Webサービス技術動向と システム構築における考察

2002-06-12

日本ユニシス(株) 牧野 友紀

アジェンダ

- 1.技術動向
 - * 相互接続性
 - **★ Webサービス・フレームワーク**
- 2.システム構築の考察

1.技術動向

Webサービス発展の歴史



SOAP1.1(W3C Note) 2000/5/8

UDDI Project発足2000/9/6

W3C XML Protocol WG発足2000/9/13

WSDL1.1 (W3C Note) 2001/3/15

XMLコンソーシアム本格活動開始 2001/4/1



>W3C Web Services WorkShop 2001/4/11-12

UDDI Business Registory運用開始 2001/5/2 XML Shema勧告2001/5/2

> WS-Inspection 2001/11/1 W3C Web Services Activity開始 2002/1/25

技術的な視点上のトピック

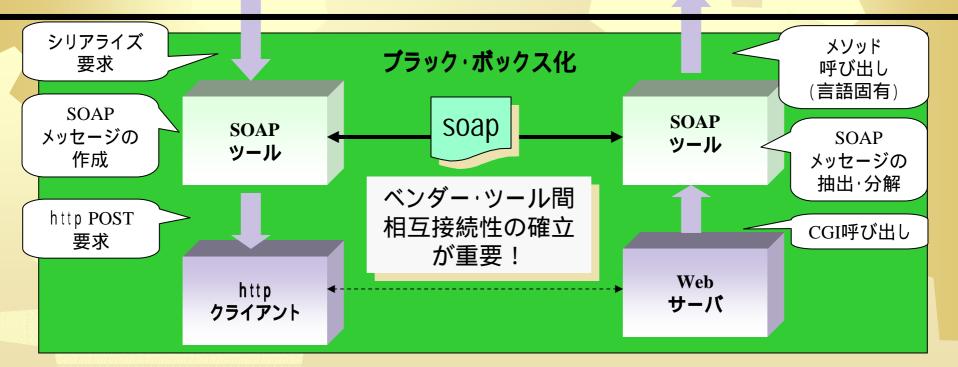
- 相互接続性
 - ★ Webサービス普及の鍵!
 - *SOAPツール·ベンダー間の人的コミュニケーション
 - Webサービス・フレームワーク
 - *現状の課題解決 = > 次世代Webサービス技術
 - ●実現のプロセス
 - フェーズ1:S/Wベンダーのアイデア,提案の表明
 - フェーズ2:標準化作業
 - フェーズ3:ツール、プロダクトの提供
 - 現状はフェーズ1が活発化し、フェーズ2が開始された状況



Webサービス実装

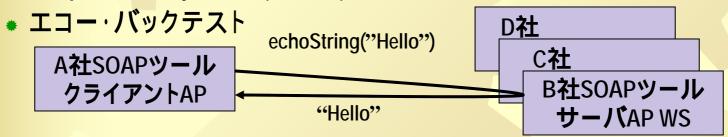
サービス利用 AP プロキシ オブジェクト リモート・オブジェクト 呼び出し

> サービス提供 A P



Webサービス 相互接続性

- SOAPBuilders
 - * IBM, Microsoft, Apache等主力ベンダーから個人ビルダーまで
 - Interoperability Lab(ILAB)



- * テスト結果(or x)マトリックスとメッセージ・ダンプをWebで公開
- ☀ Round1,2実施
 - SOAP1.1仕様書 5章 SOAP Encodingの各データ型送受信確認
 - string,integer,float,struct,dateTime,base64Binary string[],integer[],float[],struct[]
- * Round 3
 - 2002/2/27 ~ 2/28
 - IONA社 Waltham Officeにて開催
 - Document/Literalスタイルによる接続など

テスト結果一覧例

Apache Soap+Axis client Interop Results - Netscape 編集(E) 表示(V) ジャンブ(E) Communicator(C) ヘルブ(H)

Apache Soap 2.2+ and Apache Axis Client Interop Results

The source to the clients can be found at:

- Scap 22+
- Axis post-alpha3

For a definition of these tests, please see the SDAPBuilders Interoperability Lab "Round 2"

Questions or comments should be directed either to the SOAPBuilders Mailing list or to Sam Ruby

