SPI Japan 2013

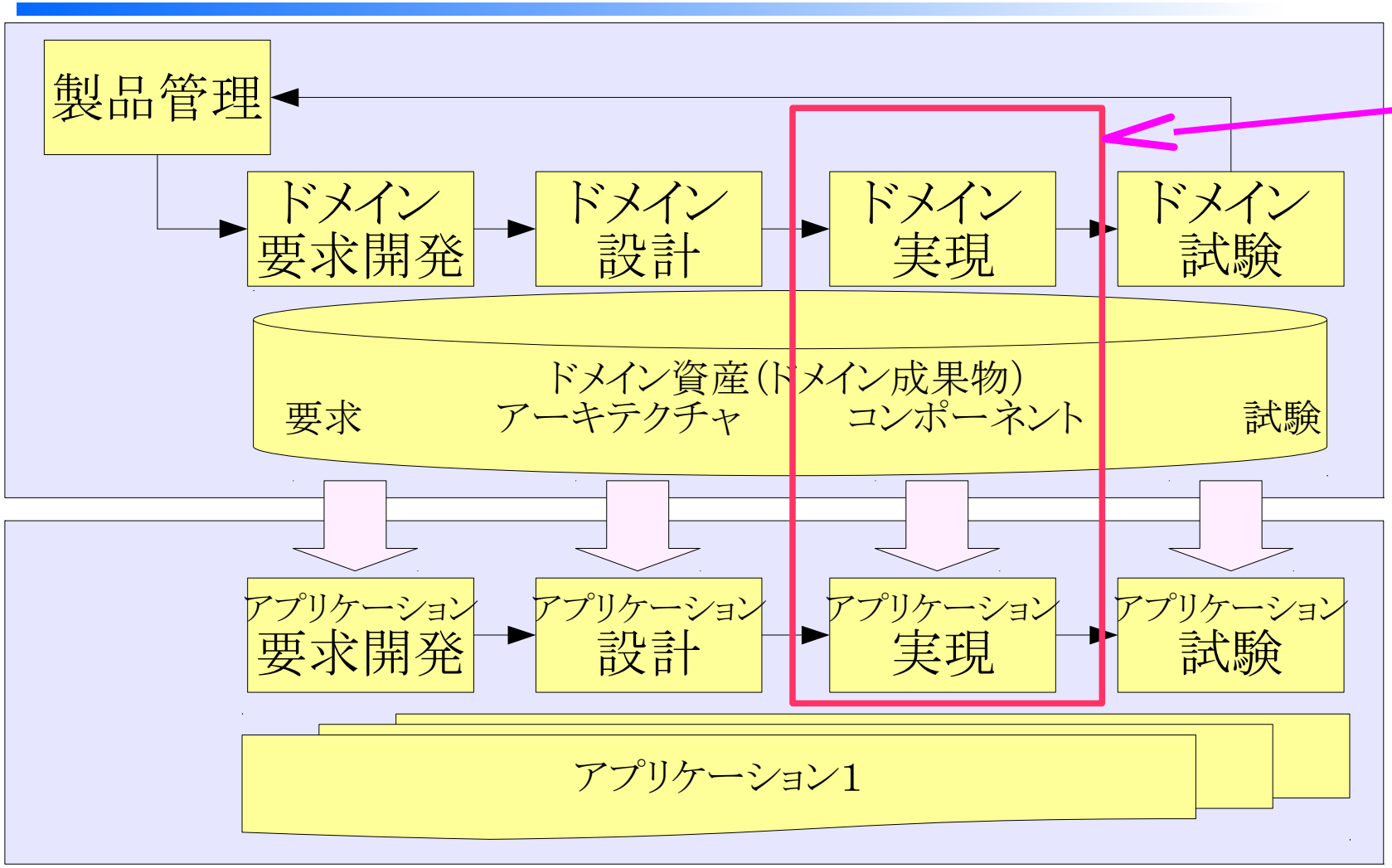
Software Product Line の実践

プログラム開発の効率化を目指した設計資産の構築

住友電気工業株式会社
情報システム部 情報技術部
システム技術グループ
川口　晃史
2013-10-17

背　景

『ソフトウェアプロダクトラインエンジニアリング』より

