

## 気象情報Webサービス

2005年 6月10日

日本ユニシス(株)

鮫島 荘介

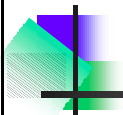
日本アイオナテクノロジー(株)

江川 潔




## アジェンダ


- ⊕ 気象情報Webサービス概要
- ⊕ 気象情報について
- ⊕ ITS世界会議2004向けシステム
- ⊕ 愛・地球博試験運用向けシステム



## 気象情報Webサービス概要



## 気象情報Webサービス概要

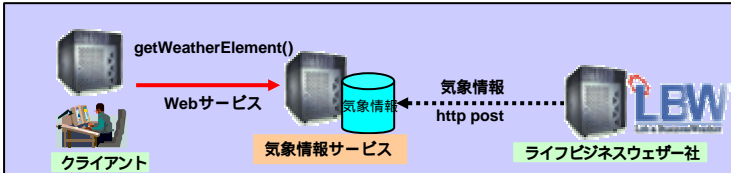


### 機能概要

ライフビジネスウェザー社提供の気象情報コンテンツをラッピングし、Webサービスとしてインタフェースを提供する。

### 特徴

インターネットを介し送信されてくる全国市区町村毎の気象情報を整理蓄積し、リクエストされた予報時刻の緯度経度に該当する市区町村の気象データを提供する。



```
graph LR; Client[クライアント] -- "getWeatherElement()  
Webサービス" --> Service[気象情報サービス]; Service -. "気象情報  
http post" .-> LBW[ライフビジネスウェザー社];
```

クライアントから気象情報サービスへは、予報時刻と予報する地点の緯度経度を指定する。気象情報サービスは、指定された情報を元に、蓄積した情報を検索し、予報結果を返す。

XML Consortium

Copyright © XMLコンソーシアム 2005 All rights reserved.

4



## 気象情報について

日本ユニシス(株)

鮫島 荘介

日本アイオナテクノロジー(株)

江川 潔

## 気象情報について

### 気象情報の提供

気象予報情報は、定時にcsvファイルがHttpのpostメソッドにより、ライフビジネスウェザー社より送信されてくる。

```
POST http://.....  
Content-Type multipart/form-data  
Content-Disposition: form-data;  
Content-Type: application/octet-stream
```

気象情報サービス

ライフビジネスウェザー社

### 気象情報Webサービスで利用する情報について

気象情報Webサービスでは、全国市区町村毎のスポット予報情報を用いており、2種類の期間の異なる予報情報を利用している。


- 短期予報(48時間以内)
- 中期予報(7日以内 = 週間予報)

Copyright © XMLコンソーシアム 2005 All rights reserved.

6



# 気象情報について



## スポット気温情報csvファイルの詳細

スポット気温csvファイルは、スポット天気.csvと同じレコード構成である。

**【スポット気温情報csvファイル概要】**


ヘッダレコード	20041008090000
データレコード	23101 +15.2,+15.0,+14.8,+14.7,+14.4,+14.4,・・・,+12.8
	23102 +15.3,+15.0,+14.8,+14.7,+14.4,+14.3,・・・,+12.7

予報気温。  
 符号付きで少数第1位まで。

XML Consortium

Copyright © XMLコンソーシアム 2005 All rights reserved. 9

# 気象情報について



## スポット週間予報csvファイルの詳細

スポット週間予報csvファイルは、同一の市区町村コードに天気レコード、気温レコードの2レコードを持つ。

**【週間予報情報csvファイル概要】**

ヘッダレコード	27 28 29 30 01 02 03	7日分の降水確率。当日は予報なし(///)
データレコード	A23101,112,101,101,101,201,200,200,///,20,20,20,30,40,40	A+市区町村コードのレコードは、天気と降水量を表す。
	A23102,112,101,101,101,201,200,200,///,20,20,20,30,40,40	
データレコード	P23101,///,22,20,19,21,21,23,///,10,8,5,8,11,12	P+市区町村コードのレコードは、最高気温と最低気温を表す。
	P23102,///,22,20,19,21,21,23,///,10,8,5,8,11,12	