This test was run at: 2002-02-04T19:00:08Z

SOAP

	Float	Float	Struct	Struct	Void	Base64	Hex Binary	Date	Decimal	Boolean	Мар	Ar
OK C	OK.	OK.	OK.	OK	OK	OK.	FAIL	OK.	FAIL	OK	Fault	Faul
OK C	OK.	OK.	Fault	Frault	OK	FAIL	FAIL	OK	OK.	ОК	Fault	Faul
Fault B	Fault	Fault	Fault	Fault	Fault.	Fault	Fault	Fault	Fault	Fault	Fault	Faul
ок с	ок	ок.	ок	ок	ОК	ОК	ок	ОК	ок	ок	ок	ок
OK C	OK .	OK.	OK	OK.	OK	OK:	Fault	OK	OK.	OK	Fault	Faul
OK C	OK.	OK.	OK.	OK.	OK.	OK	OK.	OK	OK.	OK	Fault	Faul
OK IC	OK	OK.	OK.	OK.	OK	OK	OK.	OK	OK.	OK	Fault	Faul
OK IC	OK.	OK.	OK	OK.	OK	OK.	OK	OK	OK	OK	Fault	Faul
OK C	OK.	OK.	OK	OK	OK	OK	OK.	OK	OK.	OK	Fault	Faul
OK C	OK.	OK.	Fault	Fault	OK.	OK:	Exception	OK	Exception	Exception	Exception	Exce
OK IC	OK.	OK	OK:	Frault	OK:	Fault	OK.	OK	OK	OK	Fault	Faul
OK IC	OK.	OK.	OK.	OK.	OK.	OK	OK.	OK	OK	OK	OK.	OK
Fault E	Fault	Fault	Faun	Feet	OK	Fault	Fault	Fault	Fault	Fault	Fault.	Faul
ок с	ок	ок.	ок	ок	ок	Fault	ок	ОК	ок	ок	Fault	Faul
c	ж	ок ок	ок ок	ок ок ок	ок ок ок ок	ок ок ок ок ок	ок ок ок ок ок	ок ок ок ок выше ок	ок ок ок ок ок ок			OK OK OK OK OK Fault OK OK OK Fault

WS - Lorg

- Webサービス相互運用性確保を推進するコンソーシアム
 - ★ 2002/2/6 Microsoft,IBMが中心に設立
 - ◆ 参加企業 富士通,BEA,Oracle,HP,SAP,Intelなど
 - 実装ガイド,テスト・ガイド(テスト仕様,ツール)提供
 - 相互運用性の推進
 - ◆ プラットフォーム、アプリケーション、プログラミング言語非依存
 - → 一貫性(consistent),信頼性(reliable)
- *エンド·ユーザのWebサービス構築推進
 - ◆ 構築,配置に対するガイド,ツールの提供

2つのSOAPバインディング・スタイル

- · WSDL1.1仕様内の選択肢
 - ◆ Binding要素で指定
- RPC/Encodedスタイル
 - **★ SOAP1.1 仕様のRPCモデル**
 - SOAPエンコーディング・ルールを使用
 - httpによる転送を想定
 - Document/Literalスタイル
 - ◆ 文書(DOMオブジェクト)の転送が可能
 - 整形式XMLの転送も可
 - * 今後の主流
 - * SOAPツール・キット間の相互接続性確保が大変
- 通信相手がどちらを支援するか意識する必要あり

WSDLの相違

RPC/Encoded

```
<binding name="Service1Soap" type="s1:Service1Soap">
  <soap:binding
     transport="http://schemas.xmlsoap.org/soap/http"
     style="rpc" />
  <operation name="rtnIntArray">
    <soap:operation</pre>
      soapAction="http://tempuri.org/rtnIntArray"
      style="rpc" />
    <input>
      <soap:body use="encoded"</pre>
         namespace="http://tempuri.org/"
         encodingStyle=
         "http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded"</pre>
        namespace="http://tempuri.org/"
        encodingStyle
        ="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </binding>
```

Document/Literal

```
<binding name="Service1Soap"type="s0:Service1Soap">
 <soap:binding
    transport="http://schemas.xmlsoap.org/soap/http"
    style="document" />
 <operation name="rtnIntArray">
  <soap:operation</pre>
     soapAction="http://tempuri.org/rtnIntArray"
     style="document" />
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
   <soap:body use="literal" />
  </output>
 </binding>
```

SOAPメッセージの相違

RPC/Encoded

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:tns="http://tempuri.org/"
     xmlns:types="http://tempuri.org/encodedTypes"
     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body soap:encodingStyle
     ="http://schemas.xmlsoap.org/soap/encoding/">
  <tns:rtnIntArrayResponse>
    <soapenc:Array soapenc:arrayType="xsd:int[2]">
      <ltem>int</ltem>
      <ltem>int</ltem>
    </soapenc:Array>
  /tns:rtnIntArrayResponse>
</soap:Body>
</soap:Envelope>
                             メッセージ層にて
                            データ型を評価する
```

Document/Literal

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <rtnIntArrayResponse xmIns="http://tempuri.org/">
   <rtnIntArrayResult>
    <int>int</int>
    <int>int</int>
   </rtnIntArrayResult>
  </rtnIntArrayResponse>
 </soap:Body>
</soap:Envelope>
                           メッセージ層にて
                        データ型を評価しない
```

public int[] rtnIntArray{
// int型arrayを戻すメソッド



Webサービス発展の歴史



SOAP1.1(W3C Note) 2000/5/8

UDDI Project 発足2000/9/6

W3C XML Protocol WG発足2000/9/13

WSDL1.1 (W3C Note) 2001/3/15

XMLコンソーシアム本格活動開始 2001/4/1



>W3C Web Services WorkShop 2001/4/11-12

UDDI Business Registory運用開始 2001/5/2 XML Shema勧告2001/5/2

> WS-Inspection 2001/11/1 W3C Web Services Activity開始 2002/1/25

Webサービスの課題

- * 相互接続性の確立
 - * 各ソフトウェア・ベンダーが提供するツール間の相互接続性 (SOAPメッセージ送受信)の確立・維持
 - * 米国SOAPBuilders, WS-I.org, 分散オブジェクト協議会(計画中)
- セキュリティ
 - * 暗号化,認証,電子署名,アクセス制御,権限委譲
 - * 利用ノウハウの共有,ベスト・プラクティス,典型的な実装モデル(ベスト・プラクティス)の確立が必要
- サービス品質(Webサービス自体)維持
 - ★ メッセージ転送の信頼性(再送,二重送信防止)
 - ◆ 障害回避や負荷分散のための経路変更(ルーティング)