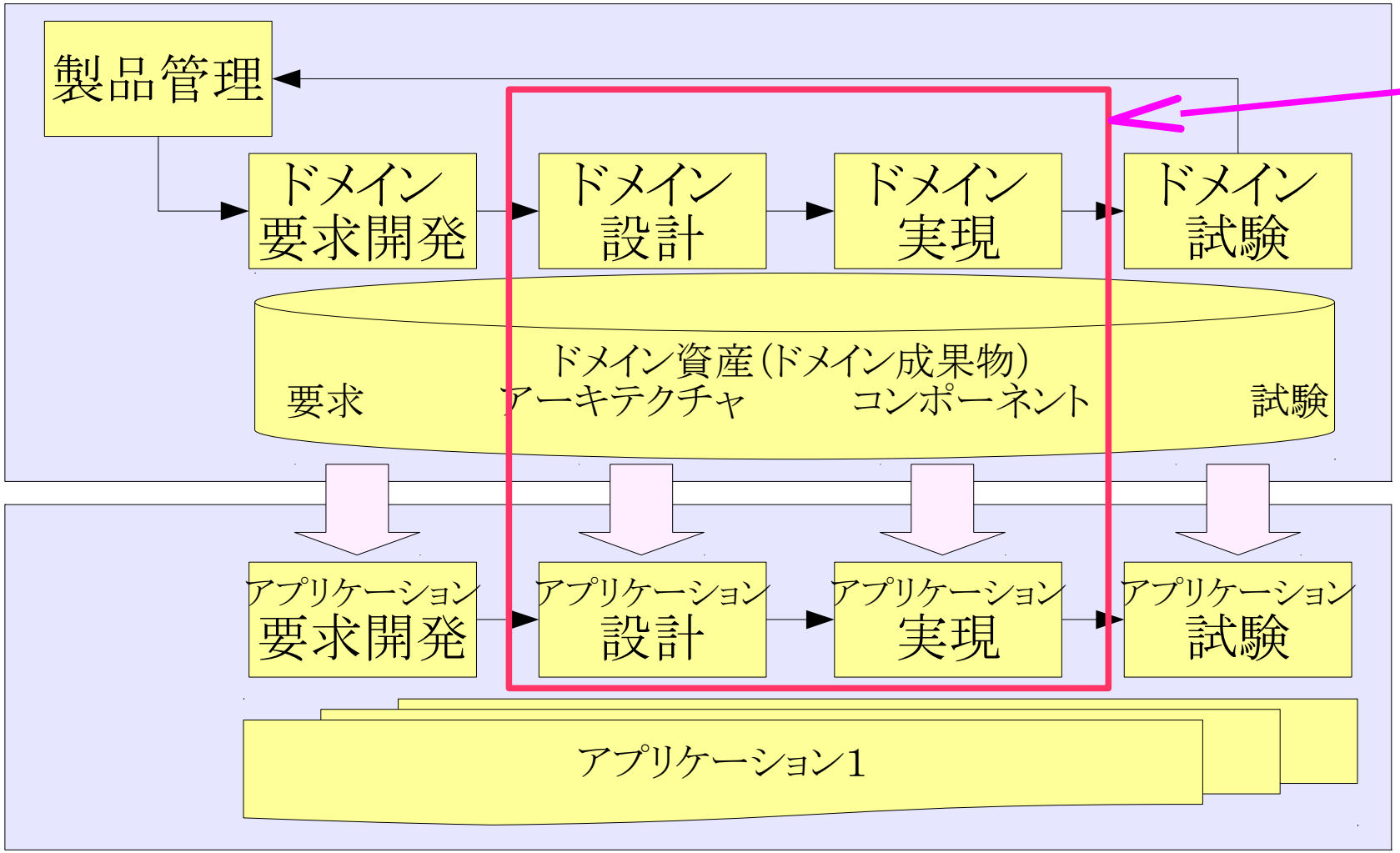


これまでの適用範囲

10年間取り組んでいる近年改善度合いが鈍化

引用元:『ソフトウェアプロダクトラインエンジニアリング　―　ソフトウェア製品系列開発の基礎と概念から技法まで』
クラウス・ポール (著), ギュンター・ベックレ (著), フランク・ヴァン・デル・リンデン (著), 林 好一 (翻訳), 吉村 健太郎 (翻訳), 今関 剛 (翻訳)

