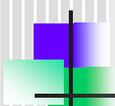


SOA実現に向けた いまどきのテクノロジーを含めた実装への 挑戦

XMLコンソーシアムWeek

関西部会

株式会社アークウェイ 丸山 和秀
株式会社プレイネットワークス 中川 聖悠
株式会社プレイネットワークス 山田 恭子



はじめに

フラワーアレンジ販売のインターネット新規出店

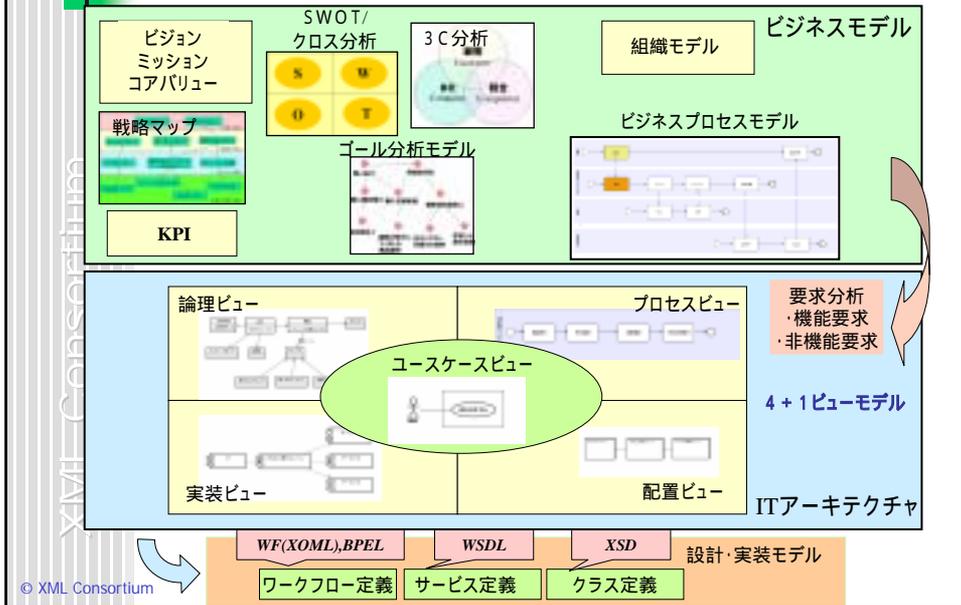
SOA実現に向けたビジネスモデルを策定

ビジネスモデルからITアーキテクチャを導く



導かれたITアーキテクチャから実装に挑戦

ビジネスモデルから導かれた ITアーキテクチャからさらに実装へ



実装モデルの作成

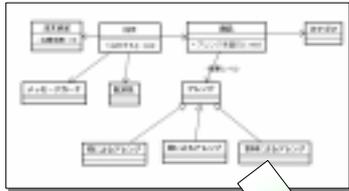


■ 導出されたITアーキテクチャ上に 実装モデルの定義を行った

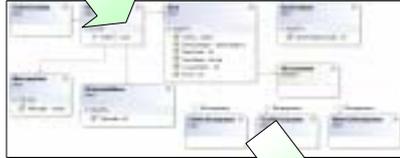
- データモデル(クラス)定義
XSDで表現可能
- サービスモデル定義
WSDLで表現可能
- ワークフローモデル定義
BPEL、XOMLで表現可能

いずれもXMLの定義

データモデルの定義



論理ビューをもとに
クラスを定義



Webサービス種類



- 商品提示サービス
 - 商品を検索する
 - クライアントへの商品一覧を返す
 - クライアントからの質問内容に応じて商品一覧を返す
- 受注サービス
 - クライアントからの受注処理を行う
 - 受注処理の窓口
- 受注登録サービス
 - 受注内容を履歴登録する(受注サービスから呼び出される)
- 在庫確認サービス
 - 在庫を確認する(受注サービスから呼び出される)
- 受注伝票発行サービス
 - 受注に対して、受注伝票を発行する(受注サービスから呼び出される)

Webサービスを実装



- WCFを採用してWebサービスを実装
 - WCF: WindowsCommunityFoundation
 - NET Framework 3.0で導入された統合通信フレームワーク

