

グリッド協議会 資料

Webサービスの技術動向の紹介

2002年12月19日
NECソリューションズ
インターネット基盤開発本部
藤田 悟

内容

1. Webサービスとは
2. Webサービスを支える技術
3. Webサービスの技術課題
4. NEC の製品群
5. まとめ
6. デモの説明



1. Webサービスとは



Webサービスの2つの定義

- サービス指向に基づいた動的なビジネス統合基盤
 - IBMのWebサービスの定義
 - A Web service is a software component described in WSDL.
 - インターフェイス定義が重要。
 - SOAP、EJB、CORBAなど、実装は問わない
 - SOAPによる連携システム
 - SOAP を用いることで
 - HTTP による firewall 越えが容易
 - XML によるハードウェア/環境に非依存

サービス指向アーキテクチャ

The diagram illustrates the progression of architectural styles. At the top, 'ビジネス設計' (Business Design) is represented by a group of business professionals. An arrow points down to 'サービス指向' (Service-oriented), which is depicted with icons of a storefront, a delivery truck, and a person with a briefcase. A second arrow points down to 'オブジェクト指向' (Object-oriented), shown with a class diagram featuring boxes, diamonds, and lines representing relationships.

- ビジネス設計にとって、オブジェクト指向では粒度が細かすぎる。
 - 粒度の粗い設計
 - プログラムではなく、フロー中心の設計
- ビジネスはサービスの中身が重要。
 - パートナーとは、実行時に動的に連携
 - サービスを検索/選択

Copyright(c) 2002, NEC Corp. All rights reserved. 5

SOAPによるシステム連携

SOAP \cong HTTP + XML

- インターネットを経由した疎結合のシステム連携
- 社内部門間のドメインを越えたシステム連携

The diagram shows a company's internal network ('企業内') on the left, containing a '本部センター' (Main Center) with server racks and several workstations. Blue arrows labeled 'SOAP' indicate communication between the center and the workstations. On the right, a cloud labeled 'SOAP' is connected to the internet ('インターネット接続'), which in turn connects to external systems represented by computer icons. Bidirectional arrows show the flow of data between the internal network and the external systems via the internet.

メリット

- インフラを既存資産である Web と共有
- 新たなネットワーク設定/管理が不要
- 機種依存性を排除

Copyright(c) 2002, NEC Corp. All rights reserved. 6

2. Webサービスを支える技術

Webサービスの基本モデル

Webサービス上の役割を、サービスの利用者、提供者、ブローカに区分して、3者間のサービスに対する関係を規定する。

[提供者⇄ブローカ]

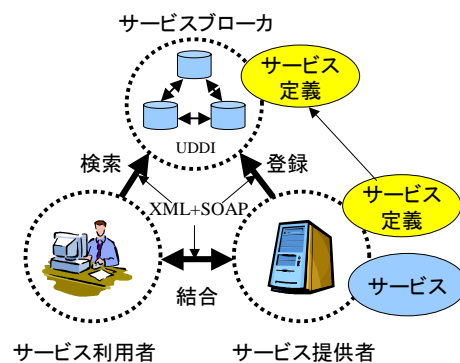
- 提供するサービスの定義を、提供者がブローカに**登録**する。

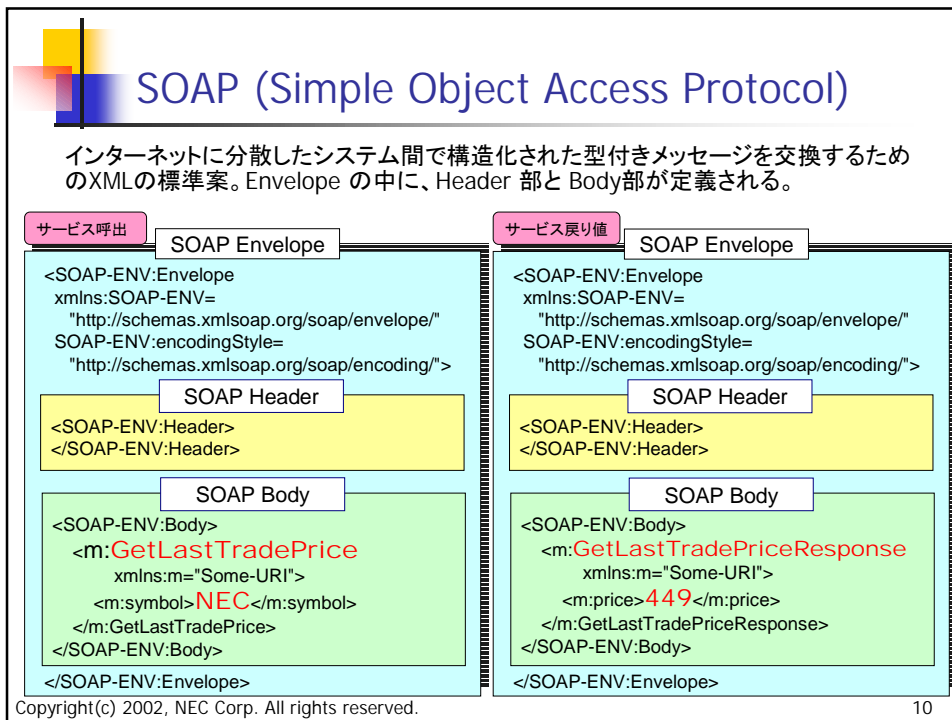
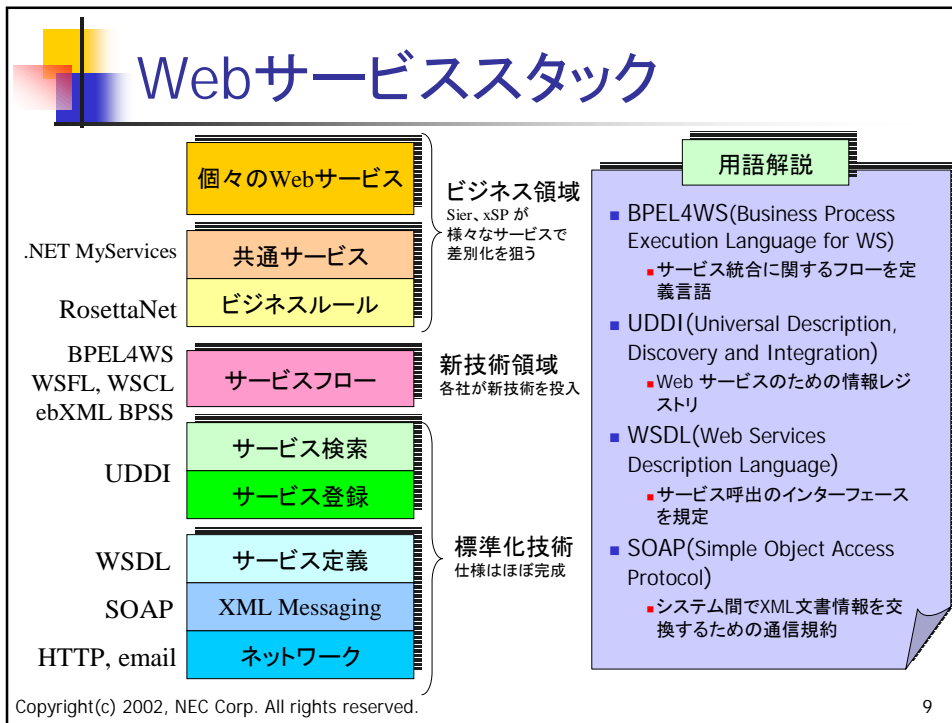
[利用者⇄ブローカ]

