

3層クライアント / サーバ設計法の適用事例*

4 N-3

高田 信一 畑 恵介 山本 修一郎
NTT ソフトウェア研究所

1 はじめに

ハードウェアとネットワークの飛躍的な性能向上を背景として、クライアント / サーバーアーキテクチャによる情報処理システムが増えている。その中で、柔軟なシステム機能追加や業務拡張を実現するためのアプリケーションアーキテクチャ、“3層クライアント / サーバーアーキテクチャ”[1] が注目されている。

しかしながら、このアーキテクチャを適用したシステムを開発しようとしたときに、従来の単一的なアプリケーションプログラムを構成していた機能を、それぞれ

- どの層の機能として(配置)
- どのくらいのまとまりで(粒度)

切り出せばよいのかが問題となる。

このため、大規模システム設計への適用を目的として、この問題を解決していく設計手法[2]を検討した。本稿では、その概略を説明する。

2 層間インターフェースの導入

3つの層間に“層間インターフェース”を導入することにより、上述の問題点を明確にした。図1にそのモデルを示す。

ここで、インターフェースとは、上位層から下位層への要求と応答の対であり、以下のような性質を持つ。

識別子 要求情報であり、インターフェースを一意に識別するもの。識別子には機能が一意に対応する。

入力 要求情報であり、機能が必要とするデータ、機能の制御を指定するパラメタなどである。省略可能である。

処理結果 応答情報であり、機能の処理結果を判定する情報である。

出力 応答情報であり、機能の処理結果により得られた情報である。省略可能である。

また、プレゼンテーション層と機能層の間のインターフェースを P-F インタフェース、機能層とデータ層の間のインターフェースを F-D インタフェースと呼ぶこととする。

一般的に層間のインターフェースとして非同期通信や peer-to-peer なども考えられるが、ここで採用したインターフェースの性質は、多くのプログラム言語の手続き呼び出しや分散処理環境におけるリモートプロシージャコールなどに共通であるため、アプリケーションアーキテク

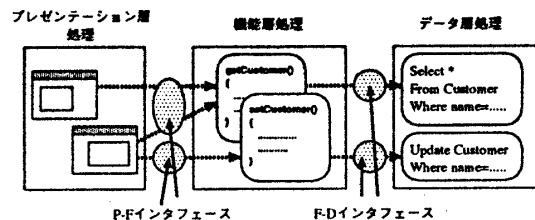


図1: 層間インターフェース

チャ上の論理的な設計結果を、さまざまなクライアント / サーバ環境上に構築することが容易である。

さて、このモデルの上で、「与えられた要求仕様に対して、適当な P-F インタフェース、F-D インタフェースを抽出する」ことが前述した設計問題の解となる。

3 層間インターフェース設計基準

以下では、配置の問題および粒度の問題を解決するために、層間インターフェース設計に基準を与える。ここで示す基準は総括的であり、実際の適用時にはさらに具体的な細目として設計者に与えられる。

基準 A ユーザインターフェースやウインドウの状態遷移に関する処理はプレゼンテーション層に配置し、機能層はそれらに依存しない汎用的な処理を配置する。

基準 B P-F インタフェースはプレゼンテーション層から汎用的に利用されるため、1回の P-F インタフェース呼び出しをトランザクションの原子性や一貫性を保証する単位とする。

基準 C F-D インタフェースは論理的なデータへの透過的なアクセスを提供する。データの物理的な格納場所、管理機構およびSQLなどのデータアクセス方法は機能層からは見えない。

基準 D 1つの F-D インタフェースは1つのデータ実体に対する1つの機能を提供する。

これらの基準はアプリケーションの論理的な構造を大きく決定するが、それにより加えられる制限も大きい。特に基準 D により結合演算を含む F-D インタフェースを使用できないが、処理性能を要求される大規模システムでは結合演算を積極的に利用しない[3]ため、実用上の問題はない判断した。

4 設計手順

これらの基準を適用し、要求仕様から P-F,F-D インタフェースを抽出する手順について概略を説明する。

*A Case Study of System Design Method based on 3-Tiered Client/Server Architecture
Shin-ichi Takata, Keisuke Hata, Shuichiro Yamamoto
NTT Software Labs.