7日分の予報天気コード。週間予報は、121種類のコードが存在する。112は、晴れ後一時雨

7日分の最高気温。当日は予報なし(///)

7日分の最低気温。当日は予報なし(///)


XML Consortium

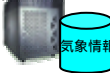
Copyright © XMLコンソーシアム 2005 All rights reserved. 10



## 気象情報について

### 市区町村コードとの対応

  
 クライアント

  
 気象情報サービス

時刻、場所(緯度経度) → 予報情報 → 気象情報

気象情報サービスへのリクエストとして、予報の対象となる時刻と場所情報として緯度経度が渡されてくる。

気象情報は市区町村コードで保持しており、緯度経度とのマッピングが必要。

↓


そのため各市区町村の役所の緯度経度情報をシステム内部に持ち、リクエストされた緯度経度に一番近い役所を探し出し、その役所の市区町村コードの予報を返している。

**【市区町村コード毎の役所の座標値】**

市区町村コード	市区町村名	役所緯度	役所経度	緯度(整数)	経度(整数)
23101	名古屋市千種区	35.09.47	136.56.57	32406272	12612352
23102	名古屋市東区	35.10.34	136.55.45	32418304	126193920
23103	名古屋市北区	35.11.27	136.54.53	32431872	126180608
23104	名古屋市西区	35.10.57	136.53.19	32424192	126156544
23105	名古屋市中村区	35.09.56	136.52.33	32408576	126144768
...	...	...	...	...	...
23623	瀬美郡瀬美町	34.37.11	137.06.45	31905536	126362880

緯度経度情報は、度.分.秒を整数化した値で管理している。

Copyright © XMLコンソーシアム 2005 All rights reserved.
11



## 気象情報について

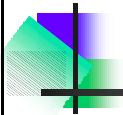
### 市区町村コードの算出方法

リクエスト場所に一番近い地区町村コードの算出には、リクエスト場所と愛知県内全ての市区町村役所の緯度経度情報から距離を算出し、もっとも近い距離の役所を採用することで導いている。

全ての市区町村役所との距離の算出は、処理としては無駄が多いが、

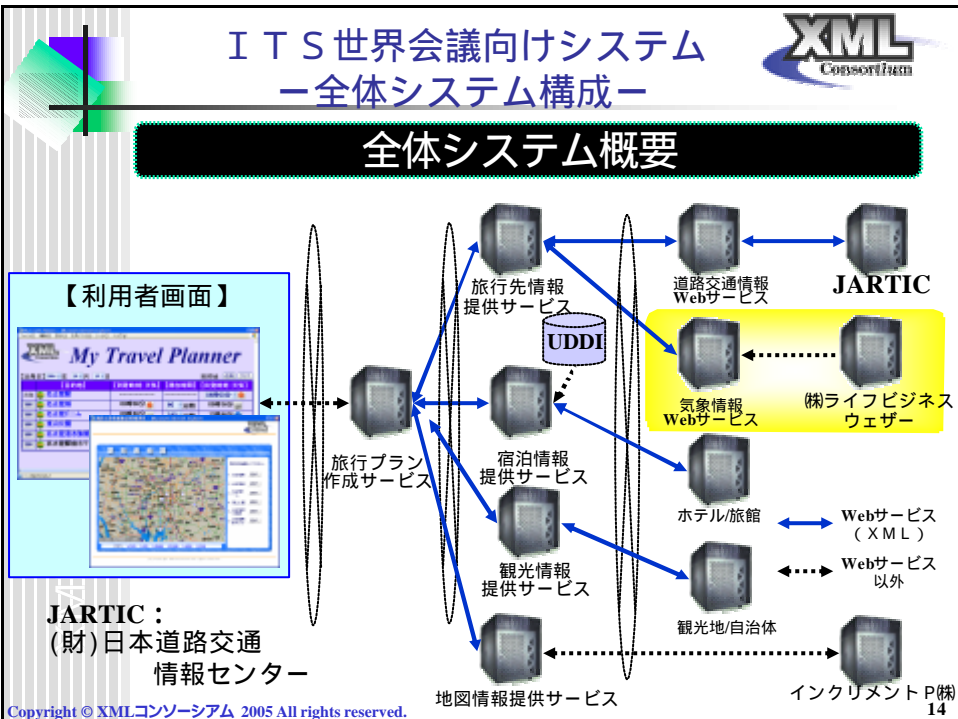
- ・ITS国際会議システムは、名古屋市内のみ16箇所
- ・愛・地球博システムにおいても、愛知県内89箇所であった為、大きなロスにはならないと判断した。