XML Webサーヒスの課題(つつき)

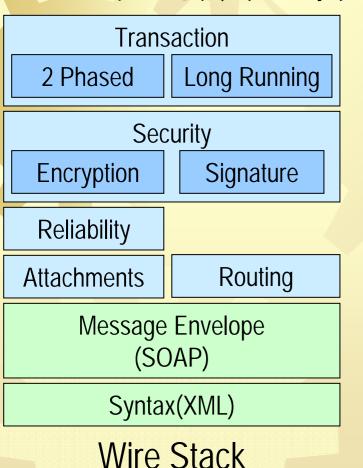
- トランザクション保証
 - 更新処理を行う複数Webサービスを連携する際のトランザクションの保証
 - * ACID属性の実現方法
- ●商用Webサービスを実現する技術の確立
 - *ライセンスの付与,契約照合の方法
 - 利用頻度の計量
 - **★課金およびライセンス処理と対象処理の分離**

W3C Webサービス・アクティビテイ

- 2002/1/25 スタート(3実働WG+調整役WG)
- Cordination Group
 - *全体の方向付け, WG間調整など, アクティビティ運営母体
 - Web Services Architecture WG
 - ★ WebサービスFrameworkの構築
 - 2001/4/11-12 Webサービス ワークショップで討議された機能要件が素 (http://www.w3.org/2001/04/wsws-proceedings/wrapup2.html)
 - ☀ 活動期間2年,2002年中頃勧告予定
- Web Services Description WG
 - ☀ 活動期間2年,2003年5月勧告予定
 - **★ WSDL1.1をベースに活動**
- XML Protocol WG(既存)
 - ☀ 活動期間1.5年,2002年中頃勧告予定

Webサービス アーキテクチャ

- 機能(技術)をモジュール化
- 各機能はXMLベース
- Webサービス・アーキテクチャはWebアーキテクチャ内に統合



W3C Web Services Workshop (2001/4/11-12) Web Services Frameworkセッション(IBM, Micorosoft)より

Process Flow Orchestration

Service Description (WSDL)

Structure (XMLSchema)

Description Stack

Service Registry (UDDI)

Inspection

Discovery Stack

フレームワーク実現に向けて

ħ	幾能	規格	団体/企業	説明
Reli	ability	httpr	IBM	httpプロトコルの上に,信頼性のあるメッセージ転送プロトコルを実現
		Reliable Message (GXA)	Microsoft	マイクロソフトが提案する次世代Webサービス体系GXA(Global XML Architecture)中の信頼性メッセージ機能
Sec	urity	XML Signature	W3C	XML文書に対する電子署名の付与
		XML Encryption	W3C	XML文書の暗号化
		SOAP Signature	W3C:Note	SOAPメッセージに含めた電子署名つきXMLデータに関連する情報の記述
		SAML	OASIS	Webサービスの内部処理で別のWebサービスを 呼び出すようなWebサービスの連鎖を想定した権 限委譲の方法
		XACML	OASIS	XML文書中のノードに対するアクセス制御。(例) 見積もり書XMLの金額に関連するノードは営業部 門のみ閲覧可能

フレームワーク実現に向けて(2)

機能	規格	団体/企業	説明
Security	ecurity XKMS W		PKIにおける証明書の発行、検査などの処理インタフェースのWebサービス化
	WS-Security IBM Microsoft VeriSign		SOAPメッセージの暗号化 , 完全性保証の機能。 SOAPの拡張を前提に , PKI , Kerberosに対応。
			セキュリティ・ポリシー(制限事項,可能事項)の記述方法の定義
	WS-Trust	同上	サービス利用者,サービス提供者(中間サービス提供者を含む)間の信頼関係を確立のモデル
	WS-Privacy 同上		Webサービスのプライバシー保護の実装方法と、 実行時の利用者に対するプライバシーに関連す る情報の提示方法
	WS-Secure 同上 Conversation		Webサービス提供者(中間提供者も含む),利用者間のメッセージ交換の管理,認証の方法。セキュリティ関連情報の交換方法や,セッション・キーの確立,配布方法も含まれる。

フレームワーク実現に向けて(3)