『ソフトウェアプロダクトラインエンジニアリング』より



「設計工程」に範囲を広げる

引用元:『ソフトウェアプロダクトラインエンジニアリング　―　ソフトウェア製品系列開発の基礎と概念から技法まで』
クラウス・ポール (著), ギュンター・ベックレ (著), フランク・ヴァン・デル・リンデン (著), 林 好一 (翻訳), 吉村 健太郎 (翻訳), 今関 剛 (翻訳)

改善の取り組み

改善前の作業概要

	外部設計工程	PG設計工程	開発工程
画面	レイアウト情報を画像と共に文書で記述	レイアウト・遷移情報をExcelで記述	レイアウト・遷移情報を部品組み立て定義ファイルで記述
入出力	入出力情報を文書で記述	データベーステーブル・項目をExcelで記述	データベーステーブル・項目を部品組み立て定義ファイルで記述
業務ロジック	日本語で記述	日本語で記述	Java言語で記述

※部品組み立て定義ファイル：
　業務システム開発で使用するフレームワークが解釈する定義ファイル
　部品の構成情報などを記述

改善点の検討

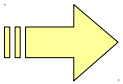
設計情報を後工程で活用すれば何が改善できるか？

- 改善点１　工程毎に同一内容を変換して記述
  - 変換作業自体が無駄
  - 変換ミスの発生
    - 不具合の原因
    - 保守段階での影響分析が困難