Copyright © XMLコンソーシアム 2005 All rights reserved.
12

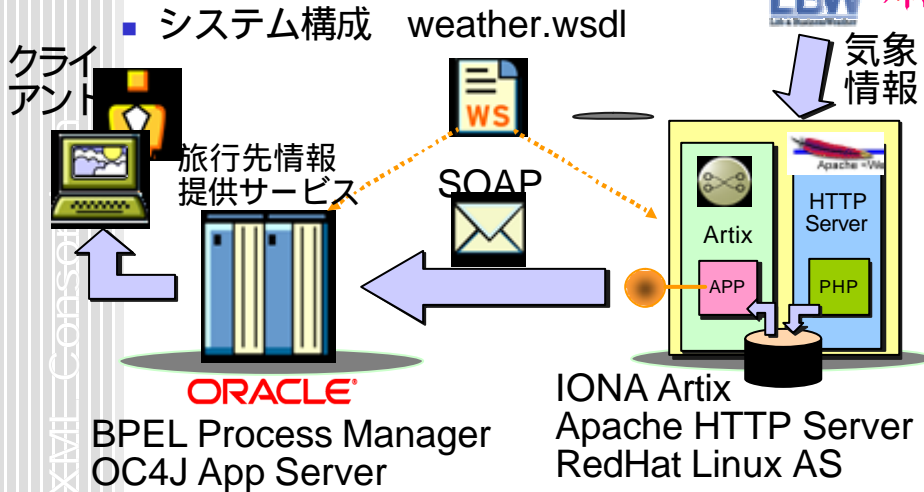


## I T S 世界会議2004向けシステム

日本アイオナテクノロジー(株) 江川 潔



ITS 世界会議向けシステム  
 - 気象情報Webサービスシステム構成 -



ITS 世界会議向けシステム  
 - Webサービス概要 -



リクエストの説明

```
<element name="requestWeatherElement">
  <complexType>
    <sequence>
      <element maxOccurs="1" minOccurs="1" name="dateTime"
        日時 : type="xsd:dateTime"/>
      <element maxOccurs="1" minOccurs="1" name="latitude"
        緯度 : type="xsd:int"/>
      <element maxOccurs="1" minOccurs="1" name="longitude"
        経度 : type="xsd:int"/>
    </sequence>
  </complexType>
```

レスポンスの説明

```
<element name="responseWeatherElement" type="xsd:string"/>
晴、曇、雨、強い雨、雪、強い雪、不明、障害」
スポット予報 (1時間天気)
```



気象情報の表示



気象情報は、(株)ライブビジネスウェザー様のサービスを利用しています  
(<http://www.wline.co.jp>)

■ 実装について

- C++Webサービス (Artix)
  - 高性能レスポンス
  - 軽量フットプリント
- 気象情報受信Apache HTTP Serverとの共存
- スポット1時間天気のみをメモリ上で展開
- 人間システム監視 (気象情報Updateの確認)
- WSDLのURL公開機能 (Artix 3.0で改善)
- OC4Jと相互運用の不具合 (Keep Alive)
  - <http-conf:server KeepAlive="Off"/>
- 気象情報
  - スポットのキーデータ (市区町村コード表)の変更
    - 市町村統廃合
  - 気象情報送受信障害 (サイズ0ファイル)



