessDBGrid(Sender) do  
begin  
  if (DataLink.ActiveRecord = (Row-1)) then  
  begin  
    mainDBGrd[n].Canvas.Brush.Color := clLtGray;           ;  
    mainDBGrd[n].Canvas.Font.Color := clBlue;  
    mainDBGrd[n].Canvas.FillRect(Rect);  
  end;  
end
```

選択した行全体をわかりやすくしたい

SOZ092	SOZ003	SOZ105	SOZ014	SOZ015	SOZ004	HNA04K	SOZ005	SOZ006	SOZ007	SOZ008	SOZ102	SOZ101	
0000003	330	430	1	1	2100	生産定尺		2.3	914	1829	0	0	
0000004	330	430	1	1	2100	生産定尺		2.3	1219	2438	0	0	
0000005	330	430	1	1	2100	生産定尺		2.3	1524	3048	0	0	
▶1752379	330	430	1	3	2100	生産定尺		2.3	1524	3048	0	0	
0000006	330	430	1	1	2100	生産定尺			914	1829	0	0	
0000007	330	430	1	1	2100	生産定尺			1219	2438	0	0	
0000008	330	430	1	1	2100	生産定尺			1524	3048	0	0	
0000010	330	430	1	1	2100	生産定尺		4.5	914	1829	0	0	
0000011	330	430	1	1	2100	生産定尺		4.5	1219	2438	0	0	

着色処理なし



素材No	元コードNo	工場	分譲元	自支	規格	メカ	品種	品名	板厚	幅	長さ	尺寸	W1	L1	数量	重量	橋梁表面	引当数	引当重	紐付部門	紐付客先	
0000016		330	430	1		新日鐵	2100	生産定尺	6	1524	3048	5 x 10	0	0	80	17520		81	17739	0		
1401115		330	430	1		新日鐵	2100	生産定尺	9	914	1829	3 x 6	0	0	99	11682		15	1770	0		