- 利用者は、ブローカに対してサービス定義に関する**検索**を行う。

[利用者⇄提供者]

- 利用者は、提供者との間で直接サービスの**結合**(実行)を行う。

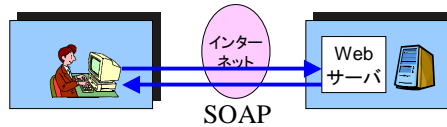




SOAP を用いた2つのメッセージ交換手法

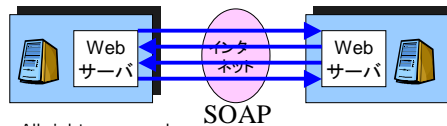
■ SOAP RPC

- サービスコールとレスポンスの組み合わせで、1つのタスクを実現
 - Web サーバは、サービス提供者側だけに存在



■ SOAP Messaging

- (論理的に)一方向のメッセージを複数やり取りすることで、1つのタスクを実現
 - Web サーバは、両者共に保有
- ビジネスプロトコルを制御するレイヤーが SOAP 上に必要 (例: ebXML Messaging Service)



Copyright(c) 2002, NEC Corp. All rights reserved.

11

SOAP RPC の仕様

- 型付きのメソッドコールの仕様
- 基本データ型(XML Schema に準拠)
 - byte, short, int, long, float, double, string
- 複合データ型
 - 配列(SOAP Encoding で規定)
 - 構造体(XML Schema を利用して表現)
 - 複数から参照される構造体は、href でリンクして表現する
- 引数/戻り値
 - 引数は、in パラメタと、in/out パラメタがある
 - 戻り値は、out パラメタと、in/out パラメタがある
- 例外処理
 - SOAP Fault によるエラー送信

Copyright(c) 2002, NEC Corp. All rights reserved.

12

SOAPの配列と構造体例

■配列宣言

```
<element name="myFavoriteNumbers"  
  type="SOAP-ENC:Array"/>
```

■配列の送信時の表現

```
<myFavoriteNumbers SOAP-ENC:arrayType="xsd:int[2]">  
  <number>3</number>  
  <number>4</number>  
</myFavoriteNumbers>
```

■複数参照される構造体の表現

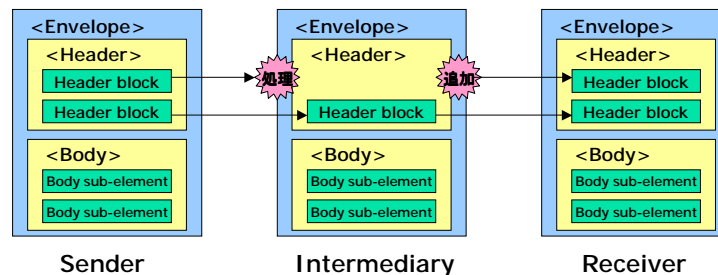
```
<e:Book>  
  <title>My Life and Work</title>  
  <author href="#Person-1"/>  
</e:Book>  
<e:Person id="Person-1">  
  <name>Henry Ford</name>  
  <address href="#Address-2"/>  
</e:Person>  
<e:Address id="Address-2">  
  <email>mailto:henryford@hotmail.com</email>  
  <web>http://www.henryford.com</web>  
</e:Address>
```

Copyright(c) 2002, NEC Corp. All rights reserved.

13

SOAP 1.2 (draft)

- W3C の XML Protocol の WGで協議中
- SOAP 1.1 との互換性を重視しながら、仕様の曖昧性の排除と、機能拡張を試みる。
 - マルチホップした時の処理モデルの定義
 - Header 部分に、トランザクションやコンテキスト管理情報を載せることを示唆。
 - 例外処理に関するコードの充実、定義の明確化



Copyright(c) 2002, NEC Corp. All rights reserved.

14

WSDL (Web Services Description Language)

WSDLは、サービスのエンドポイント間で交わされる手続き呼出のインターフェースとプロトコルバインディングを規定するためのXMLフォーマット。

```

<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd1="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns="http://www.w3.org/1999/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="symbol" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="TradePriceResult">
        <complexType>
          <all>
            <element name="price" type="float"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>

```

```

<message name="GetLastTradePriceInput">
  <part name="GetLastTradePrice" element="xsd1:TradePr
</message>
<message name="GetLastTradePriceOutput">
  <part name="GetLastTradePriceResponse" element="xsd1:TradePriceResult"/>
</message>
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePrice"/>
    <output message="tns:GetLastTradePriceResponse"/>
  </operation>
</portType>
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>

```

データ型の定義

メッセージの定義

操作の定義

プロトコルバインディング

Copyright(c) 2002, NEC Corp. All rights reserved. 5

WSDLを利用したシステム開発

- サービスプログラムがある場合
 - 例えば、Java のインタフェース定義から、WSDLを自動生成
 - そのWSDLからSOAPのクライアントプログラム用にプロキシクラスを自動生成
 - サービスプログラムをSOAPを持つAPサーバに登録して完了。

