

# chapter - 3

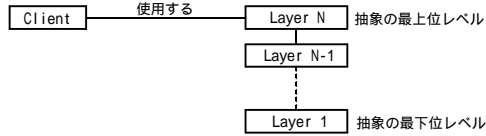
---

POSAアーキテクチャパターン一覧

[POSAのアーキテクチャパターン]

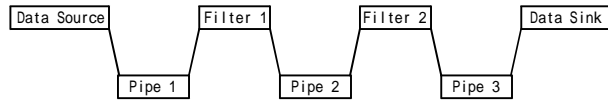
Layers

Layersアーキテクチャパターンは、アプリケーションを特定の抽象レベルに属するサブタスクのグループ群に分割するという構造化に有効である。



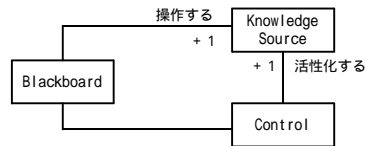
Pipes and Filters

Pipes and Filters アーキテクチャパターンは、データをストリームとして扱うシステムのための構造を提供する。個々の処理ステップは、1つのフィルタコンポーネントにカプセル化され、データはこれらの隣接するフィルタ間をパイプコンポーネントを介して引き渡される。フィルタの組み合わせを換えることにより、機能の類似した一連のシステムを作成することができる。



Blackboard

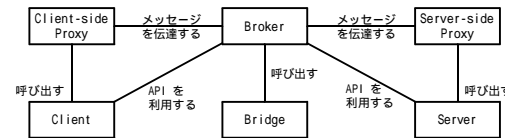
Blackboardアーキテクチャパターンは、決定論的な解決戦略が分かっていない課題に対して有効である。複数のサブシステムが持つそれぞれの知見を組み合わせることによって、部分的、もしくは、近似的に妥当な解を得ることができる。



出典：「ソフトウェアアーキテクチャ」, F. Buschmann 他, 近代科学社

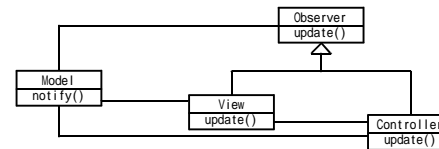
Broker

Brokerは、分散ソフトウェアシステムを構築するために利用できるアーキテクチャパターンである。互いに依存性を持たないコンポーネント群が、リモートサービスを起動することによって相互作用するという構造を持っている。このパターンの中心となるブローカコンポーネントは、リクエストの転送、レスポンスや例外の送信などの通信上の協調を行う責務を負っている。



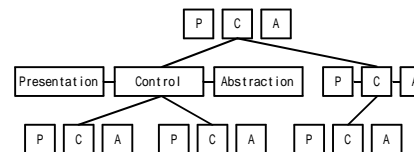
MVC(Model-View-Controller)

MVCアーキテクチャパターンは、対話型アプリケーションを3つのコンポーネントに分割する。モデルコンポーネントは、アプリケーションの中核機能とデータを含むコンポーネントである。ビューコンポーネントは、情報をユーザに表示するコンポーネントである。そして、コントローラは、ユーザの入力を取り扱うコンポーネントである。ユーザインタフェースを実現するのは、ビューコンポーネントとコントローラコンポーネントである。更新伝播のメカニズムによって、ユーザインタフェースとそのモデルとの一貫性を保証することができる。



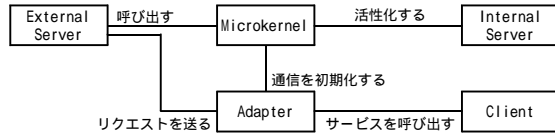
PAC(Presentation-Abstraction-Control)

PACアーキテクチャパターンは、協調するエージェントの階層という形で、対話型ソフトウェアシステムのための構造を定義する。このエージェントはその1つずつがアプリケーション機能の特定の一面を実現する責務を負っており、3つの部分から構成される。3つの部分とは、Presentationコンポーネント、Abstractionコンポーネント、Controlコンポーネントである。これらのコンポーネントへ分割することにより、エージェントとの通信から、人間とコンピュータの対話機能が分離される。



### Microkernel

Microkernelアーキテクチャパターンは、変更されるシステム要件への適合が求められるソフトウェアシステムに用いられ、システムの核となる最小限の機能を、拡張機能や顧客依存部分から分離する。また、さまざまな拡張機能を読み込んだり、その拡張機能が協調して動作できるように調停機能を提供したりする、一種のソケットとしての役割も果たす。



### Reflection

Reflectionアーキテクチャパターンは、ソフトウェアシステムの構造と振舞を動的に変更するメカニズムを提供する。このパターンを利用すると、型機構や関数呼び出し機構といった基礎的な部分からの変更が可能となる。このパターンはアプリケーションを2つの部分、メタレベルとベースレベルに分割する。メタレベルはシステム属性に関する情報を提供し、ソフトウェア自体を自己認識可能にする。ベースレベルはアプリケーションの論理を含み、メタレベル上に実装される。メタレベルの保持する情報に対する変更は、結果的にベースレベルの振舞に影響を与える。

