

派生開発カンファレンス2013

ソースコード主体の派生開発から モデル主体の派生開発へ

～ 設計の見える化による設計品質の維持と改善 ～

2013. 5. 24

派生開発推進協議会 T20研究会
(株)日立情報制御ソリューションズ

渡辺 滋

派生開発カンファレンス2013

ソースコード主体の派生開発からモデル主体の派生開発へ ～ 設計の見える化による設計品質の維持と改善 ～

Contents

- ★ 1. T20研究会のメンバと活動
2. 課題分析と解決策検討
3. サンプル事例による効果検証
4. 今後の課題と方針

1-1. T20研究会のテーマとメンバ

- テーマ: 「XDDP」とモデル駆動開発の融合

- 目的:

「XDDP」とモデル駆動開発の融合を通じて、派生開発における新たなアプローチを検討し、派生開発の現場で課題としているソフトウェア・アーキテクチャの劣化防止や改善について、解決方法を確立させる

- メンバ:

齋藤 賢一	(株)エクスマーシオン
河野 岳史	スパークスシステムズジャパン(株)
厚田 鳴海	パナソニックファクトリーソリューションズ(株)
臼井 公二彦	トヨタテクニカルディベロップメント(株)
渡辺 滋	(株)日立情報制御ソリューションズ

XDDP: eXtreme Derivative Development Process

1-2. T20研究会発足の経緯と想い

【研究会発足の経緯】

- モデリングを主体とした開発手法の適用支援を行っている中で、**開発の80%を占めている派生開発プロジェクトという大きな壁**があることに気付いた
- これを対処しない限り、**モデリングを主体とした開発手法も何故か無力**に感じてしまい、悩み始めた
- もしかすると、**世の中に同じような問題意識を持っている技術者がいるのでは**と思い始め、そのような方々と議論しながら一緒に考えた方が、より早く、より良いものができるだろうと考えた

【メンバの想い】

- XDDPは派生開発の抱える問題に的確に対処できる優れたプロセスであるが、**ソフトウェアの変更に関する設計の具体的対処法**を示してはいない
- 派生開発におけるソフトウェアエンジニアリングの部分、とくに**ソフトウェア設計をモデリング手法でカバーし、XDDPの変更設計書につなげたい**
- XDDPの適用対象外である**アーキテクチャの劣化防止や改善に関する課題解消のための実現手段**を探り、XDDPを進化させたい。

1-3. T20研究会の活動

◆ 活動方針

基本的には、メーリングリストベースでの活動
 活動計画に沿って、1回/月の頻度で秋葉原にて研究集会を実施

◆ 活動計画

- (1) XDDPへのモデリング*手法導入: 現在、検討中(今回の発表内容)
- (2) モデル駆動開発へのXDDP導入: (1)の検討完了後、検討予定
- (3) 影響範囲のモデル化: (1)の中で検討中
- (4) モデリングツールとの連携: その都度、必要に応じて検討

◆ 成果物

開発現場で活用できる具体的内容で構成したガイドライン:
 ・「モデル主体派生開発ガイドライン」(仮称)

