WS-I
Basic Profile 1.0
公開レビュー版の概説

2003年5月30日
藤田悟　(NEC)
沼田利典　(富士通)
大場みち子　(日立製作所)

発表の内容

1. Basic Profile 策定の背景
2. Basic Profile 1.0 の章構成
3. 適合性(Conformance)の基準
4. MESSAGE
5. DESCRIPTION
6. REGISTY と REGDATA
7. セキュリティ
8. 今後の計画

1. Basic Profile 策定の背景

- 各ベンダーのWebサービス技術に非互換性がある。その原因に以下のものがある。
  - SOAP や WSDL, UDDI の仕様、及び仕様中に使われている例に曖昧性や誤りがある。
  - SOAP や WSDL, UDDI の仕様に相互に冗長性や矛盾がある。
  - SOAP や WSDL が XML Schema の仕様策定以前に策定されたので、現在の XML Schema に従わない独自のエンコーディングを利用している。

[Basic Profile の目的]
以上の問題を解決し、相互運用できる標準を規定する基本仕様の集合を定めたい。

Basic Profileとは

- 基本仕様の組み合わせを規定
  - SOAP、WSDL、UDDI、HTTP etc…
  - バージョンを指定
- 各基本仕様に対して次の操作を加える
  - サブセット化
    - 基本仕様を利用する範囲のみに絞る
  - バグ修正
    - 間違いを修正する　⇒ 基本仕様へフィードバック
  - 曖昧性の排除
    - 曖昧で解釈の分かれる部分に一定の解釈を与える（基本仕様の検討と同期して）
  - オプション機能の採用/不採用を規定
  - パラメーター値の範囲を規定

Basic Profile の策定方針

- 現在利用されている仕様をもとに規定する（SOAP 1.1、WSDL 1.1 など）
- 既存の仕様を参照し、ここで新しい仕様を規定することはない
- 現在検討中の仕様に方向性を合わせる（SOAP 1.2、WSDL 1.2 などとは矛盾しないように）
- 基本仕様の要件を変更する場合、要件を制限することはあっても緩和することはない

Basic Profile の状況

- WS-I Basic Profile WG で検討中
  - 2002年4月、WG発足
  - 2002年10月、Working Group Draft 第1回公開レビュー
  - 2003年2月、Working Group Draft 第2回公開レビュー
  - 2003年4月、Board Approval Draft 公開レビュー
- ここでは、4月の Board Approval Draft 公開レビュー版をもとに説明
  - http://www.ws-i.org/Profiles/Basic/2003-03/BasicProfile-1.0-BdAD.html
- WS-I Japan SIG で日本語訳を作成中

2. Basic Profile 1.0 の章構成

- 1章: 序文
- 2章: 適用範囲
- 3章: 適合性についての定義
  - サービス提供者と利用者の適合性の定義
  - Description, Message, Registry Data の中への適合性表示
- 4章: メッセージング (SOAP)
  - SOAP中のXML表現(BOMの受容, Fault, encodingStyleの禁止, DTD添付の禁止, …)
  - SOAPの(ヘッダとフォールトの)処理モデル
  - HTTPの中のSOAPの利用方法(HTTPのバージョン, 状態コード…)
- 5章: サービス記述 (WSDL)
  - 文書構造(スキーマ定義、スキーマのimport、…)
  - Type (SOAPの定義する配列の禁止、…), Message (rpc と document, …),
  - Port Type (パラメータ順序, …)
  - Binding, SOAP Binding
- 6章: サービスの公開と発見 (UDDI)
  - bindingTemplate
  - tModel
- 7章: セキュリティ
  - HTTPSの利用

Basic Profile の要件分類

- **要件の数は、全体で158個**
- **要件レベル**
  - MUST/MUST NOT（**必須**）
  - SHOULD/SHOULD NOT（**推奨**）
  - MAY（**許容**）
- **要件の分類**

	必須	推奨	許容	計
§3. プロファイルへの適合性	9	0	3	12
§4. メッセージング（SOAP）	32	12	4	48
§5. サービスの記述（WSDL）	64	8	19	91
§6. サービスの公開と発見（UDDI）	4	0	0	4
§7. セキュリティ	1	0	2	3
合計	110	20	28	158

3. 適合性(Conformance)の基準

何に対しての適合性を判断するのか？

- Basic Profile 1.0 は「Webサービスの実装」に対する適合性を規定
- テストツールで適合性を検証
- 開発ツールやプラットフォームは Basic Profile 1.0 の適用範囲外 (out of scope)
- 将来のプロファイルでは、ツールやプラットフォームの適合性を規定する可能性あり