機能	規格	団体/企業	説明
Security	ecurity WS-Federation IB M		異なるセキュリティ体系(SSOシングル・サイン・オン)間での信頼関係の維持管理や、取りまとめ
	WS- Authorization	同上	Webサービスの権限データ,ポリシーの取扱い方法
Routing	WS-Routing WS-Referral (GXA)	Microsoft	マイクロソフトが提案する次世代Webサービス体系GXA(Global XML Architecture)中のSOAP メッセージの経路選択機能
Attachme nt	SOAP Attachment	W3C:Note	SOAPメッセージ交換における添付データ(バイナリ)の交換方法
Transaction	BTP	OASIS	<business protocol="" transaction="">トランザクション処理を持つ複数Webサービスを利用する場合のトランザクション連携。2フェーズ・コミットに基づく厳格なトランザクション・グループ(Atom)の組みあせ(Cohesion)による緩やかなトランザクション。一つのAtomのトランザクションが失敗しても残りのAtomの処理を完結することができる。</business>

フレームワーク実現に向けて(4)

機能	規格	団体/企業	説明
WFL	WSFL	IBM	Web Services Flow Language複数Webサービスの統合を柱にしたフロー言語
	XLANG	Microsoft	XML-EDIミドルウェアBizTalk Serverにおける受信メッセージの内部処理制御フローの記述が発端
	WSCL	HP	<web conversation="" language="" services=""> 軽量なフロー記述言語</web>
Inspection	WS- INSPECTION	IBM Microsoft	Webサービス提供サイトで公開するWebサービスのディレクトリ。それぞれの一元管理するUDDIビジネス・レジストリに対して、各Webサービス提供サイトに配置される分散型。
Provisioni ng	WSHT	IBM	< Web Services Hosting Technology > 有料課金Webサービスの運用も範囲に含めた, Webサービスのプロビジョニング体系
	SPML	OASIS	<service languageweb="" markup="" provisioning="">サービスの提供環境(プロビジョニング)に対する処理要求(Webサービスのサービス提供要求)と問合せ機能の標準化</service>

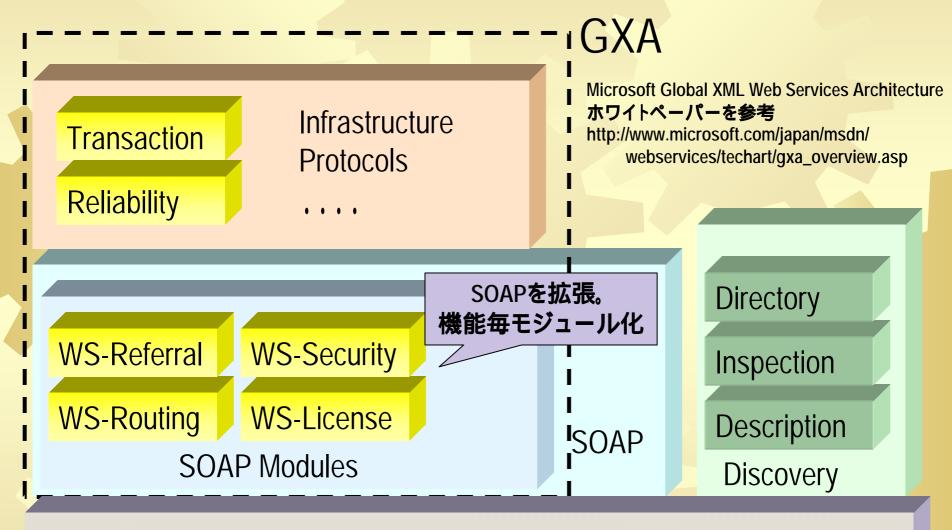
フレームワーク実現に向けて(5)

Applicatio n	WSIA	OASIS	<web application="" for="" interactive="" services=""> 対話型のWebアプリケーションの構築に, Web サービスをコンポーネントとして組み合わせるため のフレームワーク</web>
Applicatio n	WSRP		<web for="" portals="" remote="" services="">ポータルWebサイトにおいて、Webサービス技術を用い分散する動的コンテンツを簡易に集約(プラグアンドプレイ)できるようにする仕組。WSIAのフレームワークに対応。</web>