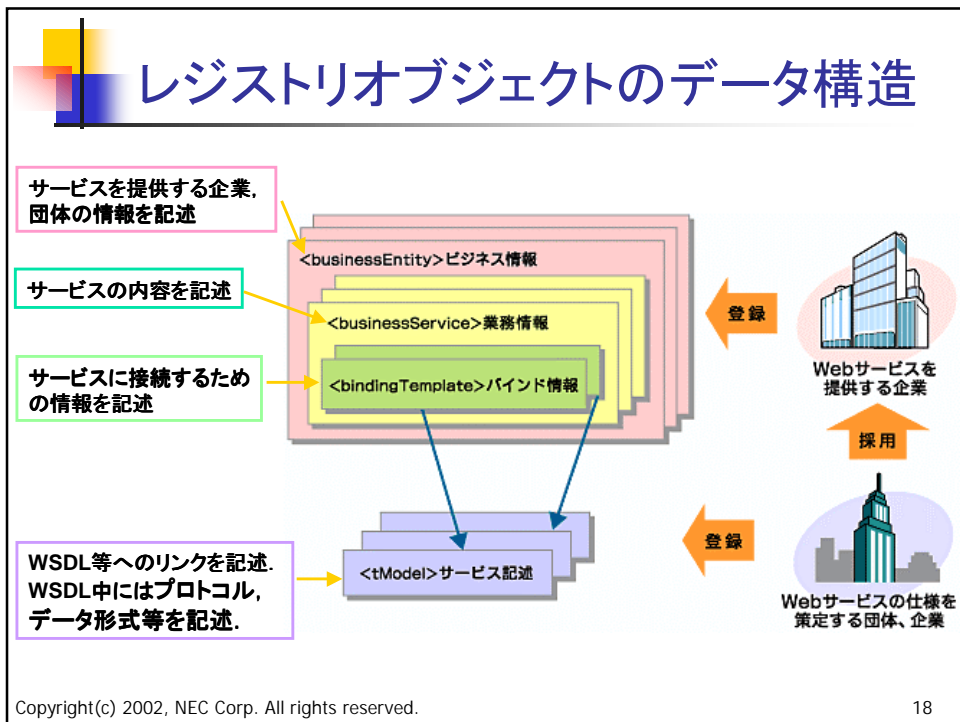
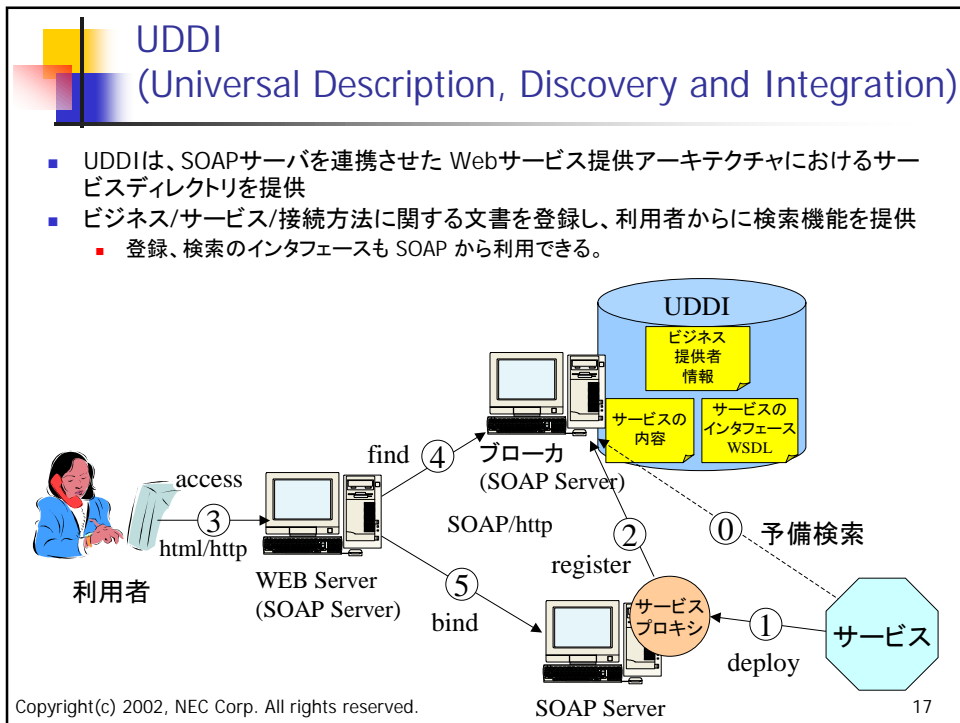
```

graph TD
    A[Java サービスプログラム] -- 登録 --> B[APサーバ]
    A -- 自動生成 --> C[Java インタフェース]
    C -- 自動生成 --> D[WSDL 定義]
    D -- 自動生成 --> E[Proxy クラス]
    D -- 自動生成 --> F[Java クライアントプログラム]
    F -- 呼出 --> B

```

※Javaでなくても、C#でも可能。相互接続も可能。

Copyright(c) 2002, NEC Corp. All rights reserved. 16



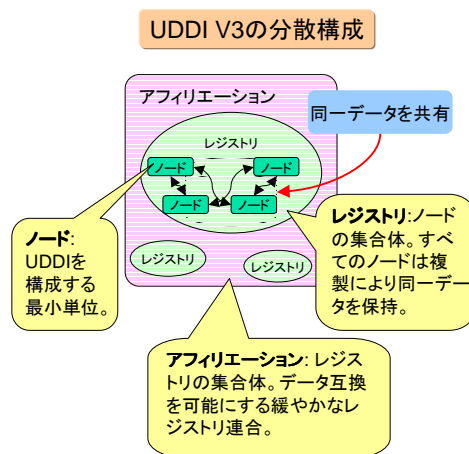
UDDIプログラマーAPI

- **参照API (Inquiry API)**
 - **find_XXX**
キーワードなどを用いた検索のためのAPI。検索の結果は対象となった要素のキーがリスト形式で返される。
 - **get_XXX**
find_XXXで検索されたキーから、その実体(内容)を取り出すときに使用。
- **発行API (Publication API)**
 - **save_XXX**
レジストリオブジェクトを保管するためのAPI。
 - **delete_XXX**
レジストリのエントリを削除する

