

## WWW とアプリケーションの新たな協調モデル\*

5U-3

藤澤快哉<sup>†</sup> 乃木篤<sup>†</sup> 黒川利明<sup>††</sup>  
(株)CSK 技術本部<sup>†</sup> 日本アイ・ビー・エム (株) <sup>††</sup>

## 1 はじめに

本稿では WWW 上にクライアント/サーバ型のアプリケーションを構築する際の手法と問題点について論じ、それらの問題点に対する解決策として WWW とアプリケーションの新たな協調モデルを提案する。

## 2 従来手法

WWW 上にアプリケーションを構築する方法は CGI を利用する方法と WWW サーバの拡張モジュールを利用する方法に分類できる。それぞれのモデルをここでは CGI モデルと拡張モジュールモデルと呼ぶことにする。

## 2.1 CGI モデル

CGI モデルはさらに、CGI プロセスが WWW サーバとアプリケーションサーバのゲートウェイとして機能するモデルと CGI プロセス自体がサーバとして機能するモデルに分類できる。

第1のモデルの欠点は WWW サーバがアプリケーションサーバにアクセスする際に CGI プロセスを経由するために発生するオーバーヘッドである。CGI プロセスはアクセスのたびに生成されるため、このオーバーヘッドは大きい。

第2のモデルの欠点は CGI プロセスは非永続的であり、データをメモリ内に保持できない点である。データを保持するためには CGI プロセスは自身が消滅する前に必ずディスク等のデバイスにデータを書き込まなければならない。これによりディスク I/O が増大し、その結果、応答時間が増大する。

さらに両方のモデルに共通する欠点として、アクセスのたびに発生する CGI プロセスの生成がサーバマシンの負荷を増大させること、クライアントと WWW サーバ間および WWW サーバとアプリケーションサーバ間で複雑なデータ構造を受渡するのが難しいことが挙げられる。

## 2.2 拡張モジュールモデル

CGI の様々な問題点の解決策として拡張モジュールを利用するためのインターフェイスを持つ WWW サーバが増えている。アプリケーションサーバは拡張モジュールとして実装され、WWW サーバと同一のプロセス内で実行される。

この方法ではプロセスを新たに生成することによるオーバーヘッドと負荷の増大を避けることができる。また、アプリケーションサーバは WWW サーバと同じ時間だけ生存することができ、その間データを内部に保持し、その内部状態に応じた動作を行うことができる。さらに WWW サーバとアプリケーションサーバが同一プロセス内で実行され、メモリを共有することにより、複雑なデータ構造を双方から参照することができるメリットもある。

## 2.3 その他の協調モデル

拡張モジュールの代わりに外部プロセスを利用する手法もある。IntelliWeb[1] では WWW サーバと外部プロセスが直接通信を行うことによって CGI モデルのような大きなオーバーヘッドを回避している。外部プロセスを用いる方法は拡張モジュールモデルと比較して柔軟性は増すが、メモリ共有のメリットが失われる。

また、WWW 上のグループウェア Blend[2] ではグループウェア・サーバが WWW サーバのデータ領域に直接データを書き込む手法を用いているが、動的にコンテンツを生成して提供することはできない。

## 3 システム事例

我々は拡張モジュールインターフェイスを持つ WWW サーバである JavaWebServer [3] を用いて WWW 上の自習システム ITS(Interactive Training System) を設計、実装した。JavaWebServer は Java 言語で記述されており、Servlet と呼ばれる Java オブジェクトを拡張モジュールとして利用することができる。

ITS サーバは Servlet として実装され、JavaWebServer 本体を通じて学習教材を HTML ファイルとして提供する。教材の HTML はテキストや画像などを動

\*New Model for Cooperation between WWW and Applications

<sup>†</sup>Kaiya Fujisawa, Atsushi Nogi  
Technology Division, CSK Corporation

<sup>††</sup>Toshiaki Kurokawa  
IBM Japan

的に組み合わせて作成される。これは個々の学習者の要求や学習進度、理解度などに柔軟に対応するためである。ITS サーバは単に学習教材を提供するだけでなく、学習者の学習進度と成績のレポート作成、用語辞書と検索機能、学習者間で情報交換を行うためのフォーラム、ユーザ管理と教材管理などの機能を持っている。また、クライアントは Java アプレットとして作成されており、学習者はブラウザ上で学習を行うことができる。

