

通信アプリケーションのフレームワーク化の検討

キヤノン株式会社

沼上 幸夫

numakami.yukio@canon.co.jp

開発における問題点

【背景】通信技術の進展、通信プロトコルの乱立、組み込み機器における通信アプリケーションが拡大

【課題】通信プロトコル毎に「似て非なる通信アプリケーション」を開発してしまい、重複コードが存在（複数コードの修正による保守コストの増大）

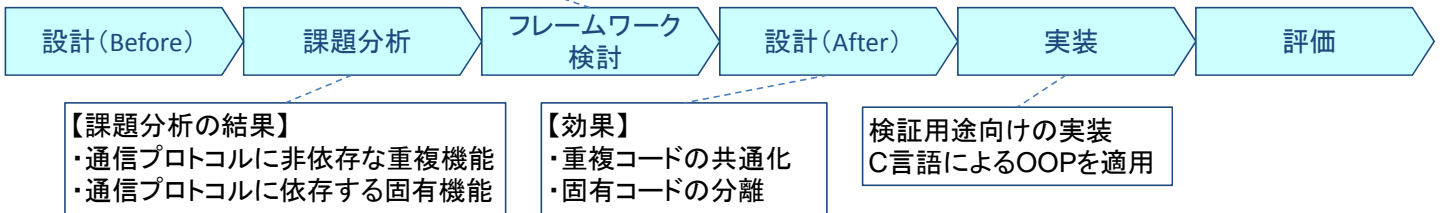
手法・ツールの適用による解決

【目的】重複コードを共通化

【解決手法】フレームワークを適用
・ソフトウェアの骨格（構造、振る舞い）を再利用し、可変部分を拡張（分離）可能にする手法

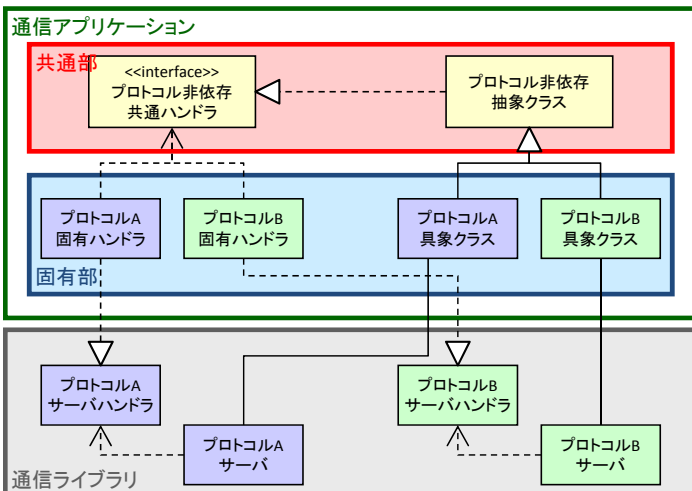
アプローチ

Template Methodパターンを応用



フレームワークの特徴

- ◆【特徴1】通信プロトコル非依存の抽象クラス
 - ▽ 重複機能を「共通機能」として括りだし
 - ▽ 通信プロトコル依存の具象クラスにて「固有機能」を実装
- ◆【特徴2】通信プロトコル非依存の共通ハンドラ/フ
 - ▽ 通信プロトコル依存の固有ハンドラクラスを定義し、共通ハンドラ/フとのマッピング



評価

- ◆ 1. フレームワーク導入時の初期コスト
 - ▽【方法】現状コードサイズのBefore/After比較
 - ▽【結果】通信アプリケーションのコードサイズが2倍に増加
 - ▽【考察】検証用途向けで最低限のロジックしか実装せず、仕組み（ハンドラ処理）が相対的に大きくなった。製品実装なら初期コストは相対的に小さくなる。
- ◆ 2. フレームワーク導入後の再利用による削減効果
 - ▽【方法】将来コードサイズ（見積り）のBefore/After比較
 - ① 既存通信アプリケーションに対する機能拡張
 - ② 新規通信アプリケーションの追加
 - ▽【結果】①約40%削減（プロトコル数4の場合）
②約15%増加（2プロトコル追加時）
 - ▽【考察】①対応プロトコル数が多い程、削減効果大きい
②1.と同様の理由。製品実装なら削減効果が大

今後の課題

- ◆ 新規固有機能の追加に伴うフレームワークへの影響
- ◆ 修正に伴う機能の妥当性の検証
- ◆ 関数呼び出し増加に伴うリソース増大・実行速度低下
- ◆ アスペクト指向手法との比較・評価