- 改善点２　開発工程での付加価値を生まない作業
  - Excelの仕様書から実装箇所を探しだす作業
  - 項目名などの変換ミスによる不具合対応

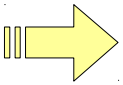
対応策と評価点

- 改善点１　工程毎に同一内容を変換して記述



設計情報をデータベースで一元管理

- 改善点２　開発工程での付加価値を生まない作業



Java開発環境への設計情報を提供

- 評価点
  - (a) 設計工程の生産性を維持
  - (b) 変換作業なく設計開発ができる
  - (c) 業務ロジック開発工数が削減

設計情報のデータベース化

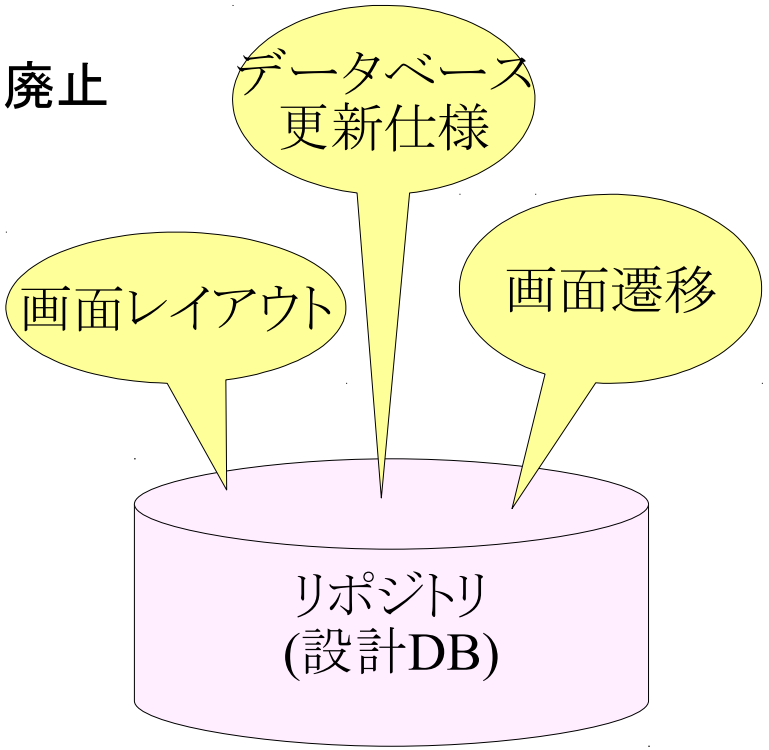
設計情報のデータベース化（リポジトリ構築）

- リポジトリで設計情報を一元管理