Webサービス実装: WCF



- ABC: WCFの重要概念
 - Address (アドレス)
 - メッセージを受信する場所を定義
 - どの(WHERE)アドレスに対してメッセージ送受信するか
 - Binding (バインディング)
 - エンドポイントが外部と通信する方法を定義
 - どのように(HOW)メッセージを送受信するか
 - Contract (コントラクト)
 - 外部に公開するインターフェイス
 - どのような(WHAT)サービスが提供されるか

■ ABCの定義 (Web.config)

■ 例: 商品提示サービス

```
<configuration>
... 中略 ...
<system.serviceModel>
  <services>
    <service behaviorConfiguration="XmlKansai2008.WebServices.FlowerServiceBehavior"
      name="XmlKansai2008.WebServices.FlowerService">
      <endpoint address="" binding="basicHttpBinding" bindingConfiguration=""
        contract="XmlKansai2008.WebServices.IFlowerService">
        <identity> <dns value="localhost">
        </endpoi>
        <endpoint address="mex" binding="mex" bindingConfiguration=""
        </service>
      </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="XmlKansai2008.WebServices.FlowerServiceBehavior">
          <serviceMetadata httpGetEnabled="true">
          <serviceDebug includeExceptionDetailInFaults="false">
          </behavior>
        </serviceBehaviors>
      </behaviors>
    </system.serviceModel>
  </configuration>
```

address:

- ・アドレスはHTTPプロトコルベース
- ・仮想ホスト側で割り当て

binding

- ・HTTP プロトコル バインディング

contract

- ・ServiceContract属性を付加したインターフェイス(クラス)を指定

■ Contract(コントラクト)定義詳細

■ サービスコントラクト

- サービス全体の構造に関する定義
 - サービスを提供するインターフェイスに ServiceContract属性を適用
 - 関連する複数の操作を 1つの機能単位に関連付け
 - 操作コントラクト
 - 操作のパラメータと戻り値の型を定義
 - サービス コントラクトを定義するインターフェイスには、メソッド定義に OperationContract 属性を適用

■ データ コントラクト

- データ構造の定義
 - サービスが使用するデータ型に DataContract属性を適用
 - データ型を他のサービスとの相互運用が可能

Webサービス実装: WCF



- インターフェイスの定義とそれを実装したサービスクラスの定義
 - 例: 商品提示サービス

サービスコントラクト

```
[ServiceContract]
public interface IFlowerService
{
    [OperationContract]
    List<Item> FindItemsByCategory(int categoryID);
}
```

データコントラクト

```
[DataContract]
public class Item
{
    [DataMember]
    public string Name { get; set; }

    [DataMember]
    public int Price { get; set; }
}
```

サービスクラス: Interfaceを実装したクラス

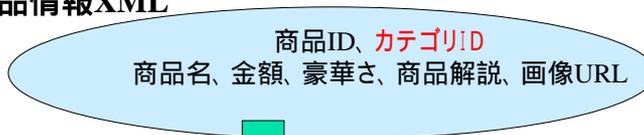
```
public class FlowerService : IFlowerService
{
    public List<Item> FindItemsByCategory(int categoryID)
    {
        //カテゴリIDで商品を検索して、商品一覧を返す
    }
}
```

Webサービス実装: ビジネスロジック



- 商品検索機能
 - 例: カテゴリIDで商品を検索して、商品一覧を返す

商品情報XML



- 商品情報XMLからデータ取得
 - カテゴリIDにマッチする商品の情報を検索
 - LINQの利用

戻り値: 商品一覧 List<Item>

商品ID	カテゴリID	豪華さ	商品名	画像URL	金額	商品解説
1	1	4	カーベラのブー	http://localhost:5959/	3150	優しい気持ちや想いやり
2	1	2	ひまわりのアレ	http://localhost:5959/	3150	ひまわりのお花もお部屋に
3	1	5	ユリの花束	http://localhost:5959/	7350	大人の女性を感じる・・・魅

- LINQ(Language Integrated Query)とは
 - SQL文のような構文でデータベース・XML・オブジェクトなどに一貫した方法でクエリを発行できる.NETFramework 3.5の新機能
- LINQ to XML
 - XMLの要素、属性値を指定し、値を取得するクエリを記述

