2R－3

Virtual Web Space Access を実現したWWWプロセスサーバのデザインと実装

中津 利秋　斎藤 淳　堀切 和典　川邉 恵久
富士ゼロックス株式会社 IT 事業開発部

1 はじめに

我々は、CGIより柔軟で分散的に協調動作する新しいWWWサーバの方式として、Virtual Web Space Access（以下VSA）を考案しVSAサーバとして、実装を行った。本稿では、VSAサーバの実現における、デザインと実装について報告する。

2 VSAの概要

インタネットやイントラネットでは、企業ごと、部門ごとに管理されるデータベースやWWWサーバについて、より上位のレイヤーで企業間や部門間にまたがるヴァーチャルな統合サービスを構築するというニーズがあり、WWWサーバ上の複数のサービスプログラムを分散的に連携させることが求められている。VSAは、このようなニーズに応える分散コンポーネントプログラム環境を提供する。

VSAは、コンポーネントプログラムを遠隔メソッドとして、順に起動するプロシジャをURLに埋め込む。このURLをヴァーチャルURL(VURL)と呼ぶ。

各VSAサーバは、VURLを分散プログラムとみなして、段階的に解析し、HTTPによって互いに受け渡しながら評価・実行する新しい計算モデルに基づく。

CGIプログラム・ブラウザ間のピアツーピアの結合と、VSAサーバ間のVURLによる遠隔メソッドの連鎖的な起動との関係は、ちょうど、TCP/IPにおけるソケットと、RPCの関係に相当する。

この計算モデルは、分散ネーミング機構が、分散プログラムの評価実行系と連動するように拡張したもので、設計の自由度が高い。およそ、名前を用いてアクセスするリソースすべてに適用できる。筆者らは、これをWWWに適用して実現を試みた。

実現にあたっては、WWWページの名前=VURLの表現能力が貧弱であると、VSAの計算能力を活かせない。また、コンポーネントプログラムとVSA機構とのインターフェースが特殊なものであると、従来のソフトウェア資産が活かせない。

そこで、VSAのコンピューティングモデルを反映した柔軟なVURLを実現することと、開発者がCGIやサーブレットなどのHTTPサーバとのインターフェースを知らなくても、コンポーネントプログラムの開発ができるプログラム環境の実現を、本研究の課題とした。

3 VURLの書式

VURLに、VSAのコンピューティングモデルに従って、階層的に分散コンポーネントのメソッド呼び出しをどのように埋め込むかを検討した。

オブジェクトによって分散処理サービスを提供するサーバを本稿ではプロセスサーバと呼ぶ。プロセスサーバのサービスをURLで指定するために、「オブジェクト名」「メソッド名」「引数」がURL内の式から得られる必要がある。

一方WWWでは、プログラム名と引数を含んだ以下のようなURLがCGIの呼び出しに利用されている。

```
http://hostname:port/cgi-bin/program?
arg1=val1&arg2=val2
```

上記URLでは、programが起動されるプログラム名、?以降の&で区切られた各文字列はそれぞれ「変数名=値」とみなされる。変数名と値を示す文字列はMIMEフォーマット「x-www-form-urlencoded」でエンコードされている必要がある。また、HTMLの<FORM>タグを利用すると、WWWブラウザから入力した値を適切にエンコードして、上記書式のURLでリクエストを送ることができる。

本方式では、既存のWWWブラウザからメソッドの引数を容易に変更できるように、CGIに習って以下のように書式を定めた。

```
http://hostname:port/context/object?
method=name&arg1=val1&arg2=val2
```

4 VSAサーバの設計

本実装では、HTTPサーバからJavaオブジェクトのメソッドを呼び出すために、Java Web ServerのServlet機構を利用してVSAサーバを実現した。VSAサーバで外部にメソッドを公開しているJavaオブジェクトをコンテキストと呼ぶ。

図1にVSAサーバの内部構成を示す。JWSはVURLによるリクエストを受けると、ServletのAPIでContextInvokerを起動する。ContextInvokerは、URLParserとVURLオブジェクトを利用してVURLを解析し、(オブジェクト名　メソッド名　引数)を得ると、それに該当する開発者定義コンポネントとメソッドを検索して呼び出し、結果をHTTPレスポンスとしてJWSへ返す。

開発者定義コンポネントの処理結果はSGMLComponentクラスのインスタンスからなるHTMLのドキュメントオブジェクトであるが、ContextInvokerはドキュメントオブジェクトからHTMLストリームを生成し、それをServletの呼び出し結果としてJWSへ渡す。

5 開発者定義コンポーネント

VSAサーバで公開するメソッドは、その引数の型として「整数(int)」、「文字列(java.lang.String)」、「URL(java.lang.URL)」、「ドキュメントオブジェクト(SGMLComponent)」あるいはこれらの配列が利用でき、メソッドの返り値はSGMLComponentとする。ContextInvokerはVURL中の引数をこれらの型に変換し

Design and Implementation of WWW process server based on Virtual Web Space Access
Toshiaki NAKATSU, Jun SAITO, Kazunori HORIKIRI, Shigehisa KAWABE
IT Bussiness Development, Fuji Xerox Co.,Ltd.