## I T S 世界会議向けシステム — 開発を終えて —



XML Consortium

- 各社のソリューションの中でArtixの位置づけを実感 (シンプルな操作性、相互運用への対応性)した。
- デモとして既存の気象情報から処理を簡素化 (近い場所のデータ)したが、実際はもっと気象のノウハウや経験データを導入すべきであり、サービス指向の中でサービスのレベルをあげていくことができる。
- C++Webサービスの高速性を考えて、データはオン・メモリとし、多重のWebサービス利用時のレスポンスを少しでも向上させる。
- インターネット接続環境の利用 (株式会社 ノムラシステム コーポレーション様)
- 愛・地球博へ向けての長期戦はユニシス様に交代

ありがとうございました。



# XML Consortium

【XMLコンソーシアムWeek 道路交通情報Webサービス実証実験Day】

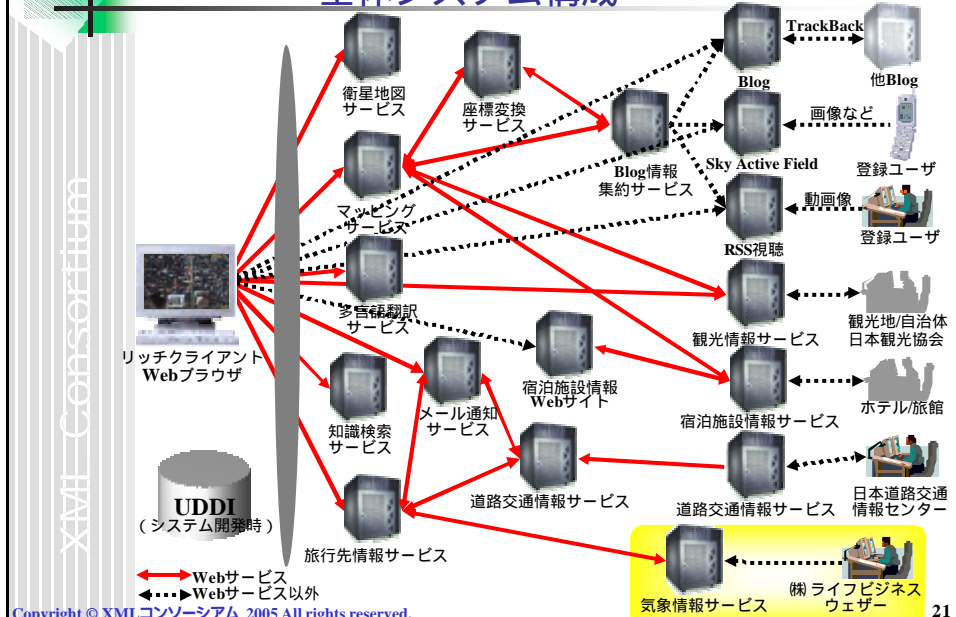
## 愛・地球博試験運用向けシステム

日本ユニシス(株)

鮫島 荘介



# 愛・地球博試験運用向けシステム —全体システム構成—

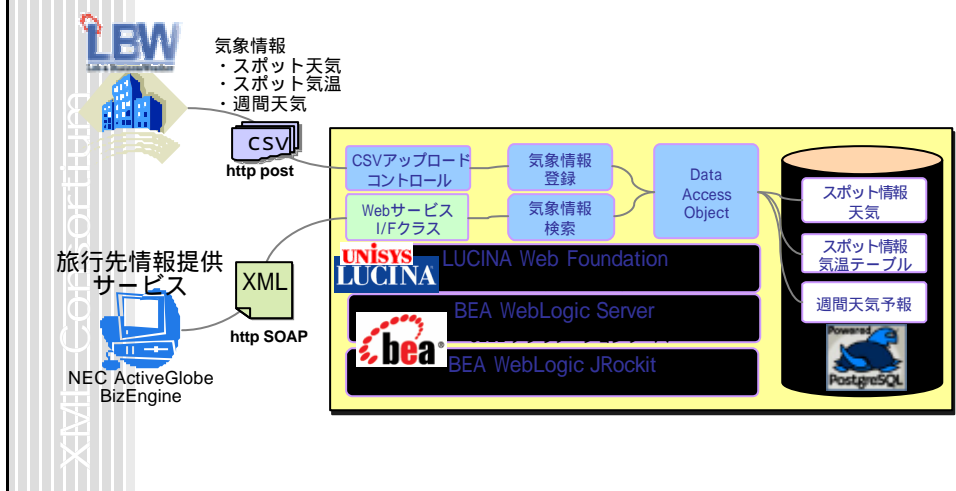


Copyright © XMLコンソーシアム 2005 All rights reserved.