- LINQtoXMLを使用し、データを操作

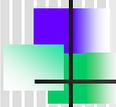
```
<items>
  <item id="1"
        categoryId="1"
        luxuriousness="4"
        name="ガーベラのブーケ(ピンク)"
        imageUrl="http://localhost:59980/image/1.png"
        price="3150"
        description="優しい気持ちや思いやりをいっぱいいっぱいお花に込めて..."/>
</items>
```

```
var selectedItem = from item in itemDataFromxml.Descendants("item")
                   where item.Attribute("id").Value.Equals(itemId.ToString())
                   select new Item
                   {
                       ID = (int)item.Attribute("id").Value,
                       Name = item.Attribute("name").Value,
                       ImageUrl = item.Attribute("imageUrl").Value,
                       Price = (int)item.Attribute("price").Value,
                       Luxuriousness = (int)item.Attribute("luxuriousness").Value,
                       Description = item.Attribute("description").Value,
                       Category =
                           GetCategoriesByCategoryID(item.Attribute("categoryId").Value)
                   };
```

select部分(射影)
インスタンスを生成

where句
XMLの属性値と
引数を比較する条件句

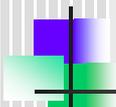
```
new Item
{
    ID = (int)item.Attribute("id").Value,
    Name = item.Attribute("name").Value,
    ImageUrl = item.Attribute("imageUrl").Value,
    Price = (int)item.Attribute("price").Value,
    Luxuriousness = (int)item.Attribute("luxuriousness").Value,
    Description = item.Attribute("description").Value,
    Category =
        GetCategoriesByCategoryID(item.Attribute("categoryId").Value)
};
```



ワークフローモデル定義



- ビジネスプロセスを
継続的に改善するために
システム側で柔軟に対応できる技術を検討
 - ビジネスプロセスエンジン / ワークフローエンジン
(例)
 - Oracle BPEL Process Manager
 - IBM WebSphere Process Server
 - Microsoft BizTalk Server
 - Microsoft Windows Workflow Foundation



ワークフローエンジン (WF) 採用

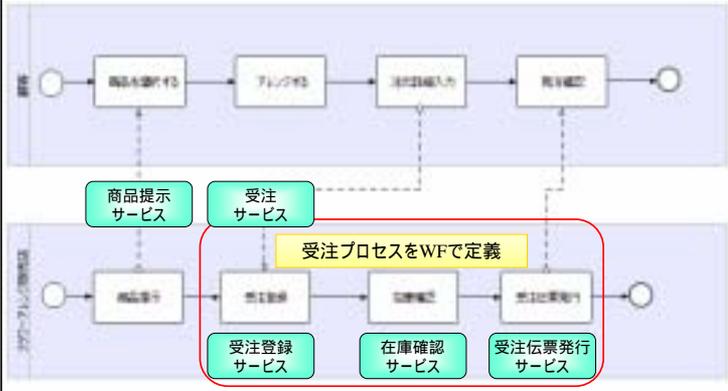


- 今回はWindows Workflow Foundation(WF)を
採用して実装
- WFとは
 - ワークフローを.NET Framework上で
取り扱うためのフレームワーク
 - WFでは、大きく分けて
2種類のワークフローを扱うことができる
 - シーケンシャルワークフロー
 - ステートマシンワークフロー



受注のフロー





- WFで受注のプロセスを定義
- WCFで実装したWebサービスをWFから呼び出す

© XML Consortium

WFとWebサービスの連携

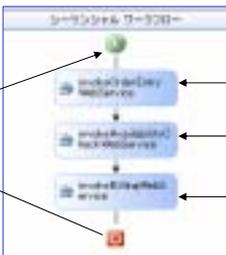


- WFを公開
 - クライアントからWFを実行するためのWebサービスを公開
- WFにてWebサービスを順番に実行
 - 受注登録 在庫確認 受注伝票発行

ワークフロー (WF)

受注サービス
クライアントから呼ばれる

start()