  - 外部設計段階から設計情報を登録
  - 後工程では既存情報をそのまま利用する

  ⇒ Excelや部品組み立て定義ファイルを廃止
  変換作業の無駄をなくす

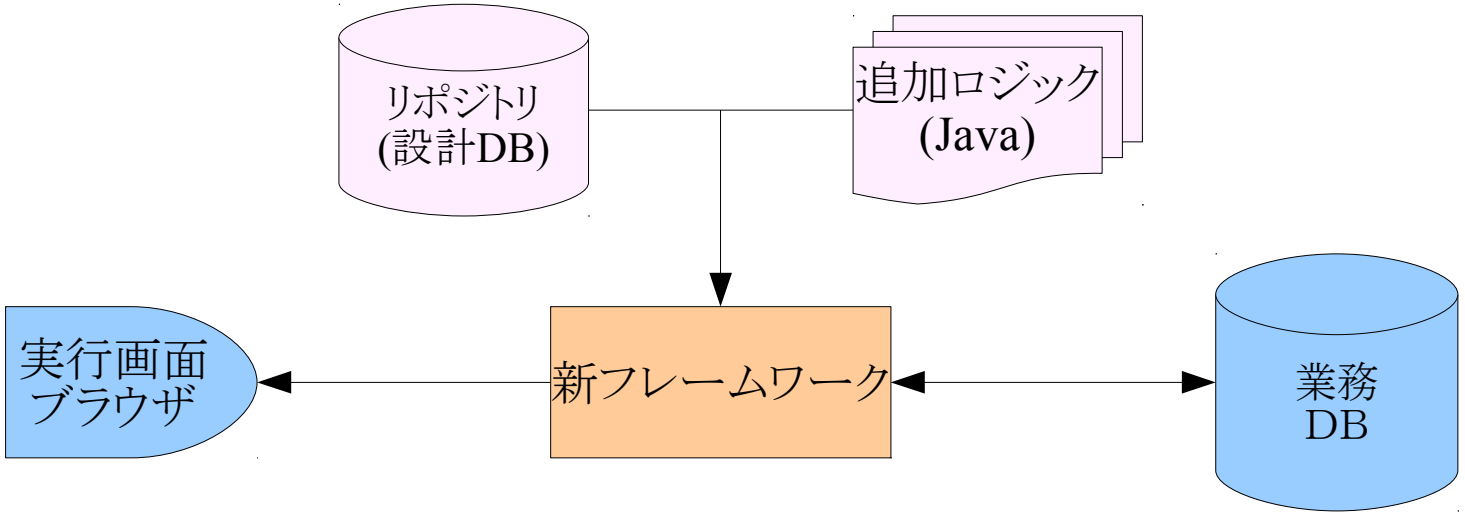
- 設計情報の登録をシステム化
  - 設計者はブラウザから設計作業が可能

  ⇒ 設計作業をスムーズに

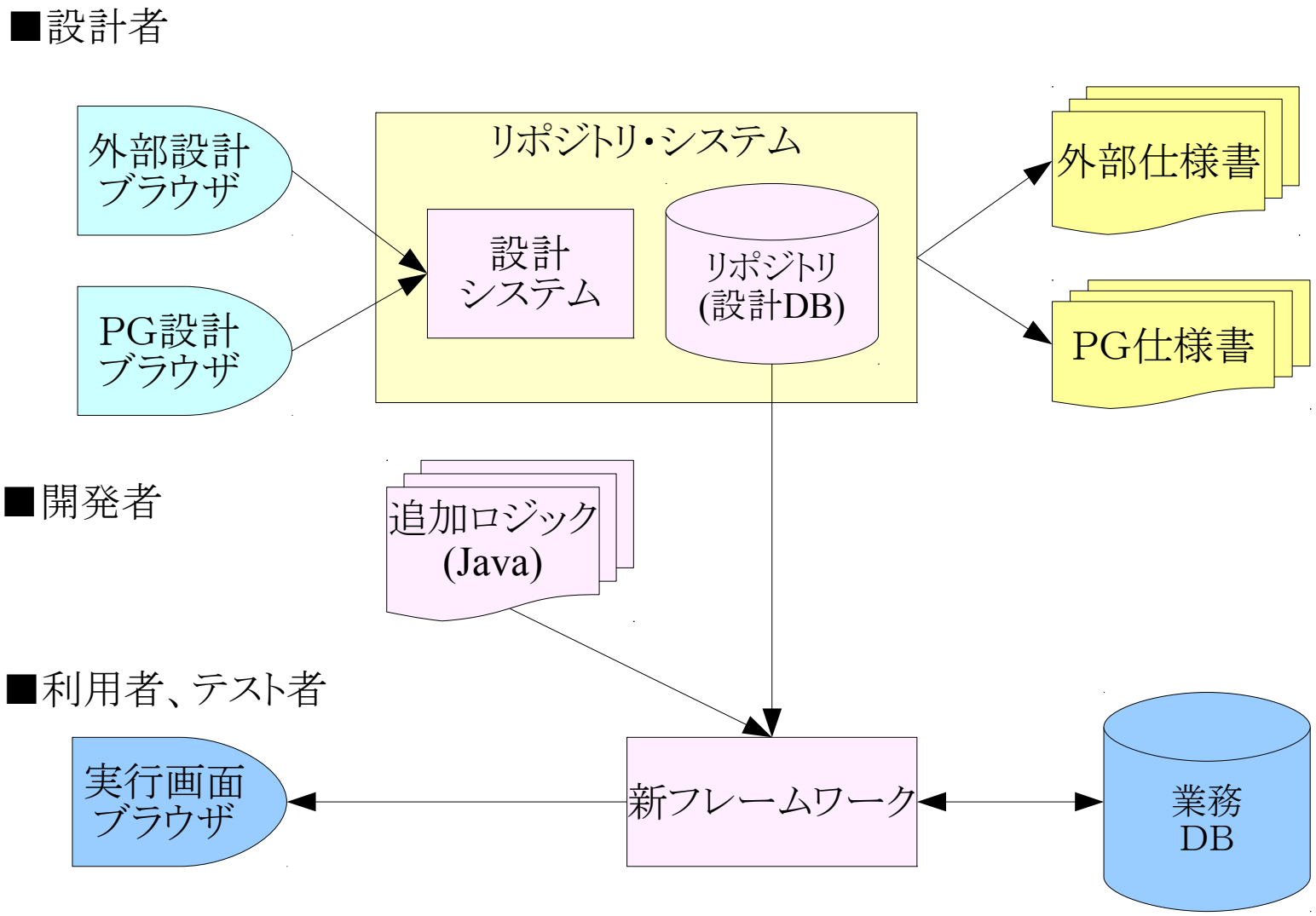


新フレームワークの構築

- リポジトリの情報で動作
  - 画面レイアウト・画面入出力・データベース入出力情報で基本的な動作を行う
    - 画面設計が終わった段階の情報だけでも画面表示や遷移が可能
  - 複雑なロジックがなければコーディングは不要
    - 実現できないロジックはJavaプラグインで追加