▶1771622		330	0	1		新日鐵	2100	生産定尺	9	914	1829	3 x 6	0	0	1	118		0	0	0		
0000021		330	430	1		新日鐵	2100	生産定尺	9	1219	2438	4 x 8	0	0	194	40740		5	1050	0		
0483655		330	430	1		新日鐵	2100	生産定尺	9	1524	3048	5 x 10	0	0	17	5576		50	16400	0		
1770845		330	0	1		新日鐵	2100	生産定尺	9	1524	3048	5 x 10	0	0						0		
0000025		330	430	1		新日鐵	2100	生産定尺	12	914	1829	3 x 6	0	0						0		
0000027		330	430	1		新日鐵	2100	生産定尺	12	1524	3048	5 x 10	0	0						0		
0000030		330	430	1		新日鐵	2100	生産定尺	16	914	1829	3 x 6	0	0	24	5040		24	5040	0		
0000032		330	430	1		新日鐵	2100	生産定尺	16	1524	3048	5 x 10	0	0	10	5830		0	0	0		

着色処理実施

5. 開発のポイント紹介 – DBChartの活用方法

① 動的且つ汎用性の高いグラフ

→ 系列のデータソースは
DBCrossTabSourceが便利

- ・自動的に集計
- ・系列も自動生成
(系列毎の定義不要)



DBCrossTabSourceが
生成される



- ・ラベルフィールド(横軸)
- ・集計フィールド(縦軸)
- ・グループフィールド
(系列)

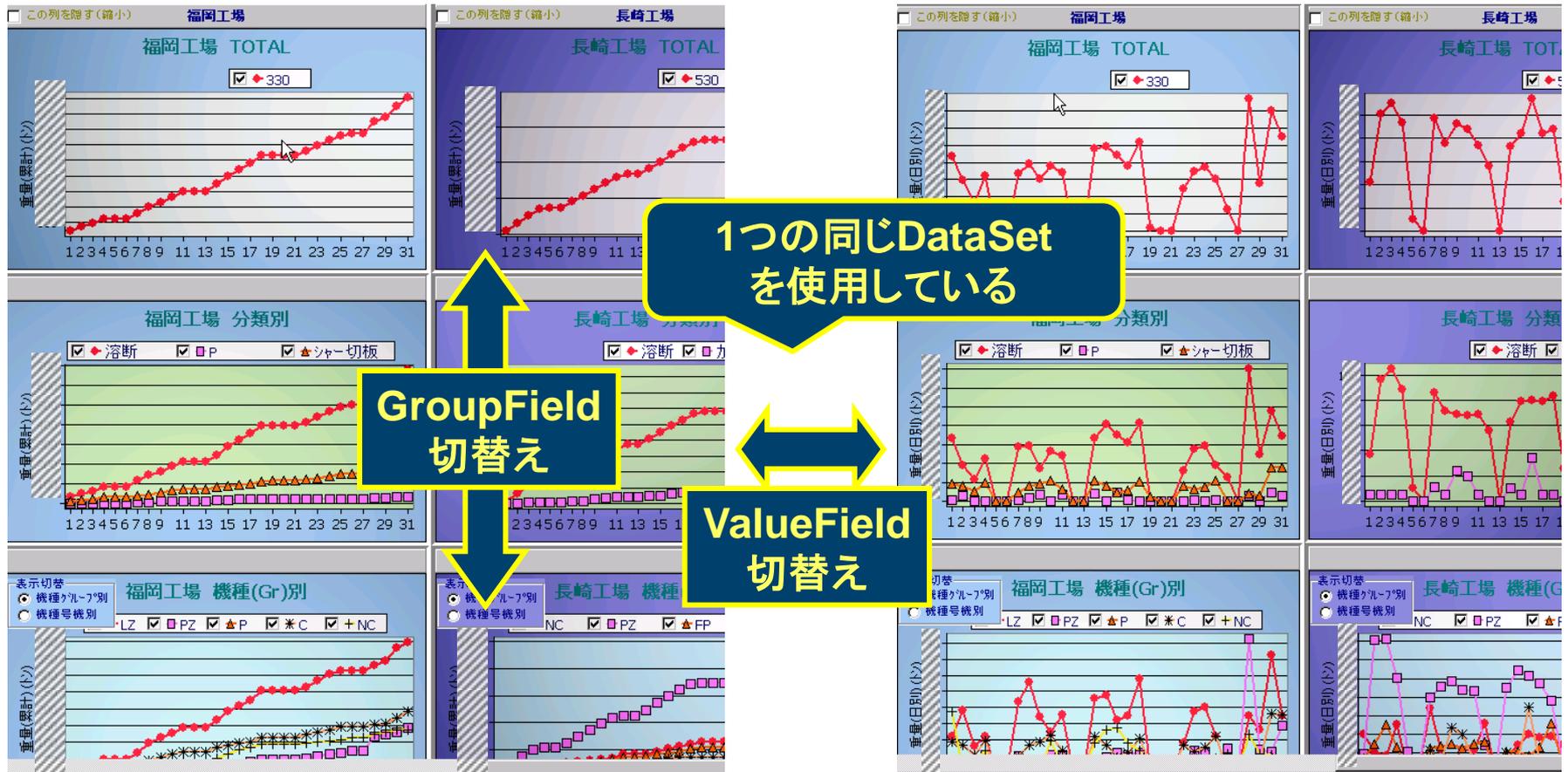
ソースで指定 →
ダイナミックに変更 (※)



(※) 変更前に一旦ActiveプロパティをFalse要

5. 開発のポイント紹介 – DBChartの活用方法

DBCrossTabSourceを利用したグラフの例



5. 開発のポイント紹介 – DBChartの活用方法

② グラフ上の着目点に関連する
データを確認するには？

→ 系列のイベント記述活用

折線グラフ上でクリックした部分の内容確認例

OnClickイベント使用