受注サービス

在庫確認サービス

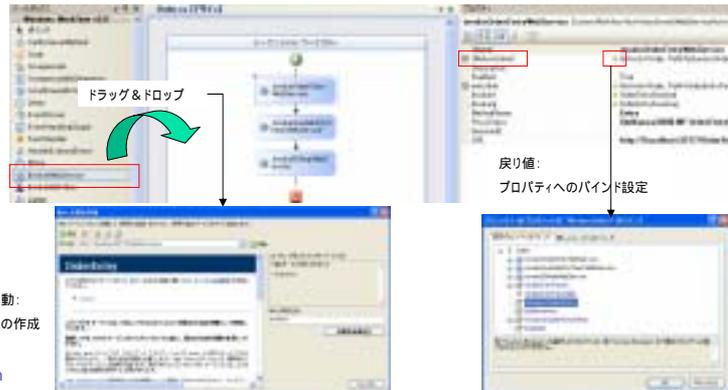
受注伝票発行サービス
WFから呼ばれる

© XML Consortium

WFとWebサービスの連携



- ワークフローからのWebサービス実行
 - InvokeWebServiceアクティビティを利用
 - ウィザードによる設定 Proxyクラスの生成が可能
 - プロパティ・ウィンドウでパラメータを設定



© XML Consortium

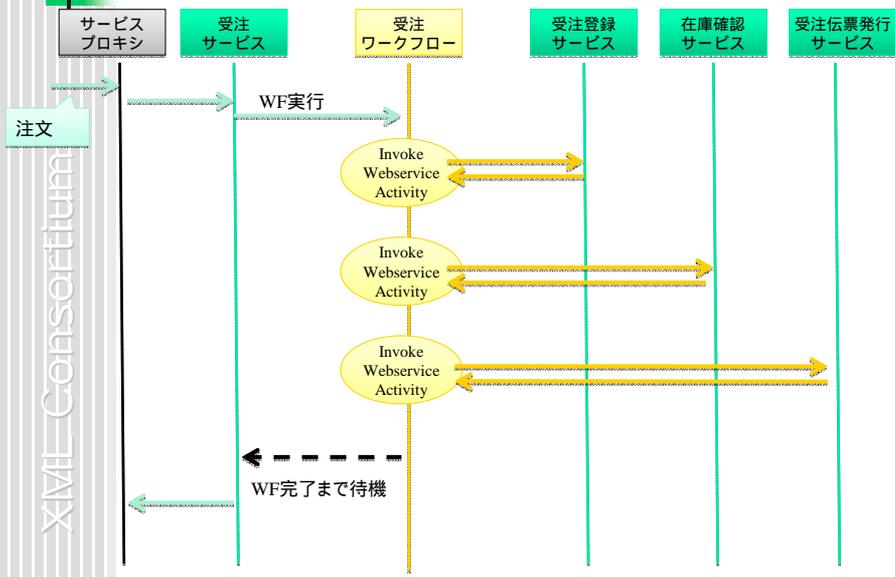
WFとWebサービスの連携



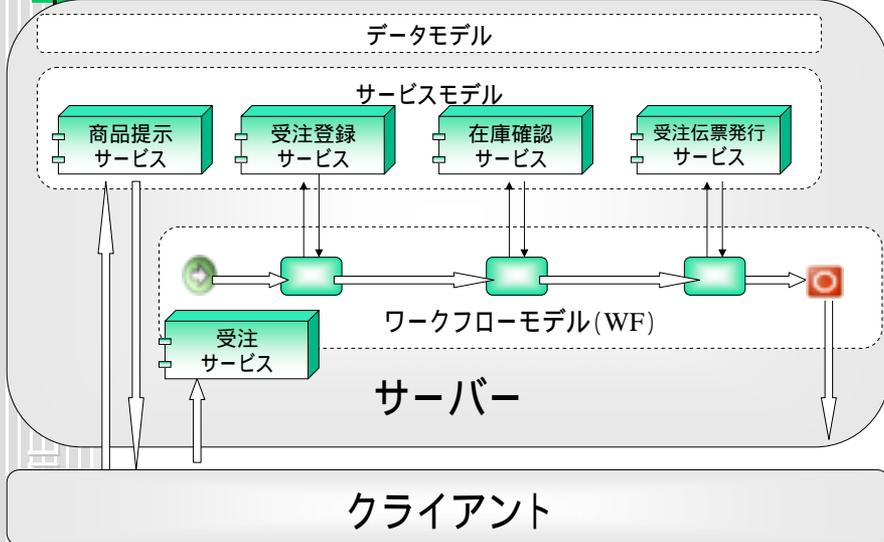
- WebサービスからWFを実行(WFの公開)
 - WCFを利用し、Webサービスを公開
 - 公開したWebサービスからWFを実行
 - System.Workflow.Runtime.WorkflowRuntimeクラスを利用
 - ワークフローを実行するためのランタイム
 - 1つのプロセスにつき1つだけインスタンスを生成し、その上でワークフローを実行
 - Webサービスでは、WFの完了を待つ
 - WorkflowCompleted イベント
 - ワークフロー インスタンスが完了すると発生
 - 完了イベントの処理を実装
 - WFで処理された値を取得し、Webサービスの戻り値に設定