適合性の対象

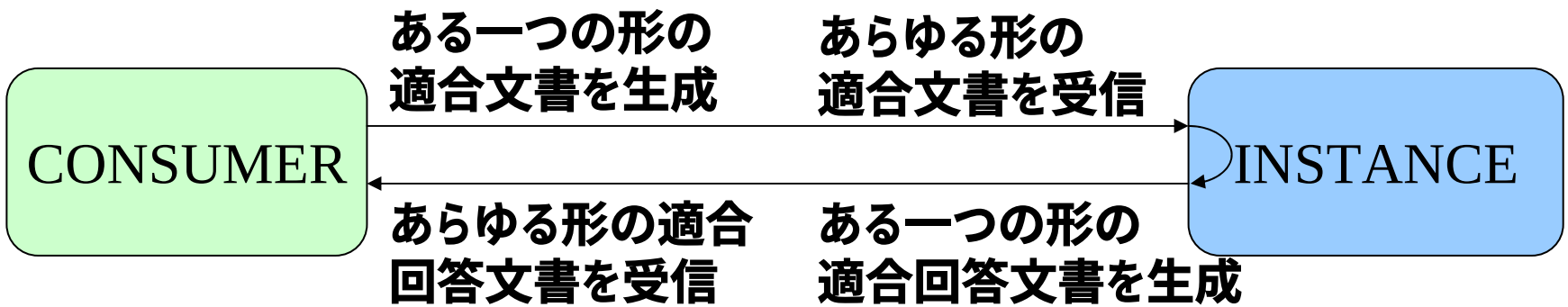
- MESSAGE
  - Web サービスを呼び出すにあたり、ネットワーク上で交換されるプロトコルの構成要素(すなわち, SOAP/HTTP メッセージ)
- DESCRIPTION
  - Webサービスにアクセスするためのインタフェース情報(データ型, メッセージなど)と、その具体的なプロトコルバインディング、アクセスポイントを定義する記述 (すなわち, WSDL 記述)
- REGISTRY
  - REGDATA の登録/検索を管理する レジストリ(すなわち、UDDI)
- REGDATA
  - REGISTRYに登録するWeb サービスについての記述 (たとえば, UDDI tModel)

Basic Profile 1.0の参照する仕様

- SOAP 1.1
- WSDL 1.1
- UDDI 2.0
- XML Schema Part 1, 2
- XML 1.0
- HTTP 1.1
- TLS 1.0
- SSL 3.0
- X.509 Public Key Infrastructure Certificate