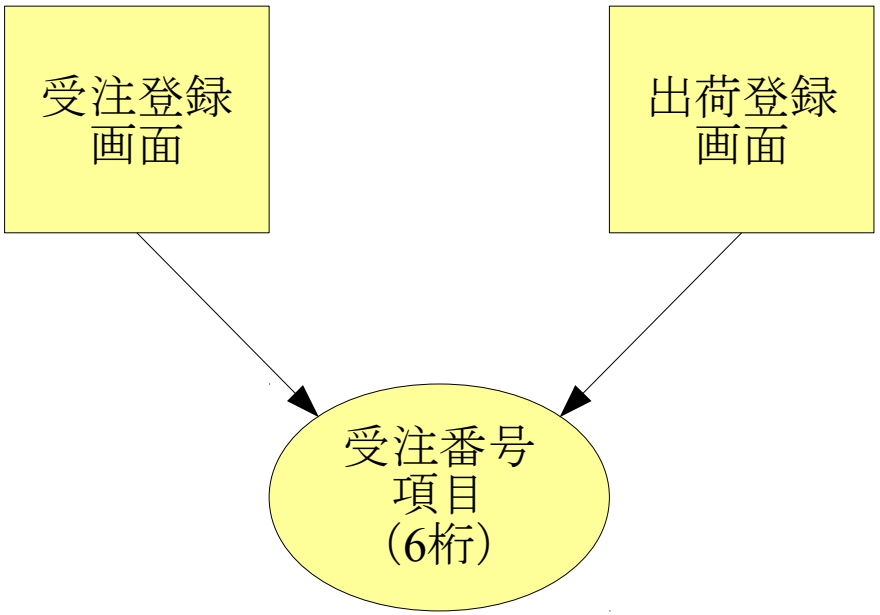
リポジトリ・新フレームワークを用いた開発イメージ



データベース化の課題

- ユーザー合意済みの仕様に対する変更対応
  - リポジトリは異なる画面上に同一の入力項目があれば同一項目部品を紐づける

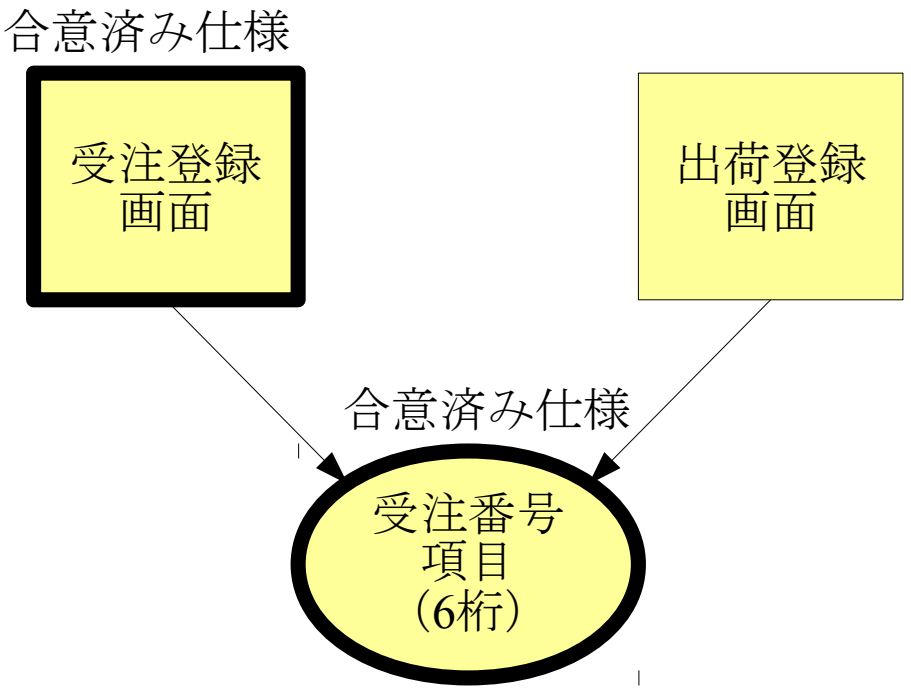
