

Webサービスの現在

XMLコンソーシアムSOA部会 天野富夫



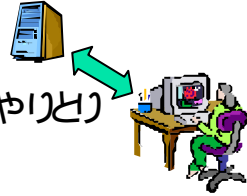
本日のアジェンダ

- ➡ ■ What's Webサービス?
- Webサービスの技術要素

Webアプリケーション VS Webサービス

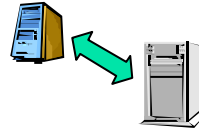
■ Webアプリケーション

- アプリケーションソフトと人間の間のやりとり
- 結果をブラウザ上に表示
- HTTP, HTMLの活用



■ Webサービス

- ソフトウェアモジュール間のやりとり
- 人間の介在無し
 - WebアプリがWebサービスを利用することが多い
- HTTP, XML, SOAP, WSDLの活用

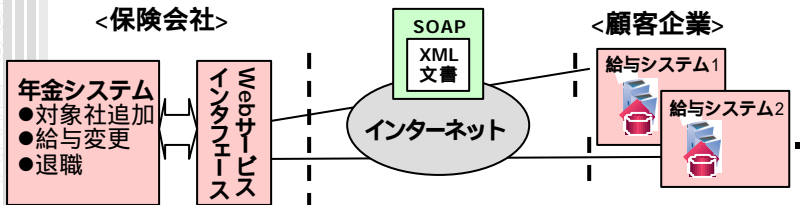


さまざまなWebサービスの定義

- ネットワーク上で**部品**として利用できるサービスであり、決められたフォーマットのメッセージを交換することによって実行されるもの。
- ネットワーク上でアプリを開発するための**部品**(コンポーネント)でメッセージのやりとりによって実行される。インターネットの標準的なプロトコルによって記述、発見、接続ができる。
- ネットワーク(Web)を通して記述、公開、配置、および起動することのできるアプリケーション**部品**。
- URIによって識別され、そのインターフェースおよびバインディング情報はXMLによって定義・記述され、それらの情報は他システムから検索が可能であり、インターネット・プロトコル上のXML書式のメッセージを使用して、他のWebサービスと交信する、**ソフトウェアシステム**である(W3C)

Webサービスの実施例 SOAP over HTTP

- インターネットを介しての給与情報の自動登録
 - 通信のプロトコルはファイアウォールと相性のよい HTTP/HTTPSを使用
 - メッセージを包み込む封筒の役割を果たすSOAP
 - 保険業界の規格に沿って記述され、システムで処理されるXML文書



Webサービスの歴史

- 1998年2月 XML1.0勧告
- 1999年4月 SOAP1.1提案
- 2000年9月 JDDI設立, Ver.1.0公開
- 2000年9月 WSDL1.0 公開
- 2000年中には主要ITベンダーがWebサービスコミット
- 2002年2月 WS-I設立
- 2002年7月 BPEL4WS 1.0公開
- :



Webサービスに対する見方 その1

- ネットワーク透過な機能呼出し
 - プログラムがメソッドをローカルなメソッドと同様に呼出すことができる
 - 他の言語で書かれたモジュールのメソッド
 - 他のハードウェア/OS上で動いているモジュールのメソッド
 - 有用な実装の選択肢
 - ワイヤフォーマットレベルでの相互運用性確保
 - SOAP/HTTPはオーバーヘッドは大きいインターネット経由での機能呼出しには向いている



XMLをベースにすることの意義

- 情報(データやプロセス)を記述する道具としてXMLは、...
 - オープンである
 - プラットフォーム独立
 - 良質で安価なツール
 - 記述能力が高い
 - 目的に応じて文法構造を定義可能
 - 名前空間によるモジュール化
 - 人間/機械双方にとっての可読性
 - 機械処理可能な厳密な記述
 - いざとなったら人手でも記述できる敷居の低さ
- **新たに情報表現の規格を策定するのならXML以外ない!**



Webサービスに対する見方 その2

- インタフェースを記述する一元的手段
 - インタフェース記述レベルでの相互運用性の確保
 - 機械処理可能なインタフェース記述
 - インタフェース記述の自動変換
 - インタフェース記述からプログラムコードの生成
 - ワイヤフォーマット/プロトコルに依らずインタフェースが記述できればWebサービス
 - JMS,RMI,EJBも見かたによってはWebサービス



- What's Webサービス?
- ➡ ■ Webサービスの技術要素,標準



Webサービス標準化の推進団体

- W3C
 - XML Schema, SOAP, WSDL...
- OASIS
 - UDDI, BPEL4WS...
- WS-I
 - 相互運用性確保のための各種プロファイル