※ここでXXXには検索の対象となるbusinessEntity、businessService、bindingTemplate、tModelなどが入る。

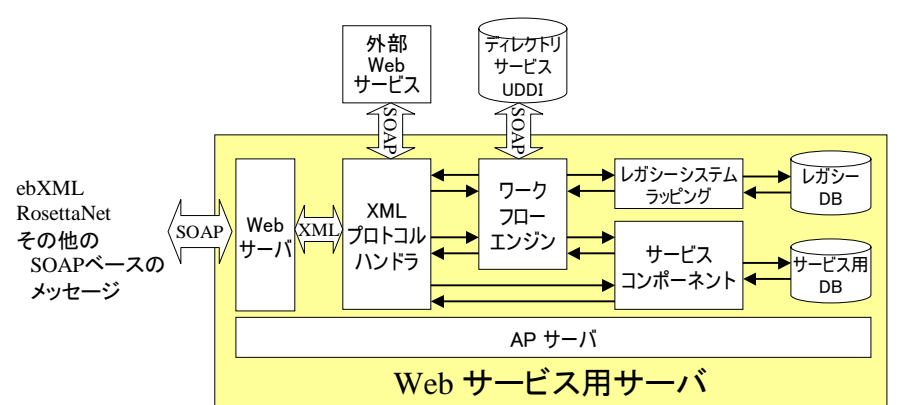
UDDI 3.0 レジストリ

- UDDI 3.0
 - Webサービス向けディレクトリサービスの最新仕様(2002年7月承認)
 - V2.0の機能を大幅に強化
 - 分散レジストリの対応
 - ノード: 一つ一つのサーバ
 - レジストリ: ノードの集合。全てのノードは同一データを共有
 - アフィリエーション: レジストリ間の相互参照関係を定義
 - プライベートディレクトリを構築するための指針
 - XML署名の導入
 - 検索機能・各種APIの強化



Webサービス向けアーキテクチャ

- XML プロトコルハンドラによる業界標準プロトコルへの対応
- ワークフローエンジンが、サービス統合のフローを管理
 一部は外部 Webサービスを呼び出し、一部は企業内サービスと統合
- UDDIへのサービスの登録、実行時におけるサービスの検索

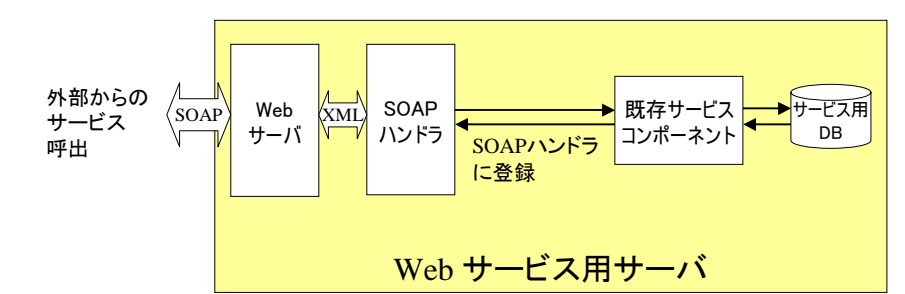


The diagram illustrates a Web Service Architecture. On the left, external services like ebXML, RosettaNet, and other SOAP-based messages are received by a 'Webサーバ' (Web Server). This server connects to an 'XMLプロトコルハンドラ' (XML Protocol Handler). The handler interacts with an '外部Webサービス' (External Web Service) and a 'ディレクトリサービス UDDI' (Directory Service UDDI) via SOAP. The handler also connects to a 'ワークフローエンジン' (Workflow Engine). The engine is linked to a 'レガシーシステムラッピング' (Legacy System Wrapping) component, which in turn connects to a 'レガシーDB' (Legacy DB). The engine also connects to a 'サービスコンポーネント' (Service Component), which connects to a 'サービス用DB' (Service DB). All these components are housed within an 'APサーバ' (Application Server), which is part of the overall 'Webサービス用サーバ' (Web Service Server).

Copyright(c) 2002, NEC Corp. All rights reserved. 21

既存サービスのWebサービス化手法

- 既にWeb上に公開していたサービスに SOAP RPCのアクセスインターフェースを作成して、Webサービスとして提供する。
 - 株価情報、天気予報のようなメソッド呼出だけで完了するサービスであれば、簡単に Webサービス化が可能
 - 優良コンテンツを保持していれば、サービス提供料金によるビジネスが可能



The diagram shows the process of converting an existing service into a web service. It starts with '外部からのサービス呼出' (Service call from outside) which goes to a 'Webサーバ' (Web Server). The web server connects to a 'SOAPハンドラ' (SOAP Handler) via XML. The SOAP handler is connected to an '既存サービスコンポーネント' (Existing Service Component) via a bidirectional arrow labeled 'SOAPハンドラに登録' (Registered in SOAP handler). The existing service component is connected to a 'サービス用DB' (Service DB).

Copyright(c) 2002, NEC Corp. All rights reserved. 22

従来技術との比較表

比較対象/項目	従来	Webサービス	必要な技術
Webシステム	人間が画面を見る	→ 自動処理	SOAP
設計対象	オブジェクト	→ サービス	サービス指向設計
CORBA	Tightly coupled	→ Loosely coupled	HTTP
相手との取決	メソッドインタフェース	→ プロトコル	BPEL4WS, BPSS
EDI	専用	→ 汎用/標準化	標準化
SCM	固定取引	→ 動的取引	UDDI
ワークフロー	企業内 アクティビティ制御	→ 企業間 プロトコル制御	BPEL4WS エージェント
取引相手	1:1	→ 1:多	SingleSignOn
XML	データ表現	→ サービス呼び出し	SOAP, WSDL

※ 全ての特徴を持つのではなく、一部の特徴を利用するだけでよい。

Copyright(c) 2002, NEC Corp. All rights reserved.

23

3. Webサービスの技術課題

Webサービスは、即、使えるか？

- 接続は本当に簡単か？
 - 楽観派
 - SOAP は HTTP を利用できるのに、新たなネットワークの構築や、管理設定が不要
 - XML なので、機種非依存の共通表現の構築が容易
 - 懐疑派
 - SOAP でつなげることはできても、その後のリモート通信の難しさは、従来と変わらない
 - トランザクション処理をどうするか
 - 通信エラー処理が簡単にできるか
 - コンテキスト管理ができるか
 - 相互接続が未検証な状態では、相互運用が疑問
- セキュリティは万全か？
- 動的なパートナー企業探しは現実的か？

Copyright(c) 2002, NEC Corp. All rights reserved.