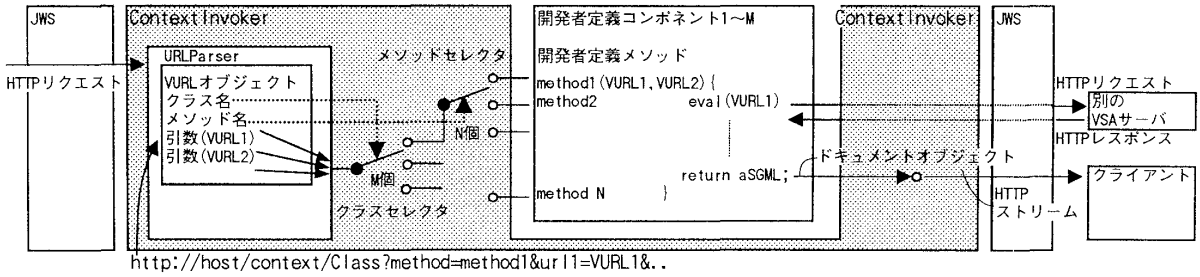

図 1: VSA サーバの構成

てメソッドを呼ぶが、特に、SGMLComponent の場合は引数の値を URL とみなして HTTP リクエストを発行し、その結果得られるSGMLComponent をメソッドへの引数とする。

開発者定義コンポネントはContextObject クラスのサブクラスとして実装する必要がある。ContextObject クラスは、プログラムの作成を支援するメソッドをいくつか提供しているが、VSA の分散処理に関係するものとしてeval とquote がある。

eval(URL *url*) は *url* で HTTP リクエストを発行して得られるSGMLComponent を返す。

quote(String *method*, Object *arg*) はそのクラスのメソッド *method* を引数 *arg* で呼び出すためのVURL を返す。

6 コンポーネントの組合せ例

図 2 に複数のコンテキストによる分散処理の例を示す。この例でブラウザは、コンテキスト B とコンテキスト C の処理結果を引数として、コンテキスト A のメソッドmerge を呼ぶ VURL でリクエストを発行している。

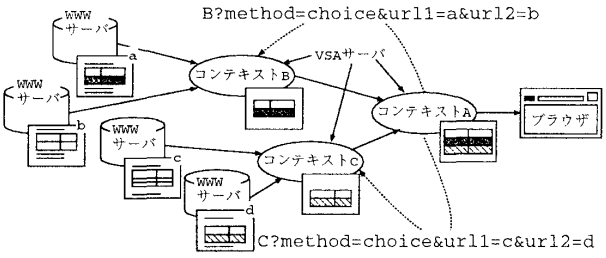

図 2: 分散処理の例

コンテキスト A は次のメソッドが呼ばれている。

```
public SGMLComponent
merge(SGMLComponent[] comps) {
 Table table = new Table();
 for( int i=0; i<comps.length; i++) {
  Table subtable = comps[i] から表を抜きだす;
  subtable の各行を table に追加;
 }
 return table.toComponent();
}
```

メソッドmerge は引数の型が配列なので VURL から任意個の引数を指定できる。また、配列要素の型はSGMLComponent クラスなので、ContextInvoker は VURL 中の引数を URL とみなし、それで得られるドキュメントをメソッドの引数とする。

コンテキスト B および C は次のメソッドが呼び出されている。

```
public SGMLComponent
choice(URL[] urls) {
 SGMLComponent result;
 for( int i=0; i<urls.length; i++) {
  SGMLComponent comp= eval(urls[i]);
  if ( condition(comps) )
   result= comps;
 }
 return result;
}
```

メソッドchoice も任意個の引数を指定できるが、配列要素の型はURL なので、引数はURL オブジェクトとしてchoice メソッドが呼ばれる。choice メソッドでは引数の URL からドキュメントオブジェクトを得るために、eval メソッドを呼んでいる。

7 まとめ

本稿では、VURL を実現し、HTTP サーバとのインターフェースに依存しないコンポーネントプログラムの開発環境を Java Web Server 上に実現した。今後の課題として、VSA サーバで公開するメソッドの引数の型の自由度を高めることを検討したい。

参考文献

[1] 堀切 和典, 川邉 恵久. Virtual Web Space Access の分散ネーム解決機構とコンピュータモデル, 情報処理学会全国大会論文集, 1999, 2R-04.

[2] National Center for Supercomputing Applications. The Common Gateway Interface, http://hoohoo.ncsa.uiuc.edu/cgi.

[3] Tim Berners-Lee and Robert Cailliau. World Wide Web Proposal for a HyperText Project, CERN European Laboratory for Particle Physics, Geneva CH, November 1990, http://www.w3.org/hypertext/WWW/Proposal.html.