```
procedure TForm1.FastLineSeries1Click(Sender: TChartSeries;  
  ValueIndex: Integer; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
var  
  meisaibi:integer;  
  kisyu:string;  
begin  
  meisaibi:=StrToInt((Sender As TFastLineSeries).XLabel[ValueIndex]);  
  kisyu:=(Sender As TFastLineSeries).title;  
  
  (Sender As TChartSeries).Marks.Visible  
    :=((Sender As TChartSeries).Marks.Visible=False);  
  //(取得したパラメータを利用した処理をここに記述)  
end;
```

横軸の値を取得
ValueIndexはクリック部分の要素番号

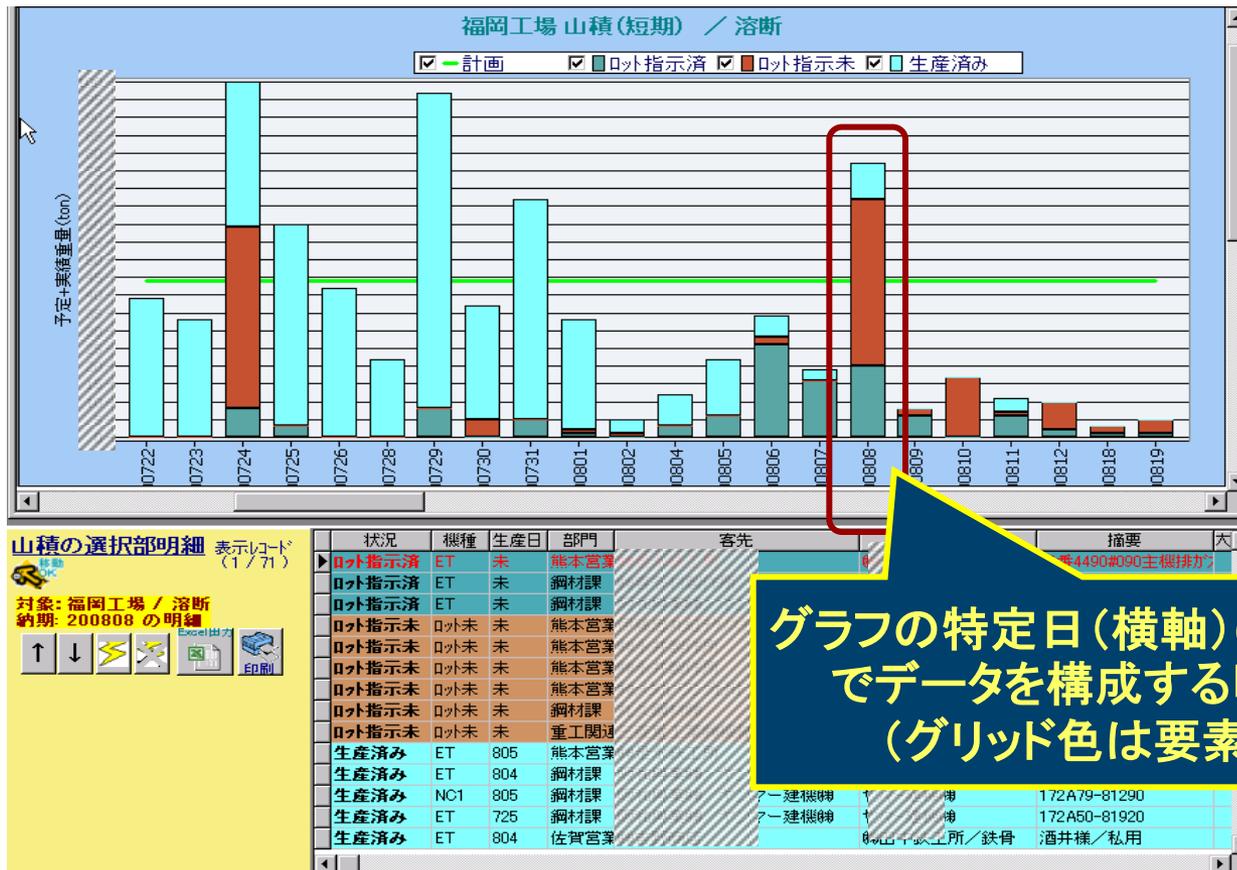
取得内容を
SQLの
パラメータ
とする

Titleプロパティ → 系列名取得
(グループ単位毎)

マーカー表示ON・OFF

5. 開発のポイント紹介 – DBChartの活用方法

グラフ上でクリックしたデータの詳細内容確認の例



グラフの特定日(横軸)の要素クリック
でデータを構成する明細を表示
(グリッド色は要素に対応)

5. 開発のポイント紹介 – WebBrowserコンポーネント

アプリケーションの中でWEBブラウザを表示するには？

- ① WebBrowserコンポーネントをフォームに貼付
(ツールパレット/Internetにある)

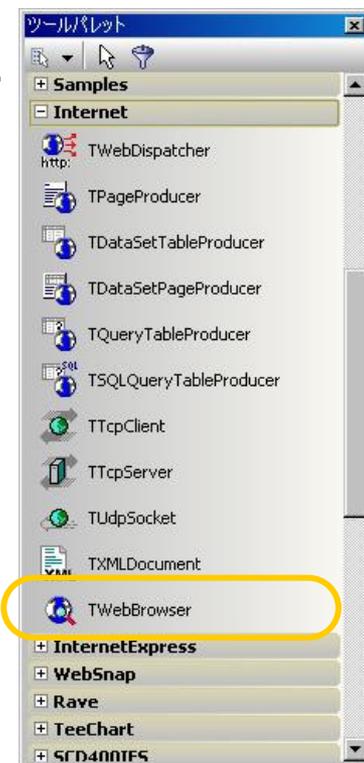


- ② 表示したいWEBページのURLをNavigate
メソッドで指定

例) // Yahoo!JAPANの場合

