

オブジェクト指向フレームワーク開発手法

早瀬 健夫, 三木 正章, 村田 由香里, 松本 一教

hayase@sitc.toshiba.co.jp

株式会社 東芝 情報・社会システム社 SI技術開発センター

1. はじめに

近年、オブジェクト指向技術は、再利用性や拡張性などのメリットから広く普及し、特に再利用技術としてオブジェクト指向フレームワーク（以下、単にフレームワークと呼ぶ）が注目されてきた。フレームワークは、アプリケーション間に共通な部分の実装を共有可能なスケルトンとしてモジュール化し、アプリケーションに固有な部分のみを拡張することで、安定した品質のソフトウェアを効率良く開発することをねらいとしている。現在では、GUI や OS などの汎用的な事例だけでなく、ドメイン向けの事例も増えている。しかし、フレームワークを開発することは、単に一つのアプリケーションを開発することに比べて、非常に高度なノウハウが必要である。

本稿では、フレームワーク開発の難しさを軽減するために、フレームワーク開発手法を提案する。

2. フレームワーク技術の現状

現在、フレームワーク開発手法に関する提案[1]があるが、必ずしも充分整備されていない。機能、構造、振る舞いの視点から次に示す問題が存在する。

第一に、機能的な側面としてドメイン分析が重要である。アプリケーションごとにカスタマイズすべき箇所（ホットスポット）を特定する必要がある。現在、ホットスポットとなる部分に対応するオブジェクト群を特定した後に、ホットスポットを満たすフレームワーク設計をサポートする技術は提案されている[1]。しかし、ホットスポットとなる部分をオブジェクトモデルにマッピングする技術について触れていない。第二に、構造的な側面としてプラットフォームの変化やトレードオフとなるシステム要件に対応する構造に関する議論が充分にされていない。

第三に、振る舞いの側面として実装メカニズム（ホワイトボックス型とブラックボックス型）を選択する基準に関する議論が充分にされていない。

3. フレームワーク開発手法

2節の課題に対し3ビューモデル手法を提案する。

3.1. 3ビューモデルとは

フレームワーク開発のための3ビューモデルを提案する。図1に着目すべき視点（ドメイン分析、レイヤ、メカニズム）を示す。既に我々は、参照モデルを導入した開発手順[2]、レイヤ構造を導入した開発手順[3]を提案しており、本稿はこれらを強化した。

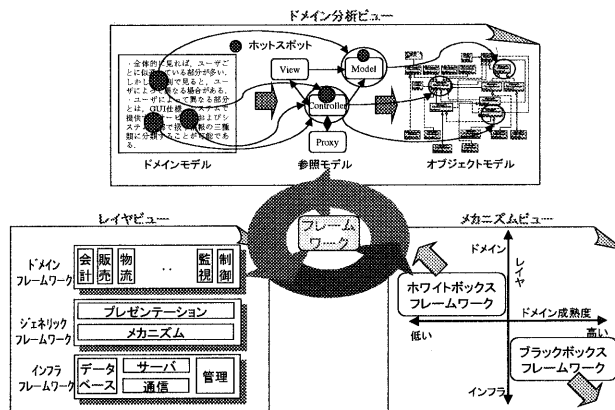


図1 フレームワークのための3ビューモデル

ドメイン分析ビュー ドメインモデルのホットスポットを最終的にオブジェクトモデルにマッピングする必要がある。両者の間に抽象度のギャップがあるため、参照モデルという新たなモデルを導入し両者のギャップを埋める。参照モデルとは、あるドメインのアプリケーション構造を端的に表現した共通の論理構成を示す再利用技術である。図1（上側）では、ドメインモデルのホットスポットを参照モデルにマッピングした上で、参照モデルからオブジェク