Webサービスのセキュリティ



Microsoft GXA

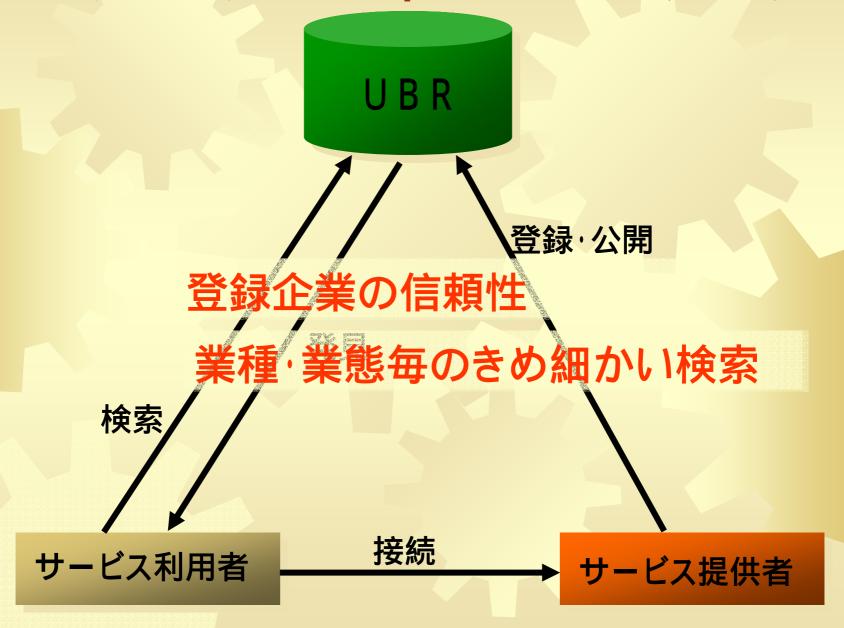


The Internet

2.システム構築の考察

UDDIレジストリの利用

UDDIレジストリ基本アーキテクチャ



レジストラ・サービス・アーキテクチャ

UBR

情報の取得

*独自分類による企業情報

- ◆製品情報(JANコード等)
- *サービス情報
- *地理情報
- *与信情報

サービスの品質保証 豊富な索引を利用した検索

サービス利用者

公開:Checked Taxnomy機能利用

ジストラ

加情報

接続

信頼の獲得 簡便な登録作業

会員登録

ビス登録

サービス提供者

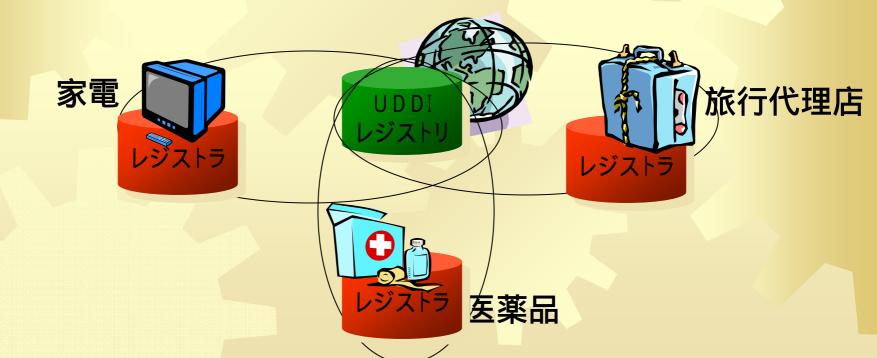
レジストリ・サービスの実現

コンセプト: UDDIビジネス・レジストリの補完

- UBRを骨格として付加機能を肉付ける

● 範囲:業種,業態,地域毎

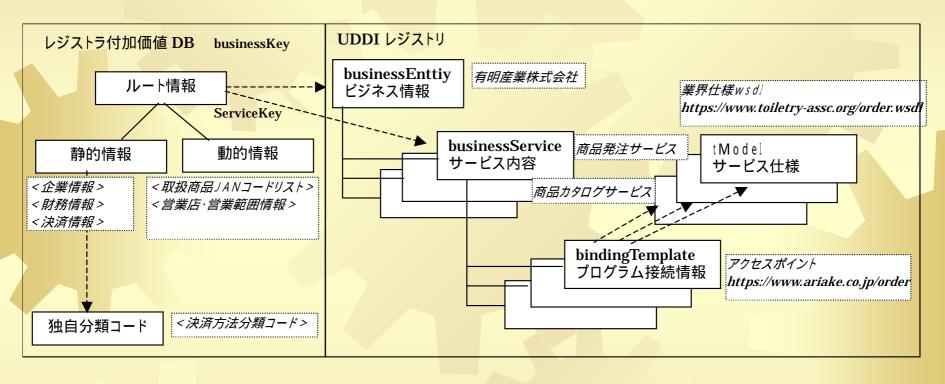
担い手:業界協会、第三者機関(企業)



レジストラ・サービス機能例

- UBRにない付加情報を保持
 - * 独自分類による企業情報
 - 取扱製品、サービス情報 (ex.料金(課金)体系,運用時間)
 - ☀ 製品カタログ
 - ★ 地理情報(ex.営業店所在地,営業範囲)
- 登録企業の与信機能
 - 登録基準によるサービス提供者の適正認定
 - 企業情報提供(事業内容,財務状況,関連企業)
 - UDDI Checked Taxnomy機能の運用
- * 豊富な索引機能
 - 付加情報による検索(ex.JANコードによる対象商品取扱ベンダーの発見)
- UBRへの代理登録機能