# 愛・地球博試験運用向けシステム —気象情報Webサービスシステム構成—



## 気象情報Webサービスシステム構成



Copyright © XMLコンソーシアム 2005 All rights reserved.

## リクエストの説明

```

<getWeatherElement xmlns="http://iplat.xmlconsortium.org/TravelPlanner/ServiceProvider/WeatherInfoService">
  <weatherElementRequest>
    <departureDateTime>2005-06-07T15:30:14.000Z</departureDateTime>
    <arriveDateTime>2005-06-10T15:32:21.000Z</arriveDateTime>
    <departurePoint>
      <pointname>カナダ館</pointname>
      <coordinate>TKY</coordinate>
      <utmPoint>
        <x>3894385</x>
        <y>690415</y>
      </utmPoint>
      <point>
        <longitude>126343369</longitude>
        <latitude>32419930</latitude>
      </point>
    </departurePoint>
    <destinationPoint>
      <pointname>グローバルループから見たカンボジア館</pointname>
      <coordinate>TKY</coordinate>
      <utmPoint>
        <x>3894339</x>
        <y>689778</y>
      </utmPoint>
      <point>
        <longitude>126336915</longitude>
        <latitude>32419660</latitude>
      </point>
    </destinationPoint>
  </weatherElementRequest>
</getWeatherElement>
    
```

出発地  
位置情報

到着地  
位置情報

出発時間と到着時間

出発地  
緯度 (latitude)  
経度 (longitude)

出発時間と出発地座標、到着時間  
と到着地座標をリクエストとして  
受ける。

## レスポンスの説明

```

<getWeatherElementResponse xmlns="http://iplat.xmlconsortium.org/TravelPlanner/ServiceProvider/WeatherInfoService">
  <getWeatherElementReturn>
    <status>
      <statusCode>0</statusCode>
      <statusMessage>成功：出発地・目的地の予報データを返します</statusMessage>
    </status>
    <departureWeatherElement>
      <weather>Cloudy</weather>
      <temperature>
        <spotTemp>+19.2</spotTemp>
        <maxTempDay>NoData</maxTempDay>
        <minTempDay>NoData</minTempDay>
      </temperature>
    </departureWeatherElement>
    <destinationWeatherElement>
      <weather>CloudyWithRain</weather>
      <temperature>
        <spotTemp>+19.4</spotTemp>
        <maxTempDay>25</maxTempDay>
        <minTempDay>18</minTempDay>
      </temperature>
    </destinationWeatherElement>
  </getWeatherElementReturn>
</getWeatherElementResponse>
    
```

処理  
ステータス

出発地  
気象情報

到着地  
気象情報

天気情報

気温情報

## 気象情報の表示

出発地	到着地	到着時刻	出発時刻	出発時刻	出発時刻
ヒマカヤ美術館	18:31	天気: ☁	27.0	18:31	天気: ☁
青柳らいうら	18:51	天気: ☀	20.0	18:51	天気: ☀
小物市 歴史通り	18:57	天気: ☀	18.1	18:57	天気: ☀
ユースホテル愛知博覧会会場	02:04	天気: ☁	21.0		

## 利用するWebサービス実装と実装方針

WebLogicには複数のWebサービス実装【runtime&ツール】が搭載されている。

WebLogic Workshop ( Apache Beehive ) の実装

- 昨年度の検証案件 (TravelXML実証実験) で利用。
- 統合開発環境WebLogic Workshop の利用により、適当にマウスを右クリックして選んでいくだけで、Webサービスの作成が可能。  
WebLogic Workshopとは、開発環境でありランタイム(含むフレームワーク)でもある。
- ただしランタイムが含まれていることから、その他J2EEフレームワークとの共存が難しい。