このシステムを開発するにあたり、WWW と教育システムがどのように協調すべきかを検討した結果、我々は新たなモデルを導入することを決定した。

## 4 新たなモデルの導入

### 4.1 拡張モジュールモデルに残る問題点

拡張モジュールモデルは CGI モデルの多くの欠点を克服したが、未解決の大きな問題がある。それは、WWW 上のクライアント/サーバ型アプリケーションは HTTP の制約を受けるということである。クライアント環境として既存のブラウザを使用する場合、学習教材をダウンロードするためのプロトコルとして HTTP を使用しなければならないため、学習システムとしてのプロトコルも HTTP の範囲内で規定しなければならない。

### 4.2 2本の通信チャネルの使用による解決策

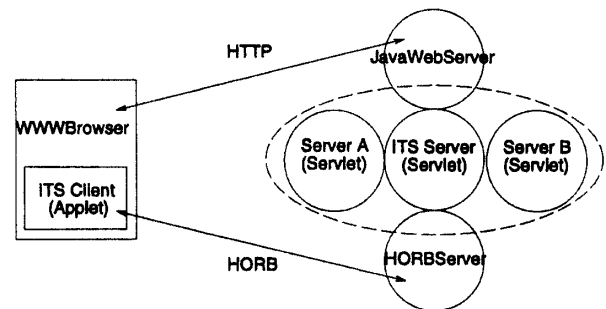
我々はクライアント/サーバ間に HTTP とは別の通信チャネルを設けることによってこの問題を解決した。教材の要求とダウンロードは HTTP を通じて行い、認証、レポート作成要求、用語検索、フォーラムの記事送信などその他全ての通信は別のコネクションを用いて行う。これによって教育システムとしてのプロトコルを規定して利用することが可能になり、そのプロトコルを用いてサーバからクライアントへの自発的な通知が可能になった。さらに、学習者に関する情報などを格納するデータ構造を WWW サーバと ITS サーバの双方から参照することができ、例えば学習者の成績によって異なる学習教材を動的に生成するといったことが可能になった。

ITS サーバが独自にサーバマシンのポートを利用することも可能であるが、より統一的で柔軟な解決策として Java 言語で記述された分散オブジェクト環境である HORB[4] を利用した。HORB は Java オブジェクトに異なる Java 仮想マシンからアクセスすることを可能にするものであり、ITS サーバをリモートオブジェクトとして HORB サーバに登録することによってクライア

ントである Java アプレットは同一の Java 仮想マシン内に存在するオブジェクトへのアクセスと同様の方法で ITS サーバのメソッド呼び出しが可能となる。

HORB を利用することにより、各アプリケーションごとにポートを用意し、従来のような形式で独自のプロトコルを規定する必要がなくなる。

我々が採用したモデルを図に示す。WWW とアプリケーションの新たな協調モデルとして提案する。



新たに導入したモデルを従来のモデルと比較すると次のような利点がある。

- アプリケーションに特化したプロトコルを規定して利用することが可能
- プロトコルはリモートオブジェクトのインターフェイスとして規定され、メソッド呼び出しによる通信が可能
- サーバからクライアントへの自発的な通知が可能

## 5 結論

本稿で提案した WWW とアプリケーションの新たな協調モデルの導入により、WWW 上のクライアント/サーバ型アプリケーションに独自のプロトコルを使用することができるようになり、より本格的なアプリケーションをより簡単に WWW 上に構築できるようになった。

## 参考文献

- [1] 牛嶋, 市来, 塩内, 高田, 毛利, 和田, 移動エージェントに対応したパーソナル WWW サーバ IntelliWeb, 情報処理学会第 54 回全国大会.
- [2] 藤澤快哉, WWW 上のグループウェアにおける添付資料機能の実現, 情報処理学会第 54 回全国大会.
- [3] <http://jserv.javasoft.com/>
- [4] <http://ring.etl.go.jp/openlab/horb/>