レジストラとUBRのデータ構造

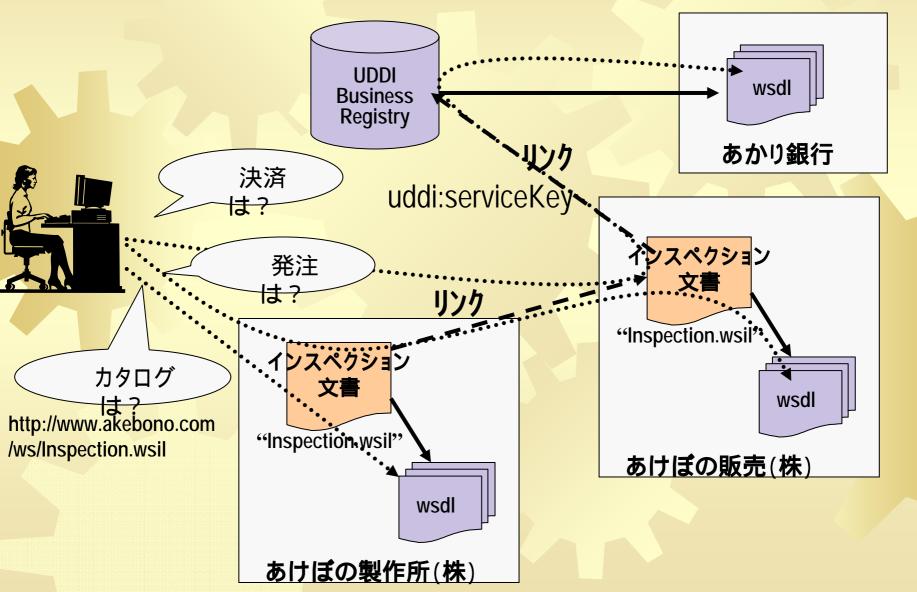


レジストラ付加価値DBの持つ属性が検索キーとなる (例) 売上高100億を越しJANコードxxxxxxxxの製品を扱うベンダーは?

WS-Inspection

- Microsoft,IBM共同提案(2001/11/1発表)
- 分散型ディレクトリ
 - ★ HTMLのWebサイトにおけるサイト・マップと比類
- シンプルな仕様
 - ★ 人手で容易に編集可能なXMLファイル (インスペクション・ファイル)
 - 運用はWeb仮想ディレクトリ配置のみ
- ・リンク,参照機能
 - インスペクション・ファイル-インスペクション・ファイル間
 - **★ UDDIエントリの参照**

WS-Inspection 利用形態



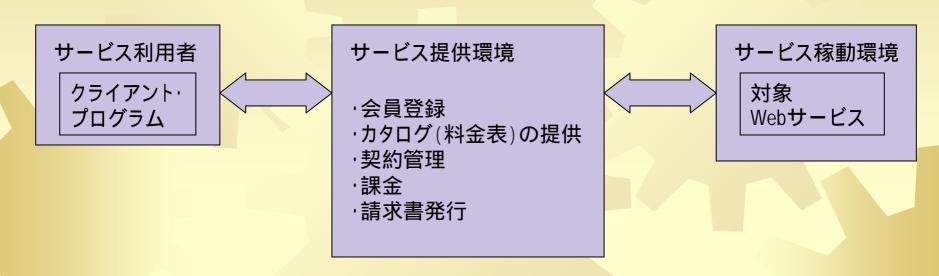
Inspectionファイル例

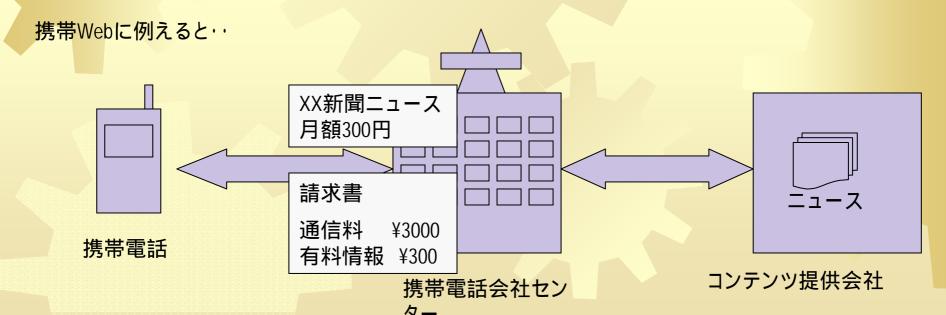
```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/wsil/"</pre>
       xmlns:wsiluddi="http://schemas.xmlsoap.org/wsil/uddi/">
 <service>
  <abstract>あけぼの商品受注サービス</abstract>
  <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"</pre>
          location="http://www.Akebono-Hanbai.co.jp/OrderService.wsdl"/>
 </service>
 <service>
  <abstract>あけぼの銀行(主要取引銀行)決済サービス</abstract>
  <description referencedNamespace="urn:uddi-org:api">
    <wsiluddi:serviceDescription</pre>
          location="http://uddi.SomeOperator.com/uddi/inquiryapi">
     <wsiluddi:serviceKey>
           4FA28580-5C39-11D5-9FCF-BB3200333F79
     </wsiluddi:serviceKey>
    </wsiluddi:serviceDescription>
  </description> </service>
 <link referencedNamespace="http://schemas.xmlsoap.org/wsil/"</pre>
    location="http://www.Akebono.co.jp/inspection.wsil"/>
</inspection>
```

