

Grid勉強会

11/05/12

5.1節～5.2節

高瀬 亘

OGSAの誕生

- Globus Toolkit2では様々なデータ形式、通信ポート等が利用され、グリッド・コンピューティング環境全体としての標準化が困難
- APIが使用するプロトコルもインターネット環境で通常使用されないものもあり、既存環境のままでは使用することが困難
- Open Grid Services Architecture (OGSA)を策定して上記問題の克服を目指す
 - GGF (Global Grid Forum)が策定
 - ビジネス分野でも使用できるようにもする

OGSAの誕生

- グリッド・コンピューティングをWebサービスアプリケーションとして提供
 - Webサービスの技術を採用
 - データ形式: XML
 - 通信プロトコル: SOAP

Webサービスとは

- コンピュータがWebを利用する仕組み
 - コンピュータが・・・
 - レジストリから検索
 - 検索結果から動的にサイトを決定
 - XML形式のメッセージ(SOAP)で自動呼出し
 - XML形式で取得したい情報が戻る
- WebとWebサービスの違い
 - Webは人が行う
 - Webサービスはコンピュータが行う

Webサービスの登場の背景

- オブジェクト指向技術の発展
 - CORBA
 - Object Management Group (OMG)
 - Distributed Component Object Model (DCOM)
 - Microsoft
 - 上記は、OSが異なり、独自のプロトコルでメッセージを送受信するため、お互いに通信することはできない
- 特定の開発手法にとらわれず、様々なプラットフォーム上で使用できるXMLベースのメッセージプロトコルが誕生
 - SOAP

Webサービスの登場の背景

- SOAP
 - 1999年にDevelopMentor、Microsoft、Userland Softwareの3社によって公開
 - 2000年にIBMとLotusが加わり、World Wide Web Consortium (W3C)に提出
- Webサービスはサービス指向アーキテクチャ (Service Oriented Architecture ; SOA)とも深い関係がある
 - 見える所(表面)の仕様だけは統一

Webサービスの目的

- 多様なプラットフォームまたはフレームワーク上で動作し、異なるソフトウェア・アプリケーション間で相互にやりとりできるようにすることが目的
 - Enterprise Application Integration (EAI)
 - Webサービスで実現すれば、システムの連携が非常にスムーズに行われ、変更も容易
 - システム開発
 - Webサービスの技術を利用すれば、OSやプラットフォーム、言語に依存することなく開発できる

XML (eXtensible Markup Language)

- マークアップ言語
- 作成者がタグを定義できる
- `<hoge attr="example">fuga</hoge>`
- 要素名が重複しないようにURIを使用
 - `<hoge xmlns:fuga="http://xxx.xxx" xmlns:piyo="http://yyy.yyy">`
 - `<fuga:foo>bar</fuga:foo>`
 - `<piyo:foo>baz</piyo:foo>`
 - `</hoge>`

XML (eXtensible Markup Language)

- XML Schema
 - XML文書のデータ構造を定義したもの
 - 自分で要素や階層構造を決めることができるため、こういったものを用意する必要がある
 - スキーマ情報を2つに分けて定義
 - 要素、属性の宣言
 - データ型の定義
 - どんな要素、属性、値から構成されるかを定める
 - データベースで、テーブルやフィールドの定義をするのと同じようなもの

SOAP

- Simple Object Access Protocol
- XMLベースのプロトコルで、HTTP上でクライアントとサーバが、データや処理結果をやりとりできるように、フォーマットやメッセージの処理ルールを定めたもの
- 次の3つから成る
 - SOAPエンベロップ
 - SOAPエンコーディング
 - SOAP-RPC

SOAPエンベロップ

- SOAPメッセージの構造

- ヘッダ

- プロトコルを表す
 - このデータを受信したサーバは、続くデータがSOAPであると分かる
 - ヘッダ自体はSOAPメッセージには含まれない

- SOAPメッセージ

- SOAPエンベロップ
 - SOAPヘッダ
 - SOAPボディ

SOAPエンベロープ構成要素

- SOAPヘッダ
 - SOAPメッセージのヘッダ
 - SOAP本体に記述されている内容を誰がどのように処理するかなどが記載
- SOAPボディ
 - SOAPメッセージの本体
 - 最終的なSOAPメッセージの受信者に行ってもらいたい処理が記載
 - RPCのメソッド名や引数等

SOAPエンベロープ

- SOAPフォルト

- SOAPボディに含まれる

- クライアントからのリクエストを受けたサーバ上でエラーが発生した場合に、クライアント宛へのSOAPボディにエラーになった情報を返すために使用

- SOAPフォルトの構成

- faultcode: SOAP違反コード
 - faultstring: エラー内容
 - faultactor: エラー発生元 (URI)
 - detail: エラー詳細

SOAPエンコーディング

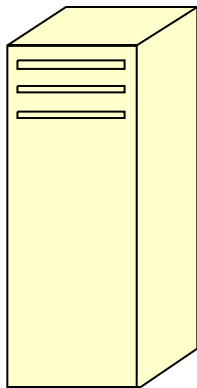
- アプリケーションで扱うデータ構造をXMLで定義するためのエンコーディング規則の定義
- 2種類のデータを定義
 - 単純型: 子要素を持たない
 - 複合型: 2種類ある
 - 構造体: C言語の構造体に似ている
 - 単純型のいくつかをひとまとめにして名前をつける
 - 配列: C言語の配列に似ている
 - 要素の集合体に対して番号でアクセス

SOAP-RPC

- RPCを実現するためのメソッド呼出しとレスポンスの記述方法を定義
 1. クライアントが呼び出したいメソッドを要素名にして引数付きでメッセージを送る
 2. サーバは指定されたメソッドを実行し、メソッド名 + Responseとした要素の中に返り値を入れてメッセージを返す