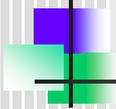
© XML Consortium

シーケンス



アーキテクチャ図

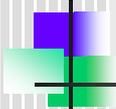




XML Consortium

クライアントアプリケーション

© XML Consortium



テクノロジー



XML Consortium

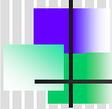


ADOBE AIR™



Microsoft®
Silverlight™

© XML Consortium



AIRとは



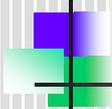
XML Consortium

アドビシステムズが開発する、**デスクトップ**・リッチインターネットアプリケーション(RIA)を開発・実行するための複数のオペレーティングシステムに対応したランタイムライブラリである。Adobe Flash、Adobe Flex、HTML、Ajaxなどの**ウェブ開発技術**を利用することができる。

by http://ja.wikipedia.org/wiki/Adobe_Integrated_Runtime



© XML Consortium



Silverlightとは



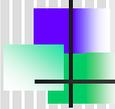
XML Consortium



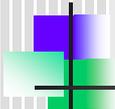
Microsoft Silverlightは、クロスブラウザ、**クロスプラットフォーム**、およびクロスデバイスに対応する**Webブラウザのプラグイン**です。**.NETを基盤**とし、Webにおける次世代のメディア体験とRIAを実現します。

by <http://www.microsoft.com/japan/silverlight/>

© XML Consortium

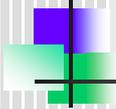


AIRによる デスクトップアプリケーション



アプリケーション概要

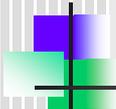
- AIRによるデスクトップアプリケーション
- コンセプトは「そこにある」
- 生活に溶け込んだアプリケーション
- デスクトップに常駐し、花のイメージを表示することで、あたかもアルバムを見ているかのような印象を与える
 - 花を購入しよう意識する必要はない
 - 気に入ったものがあれば、そこで初めて購入プロセスに移る
- 自然言語により気分などを設定可能
- 対応したアレンジのイメージを表示する



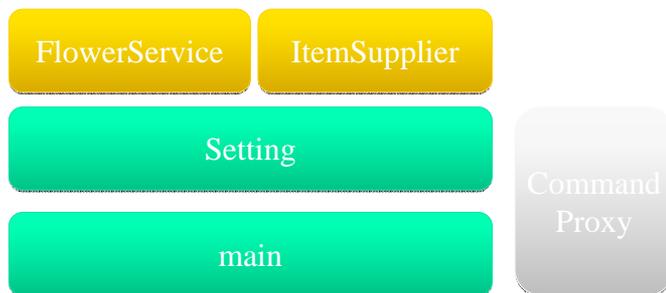
AIR選定理由

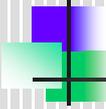


- 違いのあるアプローチ方法
 - 「購入するためのアプリケーション」ではなく「生活に溶け込んだアプリケーション」を実現するため
 - デスクトップアプリケーションとして常駐させることで潜在的な購入機会をふやす
- 技術的な導入の容易性
 - クロスプラットフォーム
 - 容易なインストール



