

【T6】ケーススタディ



DEVELOPER CAMP

業務管理システム開発の実例にみる 3層スタイルC/Sアプリ作成術

有限会社 シンクソフト
代表取締役 安達 直人

アジェンダ



DEVELOPER CAMP

- 3層スタイルC/Sアプリ開発方法をとった背景
- 3層スタイルC/Sアプリとは
- 3層スタイルC/Sアプリの応用

3層スタイルC/Sアプリ開発方法をとった背景

3層スタイルC/Sアプリ開発方法をとった背景(1)

・ シンクソフト概要

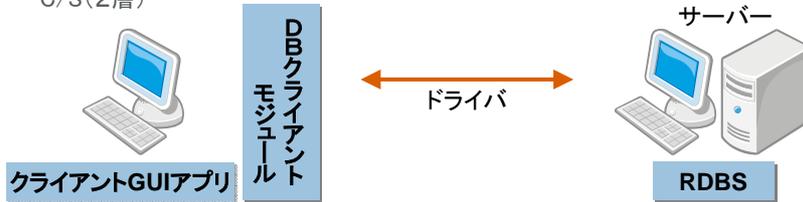
- ・ 主として業務システムの受託開発
 - ・ 要件定義～、外部設計～、内部設計～、製造～ 各種工程から対応。
主として、中小企業向け・大手企業の部門の業務システム、
パッケージソフトのカスタマイズなどを受託し、開発を行っています。
- ・ 言語は、基本的に **Delphi**
- ・ データベース、InterBase、FireBird、Oracleなど多岐。
バージョンもいろいろ。

- データベースやデータベースコンポーネントの依存を減らしたい
 - ・ InterBase、FireBird、Oracle、PostGreSQL etc
 - ・ BDE、dbExpress、IBX etc
- Web化への対抗。ブラウザベースのアプリケーションが好きになれない（あくまでも個人的な感想です）
 - ・ 使い勝手がいまいち
 - ・ ブラウザーの種類・バージョンに振り回される
 - ・ 3層 SOAP などを検討
- Corba3層開発との出会い
 - ・ Corbaを使った3層アプリの開発案件を受託

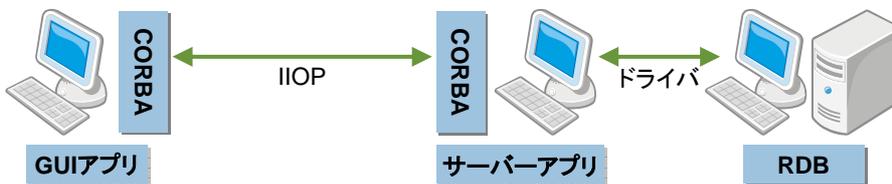
3層スタイルC/Sアプリとは(1)

・ まず3層アプリとは

- ・ C/S(2層)



- ・ 3層例 (Corba)

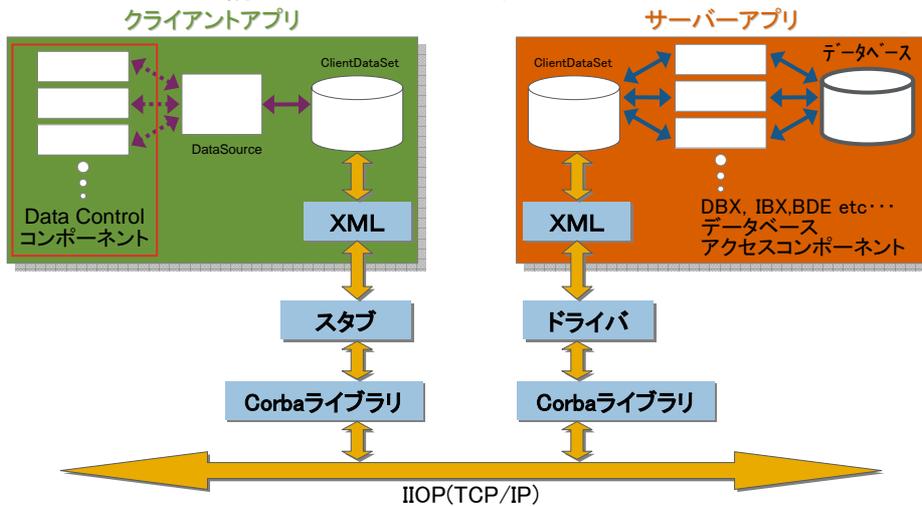


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

7

3層スタイルC/Sアプリとは(2)