25

Webサービスは、即、使えるか？

- 接続は本当に簡単か？
- セキュリティは万全か？
 - 楽観派
 - SSL で暗号化するから安心
 - XML 署名で証拠が残り、否認防止が可能
 - WS-Security (暗号、署名)も近々標準が制定される。
 - Liberty Alliance で認証連携もサポートするらしい
 - VPN で通信すれば良い
 - 懐疑派
 - DMZ(非武装地帯)を安全に作れるか
 - SOAP は、外向きの通信は簡単だが、内向きの通信をどうやって安全に行うのか
 - 通信エラーなどの例外処理にも堅牢か
- 動的なパートナー企業探しは現実的か？

Copyright(c) 2002, NEC Corp. All rights reserved.

26

Webサービスは、即、使えるか?

- 接続は本当に簡単か?
- セキュリティは万全か?
- 動的なパートナー企業探しは現実的か?
 - 楽観派
 - UDDI を使えば、その時の最良のパートナーを検索可能
 - サービスの信頼性は、レジストラが保証できる
 - 必要な部分だけでも、動的な連携を利用すれば良い
 - 懐疑派
 - 不特定の相手とのビジネスはリスクが高過ぎる
 - UDDI の検索は、最初の一回だけ。次からは、同じ相手と連携するので、毎回、動的に検索する必要はない。

Copyright(c) 2002, NEC Corp. All rights reserved. 27

課題を克服するための技術動向

- ビジネスで利用するために Simple から Complex に対応する技術開発、標準化が進展
 - トランザクション、プロセスフロー、セキュリティ、高信頼メッセージング
- 相互運用性確保のためのベンダー間の協調

Copyright(c) 2002, NEC Corp. All rights reserved. 28

ebXML Message Service

- B2B取引に必要な、コンテキスト管理、高信頼メッセージング(再送、到着確認、順序保証など)、パートナー契約、署名機能等を SOAP 上に定義
- ebXML Message Service 自身は、メッセージヘッダのようなもの
 - あらゆる型のビジネス文書を添付して送信
- V2.0 仕様が承認される(2002年9月)
 - 仕様を必須機能と、オプション機能に分離
 - 必須機能: 基本メッセージ、署名、エラー処理、同期応答
 - オプション機能: 高信頼メッセージ、順序保証、状態通知サービス、マルチホップ

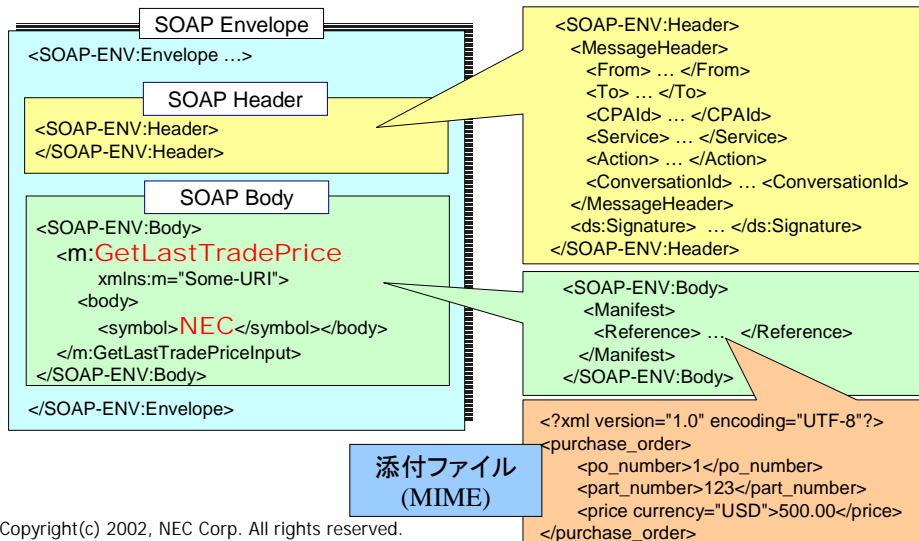
Copyright(c) 2002, NEC Corp. All rights reserved.

29

SOAP と ebXML メッセージ

簡単な SOAP メッセージ

ebXML メッセージ



Copyright(c) 2002, NEC Corp. All rights reserved.

ebXMLの高信頼メッセージ

- メッセージ到着確認
 - Acknowledgment メッセージ
- メッセージ再送
 - 再送回数、再送間隔の設定
- 重複受信回避
 - メッセージIDによる重複検出
 - 重複時に、過去のレスポンスを返答するための persist storage および、その有効期限の定義
- メッセージ順序保証
 - オーダー番号による整列

送信者 送信 受信者
受信者 HTTPレスポンス 送信者
送信者 Ack送信 受信者
受信者 HTTPレスポンス

非同期の到着確認

送信者 送信 受信者
受信者 HTTPレスポンス 送信者
送信者 Ack 受信者
受信者 HTTPレスポンス

再送
再送間隔
消失
送信

非同期の再送処理

送信者 送信 受信者
受信者 HTTPレスポンス 送信者
送信者 Ack 受信者
受信者 HTTPレスポンス

再送
再送間隔
消失
送信

処理
返答保存
復元
「一度だけ実行」を保証

重複処理と返答保存

Copyright(c) 2002, NEC Corp. All rights reserved. 31

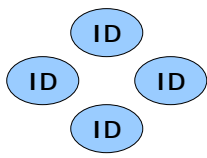
ebXML の相互接続実験

- 電子商取引推進協議会(ECOM)が中心となり、ebXML のメッセージサービスの相互接続実験を実施。
 - ebXML 相互接続テスト共通仕様書を公表
 - T1: 基本メッセージ
 - T2: 同期応答
 - T3: セキュリティ(SSL, HTTP認証)
 - T4: XML署名
 - T5: 高信頼メッセージ
 - T6: エラー処理
 - 富士通、日立、NEC、インフォテリア、NTTの5社が参加
 - 基本メッセージ、高信頼メッセージの接続に成功
 - 2002年9月30日にニュースリリース
 - 2002年12月4日に、アジア地区での相互接続実験開始を報道