適合サービスの提供者と利用者の関係

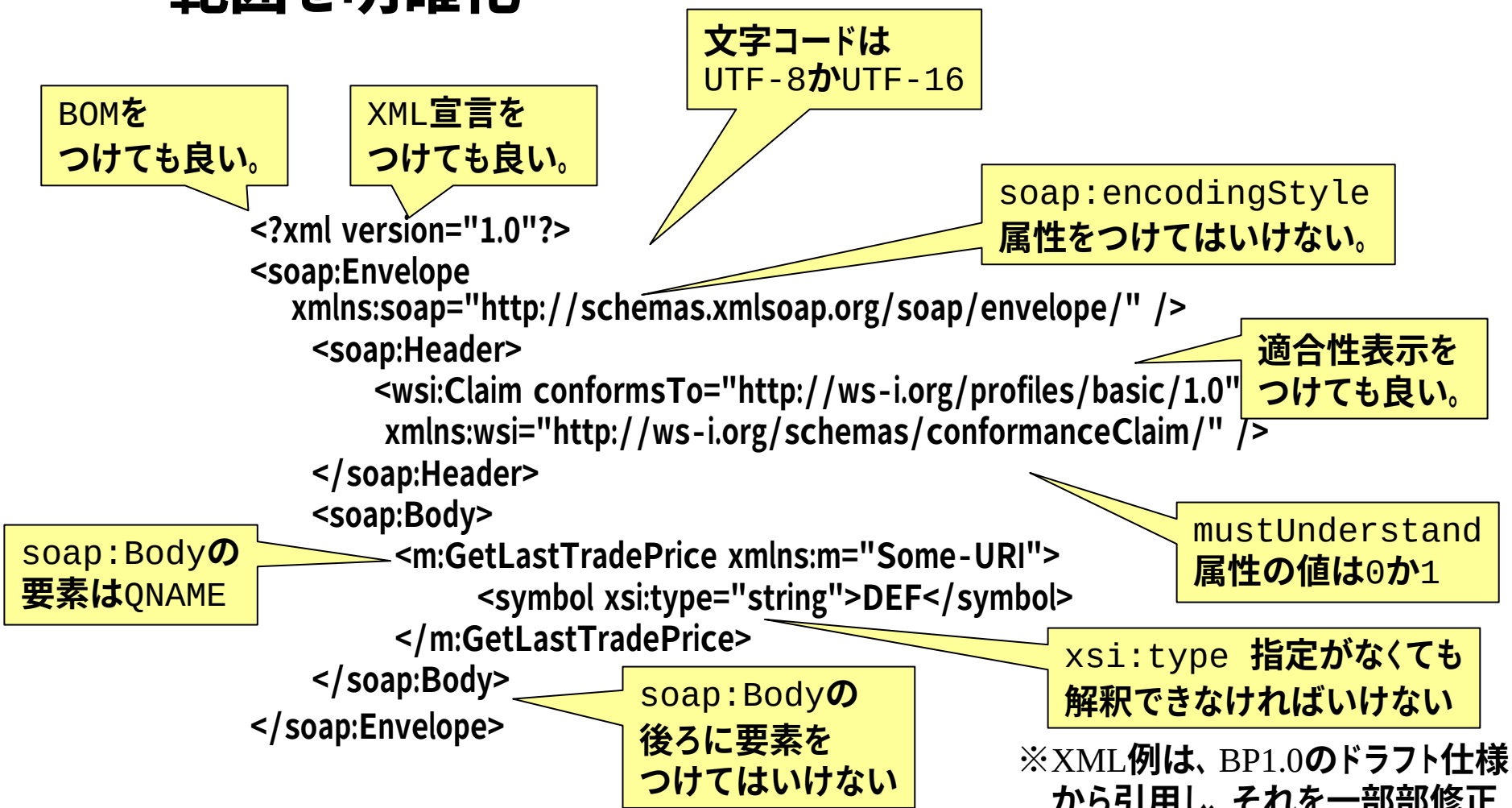
- INSTANCE: デプロイ された Web サービスそのもの (wsdl:port 又は uddi:bindingTemplate で指定されるもの)
- CONSUMER: サービスのインスタンスの利用者



例えば、MESSAGEの文字コードについて、
UTF-8とUTF-16が許されるとき、
送信側: UTF-8かUTF-16のどちらかのMESSAGEを送信して良い
受信側: UTF-8とUTF-16の両方のMESSAGEを受信できなければならない。

4. MESSAGE

- SOAPの基本フォーマットについての制限/許容範囲を明確化



※XML例は、BP1.0のドラフト仕様から引用し、それを一部部修正

Fault について

- faultについては、フォーマットの問題と、処理(faultがあったら、それ以上処理を進めない等)の問題について規定。
- HTTP の状態コードに関わらず、soap:Faultがあるときだけを、Faultの状態と判断する。

faultcodeは、SOAP1.1のfaultcodeが望ましい。

soap:Faultの子要素は、namespaceで修飾しない。

soap:Faultの子要素は、faultcode faultstring faultactor detailだけ

```
<soap:Fault
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' >
    <faultcode>soap:Client</faultcode>
    <faultstring>Invalid message format</faultstring>
     <faultactor>http://example.org/someactor</faultactor>
    <detail>
        <m:msg xmlns:m='http://example.org/...
            There were <b>lots</b> of elemen
            that I did not understand</m:msg>
     </detail>
</soap:Fault>
```

detailの中には、いかなる要素も許される。

※XML例は、BP1.0のドラフト仕様から引用

SOAP in HTTP

- HTTPのバージョン: 1.0か1.1。1.1が望ましい。
- HTTPのPOSTメッセージを利用
- HTTPヘッダのSOAPActionは、引用符(“”)で囲む
- HTTP 状態コード
  - 成功時には、2XX
    - 返信内容がある時は200
    - 返信内容がない時は200または202
  - リダイレクトは、307
  - リクエスト形式に問題があるときは、4XX
  - サーバのエラーは、5XX