メリット:　すべての画面で項目仕様の整合性が取れる



データベース化の課題

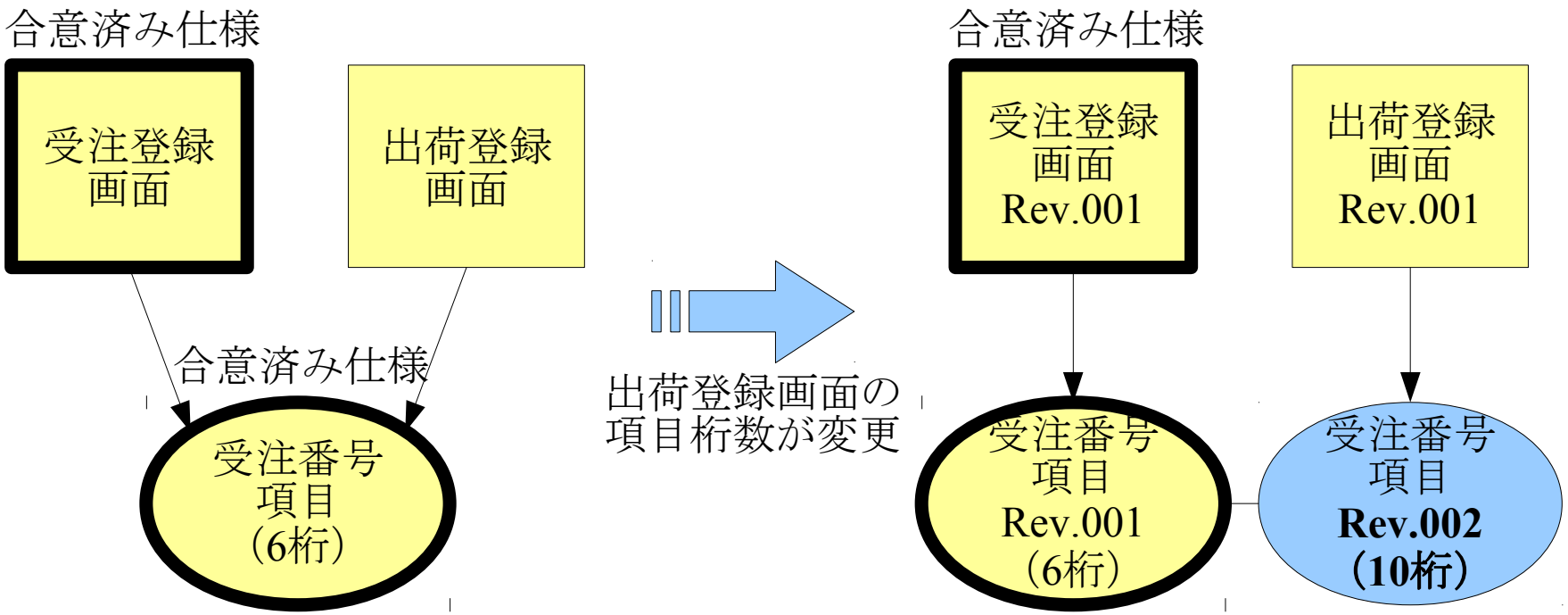
- ユーザー合意済みの仕様に対する変更対応
  - リポジトリは異なる画面上に同一の入力項目があれば同一項目部品を紐づける