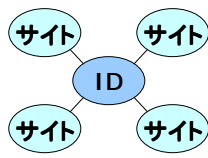
Copyright(c) 2002, NEC Corp. All rights reserved. 32

セキュリティ技術

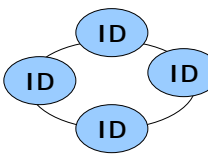
- 暗号や署名技術との統合
 - WS-Security: SOAP に暗号や署名情報を添付する標準
 - IBM, Microsoft, Verisign が共同で、OASIS に提案
- シングルサインオン(SSO)に向けた認証連携
 - SAML: 認証結果を交換するための標準
 - Liberty Alliance: SSO のプロセスの標準
 - Sun が、Network Identity というキーワードで、複数の認証基盤を連携したID管理方式を提案



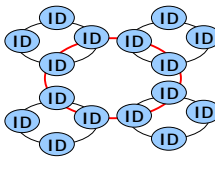
サイト毎のID管理



IDの集中管理



信頼サークル

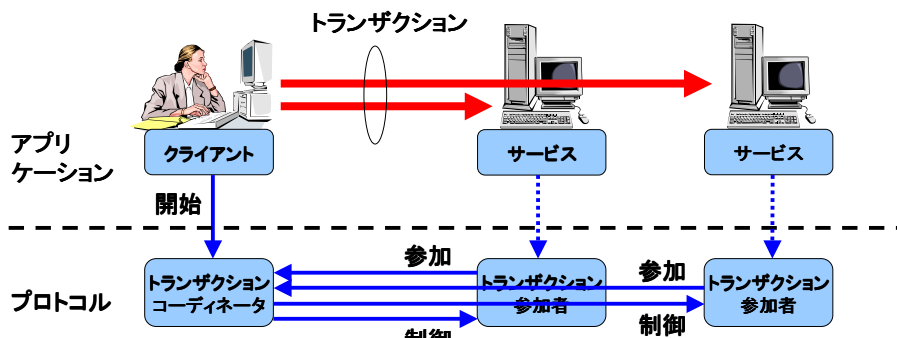


フェデレーション

Copyright(c) 2002, NEC Corp. All rights reserved. 33

BTP (Business Transaction Protocol)

- HP, BEA を中心に、OASIS にて策定中の Webサービス向けトランザクション仕様
 - 疎結合、長時間のトランザクションを実現
 - トランザクション制御用メッセージにもXMLを利用



Copyright(c) 2002, NEC Corp. All rights reserved. 34



WS-I

(Web Services Interoperability Organization)

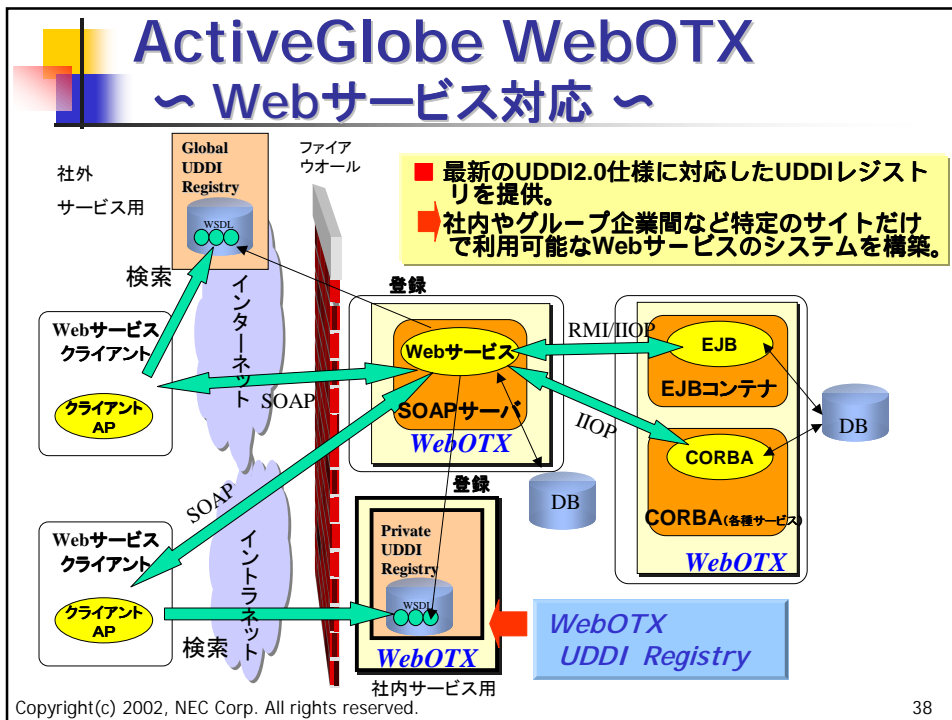
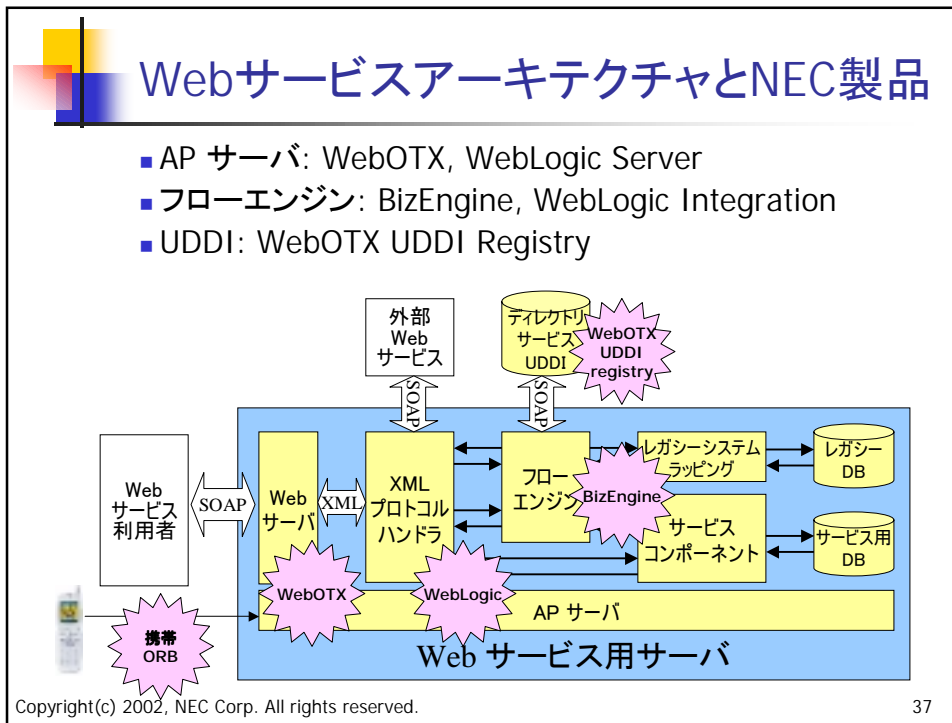
- 相互運用が可能なWebサービスの開発および普及を目的とした業界横断の団体
 - IBM、MS、HP、Oracle他9社が、2002年2月に設立
- WG活動
 - Webサービスの基本Profile を決定するWG
 - XML Schema 1.0, SOAP 1.1, WSDL 1.1, UDDI 2.0 を中心に共通仕様を議論中
 - WebサービスのサンプルアプリケーションWG
 - SCMを題材に、サンプルアプリケーションを構築
 - Webサービスの相互接続テストの構築WG