・ プログラムの構成は (Corbaでの例)



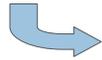
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

8

3層スタイルC/Sアプリとは(3)

- クライアントアプリは、ClientDataSetをベースに作成。
- サーバーアプリは、データベースアクセスに特化

データベースやデータベースコンポーネントの依存を減らしたい



この方法だと、依存部分は、
サーバーアプリのみである



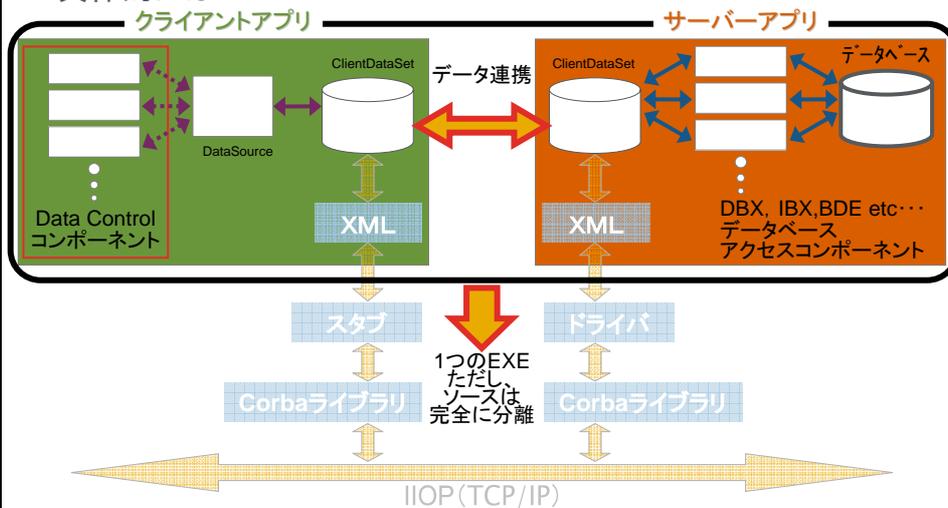
C/Sのシステムでも、同様に分けることができるのでは



1つのプロジェクトの中で、ユニットを
それぞれの役割に分けることにより
可能では。

3層スタイルC/Sアプリとは(4)

- 具体的には



クライアント側コード

```
OdsKokiyaku.Data
:= DMSvrKokiyaku.GetKokiyakuList( Shimei, Addr, Post, FlgDel, StDay, EdDay );
```

クライアント側 ClientDatasteの
CdsKokiyaku にサーバー側でデータ
ベースより取得したデータを設定する
処理のサンプル

サーバー側コード

```
function TDMSvrKokiyaku.GetKokiyakuList(ASimei, AAddr, APostCode:string;
ADelReadBoolean=False; ASDay:TDate=-1; AEdDay:TDate=-1); OleVariant;
var qryAdd: String;
    QryWk: TIBQuery;
    DspWk: TDataSetProvider;
begin
    DspWk := TDataSetProvider.Create(Self);
    QryWk := TIBQuery.Create(Self);
    QryWk.Database := DMSvrDataBase.DB;
    try
    try
        if not DMSvrDataBase.InTrm then
            DMSvrDataBase.TrmStart;
        DMSvrDataBase.TrmCommit;
        qryAdd := '';
        // // // ここ、qryAd に検索条件を設定
        QryWk.SQLtext := 'select * from "顧客マスタ" where ' + qryAdd;

        QryWk.Open; // // // テーブルオープン
        DspWk.DataSet := QryWk; // // // TDataSetProviderのDataSetプロパティに
        // // // 上記テーブルコンポーネントを設定
        Result := DspWk.Data; // // // クエリーの結果をDataSetProvider経由で
        // // // OleVariant型で返す
    except
        raise;
    end;
    finally
        QryWk.Close;
        QryWk.Free;
        DspWk.Free;
    end;
end;
```

- 効果は

- DB移植時、サーバー側ユニットのみの変更対象となる
- コードが標準化できる(他プロジェクトへの流用度が高くなった)
 - クライアント側ClientDataSet と DataControlコンポーネントなどを使ったクラス/ライブラリの開発。特に、DataControlコンポーネントを多用しても、データベースの変更時に問題とならない。
 - サーバー側ClientDataSet と各種DBアクセスコンポーネントの組み合わせでのクラス/ライブラリの開発。(DBアクセスコンポーネント: dbExpress、IBX など)