  デメリット：ユーザー合意済みの項目を意図せず変更する恐れ



データベース化の課題

- ユーザー合意済みの仕様に対する変更対応
  - リビジョン管理を導入

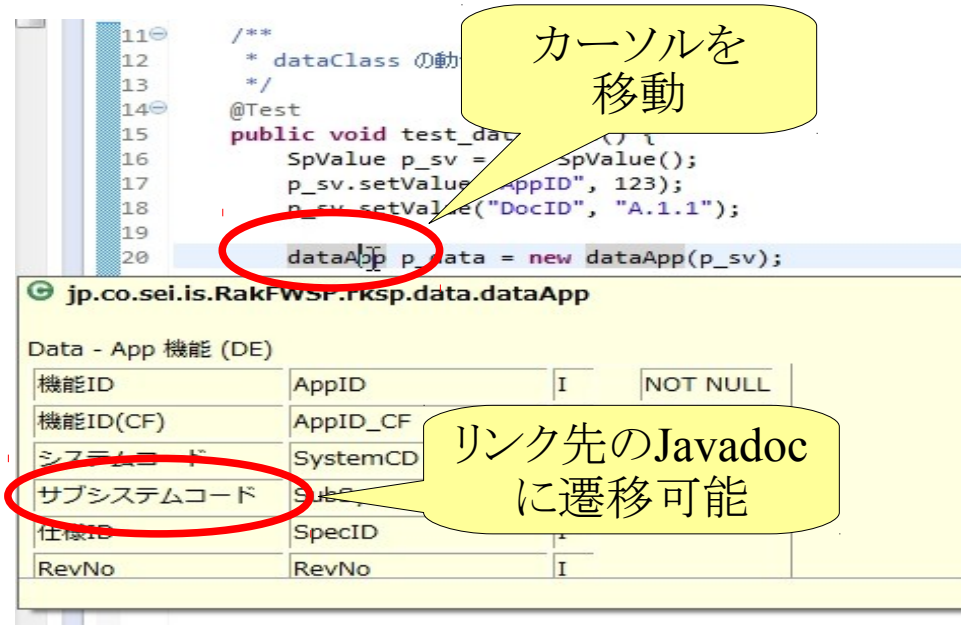


  - 仕様変更時には影響を受ける設計者に通知
  - リビジョン整合チェック機能で不整合解消をサポート

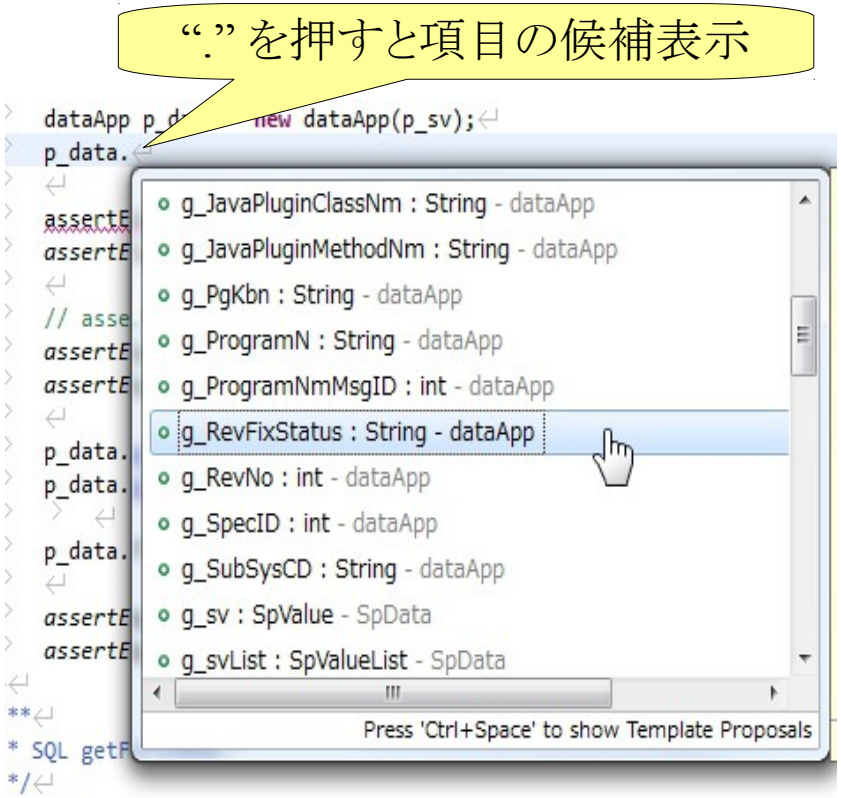
Java開発環境への設計情報提供

Eclipse機能の活用

- オープンソースの統合開発環境Eclipse
  - Javaメソッドやクラスの説明（Javadoc）の表示
  - Javadocのリンク遷移
  - 入力支援機能



Javadoc表示



入力支援

Javaソースの自動生成機能

- リポジトリの設計情報を基にJavaソースを自動生成
  - 追加業務ロジックに必要なJavaメソッド
  - メソッド単位の仕様をJavadocに出力
  - 画面遷移・レイアウトの仕様参照用ソース
  - データベース項目の入力支援用ソース
- 各Javadocにリンクを出力し、関連項目間の相互参照可能
  - 必要な仕様へ簡単にアクセス
- 入力支援ソースによる入力ミス撲滅
  - 一覧から選択入力
  - 万一書き間違えてもEclipseが警告

取り組みの結果と評価

設計資産

- 構築した設計資産
  - 仕様を一括管理するリポジトリ
  - リポジトリで動作する新フレームワーク

- 設計資産により可能になったこと
  - 工程毎の変換作業なしで設計開発
  - Javaソース自動生成
    - 仕様へのアクセス性向上
    - 項目名入力ミスの撲滅

取り組みの評価

- プロジェクトでの試行
  - 開発規模は小さいが、実プロジェクトでリポジトリ・新フレームワークを利用してシステム開発を実施中

評価点	結果	評価
(a) 設計生産性の維持	外部・基本設計　＋1% プログラム設計　－38% コーディング　　途中段階	○
(b) 変換作業なしの設計開発	変換作業はなくなった ただしリポジトリ設計システムや新フレームワークに改善要望あり	△
(c) 業務ロジック開発工数削減	90%のケースで設計情報が活用され、開発がスムーズに進められている。	○

今後の課題

- 保守での活用
  - 自動で影響分析できる機能など
- リポジトリ設計システム操作性の改善
  - ウィザード形式で簡単に登録できるように