WebLogic Serverの実装

- 個人的に使ったことが無いので使いたい。
- 弊社J2EEフレームワークLUCINA Web Foundation上に実装される業務コンポーネントを呼び出しできるか試したい。

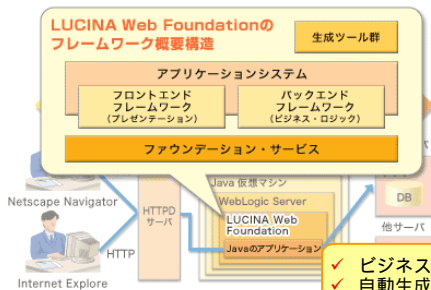
方針

WebLogic ServerのWebサービス実装でサービスを作成し、LUCINA Web Foundation上の業務コンポーネントを利用して実装する。

# 愛・地球博試験運用向けシステム —実装について—



## LUCINA Web Foundation



LUCINAWebFoundationは、以下の4部品でなるJ2EEフレームワーク製品。

- ▶ フロントエンドフレームワーク  
プレゼンテーション実装のベース
- ▶ バックエンドフレームワーク  
ビジネスロジック実装のベース
- ▶ ファウンデーションサービス  
システム全体で利用する基盤部品
- ▶ 生成ツール群  
Javaソースコード自動生成ツール

- ✓ ビジネスロジックのプレゼンテーションからの分離を強力に推進し、自動生成および基盤ライブラリにより生産性向上を図る

### 【主な機能】

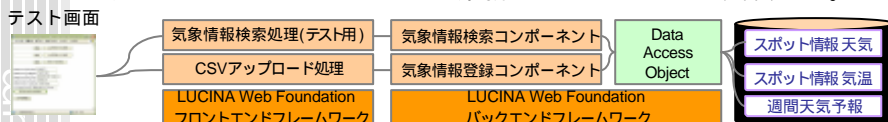
分類	機能名称	機能概要
フロントエンドフレームワーク	Front Controller	クライアントからの入力を受け付け、必要な業務モジュールやプレゼンテーションを起動する。
	Tag Library	JSP用のプレゼンテーションの基本的な部品群を提供。
バックエンドフレームワーク	Business Delegate	プレゼンテーションと業務ロジックを分離し、トランザクションを隠蔽。柔軟なシステム変更を実現。
	DAO Generator	DBのスキーマ情報からパフォーマンスを向上させたDAO(Data Access Object)のソースコードを自動生成する。
ファウンデーションサービス	Message管理	柔軟で統一したインターフェースのアプリケーション・メッセージの管理機構を提供。
	例外処理とロギング	アプリケーションシステム全体にわたる例外の処理機構を提供し、その記録を行う。