- ・ クライアント側 クラス/ライブラリの例
 - ・ ClientDataSetを機能を使ったDBGridの拡張
 - ・ CSV出力クラス
 - ・ 帳票出力クラス

- ・ サーバー側 クラス/ライブラリの例
 - ・ DB登録クラス
 - ・ ClientDataSetから、各種クエリコンポーネントを使った処理を、Overloadを使って、同一インターフェイスで作成。

```
//ClientDataSetのデータを、各クエリコンポーネントを使って登録するメソッドの宣言例  
procedure Cds2Touroku( aCdsSrc:TClientDataSet; aCHKQry,alnsQry,aUpdQry:TSQLQuery); overload;  
procedure Cds2Touroku( aCdsSrc:TClientDataSet; aCHKQry,alnsQry,aUpdQry:TIBQuery); overload;
```



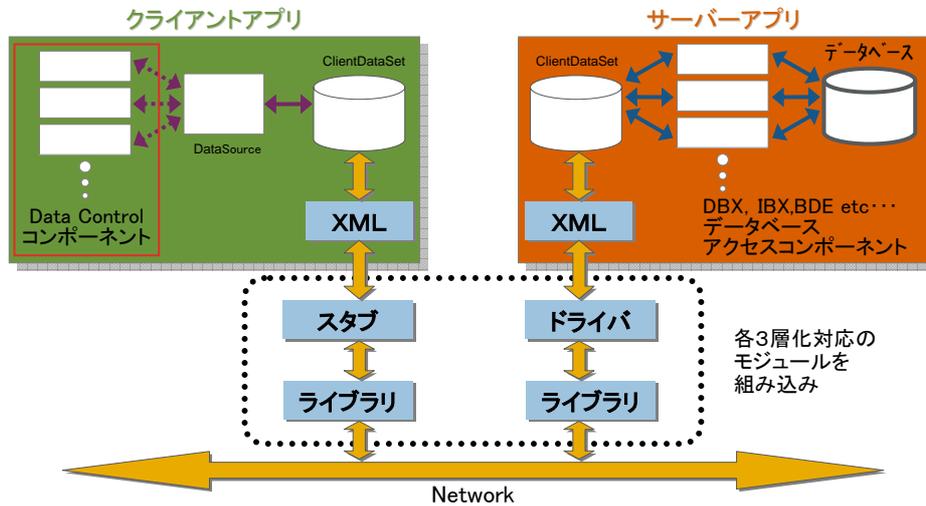
3層スタイルC/Sアプリの応用

3層スタイルC/Sアプリの応用(1)



- ・ 3層スタイルC/Sアプリの応用例
 - ・ 3層化への展開 当然！
 - ・ 画面設計・帳票設計への応用と、実装への展開

3層化



画面設計・帳票設計への応用と、実装への展開

- ・ ClientDataSetは、XML形式のファイルの読み込みができる。
 - ・ XML形式のダミーデータを持つファイルを作成することで、画面等に表示イメージを出力することが可能
 - ・ XML形式ファイルは、エディタで簡単に編集が可能。
 - ・ Delphiのライブコネクト機能のより有効な活用が可能
 - ・ それまでは、Paradoxなどを使って行うこともあった
 - ⇒BDEやツールのインストールが手間で、なかなか有効に使えない。
 - ⇒DBアクセスコンポの変更も必要。(BDEから、IBXやDBX)
 - ・ この方法では、
 - ⇒ツール不要。ドライバーなども不要。
 - ⇒クライアントソースは、そのまま活用することが可能。
- ClientDataSetのデータ取得を、
XML形式のファイルの読み込み から
サーバーモジュールメソッドの取得値を設定するように
変えればよい

- 参考資料

- ・ 第2回デベロッパーキャンプ
『オープンソースを利用した3層C/Sシステムの構築方法』資料

ありがとうございました

ご意見・ご感想などいただければ幸いです。

メール: adachi@thinksoft.co.jp

(件名は、日本語をお願いします)

- ・ 宣伝のページ

- ・ 有限会社 シンクソフト では、 Delphi を使ったシステム開発の受託開発を行っています。
- ・ 新規開発、既存システムのメンテナンス・機能追加など各種対応をいたします。既存システムのメンテナンスは仕様をがなくても対応可能です。Delphiのバージョンも不問です。
- ・ 詳細は
<http://www.thinksoft.co.jp/>