コンポーネント

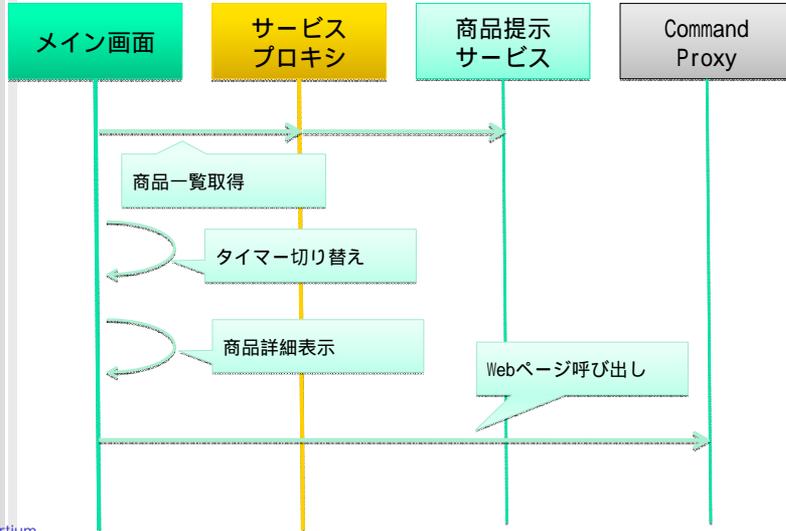




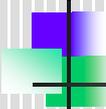
シーケンス



XML Consortium



© XML Consortium



XML Consortium

Silverlightによる ブラウザアプリケーション

© XML Consortium

アプリケーション概要



- SilverlightによるWebブラウザアプリケーション
- 購入方法に大きな特徴がある

XML Consortium

- 選択式の質問に回答してもらうことにより、その時の気分などの状態を分析し、最も適したアレンジを提供する

Feel

- 誕生日祝いや、結婚祝いなど状況に応じて最適なアレンジを提供する

Recommend

- テンプレートとなるアレンジに、「もっと豪華に」や「かわいらしく」などの感覚的な指定を行うことで、カスタマイズされたアレンジを作成する

Custom

今回はこの購入方法を構築

© XML Consortium

Silverlight選定理由



- 自社の強みであるアレンジメント能力を表現するためにはRIA技術が必須
 - 感覚的な指定に反応して変化できるUIと表現能力
 - HTMLベースでは限界、もしくは複雑になる
- サーバー側技術(WF / WCF)との親和性を考慮

XML Consortium

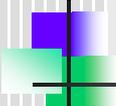
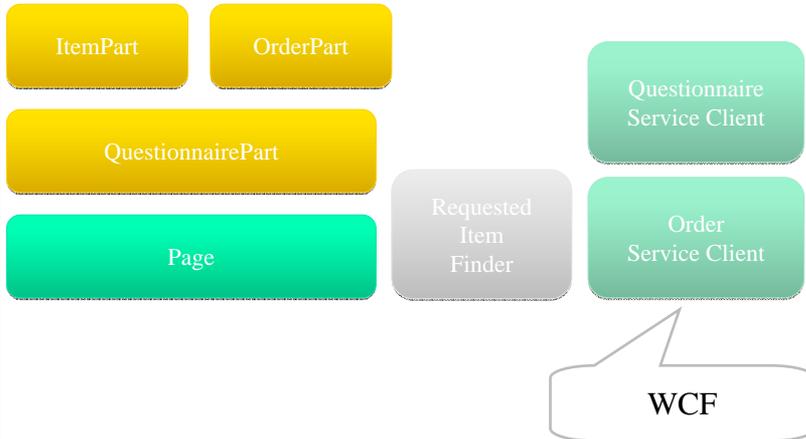
© XML Consortium



