

## 三層クライアント/サーバ・システムにおけるミドルウェアの開発

4F-3

### (1) システムアーキテクチャ

鈴木 勇至 成田 秀明 松本 諭 関根 徹  
松下電器産業(株) マルチメディアシステム研究所

#### 1. はじめに

WS および PC によるダウンサイジングの広まりの中で、従来までのクライアント/サーバによる二層型のソフトウェアアーキテクチャから、三層モデルと呼ばれる新しいアーキテクチャが注目を集めている。本稿では、この三層モデルを採用した三層クライアント/サーバ・システムとその中で稼働するミドルウェアの開発について報告する。

#### 2. システムの概要

三層クライアント/サーバ・システムの構成を図1に示す。

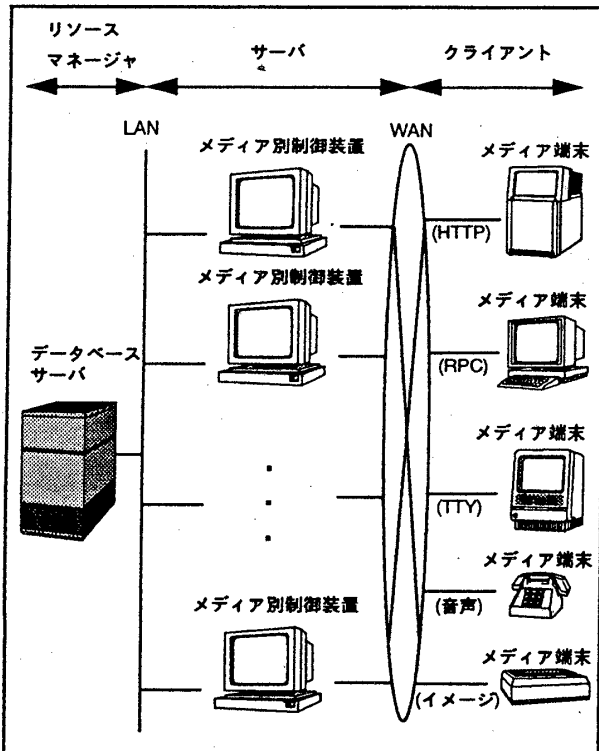


図1 三層クライアント/サーバ・システムの構成

リソースマネージャは、データ及びデータの操作手続きの管理を行なう。今回は、Oracle7 を搭載したデータベースサーバ (UNIX WS) をリソースマネージャに採用した。

サーバは、クライアントからの要求に応じて、リソースマネージャのデータに必要な処理を加えたり、クライアントに対しデータの提供を行なう。各種メディア端末ごとに対応した、メディア別制御装置と名付けた UNIX WS をサーバに採用した。

クライアントは、エンドユーザに対しサービスの提供を行なう。音声、イメージ、テキスト、ビットマップといったメディアごとに用意されたメディア端末を、クライアントと呼んでいる。

今回開発を行なったミドルウェアは、上記サーバ上で稼働するものである。

図2に、ミドルウェアの内部構成を示すとともに、以下に各構成要素について説明する。

#### (1) エージェント

エージェントはクライアント (端末) の制御を行なうプロセスで、端末の数と同数存在し、端末メディアに依存するアプリケーションと、トレーダとやりとりを行なう共通インタフェース部とからなる。共通インタフェース部はサービス (データベースに対する任意の処理) 単位で構成されるため、アプリケーション開発者は、データベースアクセスの複雑な作法を知らなくても、アプリケーションの開発が可能である。

#### (2) トレーダ

トレーダは、エージェントからの要求に従い、データベースサーバにアクセスを行なうプロセスである。トレーダは、サービス単位でデータベース内に用意されたストアプロシジャをコールする、データベースドライバで構成される。トレーダの数は、システムのコンフィギュレーションで変更することが可能である。

#### (3) キュー

キューは、エージェントとトレーダ間でメッセージ交換を行なうためのもので、共有メモリ上に実装され、独自の排他制御とメモリ管理機構を持つ。

Developing a Middleware in the Three Tier Client-Server System

(1) System Architecture

Yuji SUZUKI, Hideaki NARITA, Satoru MATSUMOTO, Toru SEKINE  
Matsushita Electric Industrial Co., Ltd.