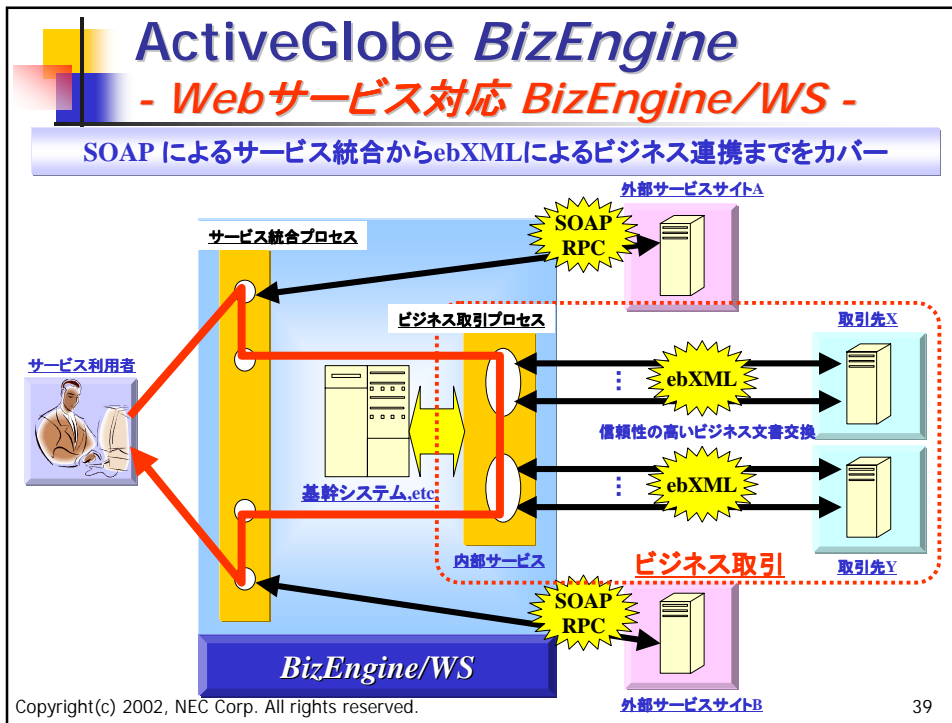
Copyright(c) 2002, NEC Corp. All rights reserved.

35

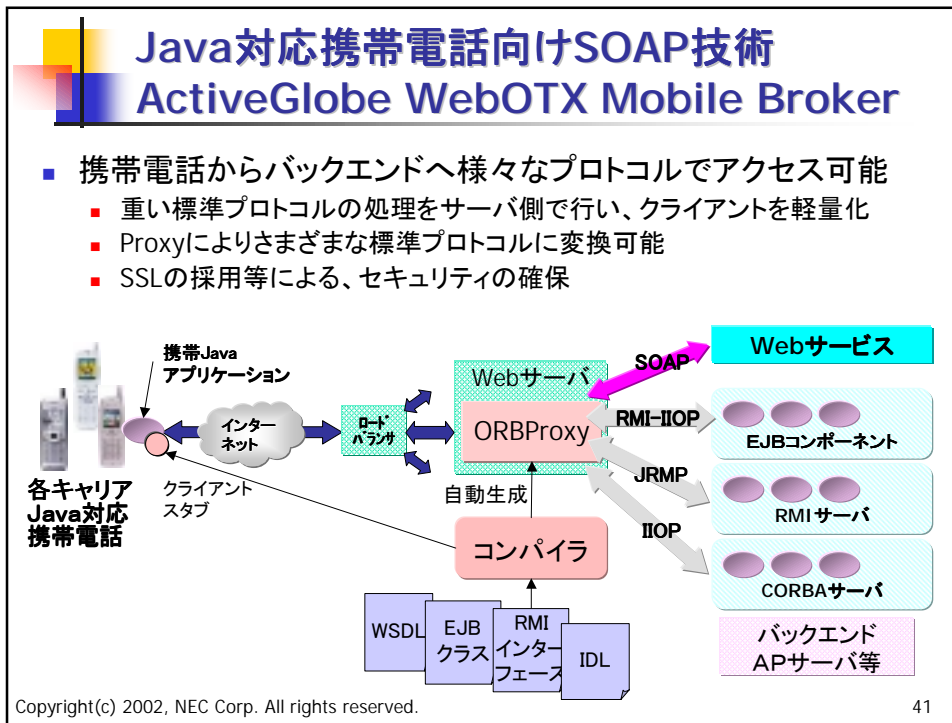


4. NEC の製品群





- ## BizEngine/WS の特徴
- フローエンジン/エディタを使ったビジネス設計
 - 簡単な Webサービス構築のために
 - SOAP サポート
 - SOAP からのフローエンジンの起動
 - フローエンジンから、SOAP呼出し
 - ビジネスWebサービスの構築のために
 - ebXML Message Service V2 仕様をサポート
 - メッセージ送受信するフローを定義し、実行
 - 高信頼メッセージ、CPA V2(契約)サポート
 - XML署名機能サポート
 - DMZ 対応 proxy も準備
- Copyright(c) 2002, NEC Corp. All rights reserved. 40