```
WebBrowser1.Navigate('http://www.yahoo.co.jp/');
```

これだけでOK



5. 開発のポイント紹介 – WebBrowserコンポーネント

5-1 お客様住所をGoogleMapで表示したい

考え方)

注) GoogleMapのAPIキーを取得、記述要

GoogleMapを表示するhtmlファイル(ローカル保管で可)
の内容をデータセットの内容により動的に編集

テクニック

雛形のhtmlを準備 → 編集すべき箇所を特定の文字列とする
→ 目的の内容(お客様名、住所)に文字列を置換



注) 住所の内容によってはエラーとなる。
ビル名含む場合、旧漢字等

編集済みのhtmlファイル(フルパス)をNavigateメソッドで指定

雛形のhtmlの一部分(編集される部分、2段表示の例)

//(前略)

```
if (geocoder) {
    geocoder.getLatLng(
        "KYAKUADD",
        function(point) {
            if (!point) {
                alert("KYAKUADD" + " では表示不能、下のボックスの住所を少し短くして試して下さい  
(又は古い漢字修正!!)");
            } else {
                map.setCenter(point, 12);
                map2.setCenter(point, 15);
                var marker = new GMarker(point);
                var marker2 = new GMarker(point);
                map.addOverlay(marker);
                map2.addOverlay(marker2);
                marker.openInfoWindowHtml("KYAKUNAME");
                marker2.openInfoWindowHtml("KYAKUNAME");
            }
        }
    );
}
// (後略)
```

実際の住所に置換

実際のお客様名に置換

雛形のhtmlのAPI Key記述について

// (先頭)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>Google Maps 配送先 住所</title>
```

```
<script src="http://maps.google.com/maps?file=api&v=2&key=XXXXXXXXX"  
  type="text/javascript" charset="utf-8"></script>
```

```
</head>
```

```
<body onload="load()">
```

// (後略)

ここに取得したAPI Keyを記述

API のバージョン、
最新では安定せずエラーの場合有
2、2.sであれば安定？

参考)

Google MAPS APIプログラミング (API Key取得方法もあり)

<http://www.geekpage.jp/web/google-maps-api/>

逆引きGoogle Maps APIリファレンス

<http://www.openspc2.org/reibun/Google/Maps/API/>

マップを2つ表示する(デュアルマップ)

<http://www.openspc2.org/reibun/Google/Maps/API/ouyou/ver2/005/index.html>

Javaのスクリプトエラー
→IEのキャッシュクリア
でOKの場合有り

お客様の住所のGoogleMap表示例(2段表示)

印刷も可能

お客様名をバルーンに表示

同じ住所を2段階の縮尺で同時表示(1ブラウザ)

予定	受発注	行	客先	持込先	次工程	品種
E	25190	1	閉資		ウエ	
E	25190	2	閉資		ウエ	
E	25190	3	閉資		ウエ	
E	25190	4	閉資		ウエ	
E	25190	5	閉資		ウエ	
E	25190	6	閉資		ウエ	
E	25189	1	閉資		シャ	
E	25189	2	閉資		シャ	
B	14418	1	群特		ガス	
B	22463	1	工ス		生産	
E	5069	1	閉資		フレ	
E	5069	2	閉資		フレ	
E	5069	3	閉資		フレ	
E	5069	4	閉資		フレ	
E	5068	1	閉資		フレ	
E	5068	2	閉資		フレ	
E	5068	3	閉資		フレ	
E	5068	4	閉資		フレ	
E	26244	1	工機 東陽建設		ウエ	
E	26245	1	閉資		普通	
E	25192	1	閉資		ガス	
E	25192	2	閉資		ガス	
E	25192	3	閉資		ガス	
E	5072	1	甘洋		織板	
E	5072	2	甘洋		織板	
E	26217	1	工機 東陽建設		特殊	
E	26217	2	工機 東陽建設		特殊	
E	26217	3	工機 東陽建設		特殊	
E	26217	4	工機 東陽建設		特殊	
E	26217	5	工機 東陽建設		特殊	
E	26217	6	工機 東陽建設		特殊	
E	25194	1	閉資		シャ	
A	33757	1	ガラ 南吉開製		切所	
A	33758	1	ガラ 南吉開製		切所	
E	25693	1	工ス		生産	
E	25198	1	閉資		シャ	
E	25198	2	閉資		シャ	
E	25198	3	閉資		シャ	
E	25198	4	閉資		シャ	
E	25695	1	所専 山口全機		生産	
E	25264	1	鉄工 宇美工場		生産	
E	25264	2	鉄工 宇美工場		生産	