トモデルへマッピングする様子を示している。
レイヤビュー システムによっては、再利用性だけでなくリアルタイム性などの実装上の制約を考慮する必要がある。そこで、図 1 (左側) に示すように、フレームワークにレイヤ構造 (インフラ, ジェネリック, ドメイン) を持たせる。インフラレイヤは、インフラに依存した機能を実現する。ジェネリックレイヤは、汎用的に用いる機能を実現する。ドメインレイヤは、ドメインに依存した機能を実現する。
メカニズムビュー 図 1 (右側) にメカニズムを選択する基準を先の二つのビューから示す。一つはドメインの成熟度である。ホワイトボックスは未成熟なドメイン向き、ブラックボックスは成熟したドメイン向きである。もう一つは対象とするレイヤである。インフラやジェネリックレイヤでは普遍的な機能を実装するのでブラックボックス向きである。ドメインレイヤでは一般に将来のドメインの変化を特定することは難しくホワイトボックス向きである。

3.2. フレームワーク開発手順

3.1節のモデルに基づく開発プロセスを図 2 に示す。

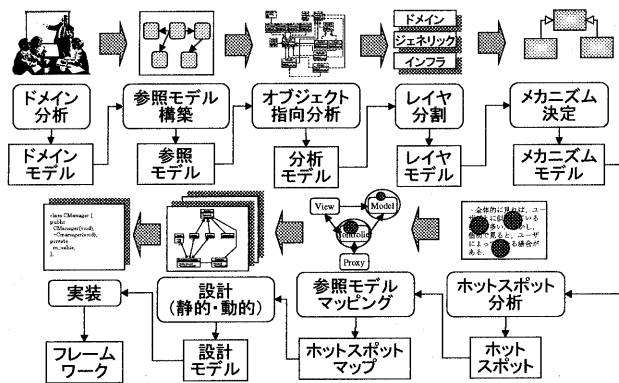


図2 フレームワーク開発プロセス

1. **ドメイン分析** ドメイン知識を整理する。
2. **参照モデルの構築** ドメインモデルに基づき、システム構造を参照モデルとして構築する。
3. **オブジェクト指向分析** 参照モデルに沿って、オブジェクトを抽出する。
4. **レイヤ分割** 必要なレイヤを定義する。
5. **メカニズム決定** レイヤごとあるいはレイヤ内部のカテゴリごとにメカニズム方式を決定する。

6. **ホットスポット分析** ホットスポットを特定する。
7. **ホットスポットマッピング** ホットスポットと参照モデルのモデル要素の対応を明らかにする。
8. **フレームワーク設計** ホットスポットをマッピングした参照モデルに基づき設計する。
9. **フレームワーク実装** オブジェクト指向言語を用いて実装する。

4. 適用事例

3節の手法を監視システムのフレームワーク開発に適用した。ここでは、「2. 参照モデルの構築」に関し述べる。MVCP (Model-View-Controller-Proxy) と名づけた図 3 に示す参照モデルを導入した。Model はドメインのデータを表す。View は Model の視覚的な情報を表す。Controller はシステムのサービスを提供する。Proxy は監視対象機器との通信を担う。参照モデルにより、ドメイン分析で抽出したホットスポットをソフトウェア構造レベルで端的に表現でき、ホットスポット分析からフレームワーク設計へスムーズに進めることができた。

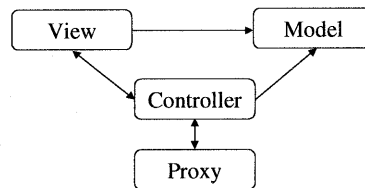


図3 MVCP モデル

5. おわりに

本稿では、3 ビューモデルに基づくフレームワーク開発手法について述べた。本手法により開発をスムーズに進めることができる。今後は、実開発に対する適用をさらに進め、手法の改良・強化を行う。

参考文献

- [1] Free, W. : Design Patterns for Object-Oriented Software Development, Addison-Wesley, 1994.
- [2] 早瀬健夫 他:産業分野の IT ソリューションー参照モデルに基づくフレームワーク開発手法一, 情報処理学会第 60 回全国大会, No.4, pp. 265-266, 2000.
- [3] 早瀬健夫:オブジェクト指向フレームワークと製品特化 CASE による組み込みシステム開発, 電気学会論文誌, Vol.119-C, No.12, pp.1573-1581, 1999.