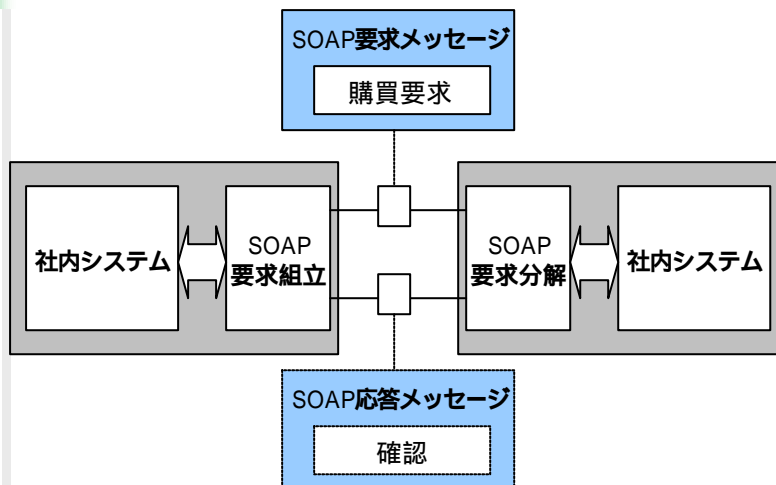
Webサービス基本規格

- SOAP
 - Webサービスの呼び出し実行
- WSDL
 - Webサービスインタフェースの記述
- UDDI
 - Webサービスの登録、検索
- WSS(WS-Security)
 - セキュリティ

SOAPとは

- SOAP (Simple Object Access Protocol)
 - 分散環境での軽量情報交換プロトコル
 - メッセージ内容を入れる封筒の役割
 - プラットフォーム独立なRPCも実現
 - SOAP関連規格による拡張のベース
 - SOAPのヘッダ部分に各種の情報を記述
 - WS-Security, WS-Addressing, WS-ReliableMessageなど

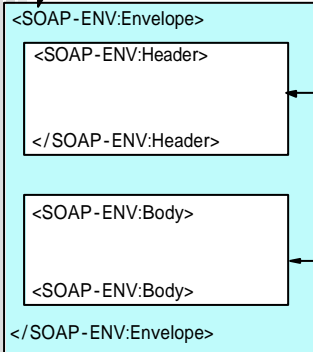
SOAPを用いたデータ交換



SOAPエンベロープの構造

XML Consortium

- ▶ Envelopeの構造を記述する要素は名前空間 `http://schemas.xmlsoap.org/soap/envelope/` に属している
- ▶ 慣習的に使われるプレフィクスはSOAP-ENV:



- ▶ メソッドのシグニチャにない情報(Transaction IDなど)の受渡しに使用
- ▶ 中に書かれる各エントリの要素名はアプリ依存
- ▶ SOAP-ENVに属するグローバルな属性
 - 当該要素の受取人を指定するactor
 - 受取人が当該要素の意味を理解していなければならないことをしめすmustUnderstand
- ▶ 要素/属性は名前空間が限定されている

- ▶ アプリケーションデータの受渡しに使用
- ▶ 中に書かれる各エントリの要素名はアプリ依存
- ▶ 構造や型の記述用の語彙が定義されている
- ▶ エラー情報を示す語彙が定義されている

```

<SOAP-ENV:Fault>
  <faultcode>...</faultcode>
  <faultstring>...</faultstring>
  <faultactor>...</faultactor>
  <detail>...</detail>
</SOAP-ENV:Fault>
  
```

SOAPメッセージの例

XML Consortium

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
  
```

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <t:Transaction
      xmlns:t="some-URI"
      xsi:type="xsd:int" SOAP-ENV:mustUnderstand="1">
      5
    </t:Transaction>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse
      xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
  
```




SOAPメッセージの内容

- SOAPはアプリケーションデータについては何も規定していない
- アプリデータを規定する2つの考えかた
- ビジネス機能を提供するサービスではDocument/Literal型のメッセージが主流

	既存API型(RPC/Literal)	新規メッセージ型(Document/Literal)
方式	既存処理モジュールのAPIをメッセージにマッピング	業務モデルからメッセージフォーマットを決定し、処理モジュールを設計
フォーマット	元のAPIに依存した 独自 なものになる	共通に使える 標準 フォーマットを策定可能
設計上の重点	既存資産の有効活用	標準への準拠、汎用性
開発	比較的容易、SOAPへのマッピングツール等有り	関係者間の合意形成に時間がかかることも