(4) モニタ

モニタは、トレーダの起動やキューの作成といったシステムの初期化処理や、トレーダとキューに関する情報管理、統計情報の記録や表示といったシステム全体の管理機能をつかさどるプロセスである。

(5) エージェント管理サーバ

エージェント管理サーバは、エージェントの起動や監視といった、エージェントの管理機能をつかさどるプロセスである。

主な処理フローは次のようになる。

- クライアントから発せられた要求は、クライアントのメディアの種類に対応したエージェントに受け取られる。
- エージェントは要求の内容に応じて、どのトレーダに対し要求を送れば良いかをモニタに問い合わせた後、共通インタフェース部の中から適当なインタフェースを選択し、適当なトレーダの持つキューに要求データをエンキューする。
- トレーダはキューに入れられた要求データを取り出し、その内容に応じてデータベースドライバを用いてデータベースサーバに処理要求を送る。
- データベースサーバからトレーダに返された結果は、逆の順序を経て端末に返される。

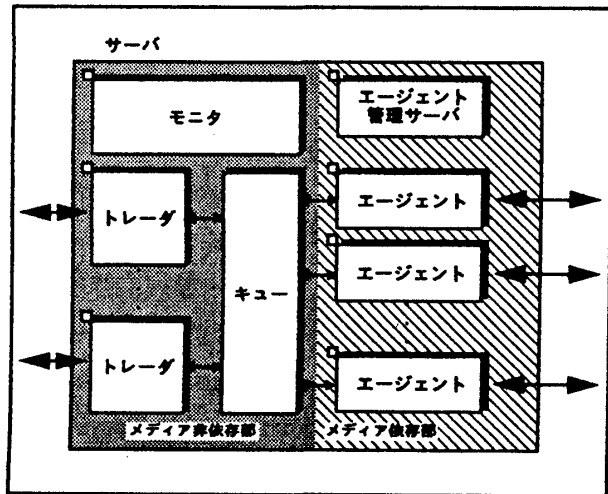


図2 ミドルウェア内部構成

3. システムの特徴

(1) マルチメディア端末のサポート

端末のメディア種別による処理手順およびデータ変換の差異は、エージェントで全て吸収され、トレーダはデータベースアクセスというメディアに依存しない働きをするため、新たなメディアの対応には、必要なメディアを制御するエージェン

トを開発するだけでよい。

(2) 開発コストの軽減と品質の向上

サービスの追加、及び新規システムの開発は、業務処理に依存した部分、すなわち必要なストアードプロシジャの作成とそれに対応した共通インタフェース、及びデータベースドライバをそれぞれエージェント、トレーダに組み込む作業に集約され、それ以外の部分はミドルウェア部品を共有できるため、開発コストの軽減と品質の一定化が図れる。

(3) システムの拡張性

端末数の増加に対しては対応するメディア別制御装置の増設、端末メディアの新設に対しては対応するメディア別制御装置を追加することで容易に対応できる。すなわちメディア別制御装置単位で対応ができるため、柔軟な拡張が可能である。

(4) システムの柔軟性

端末からの要求を処理するエージェントと、データベースサーバにアクセスするトレーダの数は m:n の関係にあり、トレーダの数は要求数には直接左右されない。すなわちシステム全体の状況 (CPU 負荷、リソースの空き状況等) に応じてトレーダの起動数を判断することが可能であり、様々な状況に応じた最適なシステムのコンフィギュレーションが可能となる。

(5) 負荷の軽減

データベースに対する要求の数は、常にトレーダ数の一定値となり、データベースサーバの負荷の軽減につながる。サーバ上でも、データベースアクセスを行なう比較的重いプロセスであるトレーダを一定数に保つことは、メディア別制御装置内の資源の節約や負荷の軽減につながる。

4. 今後の展開

今後は、以下のような拡張を行なう予定である。

- TP モニタ等の導入によるトランザクション機能の充実
- 他の DBMS や標準のファイルシステム、あるいは全文検索サーバ等によるリソースマネージャの多様化
- マルチスレッド化による高速化及び負荷の軽減

【参考文献】

[1] 成田 他、「三層クライアント/サーバ・システムにおけるミドルウェアの開発 (2) ~ (5)」情報処理学会第 51 回全国大会講演論文集