コンポーネント



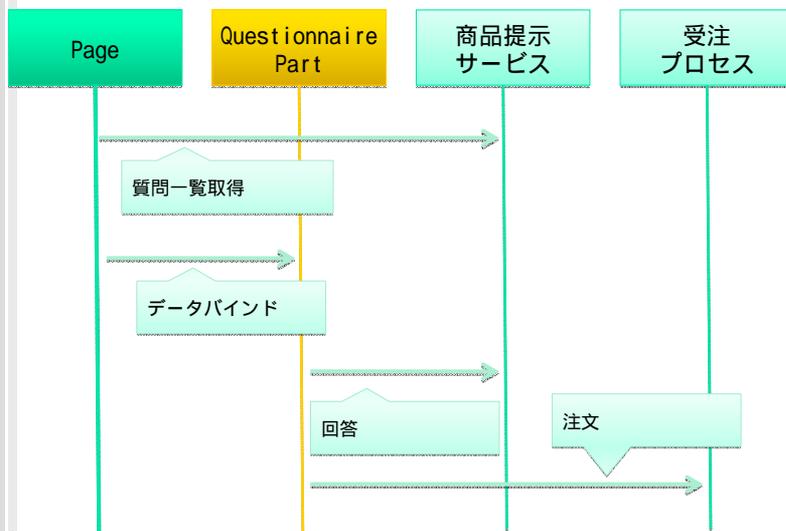
XML Consortium

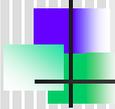


シーケンス

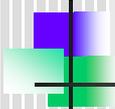


XML Consortium





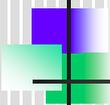
デスクトップとWEBの融合



概要



- クライアントアプリケーションで気に入った花があった時に、その詳細をブラウザから確認できる
 - あくまでもクライアントアプリケーションはガジェットという位置づけ

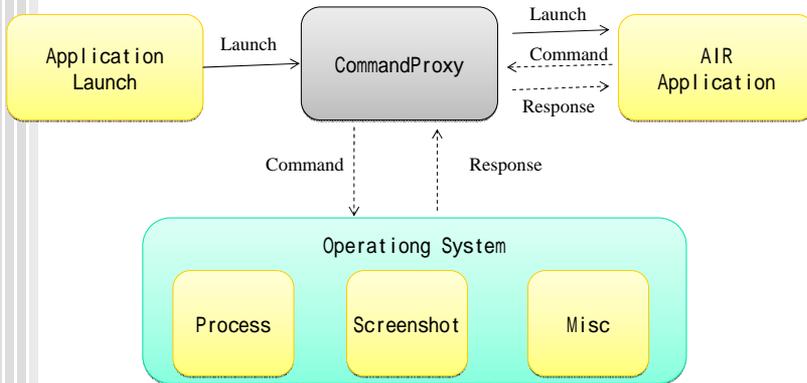


CommandProxy



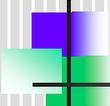
XML Consortium

- AIRからデスクトップにはアクセスできないため CommandProxyを仲介して、ブラウザを立ち上げる



<http://www.mikechambers.com/blog/2008/01/22/commandproxy-its-cool-but-is-it-a-good-idea/>

© XML Consortium

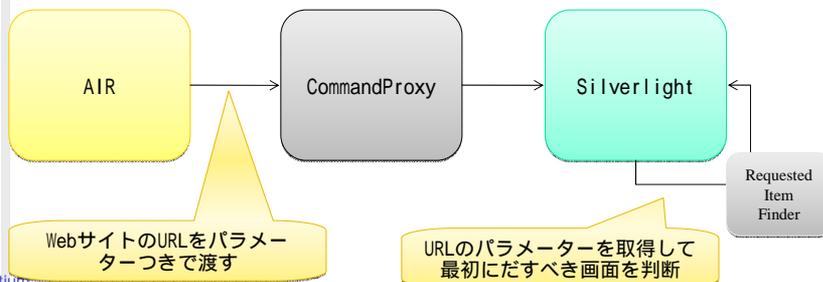


AIRとSilverlightの接続

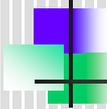


XML Consortium

- AIRからCommandProxyにパラメータ付きのURLを渡す
 - `http://localhost/flowershop/?item=6`
- CommandProxyは渡されたURLでブラウザを起動する
- Silverlightは起動したときのURLを確認して初期に表示する画面を決定



© XML Consortium



まとめ



- サービス(WCF/WF)
 - WCFやWF、LINQなど技術を組み合わせる時に、構造が複雑になってしまった
 - しかし、個々の技術単体で見た時はそれぞれに便利
 - LINQはプログラムの書き方が変わった
- Silverlight
 - .NETベースの開発
 - 情報がまだまだ少ない
- AIR
 - Flexも含めて情報が豊富
 - どちらの情報か区別がつかなくなる可能性も
 - “AIR”でググれない
 - 代官山 AIR (エアー) 2008/6/21 14:05