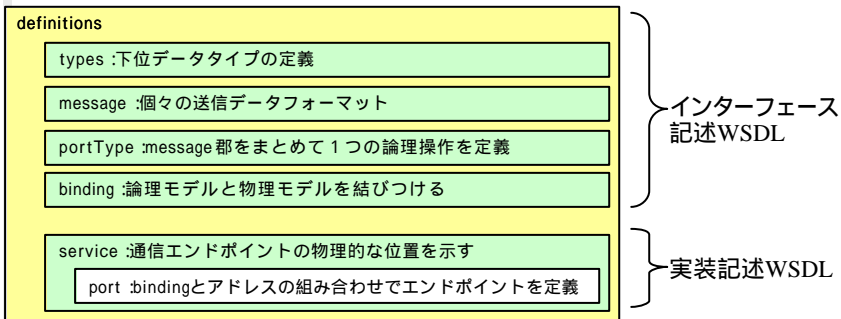
インタフェースの規格 WSDL

- Web Services Description Language
- サービスに関する最も基本的なメタデータの記述
 - SOAPやHTTP専用というわけではない
- Webサービスのインターフェース定義記述
 - インターフェースを公開するため
 - 接続プロキシーを自動生成するため
 - 動的に接続するため
- どういうXMLを受取って
- どういうXMLを返すか
- どういう通信プロトコルを使うか



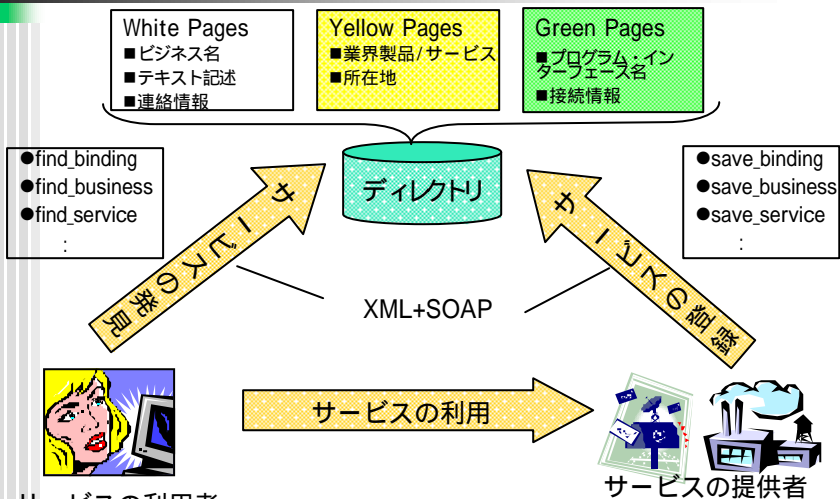
WSDLの構造

XML Consortium



UDDIによる取引相手/サービスの発見

XML Consortium





UDDIにおける検索とは？

XML Consortium

- 地理的な情報や業種などのタクソノミー (分類体系) に基づく
- タクソノミーの分類(tModel)が一致したかどうかで検索
- カスタムメイドのタクソノミーを用意することも可能(プライベートUDDI)
- 部分一致や数量比較等の検索APIは持たない
- アプリに応じた内容検索には別途検索用Webサービスが必要

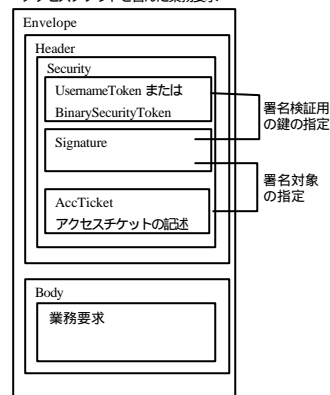


WS-Security

XML Consortium

- SOAPでセキュアなメッセージ交換(End-to-End)を行なうためのSOAP拡張
- 暗号化や署名の情報をSOAPヘッダに記述するための語彙を規定
- 暗号化や署名済みXMLデータの表現自体は別規格で規定される

アクセスチケットを含んだ業務要求



SSLとの比較

- SSL: Point-to-Point
 - トラフィック層のセキュリティ
 - 中継者へのセキュリティ確保が困難



- WSS: End-to-End
 - メッセージコンテンツのセキュリティ
 - 中継者にもセキュリティを確保



WS-*規格の今後

- WS-Addressing
 - Webサービスメッセージの宛先、意図の指定
- WS-ReliableMessage
 - メッセージの送達保証
- WS-Policy
 - Webサービスの非機能的な能力・要件の記述
- WS-MetaDataExchange
 - WS-PolicyやWSDLなどメタデータ交換用サービス
- WS-Coordination, WS AtomicTransactions, WS-BusinessActivity
 - ACIDトランザクション、ビジネストランザクションの実現



Webサービスの現在

- アーキテクチャと実現手段の概念の整理
 - サービス指向アーキテクチャ(SOA)
 - 実現手段としてのWebサービス