複数の配送先をマーカー表示するには？

The screenshot shows a web-based map application interface. At the top, there are tabs for '配車・出荷準備' (Vehicle/Shipping Preparation), '受領確認処理' (Receipt Confirmation Processing), and '配送先地図' (Delivery Point Map). Below the tabs is a navigation bar with buttons for '検索ページ' (Search Page), '戻る' (Back), '進む' (Forward), '更新' (Refresh), '閉じる' (Close), and '印刷' (Print). The main area is a map of the Kumamoto region, showing several red location markers. A callout box points to one of the markers with the text '3箇所の配送先を一度に確認' (Check 3 delivery points at once). The map includes labels for various locations such as 鳥栖 (Tsuiki), 日田 (Hideta), 別府 (Beppu), 大分県 (Oita Prefecture), 熊本県 (Kumamoto Prefecture), 山鹿 (Yamaguchi), 玉名 (Tamana), 菊池川 (Kikuchi River), 阿蘇山 (Mount Aso), 島原 (Shimabara), 南島原 (Minamidomari), 島原半島 (Shimabara Peninsula), 島原湾 (Shimabara Bay), 上草 (Ugokusa), 緑川 (Kikuchi River), 普賢岳 (Fugen-dake), 雲仙 (Unzen), 筑後川 (Kikuchi River), 有明海 (Aomori Bay), 小城 (Miyama), 佐賀 (Saga), 賀県 (Kagami), 山鹿 (Yamaguchi), 阿蘇山 (Mount Aso), 高 (Takano), 長崎 (Nagasaki), 熊本 (Kumamoto), 宮崎 (Miyazaki), 鹿児島 (Kagoshima), 山口 (Yamaguchi), 広島 (Hiroshima), 福岡 (Fukuoka), 大分 (Oita), 松山 (Matsuyama), 長崎 (Nagasaki), 熊本 (Kumamoto), 宮崎 (Miyazaki), 鹿児島 (Kagoshima).

複数の配送先をマーカー表示するには？

雛形htmlの該当部分を配送先のみだけ繰返し記述

```
sl:= TStringList.Create;  
sl.LoadFromFile('雛形.html');
```

テクニック

繰返し部分の開始・終了Indexを取得(雛形html)

```
idx_s := sl.IndexOf(' //ここから');
```

```
idx_e := sl.IndexOf(' //ここまで');
```

→ テキストファイル書込み

```
for c:=1 to xx do // 配送先分繰返し
```

```
for i := idx_s + 1 to idx_e - 1 do
```

```
Writeln(OutF,sl.Strings[i]);
```

編集箇所は
適宜置換必要

//(前略)

//ここから

```
geocoder.getLatLng(  
"KYAKUADD",
```

```
function(point) {
```

```
if (!point) {
```

```
alert("KYAKUADD" + " は表示不能、住所情報が正しくない!!");
```

```
} else {
```

```
map.setCenter(point, 9);
```

```
var marker = new GMarker(point);
```

```
map.addOverlay(marker);
```

```
}
```

```
}
```

```
);
```

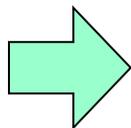
//ここまで //(後略)

5. 開発のポイント紹介 – WebBrowserコンポーネント

5-2 リアルタイムの工場映像を見たい

普通に見るだけであれば、ブラウザでネットワークカメラのIPアドレスをURLに指定するだけで可能

- しかし
- ・誰にでもは見せたくない
 - ・繋ぎ放題ではネットワーク負荷も問題
 - ・誰が見ているかも確認したい
 - ・カメラの追加・変更でアプリ変更したくない



