

ユースケースを基に変更容易性の高い設計を行う

株式会社日立アドバンストシステムズ
株式会社東証システムサービス
NECソリューションイノベータ株式会社

山科元希 genki.yamashina.bg@hitachi.com
坂田宏輔 k-sakata@tssx.co.jp
保坂広樹 hir-hosaka@xr.jp.nec.com

開発における問題点

システム開発では、近い将来にはワークフローを含むビジネスモデルの変更を予定していたり、その変更が確定ではなかったりする。そこで、将来のビジネス拡大によるシステム変更を考慮した設計を当初より組み入れる必要がある。

手法・ツールの適用による解決

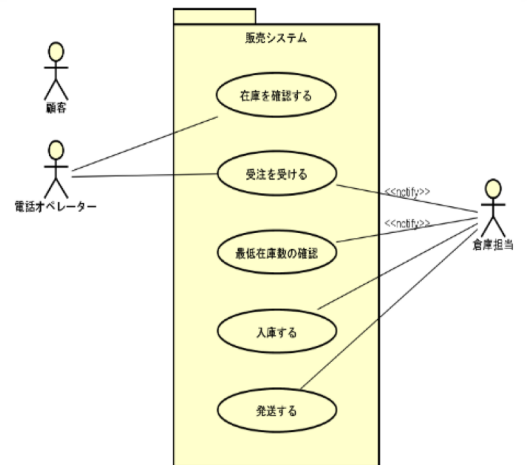
ドメイン及びシステムの現状の振舞いを整理し、対象システムの変更可能性を分析した。分析した結果を用いて、ソフトウェアデザインパターンを適用し、変更に強い設計を行った。

アプローチ

1. ユースケースを基にドメイン分析を行い、システム化の範囲を決め設計モデルを作成
2. ビジネスモデルの変更可能性を考慮して、デザインパターンを適用
3. 仕様変更提示 (講師から提示される)
4. 仕様変更の影響範囲を確認し、設計モデルを修正する

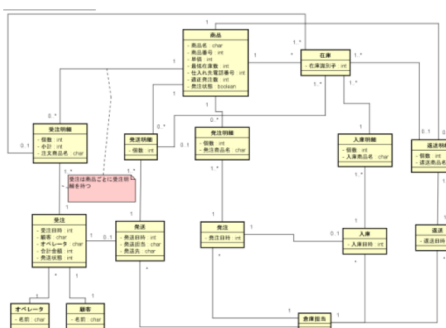
対象システム

- 通信販売の受注と発注のシステム化
- 受注時に在庫を確認し、在庫がないものは販売しない
- 決められた数値以下の在庫になったときに発注をかける
- 支払いは全て代引き

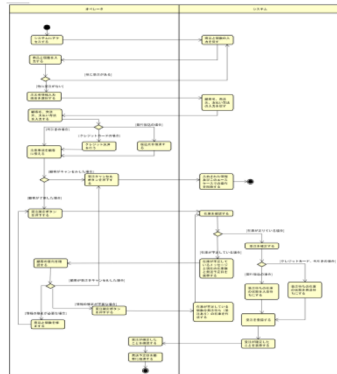


想定した変更と適用したデザインパターン

ドメイン分析



振る舞い分析



ユーザインタフェースの変更を局所化したい

保存先の変更を局所化したい

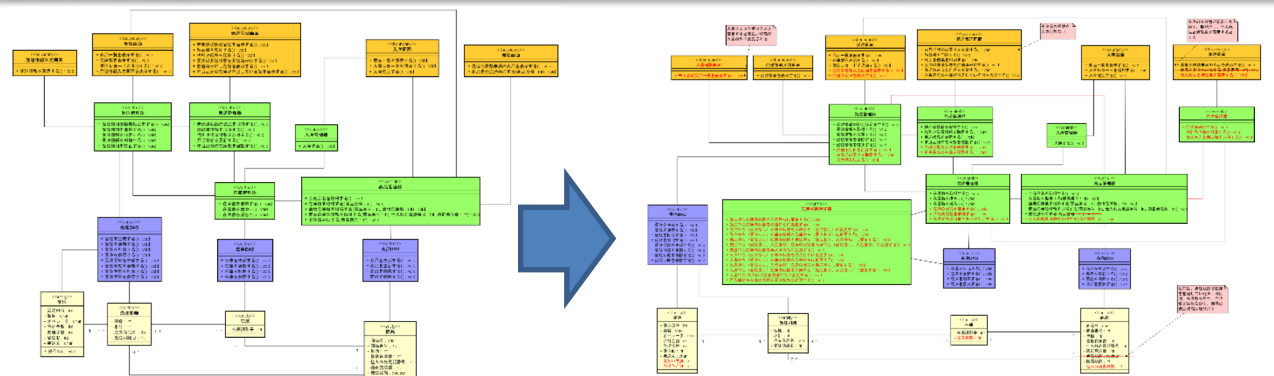
Layersパターン

DAOパターン

仕様変更内容とデザインパターンの効果

【変更内容】

- 在庫に関係なく受注可能にする
- 支払方法に銀行振込、クレジットカード払いを追加
- 銀行振込は入金確認後、クレジットカードは決済後、代引きは即時に出荷する



- Layersパターンの効果で影響範囲を自レイヤもしくは隣接したレイヤ内で吸収できており、局所化ができた
- 保存先は変更なかったためDAOパターンの効果はなし