6S-05

XMLを用いた分散オブジェクトのアクセス方式

堀内　浩規　　茂木　信二　　榎本　大　　小田　稔周

（株）KDD研究所

1. はじめに

近年、分散しているアプリケーションやデータを統合化するプラットフォームとしてCORBA (Common Object Request Broker Architecture)[1]が普及しつつある。一方、インターネット上での文書やデータのやりとりをする際に使用される文書記述言語としてXML(Extensible Markup Language)[2]が注目されている。今後は、CORBAとXMLとの連携が重要になると考えられ、筆者等はCORBAサーバへのXMLによるアクセス方式や、CORBAサーバとXMLサーバとの相互接続方式等の検討を行っている。本稿では、XMLを用いてCORBAサーバにアクセスする方式を提案する。

2. CORBAとXMLの概要

2.1 CORBAの概要

CORBAのアプリケーションは、クライアントとサーバから構成され、サーバは**オブジェクト**を持ち、クライアントはサーバ上のオブジェクトが提供するサービスを利用する。オブジェクトが提供するサービスの**インタフェース**は、インタフェース定義言語(IDL)を用いて、オブジェクトの**操作**と**属性**、**データ型**等が定義される。

クライアントが操作や属性を呼出す際には、対象となるオブジェクトを**オブジェクトリファレンス**により指定する。オブジェクトの名前とオブジェクトリファレンスの組は、ネーミングサービスで管理される。

2.2 XMLの概要

XMLは、文書を**要素**という単位から構成し、開始と終了の対からなる**タグ**を使って表される。文書の構造は、要素を階層化することにより表現する。このタグ付き文書をXML**インスタンス**と呼ぶ。XML文書は、XMLインスタンスに加えて、XMLのバージョンや文字コード等を宣言するXML宣言と、文書の型を表す文書型定義(DTD)から構成する。DTDは、タグ付き文書に現れる要素、属性、エンティティの定義を行う。要素の定義では、名前、出現順序、出現回数について定義する。

3. XMLを用いたCORBAサーバのアクセス方式

3.1 基本方針

(1)XMLを用いてCORBAサーバにアクセスする形態を実現する。このため、両者の手順を変換するゲートウェイを設ける(3.2節参照)。

(2)既存のCORBAサーバに変更を加えることなく収容可能とする。このため、CORBAサーバのアクセスに使用するXMLインスタンスは、CORBAサーバが提供するインタフェースに対応させて定義する(3.3節参照)。

(3)ゲートウェイは、様々なCORBAサーバに柔軟に対応可能とさせる。また、クライアントが、XMLを用いて容易にアクセス可能とする(3.4節参照)。

3.2 アクセス形態

基本方針(1)に従った、XMLによってCORBAサーバをアクセス可能とする形態を図1に示す。具体的には、クライアントは、分散オブジェクト環境にあるゲートウェイに対して、HTTP(Hypertext Transfer Protocol)の通信プロトコル上で転送するXMLインスタンスを送信する。ゲートウェイは、それを変換して、IIOP(Internet Inter-ORB Protocol)の手順でCORBAサーバに操作要求を送信する。さらに、ゲートウェイは、CORBAサーバから受信した操作応答をXMLインスタンスに変換してクライアントに返送する。

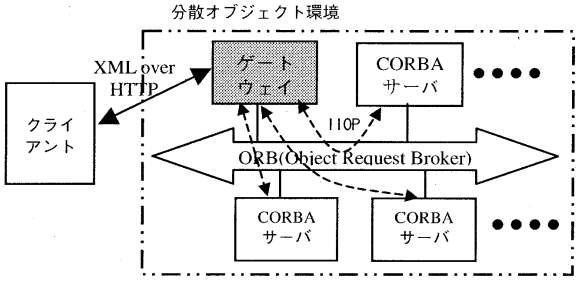

図1　XMLを用いたCORBAサーバのアクセス形態

3.3 IDL定義とXMLとの対応付け規則

基本方針(2)に従い、以下の規則でIDL定義をDTDに対応付ける。

(1)各モジュールを文書に対応させ、モジュールに含まれる全てのインタフェースを要素とする。

(2)継承しているインタフェースがある場合には、継承元の定義を展開する。

(3)インタフェースの要素では、定義中の属性と操作に着目し、属性の取得・設定要求(readOnly属性は取得のみ)、操作起動要求、ならびに、これらの要求の応答を要素とする。

(4)上記(3)の属性の要求・応答は、オブジェクトの名前(またはオブジェクトリファレンス)、属性名、属性型・値から構成する。操作の要求・応答は、オブジェクトの名前、操作名、パラメータ名、パラメータ型・値から構成する。

(5)属性型・値、パラメータ型・値の対応付は、表1のCORBAデータ型の対応付け規則に従う。ここで

Access Method for Distributed Objects using XML
Hiroki HORIUCHI, Shinji MOTEGI,
Masaru ENOMOTO and Toshikane ODA,
KDD R&D Laboratories Inc.

