

実践! 次世代



C/S

クライアント/サーバー

システム

スマートクライアント&Webサービスの実際

中垣 健志
NAKAGAKI, Kenji
株式会社CSKシステムズ
IT生産技術部

第2回

レイヤーアーキテクチャの雛形を作成する

Client / Server System



アーキテクチャの欠落

一時期、テレビや新聞は欠陥マンションの話題でもちきりでした。正しく設計されていないため、震度5の地震で倒壊してしまうというマンション。補強しようにも、そのようなことを想定して設計されていないため補強できず、最悪の場合は取り壊しになった物件もあるそうです。

ひるがえってエンタープライズシステムについて考えてみると、実は同じような状況かもしれません。開発前に

想定していた機能を実現できず、ほとんど稼動することも改修されることもないまま使われなくなってしまうシステムは決して少なくありません。そのなかには設計内容そのものが業務にマッチしていないために、使われなくなったシステムもあるでしょう。しかし設計自体は正しかったにもかかわらず、実装が正しく行なわれなかったために使えなくなってしまったシステムも数多く存在します。

犬小屋程度の小さな建築物であればDIY感覚で簡単に作るができるかもしれませんが。しかし高さ100m

の高層ビルを建築するためには、強度計算や耐震構造の仕組みなどを綿密に設計する必要があります。これはエンタープライズシステムも同様です。大きなシステムを構築するためには「アーキテクチャ」と呼ばれる基本的な構築方法を検討する必要があります。アーキテクチャを検討/設計するための作業は「方式設計」と呼ばれています。



レイヤーアーキテクチャを使った方式設計

前回の最後にお話ししましたが、Microsoftはエンタープライズシステムを設計/実装するための標準的なアーキテクチャとして「Application Architecture for .NET」(以下AAfN)を提供しています。これは.NET Frameworkに限らず、一般的に変化に強いエンタープライズシステムを構築するためによく使われるレイヤー構

レベル >>> Level

1 2 3 4 5

言語 >>> Language

- Visual Basic
- C#

ツール >>> Tool

- Visual Studio 2005 Professional
- SQL Server 2005

サンプル >>> Sample

この記事で取り上げたソースコードおよびサンプルプログラムは、<http://www.shoehisha.com/mag/windev/>からダウンロード可能です。

造を採用したアーキテクチャです。

レイヤーアーキテクチャを適用できるかどうかはシステム特性により判断しなければなりません。この連載で構築するシステムは「レイヤーアーキテクチャが妥当」と仮定して話を進めていきます（コラム「レイヤーアーキテクチャのメリットとデメリット」参照）。

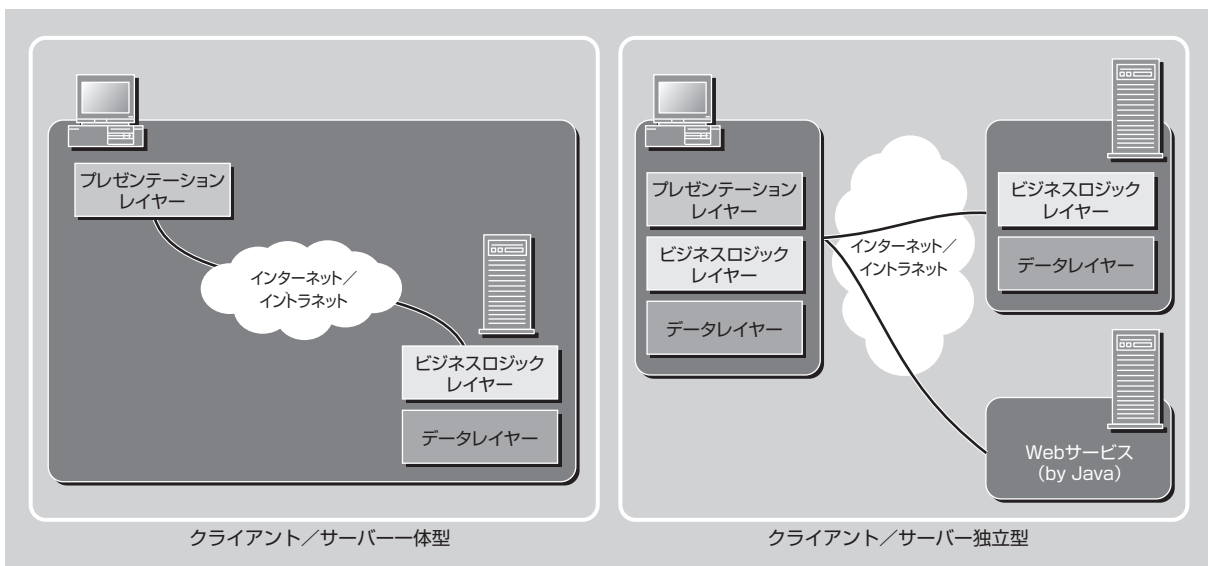
＊クライアントとサーバーの関係

方式設計の初期段階で、今回のシステムにレイヤーアーキテクチャとスマートクライアントの組み合わせを採用した場合、その次に決めなければならないのは「レイヤーをどのように構成するか」です。とはいえ、おおまかなレイヤー構造はAAfNにより決まっています。そのレイヤー構造は、クライアントとサーバーの関係という点から、以下の2つの形式に分かれます（図1）。

クライアント/サーバー一体型（本連載で取り上げる形式）

クライアントとサーバーをひとつのシステムとして見立てる形式を、ここでは「クライアント/サーバー一体型」として定義します。プレゼンテーションレイヤーとビジネスロジック/データレイヤーはインターネットまたはイントラネットで隔てられていますが、あわせてひとつのシステムになります。

図1：2つのレイヤーアーキテクチャ



レイヤーアーキテクチャのメリットとデメリット

レイヤーアーキテクチャはすべての開発に対して万能ではなく、メリットとデメリットが存在します。

メリットは、あるレイヤーの修正が他のレイヤーへ与える影響を少なくすることでアプリケーションが堅牢になることです。また、レイヤーごとに分業を行なうことも可能となります。したがって、大きなシステムを継続的に構築するとき用いると、大きな効果を上げることができま

す。逆にデメリットは、ひとつの機能が複数のレイヤーにまたがるためにコード量が増えることです。また、レイヤーの設計を事前に綿密に行なう必要があるため、設計に時間がかかります。したがって、市場のニーズに応じてすばやく機能提供しなければならないときには、レイヤーアーキテクチャでは効果を出しにくい場合があります。

クライアント/サーバー独立型

クライアントとサーバーを独立したひとつずつのシステムとして見立てる形式を、ここでは「クライアント/サーバー独立型」として定義します。クライアントとサーバーそれぞれで、レイヤー構成をとるアプリケーションとして設計します。

また、サーバーサイドでは.NET Framework以外にも、Javaで作られたWebサービスなどを利用することも可能