

ライフサイクル管理業務に注目した アプリケーションフレームワークの抽出と再利用効果の考察*

島田 圭 中所 武司†

明治大学大学院理工学研究科基礎理工学専攻情報科学系‡

{shimada,chusho}@cs.meiji.ac.jp

1 はじめに

ネットワークの普及により、分散システムの構築が注目されているなか、私達はその支援方法としてアプリケーションフレームワークに着目し、その開発方法について研究を進めている [1] .

本稿では、開発するフレームワークの問題領域として定めたライフサイクル管理業務のモデル化、フレームワークの抽出実験、さらに他のライフサイクル管理システムを構築したときのフレームワークの再利用効果について考察する .

2 ライフサイクル管理業務

2.1 モデル化

物を人に貸し出すという業務をライフサイクル管理業務としてモデル化した . 管理対象は以下の 3 種類である .

- 物 = 管理の対象となる物品
- 人 = システムを利用する人間
- 貸出 = 人が物を借りる事象

これらの登録 (購入 , 入会 , 貸出) から削除 (廃棄 , 退会 , 返却) までをそれぞれのライフサイクルとみなす .

2.2 例題システム

まずライフサイクル管理業務に該当する図書管理業務を取り上げ、その業務の一連を自動化した図書管理システムの開発から始めた . このシステムは、図書を借りる人のための利用者用システムと図書および利用者の管理のための管理者用システムからなる .

*Application Framework for Life-cycle Management and its Reusability.

†Kei SHIMADA and Takeshi CHUSHO

‡Computer Science Course, Major in Sciences, Graduate School of Science and Technology, Meiji University.

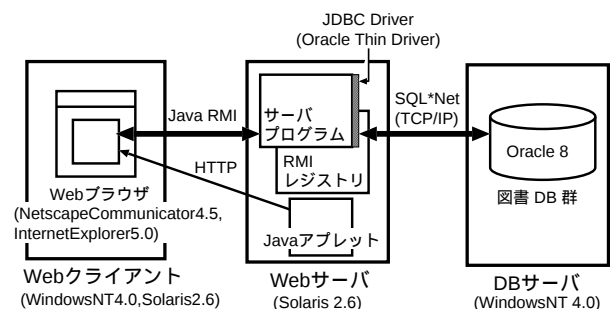


図 1: 図書管理システムの構成

2.3 構成

開発した図書管理システムは、図 1 のような構成で、アプレットとブラウザ-DB 間を取り持つサーバプログラムからなる .

アプレットでは、GUI や処理を行なう SQL 文の設定をしている . アプレットで設定された SQL 文は、サーバプログラムを介してデータベースに送られる . サーバプログラムは、SQL 文の実行や検索結果の取得を行なうメソッドを持ち、アプレットからの要求を受けてデータベースを操作する .

3 フレームワークの抽出実験

3.1 概要

アプリケーションフレームワークは、問題領域に共通なフローズスポットと個々のアプリケーションに依存するホットスポットに分けられる [2] . 本実験では、図書管理システムの開発とそのプログラム構造の改良を通じてフレームワークの抽出を行なった .

	ホットスポット	フレームワーク全体	ホットスポット占有率
初期のフレームワーク	2,683 ステップ	3,432 ステップ	約 78%
GUI クラスの抽出	1,163 ステップ	2,820 ステップ	約 41%
カスタマイズ箇所の抽出	818 ステップ	3,033 ステップ	約 27%

表 1: ホットスポット占有率の推移

3.2 初期のフレームワーク

図書管理システムでの処理は、SQL 文を用いたデータベース操作により行なわれる。その SQL 文による処理過程をアプレットで実装するのに、継承機能による差分プログラミングを用いた。

システムの登録、貸出、削除などの各機能画面における記入項目のチェックから SQL 文の実行、結果取得までの処理過程は、各機能共通である。そこで、処理過程を抽象クラスで実装し、それを継承する各機能の具象クラス（機能クラス）で処理内容を実装した。システム全体をフレームワークとして見た場合、抽象クラスがフローズンスポットになる。

3.3 プログラム構造の改良

3.3.1 GUI クラスの抽出

GUI の設定を機能クラスとした場合、同じ記述が複数の機能クラスの間に散在し、無駄なホットスポットを作ってしまう。

そこで、GUI の設定に関する記述を抽出して抽象クラスで実装し、それを機能クラスが継承する形にプログラム構造を変えた。このとき、抽出されたクラスがフローズンスポットになる。

3.3.2 カスタマイズ箇所の抽出

フレームワークのホットスポットとフローズンスポットはクラス単位で分離されていた方がカスタマイズが容易になる。そこで、1 つのクラスにカスタマイズ箇所と再利用可能箇所が混在している機能クラスからカスタマイズ箇所を抽出して新たなクラスとした。

機能クラスでのカスタマイズ箇所は、SQL 文記述と検索結果表示の部分である。SQL 文は、1 文に対して参照メソッドを作り、それらを新たなクラス (sql クラス) にまとめることで抽出できる。検索結果表示は、メソッドごと抽出し、新たなクラス (search クラス) に持たせる。この検索結果表示は、上位の GUI クラスで実装したものに結果を渡すようにしているので、search クラスは GUI クラスを継承する仕組みにした。これ

らの抽出により、sql クラス、search クラスがホットスポットに、機能クラスがフローズンスポットになった。

3.4 実験結果

先に述べた抽出実験をした結果を表 1 に示した。本稿では、フレームワーク内におけるホットスポットの占有率を再利用性の目安とし、値が小さいほど再利用効果が高いものとした。

4 フレームワークの再利用効果

今回開発したフレームワークを備品管理システムの開発に適用し、以下の再利用効果を確認した。

- 機能設計の容易さ
問題領域を限定したことにより、業務に関する機能を網羅できた。そのためシステム開発者が機能設計する必要がない。
- カスタマイズの容易さ
sql クラスと search クラスの 148 ステップ (全体の約 5%) のプログラム開発だけで済んだ。

5 おわりに

今後は、多くの例題システムへの適用実験を行ないながら、フレームワークにおける問題領域と適用範囲の関係について検討していく。

参考文献

- [1] 藤原克哉, 中所武司: エンドユーザ向き分散アプリケーションフレームワーク wwHww の開発と適用評価, オブジェクト指向最前線'98, 朝倉書店, pp.116-123(1998)。
- [2] Wolfgang Pree 著, 佐藤啓太, 金澤典子訳: デザインパターンプログラミング, トッパン (1996)。