コード例

```

import com.nttdocomo.ui.IApplication;

public class OrderClient extends IApplication {
    ....
    public void processEvent(int type, int par) {
        ....
        try {
            Order obj = new Order(this,
                "sumida/servlet/_Hello_Proxy",
                "OrderObj");
            int nstock= obj.getStockNum("Gear");
            _setText(String.valueOf(nstock));
        } catch (Exception e) {
            err.out(e.getMessage());
        }
        ....
    }
}
    
```

```

int nstock;
try {
    HttpURLConnection conn
    = (HttpURLConnection)Connector.open(
        http://inb.kk.co.jp/Order/getStockNum,
        Connector.READ_WRITE, true);
    conn.setRequestMethod(HttpURLConnection.POST);
    DataOutputStream os = conn.openDataOutputStream();
    os.writeUTF("Gear");
    os.close();
    conn.connect();
    DataInputStream is = conn.openDataInputStream();
    if (is.readByte() == 0) {
        nstock = Integer.parseInt(is.readUTF());
    } else {
        re = new RuntimeException(is.readUTF());
    }
    is.close();
    conn.close();
} catch (ConnectionNotFoundException e) /* open */ {
    re = new RuntimeException(e.getClass().getName());
}
...
    
```

携帯JavaORBを利用しないと上の1行が左のコードに。

Copyright(c) 2002, NEC Corp. All rights reserved. 42



5. まとめ



まとめ

- Webサービスとは、インターネット環境における疎結合なサービスアーキテクチャ
 - WSDL によるサービスの抽象化
 - SOAP によるメッセージ交換
- SOAP, WSDL, UDDI とともに標準化途中
 - SOAP 1.2 で、中継ノードも含めた仕様に進化
 - WSDL 1.2では、定義の精緻化
 - UDDI 3.0 で分散環境におけるUDDIに進化
- 関連する標準化団体
 - OASIS: UDDI, ebXML, WS-Security, SAML, BTP
 - W3C: SOAP, WSDL, XML Schema
 - WS-I: Webサービス全体の Basic Profile を規定

6. デモの説明

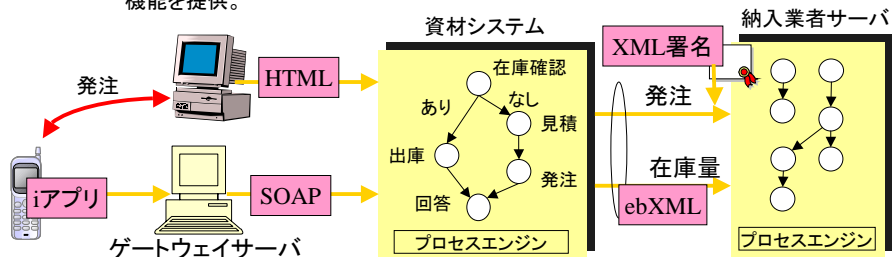
デモの説明

企業内の資材管理システムと社外のサプライヤをWebサービスで結合するデモ。

1. Webブラウザから発注する従来の企業内の資材発注システム。
2. 大量の資材発注をすると、自動的に社外の納入業者にebXML(Webサービス)で発注する。
3. ebXML メッセージ中にXML署名などがされる。
4. 同じような発注を、携帯JavaからSOAPを通して実行できる。

<< NECの製品 >>

- Webサービス統合基盤 ActiveGlobe BizEngine を用いて、資材管理システム、納入業者サーバを構築。BizEngine が ebXML プロトコルハンドラを提供。
- ActiveGlobe Web OTX Mobile Broker がSOAPのサービスを携帯から呼び出す機能を提供。



Copyright(c) 2002, NEC Corp. All rights reserved.

46

資材発注デモ画面(1)

社内資材発注画面 サプライヤ画面

(文房具一覧表示中) (受注履歴表示中)

Copyright(c) 2002, NEC Corp. All rights reserved. 47

資材発注デモ画面(3) ビジネスプロセス定義例

- 資材発注システムのバックエンドのビジネスプロセス例

アクティビティ定義用テンプレート

在庫確認

購買要求確認

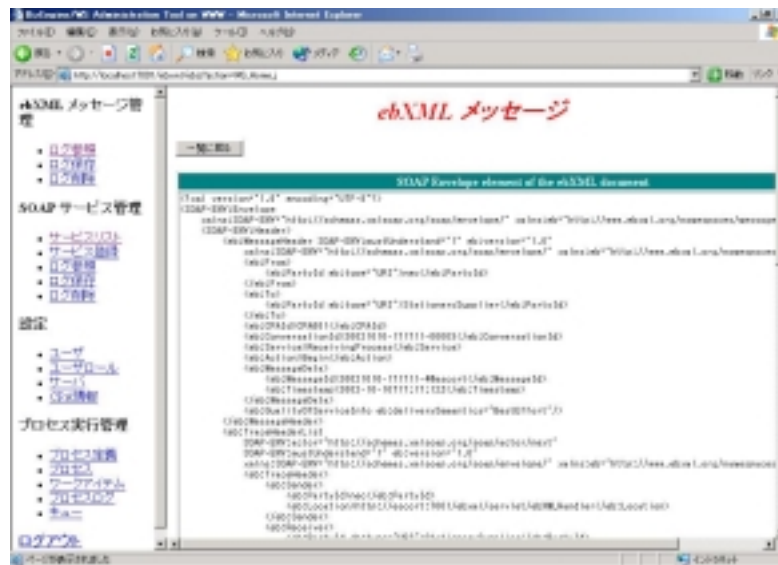
外部発注用メッセージ送信

確認用メッセージ受信

エラー処理

Copyright(c) 2002, NEC Corp. All rights reserved. 48

資材発注デモ画面(4) ebXMLメッセージ表示



Copyright(c) 2002, NEC Corp. All rights reserved.

49

資材発注デモ画面(5) 携帯電話からの発注



初期画面 データ取得 品目決定 発注数入力 発注確認 発注履歴
グラフ

Copyright(c) 2002, NEC Corp. All rights reserved.

50