トップエスイー修了制作

ソフトウェアパターンを活用した設計による、ネットワーク機器運用保守機能の変更容易性向上

富士通株式会社　小林　和敏　kobayashi.kazut@jp.fujitsu.com

開発における問題点

複数種類のネットワーク機器(NE)を組み合わせてシステムを構築すると、複数のベンダ混在により操作性が低下してしまう。そこで、操作性を維持するため、保守運用の操作を集約する機能(以降、本機能)を導入している。
本機能は**変更容易性が高い**ことが要求される。

手法・ツールの適用による解決

本提案は開発完了した実際の機能に対して、本来の設計を探求し、現状の設計に対する新たな知見を得ることを目的とする。
例えば**本機能に有効なソフトウェアパターン**が発見できれば今後の機能追加時に適用することができる。

アプローチ方法

実現機能の整理 → 機能性を満たす設計を**ICONIXプロセス**で実施

増設／減設
電源管理／温度管理
バックアップ
ファイル管理
NE監視
：

変更要件の整理 → 変更要件を満たす設計を**ソフトウェアパターン**を活用して実施（対象はA,Bに絞る）

A. NE種類追加
B. 操作追加
C. プロトコル追加
D. 自動操作の追加
（対象）

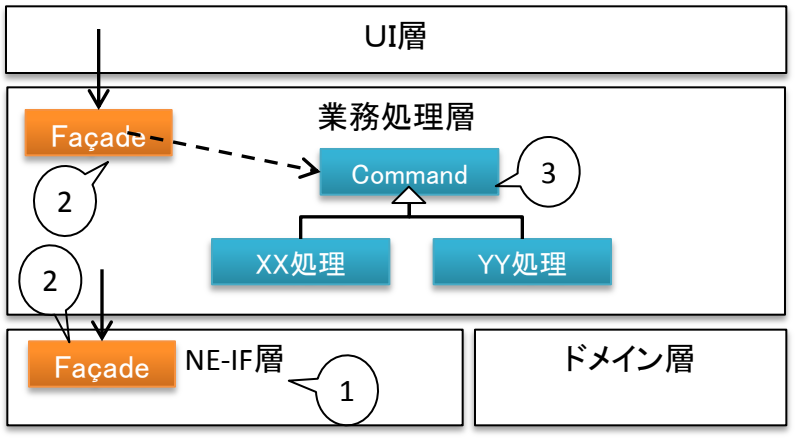
→ 機能追加をシュミレーションし、設計を**検証**

適用パターン

①アーキテクチャ：拡張レイヤーアーキテクチャ
本機能の特性と変更要件Aから、NE-IFを独立させ、各機能変更時の相互影響を最小化

②クラス設計１：Façadeパターン
レイヤーアーキテクチャによる層分割をサポートし、層間の関係を最小化

③クラス設計２：Commandパターン
「命令」と「命令の集合体」を同列に扱い、変更要件Bを更に向上



評価結果

- 変更要件Ａ：NE種類の追加
Façadeパターンによる隠ぺいにより、NE-ID層のFaçadeから呼び出す新しい種類NE機能に対応するクラスを追加し実現可能。影響範囲をNE-IF層内に止めた。

- 変更要件Ｂ：増設後バックアップ
Commandパターンより、既存機能を組み合わせた機能追加が容易に実施可能で、また影響範囲を業務処理層内に止めた。

➡ **設計効果を確認**

今後の展望

- 適用パターン発見
➡ 今後、機能追加時に適用していく

- 残変更要件への適用
➡ 変更要件CについてはFaçadeパターンで実現されているかもしない。変更要件DについてはDecoratorパターンを活用することで更に変更容易性が向上する可能性がある。