5. DESCRIPTION

BOMを
つけてよい

UTF-8又は
UTF-16

wsdl文書をimport。
namespace は、元wsdlの
targetNamespaceと一致

document,
import,
typesの順番

wsdl:importは
locationが必須

xsd:schemaには、
targetNamespaceが必須

soapbind:body
で利用する場合は、
rpc-literalは type属性
document-literalは
element属性を使用

XML Schemaをimportす
る時は、xsd:schemaの
中でxsd:importを使用

operation nameの
オーバーロード
は禁止

bidingは、
soapbindに限定

use属性の指定が
ない場合、
literalと解釈

addressのlocation
が同じportを持つ
べきではない

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/
targetNamespace="http://example.org/bar/"
xmlns:foo="http://example.org/foo/">
<document> … </document>
<import namespace="http://example.com/stockquote/base"
      location="http://example.com/stockquote/stockquote.wsdl"/>
<types>
    <xsd:schema targetNamespace="http://example.org/foo/" …>
        <xsd:import … /> …
    </xsd:schema>
</types>
<message name="BarMsg">
    <part name="BarAccessor" type="foo:fooType"/>
</message>
<portType name="BarPortType">
    <operation name="BarOperation"> <input message="bar:BarMsg"/> </operation>
</portType>
<binding name="BarSOAPBinding" type="bar:BarPortType">
    <soapbind:binding transport="http://schemas.xmlsoap.org/soap/http/"
        style="rpc"/>
    <operation name="BarOperation">
        <input message="bar:BarMsg"> <soapbind:body use="literal"/> </input>
    </operation>
</binding>
<service name="serviceName">
    <port name="BarSOAPPort" binding="bar:BarSOAPBinding">
        <soapbind:address location="http://example.org/myBarSOAPPort"/>
    </port>
</service>
</definitions>
```

※XML例は、BP1.0のドラフト仕様
から引用し、それを一部部修正

配列の問題

- DESCRIPTION 中の array 宣言は、soapenc:Array 型の拡張であってはいけない。
- DESCRIPTION 中の array 宣言は、その型宣言に wsdl:arrayType 属性を使用してはいけない 。
- DESCRIPTION 中では、array 宣言に対する要素名に、ArrayOfXXX という名前付けをすべきではない。
- MESSAGE は、soapenc:arrayType 属性を持ってはいけない。

WSDL例

```
<xsd:element name="MyArray1" type="tns:MyArray1Type"/>
    <xsd:complexType name="MyArray1Type">
    <xsd:sequence>
        <xsd:element name="x" type="xsd:string" minOccurs="0"
                              maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

SOAP例

```
<MyArray1>
    <x>abcd</x>
    <x>efgh</x>
</MyArray1>
```

6. REGISTRY と REGDATA

- UDDIの仕様は、文字エンコードをUTF-8に限定してしまっているので、UTF-16を受け付けない。この意味で、UDDI自体はWS-IのWebサービスの条件に完全適合していない。
  - UDDI は WS-I 準拠の Webサービスではないが、WS-I 構成する重要な構成要素の一つである。
- bindingTemplate
  - WS-Iに準拠する INSTANCE に対する uddi:bindingTemplate 型の REGDATA は，uddi:accessPoint 要素を含まなければならない。
    - WSDLは、addressを直接指定する必要があり、それと整合するために、REGDATEの方は、uddi:hostingRedirector を使わない。
- tModel
  - WS-Iに適合する Web サービスの REGDATA は，WSDL で記述されなければならない。
  - WS-Iに適合する Web サービスの REGDATA は，WSDL と UDDI の整合をとるために、UDDI Best Practice for Using WSDL in a UDDI Registry V1.08 に従わなければならいない。

7. セキュリティ

- INSTANCEは、HTTPS(SSL3.0, TLS1.0)の使用を必須条件としてもよい。
- INSTANCE が
  - HTTPS を使用する場合は、wsdl:port 定義中の soapbind:address 要素の location 属性に示すURIは，https でなければならない 。
  - そうでない場合は，上記URIは、http でなければならない。
- INSTANCE は，HTTPの相互認証 (mutual authentication) を必須としてもよい。

8. 今後の計画
- Basic Profile 1.1 -

- W3C Noteである “SOAP Messages with Attachments” (2000年12月11日) を取り込んだ Basic Profile 1.1 を策定中。
  - Basic Profile 1.0 に適合する INSTANCEが、Basic Profile 1.1 にも適合するように配慮。

- Basic Security Profile -

- XMLレイヤのセキュリティに関する Profile を規定する。
- Basic Profile に対する拡張プロファイルとして規定。
- Basic Profile とは別のWG (Basic Security Profile WG) で策定中。(2003/05～)
- OASISで策定中の WS-Security を取り込む。