カメラのマスター情報等のSystem i管理
+ WebBrowserコンポーネント
+ カメラ側のアクセス制御
+ アプリ側のアクセス制御

で対応

ネットワークカメラ表示画面例

4つのパネルにそれぞれ
WebBrowserコーポネントを配置

接続可能時間表示

接続中のユーザー表示

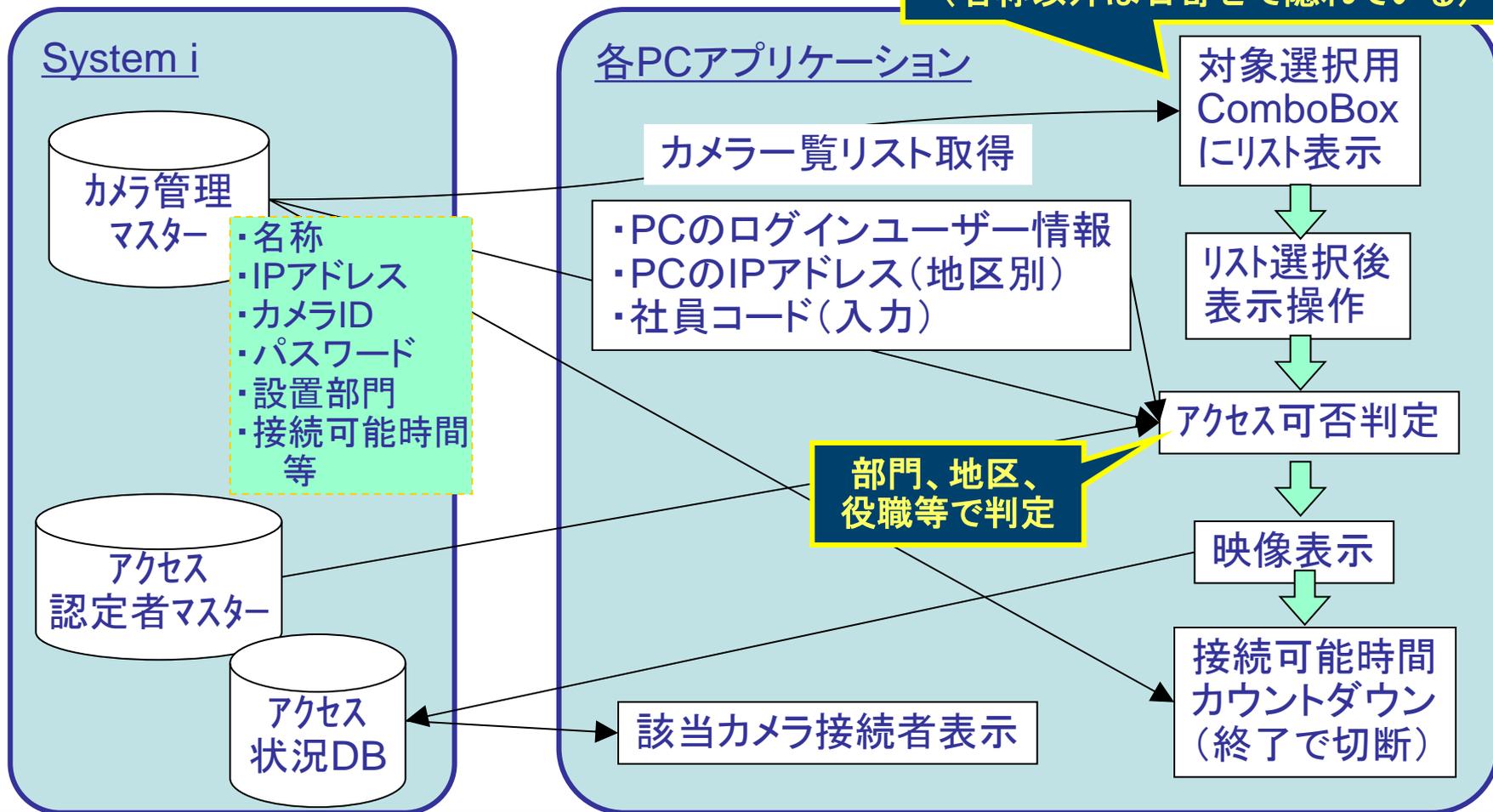
**操作可能機能の制限
(カメラ側のユーザー権限)**

**コンボボックスから
工場、会議室を選択
(IPアドレス、URLは見えない)**

The screenshot shows a software interface with four camera panels. Each panel includes a dropdown menu for camera selection, a '見る' (View) button, a '閉じる' (Close) button, and a connection time indicator. The top-left panel shows a meeting room with a connection time of 360 minutes. The top-right panel shows a factory floor with a connection time of 15 minutes. The bottom-left panel shows another factory floor with a connection time of 15 minutes. The bottom-right panel shows a dropdown menu with various camera locations such as '本社製造会議用', '福岡工場レーザ(マルチ)', and '刈田工場A棟(マルチ)'. The interface also features navigation controls like 'Scan', 'ズーム', and 'フォーカス'.