SOAP-RPC

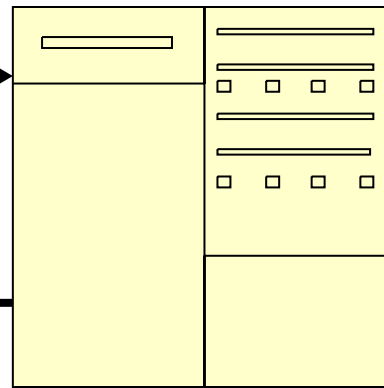
クライアント



XML形式のリクエスト



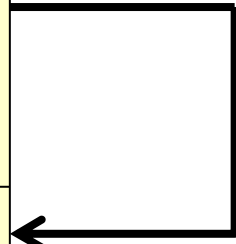
サーバ



XML形式のレスポンス



処理



SOAPの実装

- 実装の一つにApache-SOAPがある
- Globus Toolkit3ではApache-Axis (Apache-SOAP 3.0)が使用されている
 - SAX (Simple API for XML) 採用による処理速度の向上
 - ヘッダ処理と仲介者のサポート
 - JAX-RPC (Java API for XML RPC) のサポート

Apache Axis

- ヘッダ
 - ハンドラ(サーブレットのフィルタに似た仕組み)をSOAPヘッダに組み込むことで、ヘッダ処理を柔軟に扱える
 - 電子署名に関する情報を追加するようなハンドラを組み込み、受信側でも検証するハンドラを用意し、メッセージレベルでの署名検証が行える
- JAX-RPC
 - SOAP RPCを利用するアプリケーションのためのAPIを規定(Sunの決めた規格)
 - JAX-RPCハンドラを利用できる
- WSDO (Web Service Deployment Descriptor) ファイル
 - ハンドラを様々な場所に自由に配置できる

WSDL

- Web Service Description Language
- XML形式で記述されたWebサービスのインタフェースを定義
 - 発見したサービスがどういうサービスか、どのようにアクセスするかの情報を記載
- コンピュータ同士がお互いでやり取りする上での取決めを記載
 - この関数がきたらこの関数で返す等

WSDLの基本構造

- wsdl:definitions要素
 - WSDL文書の最上位要素

抽象的定義部分

- wsdl:types要素
 - メッセージで使用するデータの型を定義
- wsdl:message要素
 - Webサービスでやり取りするメッセージを抽象的に定義
- wsdl:portType要素
 - operationをひとまとめにして一つのポートとして定義
- wsdl:operation要素
 - メソッドの定義(入出力、エラーのメッセージ)

- wsdl:binding要素
 - portTypeを具体的な通信プロトコルやメッセージに結び付ける
- wsdl:service要素
 - portをひとまとめにして一つのサービスとして定義
- wsdl:port要素
 - サービスの宛先を指定

具体的定義部分

UDDI

- Universal Description, Discover and Integration
- Webサービスをインターネット上に公開し、検索できるようにする仕組み
 - サービスを発見できるようにするためのレジストリ、標準仕様を提供
- SOAPインタフェースを持つWebサービスのディレクトリ・サービス
- UDDIディレクトリを検索してサービスを利用
 - 予め、UDDIディレクトリにサービスの内容、使用方法などの情報を登録

Webサービスの課題

- まだWebサービスの利用はイントラネットが中心
- 以下の課題がある
 - 複数のSOAPメッセージを使用して一連の業務処理を実現するためのトランザクション処理
 - 処理データの機密性・信頼性を保証するためのセキュリティ
 - Webサービス利用者への課金の仕組み

WS-Securityとは

- IBM、Microsoft、VeriSignの3社が発表したメッセージセキュリティのための基本要素を定義したもの
 - SOAPヘッダとボディにどのように署名/暗号化情報、トークン、データを格納するかを規定
- Webサービスによるセキュリティの課題の解決を試みる

WS-Securityとは

- WS-Security Core
 - 署名と暗号とトークンの格納方法の規定
- WS-Security
 - Webサービスに必要なセキュリティ仕様に関する上位規定も含んだもの

※本書での区別

WS-Security Core

- SOAPヘッダ部
 - セキュリティ・トークン
 - ユーザID/パスワード、証明書等のメッセージ作成者自身を証明する情報
 - 署名に使用したデジタル署名
 - ボディ部の暗号化に使用した鍵情報
- SOAPボディ部
 - 暗号化されたメッセージ
- 具体的な認証や暗号化方法は規定されていない
 - SOAメッセージの受取先のサーバ側の実装に依る

WS-Securityの機能仕様

- WS-Security Core
 - SOAPメッセージのデータレベルでの機密性や完全性を実現するために、XMLデータに対する電子署名と暗号化の仕組みを規定
- WS-Policy
 - セキュリティに関するデータの利用方法を相互に提示/要求する仕組みを提供
- WS-SecureConversation
 - 複数のサーバにまたがる安全なセッションの維持に関する規定
- WS-Trust
 - 宛先のドメインとの有効な認証情報の受け渡しに使用
- WS-Federation
 - 宛先のドメインで有効なユーザIDを扱えるようにマッピング
- WS-Privacy
 - 個人情報特有の取り扱いを規定
- WS-Authorization
 - アクセス制御に関する規定

WS-Securityとグリッド・コンピューティング

- WS-Securityが提供する、Webサービスにおけるセキュリティ仕様が解決する課題
 - グリッド・コンピューティングにおけるセキュリティ要件と重なる部分が多い
- WS-Securityの実装によっては、グリッド・コンピューティングにおけるセキュリティの課題が解決されるかもしれない
 - WS-Security Coreは、Globus Toolkit3のセキュリティ仕様として採用されている

次回

- 11/05/17 11:00
- 5.3～5.4