

3S-06 共通コンポーネントの部品化とアプリケーション構造標準化による 企業内基幹システムの構築

松本茂[†]、今村大輔[†]、馬淵誠司[†]、川上智博[‡]、林博仁[‡]、杉浦宏幸[‡]

[†](株)東芝 [‡]中部電力(株)

1. はじめに

近年 実システム開発にも利用が進み始めたオブジェクト指向技術により、再利用性の高い部品開発が可能になりつつある。中でもデザインパターンとフレームワークの考え方が有効であると言われてい[RLP]。しかし、状況に応じてパターンを使い分けたり、再利用性を考慮したフレームワークを開発するには、オブジェクト指向開発に関するスキルが必要となる。

我々は企業内基幹システムの再構築にあたり、以下の目標を設定した。

- システムで頻繁に利用される基本機能を部品化し、プログラムの再利用性を高める
- オブジェクト指向開発経験の浅い開発者でも、適切なパターンやフレームワークを適用したアプリケーション構築を可能にする

この目標実現のため、以下のような特長を持つ部品化とアプリケーション構造の標準化を行った。

- カスタマイズ性の高いコンポーネント指向型フレームワーク[APF]による部品化
- デザインパターンを組み合わせた、アプリケーションの標準構造を定義(雛型化)

本稿では、上記の部品と雛型を伝票発行システムの開発に適用した事例について、システム全体の中の部品の位置付け、雛型の概略について述べる。

2. システム基本機能の部品化

システム内で何度も利用される共通的な機能を切り出すため、開発対象となるシステムの要件を整理した結果、以下の特性として纏められた。

- (1) 互いに機能的に独立した複数の業務が存在する

- (2) 各業務処理では、業務の種類によらず、共通して必要となる処理が存在する。即ち データエントリの GUI、データベースアクセス処理、帳票出力処理、実行ログ出力処理、リソース管理処理、など。
- (3) 上記処理は比較的カスタマイズが必要無い定型的な処理と、見た目や扱うデータ内容などで業務毎に様々なカスタマイズが必要な処理がある

これらの分析結果から、システムで共通的に利用される基本機能を以下の2種類の方法で部品化する方針とした。

- カスタマイズが少ない機能は、利用方法の修得が簡単なクラスライブラリ型の部品化
- 多彩なカスタマイズを必要とする機能は、ライブラリ型に比べると利用方法が難しいがカスタマイズの可能なフレームワーク型の部品化

フレームワークについては、継承によるカスタマイズを基本とする whitebox 型フレームワークと、複数のオブジェクトの組み合わせでカスタマイズを行う blackbox 型フレームワークが知られている。このうち、blackbox 型フレームワークは whitebox 型よりも利用方法を修得し易いがカスタマイズ性が低い、という傾向を持っている。この問題を解決する技術として、利用方法修得コストが比較的安く、しかもカスタマイズ性が高いという特長を持つコンポーネント指向型フレームワークが提唱されている。そこで本開発では、コンポーネント指向型フレームワーク技術を採用した。

上記のような方針の基に、2種類の部品化技術を用い、以下の8種類の機能を基本部品として開発した。

コンポーネント指向型フレームワーク：

画面管理部品、データ管理（リアル）部品、
印書管理部品、コード管理部品、文字コード変
換部品

クラスライブラリ：

リソース管理部品、データ管理（バッチ）部品、
接続・トレース部品

3. アプリケーション構造の標準化

アプリケーションを開発するには、適切な部品やデザインパターンを組み合わせ、クラス構造を設計する必要がある。これはオブジェクト指向開発の経験が少ない開発者にとっては困難な作業である。そこで、アプリケーション全体のクラス構造の標準化を目的として、開発対象システムの業務処理要件を分析しアプリケーションの構造を整理した。すると、データエントリ処理、事例検索処理 など、少数の種類に纏められることが分った。

そこで、複数のデザインパターンを適宜カスタマイズして組み合わせ、4種類の雛型を定義した。

4つの雛型は全て MVC パターンを中心とした基本構造を基に派生したクラス構造を持つ。雛型の一例を下図に示す。

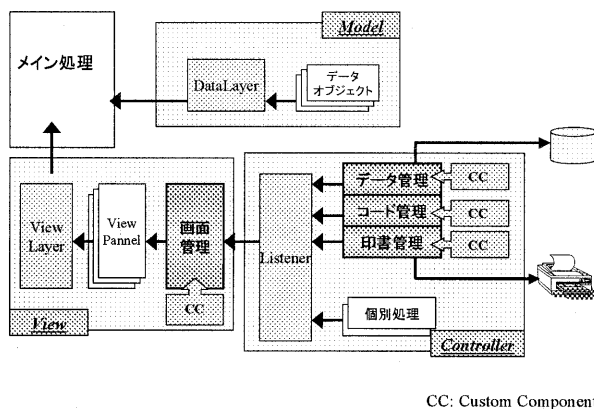


図 1 基本部品を組みこんだ雛型の一例

この雛型の基本構造は、アプリケーション全体を Model, View, Controller の各ブロックに分けるクラス構造であり、MVC の各ブロックで管理される複数の画面(View)、データ(Model)、ビジネスロジック(Controller) を管理して互いに関連付ける方法とし

て、他のデザインパターンを応用している。

4. システム開発への適用

8種類の基本部品と4種類の雛型を伝票発行システムの開発に適用した。本システムは中部電力(株)の次期事務処理系基幹システムの一部であり、部品及び雛型の有効性を評価する先行開発事例となっている。

伝票発行システムの開発開始時、開発者の中にはオブジェクト指向技術での開発経験が浅いメンバーが含まれていたが、業務要件に併せて雛型を選択し、そこに適用されているデザインパターンを利用して業務ロジックを組みこんで行くことにより、プログラム開発を進めて行くことができた。

また、業務の内容に応じてカスタマイズを行うことにより、システム内の各所で基本部品を利用することができ、基本部品の高いカスタマイズ性を確認できた。

5. まとめ

本稿では、コンポーネント指向型フレームワークを中心としたソフトウェア部品と、複数のデザインパターンを組み合わせたアプリケーション構造の雛型をシステム開発に適用した事例について、部品化と雛型化の考え方について述べた。

本稿で報告した方法によって部品化・雛型化を行うことにより、様々な業務処理固有のカスタマイズを行いつつ、基本機能部分の再利用性を高めることができた。また、アプリケーション構造の標準化を実現することができた。

今後は、基本部品の拡充及び雛型の汎用性向上を行い、他システムへの展開を図っていきたい。

[参考文献]

[RLP] Ralph E. Johnson, 中村宏明, 中山裕子, 吉田和樹:パターンとフレームワーク, 共立出版, 1999.

[APF] 斎藤 他:コンポーネントベース・フレームワーク技術(C Solution APF)全体構想とアーキテクチャ, 情報処理学会第60回全国大会, 1-241, 1999.