# 愛・地球博試験運用向けシステム —実装について—



## 業務コンポーネントの実装

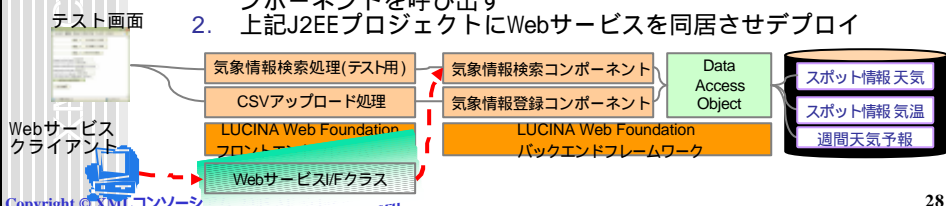
まずはJ2EEフレームワーク上に業務コンポーネントを実装した。



## Webサービスの実装

次に提供されたWSDLを元に、WebLogic Serverのツールを用いてWebサービスクラスを作成、デプロイする。

1. WSDLよりクラスを生成し、バックエンド上の気象情報検索コンポーネントを呼び出す
2. 上記J2EEプロジェクトにWebサービスを同居させデプロイ



## WSDLからのクラス生成

WSDLからのクラス生成ツールは、Antタスクとコマンドラインツールが提供されている。今回はAntタスクを利用して作成。

下記build.xmlを実行すると、以下のファイルが生成される。

- ✓XMLスキーマに定義されるデータ型クラス
- ✓WebサービスI/Fクラス（interfaceとそのimplementクラス）
- ✓web-service.xml（デプロイメントディスクリプタ）

```

<project name="buildwebservice" default="generate-from-WSDL">
  <target name="autotype">
    <autotype
      wsdl="file://weatherInfoService.wsdl"
      targetNamespace="http://iplat.xmlconsortium.org/iPlatXML"
      packageName="org.xmlconsortium.iplat.iplatxml"
      destDir="." /src"
      keepGenerated="True"
    />
    <classpath>
      <pathelement path="." />
      <pathelement path="C:\weblogic81\werve\lib\webservice.jar" />
    </classpath>
  </autotype>
</target>

<target name="generate-from-WSDL" depends="autotype">
  <wsdl2service
    wsdl="file://weatherInfoService.wsdl"
    destDir="." /src"
    generateImpl="true"
    typeMappingFile="." /src/types.xml"
    packageName="org.xmlconsortium.iplat.travelplanner.serviceprovider.weatherinfo.webservice"
  />
</target>
</project>

```

データ型クラス  
作成Antタスク

・WebサービスI/Fクラス  
・デプロイメントディスクリプタ  
作成Antタスク

## J2EE プロジェクトへのデプロイ

生成したクラスをsrc  
ディレクトリの下に  
コピー

生成したデプロイメント  
ディスクリプタを  
WEB-INFの下にコピー

フレームワークが自動生成するディレクトリ環境に、生成したファイルをコピーするだけで、既存のJ2EEプロジェクトにWebサービスを同居させることが出来た。

ear作成用Antターゲットにも変更無し。既存J2EEシステム上の業務コンポーネントを簡単にWebサービス化できた。



WebLogicには、EJBやJavaBeanをWebサービス化するツールも提供されているが、

- ✓ WSDLが別に決まっていればEJBのI/Fと一致しない
- ✓ フレームワークでBusiness Delegateパターン等を利用し、実I/Fが隠蔽されている

ようなケースであっても、簡単にWebサービス化が可能である。



# 愛・地球博試験運用向けシステム —開発を終えて—



XML Consortium

- iPlat プロジェクトについて
  - 様々なコンテンツを持つシステムをWebサービスでラップし、サービスとして外部解放することで、新たなサービスが導出されるという概念を、裏付けられたと思う。
    - コンテンツ提供というビジネスにおいて、SOAは身近なところにあると感じた。越えるべき壁はコンテンツ提供によるビジネスモデルの導出。その手順や方針も提示するコンサルテーションが必要なのだろう。
    - とはいえ今回は参照系のみ。密なるシステム間連携を、ミドルウェアの垣根を越えて疎結合に出来るのか。ミッションクリティカル度合いによっては、行わないべきラインが出てくるのではないか。そのような指針も会社として持ち、適宜お客様に対し説明できなければならない。
- WebLogic ServerのWebサービス実装について
  - LUCINA Web Foundationとの親和性が高い
    - 生成されたファイルを何も考えずにコピーするだけで、Webサービスがデプロイできる。
    - 既存J2EEシステムの業務コンポーネントをWebサービス化するという物理的手法が見えた。
  - WebLogic Workshopと比べるとやや実装が古い。今回相互に利用したWSDL、XSDでは以下の問題で一部修正を施した。WebLogic9に期待。
    - XMLSchemaでinclude、choiceを利用できない
  - RPC/Literalに対応していない?
    - Document/Literal、RPC/Encodedには対応している。
- 所感
  - 今年も大変でしたが、純粋に物作りをしているという実感があり、開発者として非常に楽しかった。
- 謝辞
  - 気象情報について、コンテンツ仕様の解説はもとより、送信テストや本資料のチェック等、ライフビジネスウェザー様に多大なご支援を頂きました。ありがとうございました。



## XML Consortium

【XMLコンソーシアムWeek 道路交通情報Webサービス実証実験Day】

# つづく