Webサービス・プロビジョニング

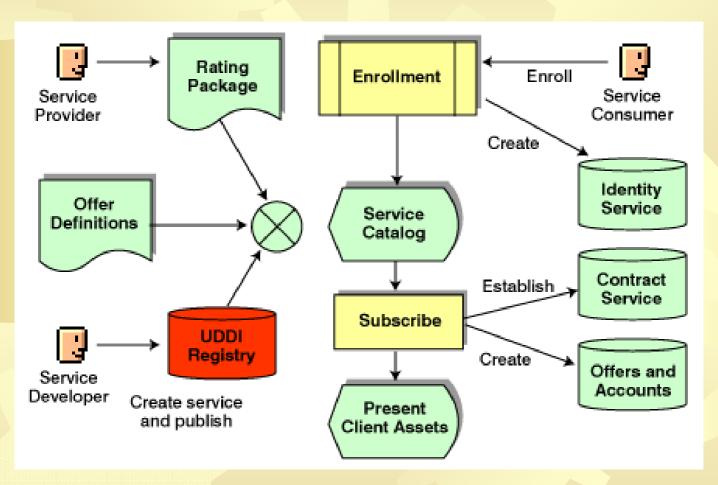
- → プロビジョニング : Webサービス提供環境へのWebサービス配置
- 三つの役割
 - ★ Webサービス所有者
 - ★ Webサービス提供者
 - * Webサービス利用者
- 三つのフェーズ
 - Webサービスの配置(UDDI登録を含む): 所有者 > 提供者
 - * Webサービスの契約:利用者 > 提供者
 - * Webサービスの運用:利用者>提供者>所有者
- Webサービス自身の配置,契約,運用からの分離
- 有料課金モデルを想定

プロビジョニング概念



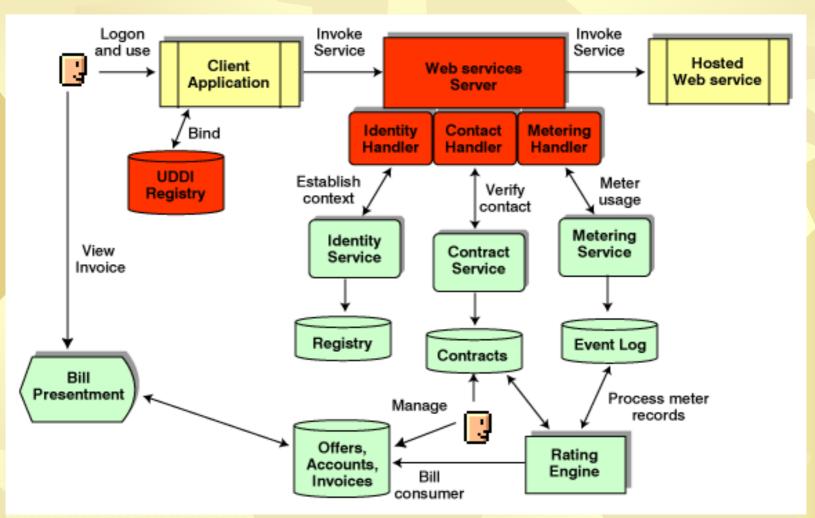


IBM WSHT (登録, 契約プロセス)



IBM developerWorks:"Web services provisioning" 2002/1 Mike Polan氏著より http://www-106.ibm.com/developerworks/webservices/library/ws-wsht/

IBM WSHT 続き(稼動プロセス)

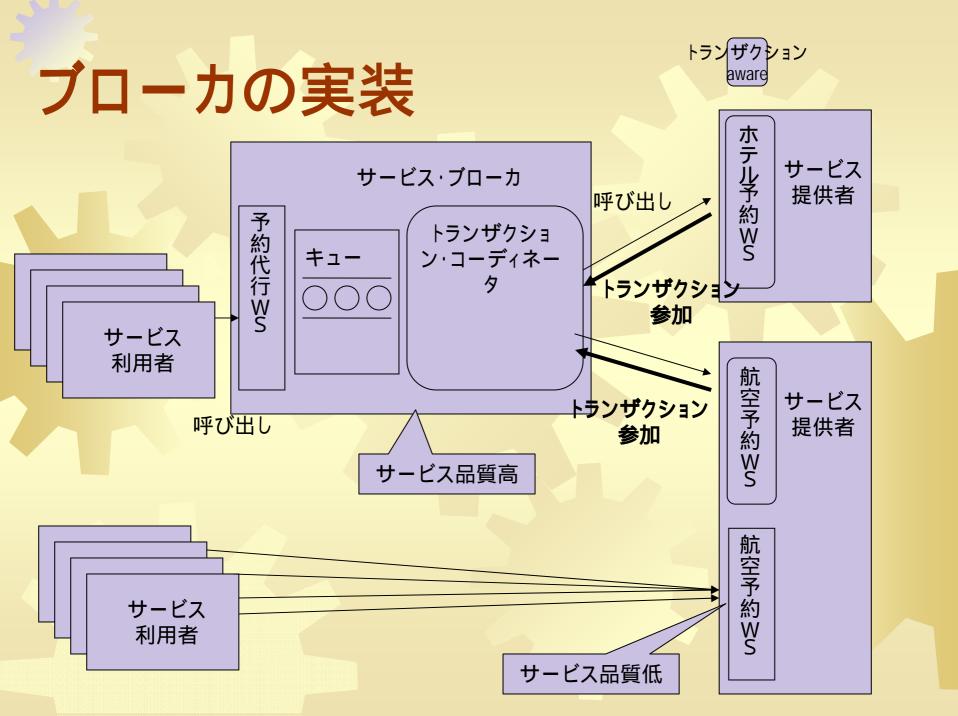


IBM developerWorks:"Web services provisioning" 2002/1 Mike Polan氏著より http://www-106.ibm.com/developerworks/webservices/library/ws-wsht/