ネットワークカメラ表示の管理イメージ

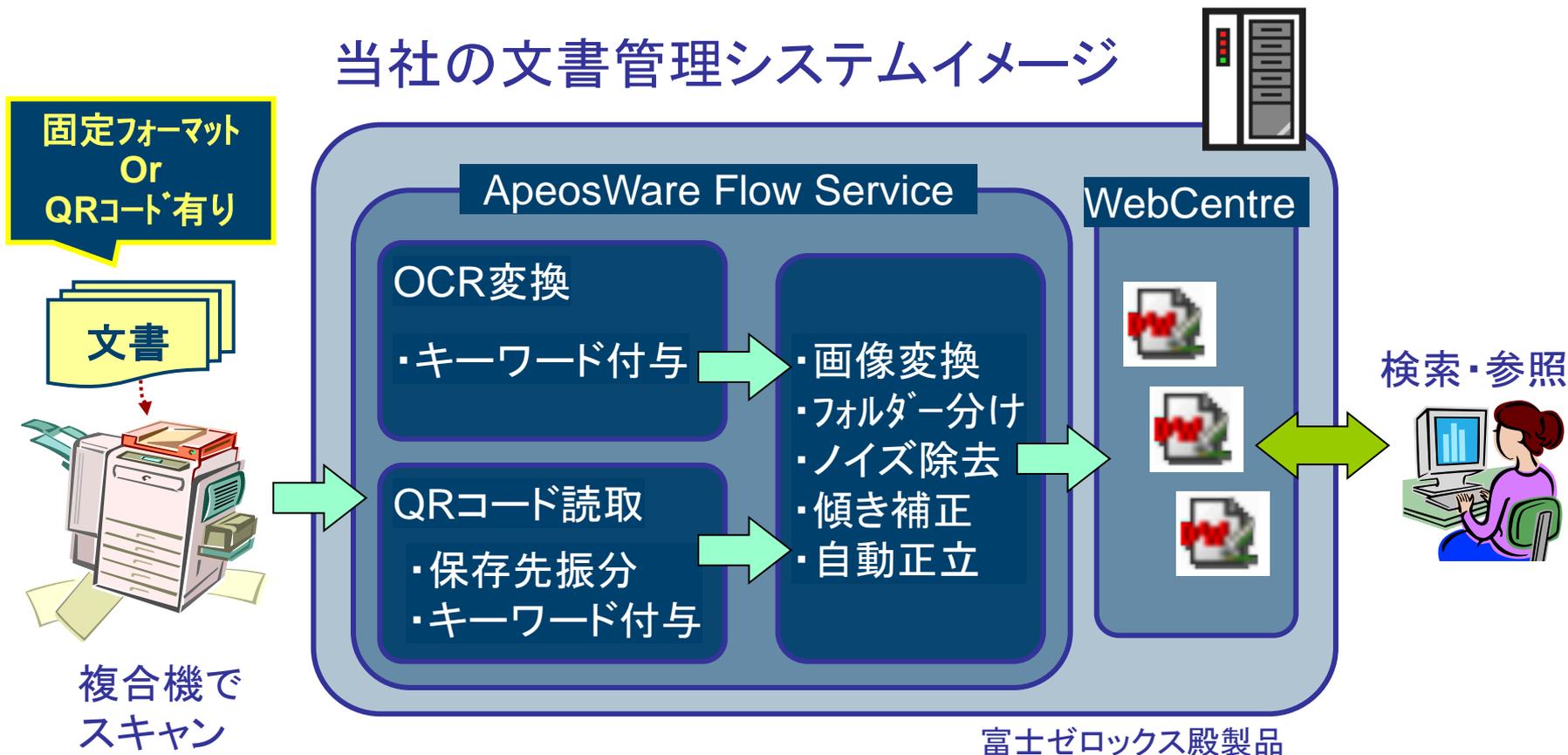
各マスタ情報を固定長でItemに追加
(名称以外は右寄せで隠れている)