まず、層間インターフェース抽出に先立ち、以下のような分析および設計を行う。

業務分析 システム化対象となる業務、業務の流れ、入出力情報の流れを明確にする。

データ分析 業務分析の結果からデータモデルを設計し、各業務でアクセスする実体、その属性、実体間の関連、一貫性を保証する制約、既存外部システムとのインターフェースを明確にする。

画面設計 業務分析およびデータ分析の結果から画面設計を行い、各業務を構成する複数の画面、それらの画面状態遷移、画面中のボタンなどと連動した機能、その入力情報および出力情報を明確にする。

この結果、システムを構成する全画面と画面ごとに必要な機能、入出力情報が明確になる。この情報から層間インターフェースを抽出する。以下の手順は画面中の機能ごとに適用していく。

P-F インタフェースの抽出 プレゼンテーション層を設計し、基準 A をもとに機能層の処理が必要となる画面中の機能を抽出する。これが P-F インタフェースとなる。あたえられた入出力情報と基準 B により P-F インタフェースの大枠が決まる。

アクセスする実体の抽出 P-F インタフェースの機能から機能層のおよびその処理を設計し、機能層からアクセスする実体(基準 D を適用)と機能を抽出する。ここでいう機能とは実体に対する更新、削除、追加、照会の処理種別と SQL の WHERE 文節に相当する処理条件である。

F-D インタフェースの設計 それぞれに基準 C を適用しながら F-D インタフェースおよびデータ層を設計する。

機能層の設計 設計されたF-Dインターフェースを使用して、P-Fインターフェースの機能に合致する機能層の処理を設計する。

最後に、全画面に対してこの作業を行い抽出した各層間インターフェースに対して、それぞれのインターフェース仕様を比較して等しいもの、包含関係にあるものを統合する。

5 考察

この手順による効率的な設計作業を支援するためにシナリオフロー図(図2)を考案し、これを利用して小規模なクライアント/サーバシステムを設計する実験を次のような手順で行った。

1. 画面設計まで、全員で検討する。
 2. 層間インターフェース抽出作業は、各作業者が独自に設計する。
 3. 全員で結果を持ちよりレビューを行い、層間インターフェースを統合する。

その結果、機能の配置や粒度が統一された層間インターフェースを抽出することができた。シナリオフロー図内の各層処理の記述内容については、詳細さに不統一が見られたが、レビュー内で記述レベルを統一することができ

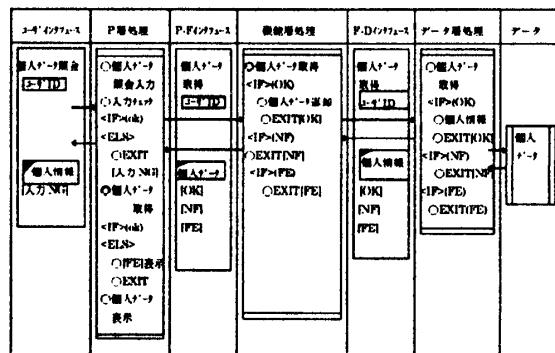


図2・シナリオフロー図

た。したがって、この設計手法を適用して、層間インターフェース設計を分担させることでより短期間で設計ができる見通しを得られた。

さらに、層間インターフェース抽出前には機能層をサバに配置する予定であったが、抽出後にクライアント側に配置するように実装したところ、層間インターフェース設計の手戻りは発生しなかった。

実験を通じて、シナリオフロー図の代表的なパターン毎に処理を概略的に記入したテンプレートを準備することで層間インターフェース抽出工数を削減できることがわかった。この結果を反映して、より効率的な手順への改善とともに、手順をさらに詳細化するための検討を今後進める。

6 あわりに

3層クライアント / サーバーアーキテクチャシステム設計において、設計基準を適用して層間インターフェースを設計する手順を提案した

設計する子順を捉えた。

今後、実験的な開発と改造を行うことにより開発効率やシステム変更に対する柔軟性の観点でこの手法を評価する予定である。

詩五

本検討を通して多大な協力を頂きました伊藤 路夫 情報システム本部部門長および平野 健一 グループリーダ、ならびに日頃ご指導頂く、細谷 優一 ソフトウェア研究所長および長野 宏宣 部長に深謝いたします。

参考文献

- [1] John J. Donovan.: "Business Re-engineering with Information Technology", Prentice Hall, ISBN 0-13-125907-5. (1994)
 - [2] 高田 信一, 山本 修一郎 : "3層クライアント / サーバシステム設計方法の提案", 電子情報通信学会技術研究報告, KBSE95-6, p.39-46. (1995)
 - [3] 赤間 浩樹, 石垣 昭一郎 : "ダウンサイジングにおけるデータベース利用の課題と対策", 情報処理学会研究会報告, 94-DBS-98, p.35-41. (1994)