は、CORBA のデータ型を、型を示すタグと、値を示す PCDATA によるテキストとして表現する。

図 2(a), (b), (c)に、IDL 定義、対応する DTD 及び、XML インスタンスの簡単な例を示す。

表 1　CORBA のデータ型と XML の対応

CORBA のデータ型	XML テキストによる表現
整数型(long, short, unsigned)	10 進数
浮動小数点型(float, double)	IEEE 形式
文字型（char）	文字またはヘキサ形式
論理型（boolean）	“TRUE”または“FALSE”
Octet 型,any 型, string 型	文字列またはヘキサ形式
構造体（struct）	構成要素の型の値を列挙
共用体（union）	選択した型の値
列挙型（enum）	選択した識別子
シーケンス型（sequence）	構成要素の型のリスト

```
module Example {
   interface Account {   //インタフェース
      attribute long account_num;   // 属性
      long deposit (in long amount);   // 操作
      long withdraw (in long amount);   // 操作 };
   interface Manager {   // インタフェース 省略 };
   // 他のインタフェース省略 };
```

(a) IDL 定義例

```
<!ELEMENT Example ((Account | Manager | … )*>
<!ELEMENT Account (cos_naming ( operation_ind |
   attribute_get_ind | attribute_set_ind | operation_cnf |
   attribute_get_conf | attribute_set_conf )* )>
<!ELEMENT cos_naming ( reference | (naming_id,naming_kind?)+)>
<!ELEMENT operation_ind (operation_name, (in_param?, out_param?,
   in_out_param?, return_value? )*) >
<!ELEMENT attribute_get_ind (attribute _name, (long | … ))>
        中略
<!ELEMENT naming_id (#PCDATA)>
<!ELEMENT operation_name (#PCDATA)>
<!ELEMENT attribute_name (#PCDATA)>
<!ELEMENT in_param (type_name, ( long | … )>
        中略
<!ELEMENT type_name (#PCDATA)>
<!ELEMENT long (#PCDATA)>
```

(b) DTD 対応付け例

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE Example
http://www.KDD-R&D.co.jp/xml-corba/example.dtd>
<Example>
   <Account>
      <cos_naming>
            <naming_id>KDD</naming_id>
            <naming_id>Horiuchi</naming_id>
      </cos_naming>
      <operation_ind>
         <operation_name> deposit </operation_name>
         <in_param>
            <param_name> amount </param_name>
            <long> 1,000,000 </long>
         </in_param>
      </operation_ind>
      <attribute_get_ind>
         <attribute_name > account_num </attribute_name>
         <long></long>
      </attribute_get_ind>
   </Account>
</Example>
```

(c) XML インスタンス例

図 2　IDL 定義と XML との対応付け

3.4 ゲートウェイの概要

3.4.1 システム構成

基本方針(3)に従い、ゲートウェイは、XML インスタンスで渡されるパラメータ値や CORBA の動的起動インタフェース(DII)を活用して、対象となる IDL 定義が変更されてもプログラムを変更せずに対応可能なシステム構成とした(図 3)。なお、本ゲートウェイの実装では、Java Servlet を用いた Web アプリケーションとするとともに、XML パーサとして XML Parser for Java[3]を使用した。

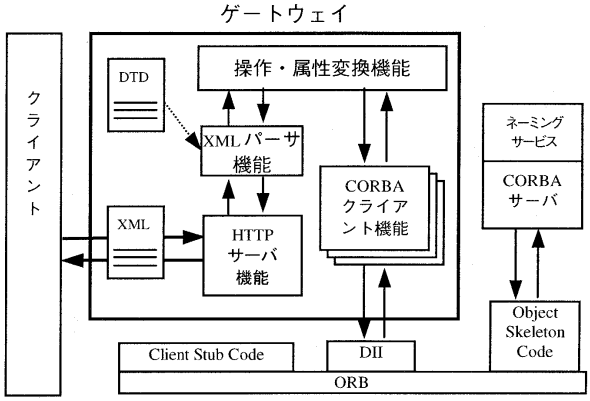

図 3　ゲートウェイのシステム構成

ゲートウェイの処理の流れを以下に示す。

(1)クライアントから受信した XML 文書に対しパースを行う。この時、DTD を使用した検証も行う。

(2)操作変換・属性変換機能では、エレメントハンドラ[3]を用いて、パース結果を解析し、パラメータ等を CORBA データ型に変換する。

(4)ネーミングサービスから処理対象のオブジェクトのオブジェクトリファレンスを取得する。

(5)DII を使用し、パラメータを設定の後、CORBA サーバに操作要求を送信する。

(6) CORBA サーバから操作応答を受信し、DOM(Documet Object Model)インタフェース[3]を用いて、XML 文書を作成する。

3.4.2 トランスレータ

本方式では、3.3 節で示したように IDL 定義の変更に応じて DTD も変わる。このため、図 4 に示すトランスレータにより、対象とするモジュール毎に DTD を用意するとともに、Web 上に公開し、クライアントによる XML 作成時の利用、ゲートウェイにおける XML インスタンスの検証時の利用を可能とした。

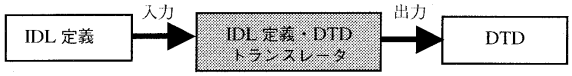

図 4　トランスレータの概要

4. おわりに

本稿では、XML を用いて CORBA サーバをアクセスする方式を提案した。ここでは、IDL 定義と XML 文書との対応付けを規定し、XML と IIOP を変換するゲートウェイを設けた。本ゲートウェイは、様々なオブジェクトに対応可能な汎用性をを持つ。最後に、日頃御指導頂く(株)KDD 研究所 村谷拓郎所長、鈴木健二副所長ならびに山本副所長に感謝します。

参考文献

[1]: Object Management Group, “The Common Object Request Broker: Architecture and Specification Rev. 2.2”, 1998. http://www.omg.org/.

[2]: W3C, “Extensible Markup Language (XML) 1.0”, 1998. http://www.w3.org/TR/REC-xml.

[3]: IBMCorporation, “XML Parser for Java”, 1998. http://www.alphaworks.ibm.com.