Webサービス・ブローカ必要性

- サービス提供者にとって
 - 業務:販売チャネルの拡大
 - *技術:トランザクション連携相手の特定化
 - サービス要求のフィルタリング
 - * 流量制限
- サービス利用者にとって
 - 業務:購買代行
 - * 技術: 複数トランザクション集約の代行
 - * : トランザクション・コーディネータ実装が不要
 - * :エージェント機能

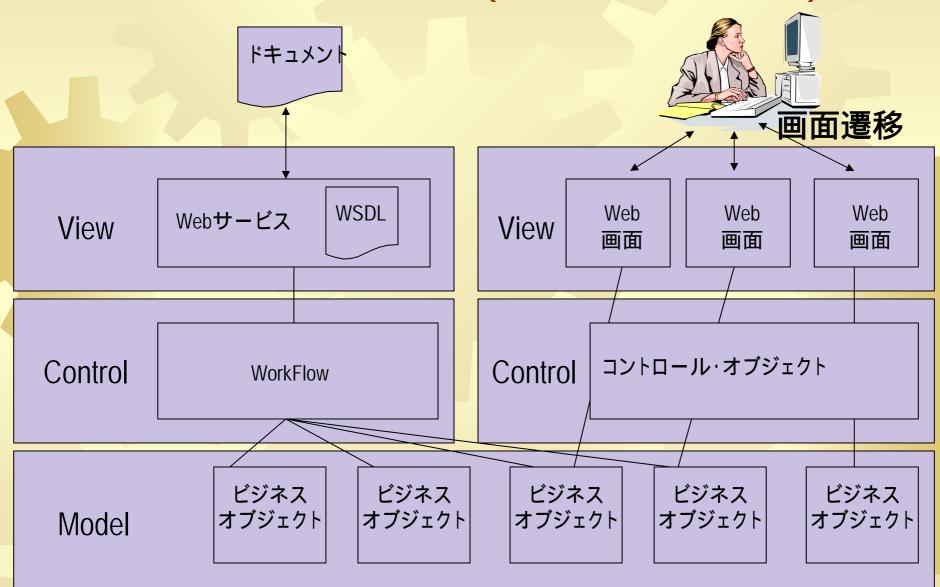




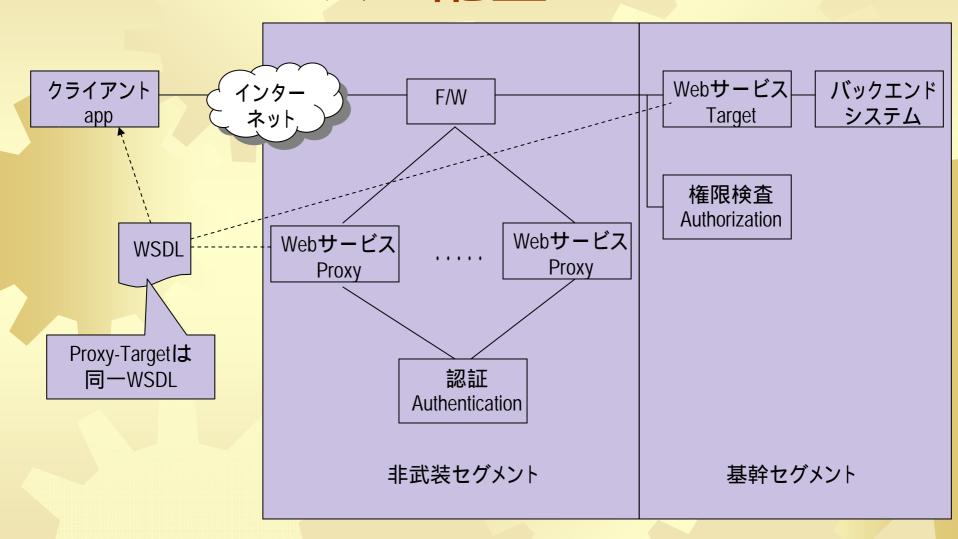
Webサービスの設計ポイント

- * 論理設計モデル分析課程で1オブジェクトとして 出現
- 疎結合分析で適用オブジェクトの抽出
 - ☀管理主体(人)境界
 - **★**プラットフォーム,アプリケーション,言語境界
- サービスの粒度
 - ★ ネットワーク遅延の考慮
 - 既存システムのオブジェクト粒度と異なる
 - * DOA(Data Oriented Approach)の適用
 - *業務視点で分割できないデータまとまりを入出力
- 既存システムとビジネス・ロジックを共有

既存システム共存(MVCモデル)



Webサービスの配置



おわりに

- Webサービスは既に実用段階
 - Webサービス課題解決を待つのは得策でない
 - Webサービスの評価は机上では困難
 - プロトタイピングによる考察が必須