5. 開発のポイント紹介 – WebBrowserコンポーネント

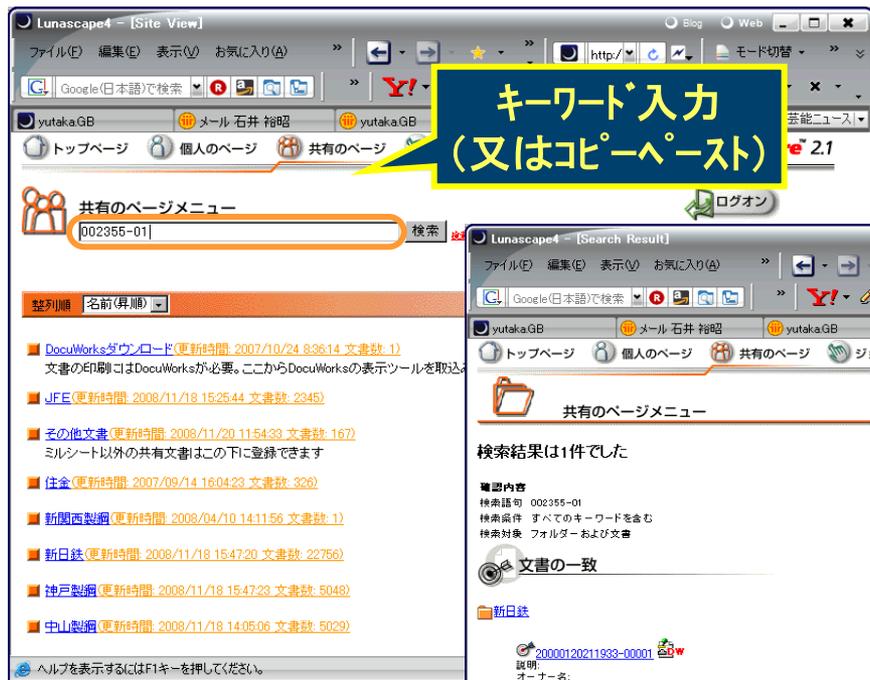
5-3 WEB型文書管理システムとの連動

当社の文書管理システムイメージ

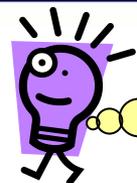


Delphi/400 Technical Seminar

従来は キーワード調査 →
ブラウザ起動し、検索ページよりキーワード入力



解決策)



ミガロ吉原様の
セミナーで
出てた!!

検索のキーワードをURLのパラメータとして渡すことで実現

WebCentre(富士ゼロックス殿)での検索例

```
SinseiBrowser.Navigate('http://(IPアドレス)//WebCentre/asp/c/SrhResult.asp ?  
Folder=(検索パス)&SEARCHOP=1&SEARCH='+SearcST);
```

WebCentreの検索パラメータ

Folder : 検索パス

SEARCHOP : 複数キーワード指示時の AND/OR 指定

SEARCH : 検索キーワード

検索の各パラメータは
検索ページにより
異なるので確認必要

パラメータの空白、特殊記号、全角文字等
は URLエンコード必要
(HTTPEncode関数使用)

検索用URLのパラメータが解れば

検索したい行のダブルクリック
のみで直接検索結果を表示
(リスト全行分のキーワードを
OR条件で一括検索も可能)

工場	部門	担当	客先	符号	受発注	行	納期	メーカー名	製鉄所	板番	連番	品種	規格	厚	幅	長	受注数	受注
330	311	松尾 広	鐵機興	E	56845	1	831117			39902	1780006	2200	S				1	450
▶ 330	311	松尾 広	鐵機興	E	53460	1	831117			520202	1798401	2450	S				1	15
430	311	松尾 広	キャリ	E	56856	1	831119			10901	1722258	2200	S				1	751
430	311	松尾 広	キャリ	E	56856	2	831119			39202	1777461	2200	S				1	841
430	311	松尾 広	キャリ	E	56856	3	831119			5923	1799964	2200	S					
430	311	松尾 広	キャリ	E	56856	3	831119			5924	1799964	2200	S					
430	311	松尾 広	キャリ	E	57412	1	831119			19503	1799772	2200	S					



通常実績 | 素材販売実績 | コイル製品実績 | 定尺実績 | 紐付き材使用履歴 | レバー生産履歴 | シート(WebCentre)

トップページ | 個人のページ | 共有のページ | ジョブテンプレート設定 | ヘルプ | WebCen

検索結果は1件でした

確認内容
 検索語句 963620202
 検索条件 一つ以上のキーワードを含む
 検索対象 フォルダerおよび文書

文書の一致

新日鉄

20081020085541-00001

説明:
 オーナー名:
 キーワード: 8-280-HI-1-9-5824 JIS G3136 SN490B AENMALL OOMUTA SHOPPINGMALL SHINCHIKUKUJI Y-32766 2008-09-30 B
 963620202 963620203 TO 963620204 6 FROM 963620205 963620206 TO 963620207 LADLE 1 2 X 2 4 3 8 X 1 2 1 9 2 12X2438X1 21 92 L

フォルダの一致

6. 今後の計画と課題

- ・WEBアプリケーションへのチャレンジ
どのような開発手法が望ましい？
- ・Y-VIS (MP)の機能追加、拡張
一部基幹業務取込み（GUI化メリット大の更新系など）
- ・照会レスポンス向上へのアプローチ
動的に変更されるSQLと同じ内容の処理を
System i側のNativeな方法で実行させられないか？
（Delphi/400側からRPG、CLソース等を動的に作成）

ご静聴ありがとうございました