【*注記】モデリングの表記には、UML(Unified Modeling Language)やMATLAB/Simulinkを使用する。

派生開発カンファレンス2013

ソースコード主体の派生開発からモデル主体の派生開発へ ～ 設計の見える化による設計品質の維持と改善 ～

Contents

1. T20研究会のメンバと活動
- ★ 2. 課題分析と解決策検討
3. サンプル事例による効果検証
4. 今後の課題と方針

2-1. 派生開発に関する問題

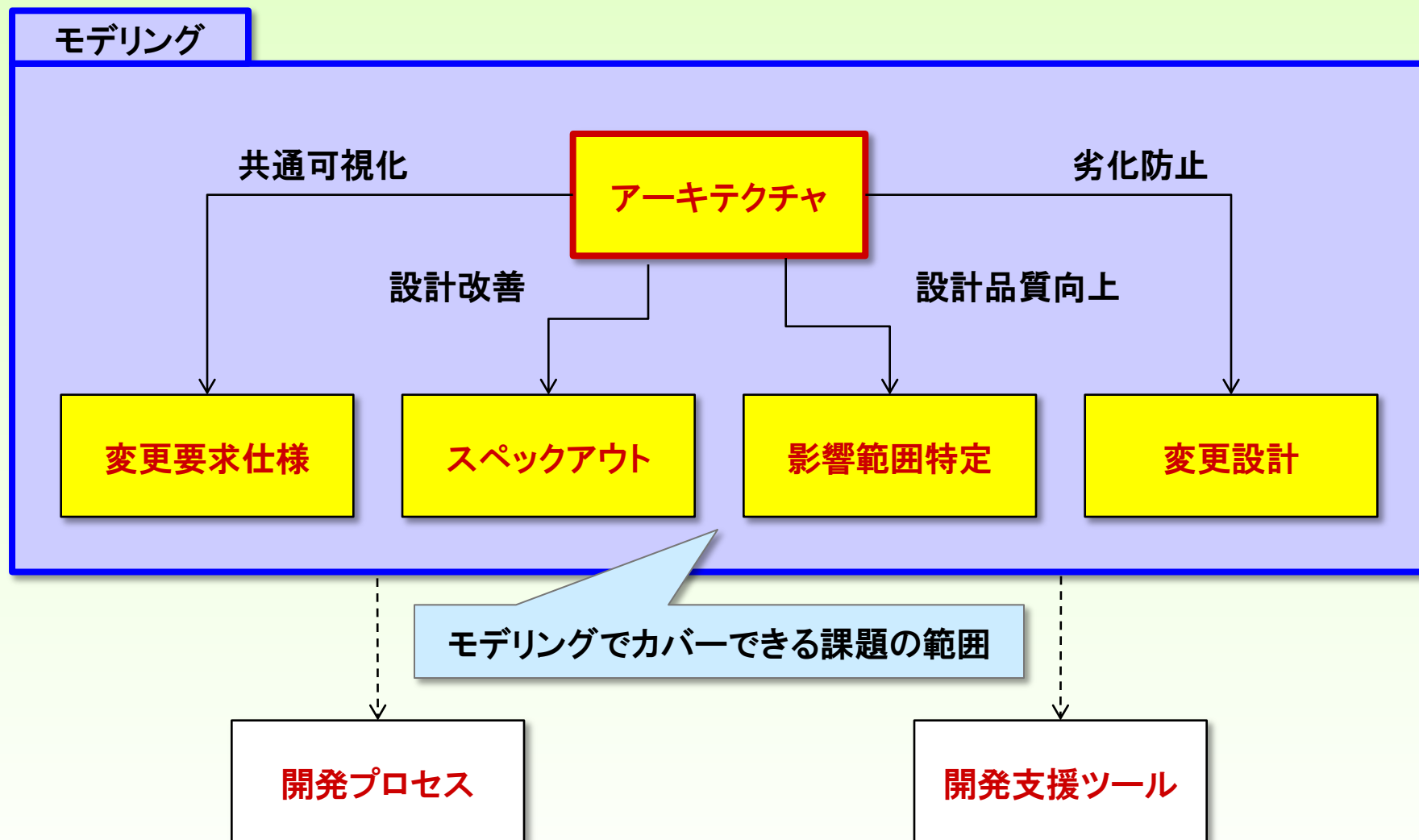
派生開発の現場では、以下のような問題が発生している：

- 派生開発を続けていると、プログラムの細かい部分に着目してしまい、**アーキテクチャ設計の視点が失われ、設計品質が低下する**
- **文章主体の記述(変更要求仕様書、変更設計書)**では、表形式により上から下へと直線的に表現されているため、**条件を把握しにくく、処理フローや条件分岐が分かりにくい**
- **スペックアウト**では、担当者がどのような図を作成してよいか分からず**個人に依存している**
- **モデル駆動開発**で派生開発を行おうとしても、**新規開発プロセスしかないため、その方法は担当者任せになっている**
- **派生開発、特に変更の分析、設計を支援するツールの必要性**を感じているが、現状は極めて少ない

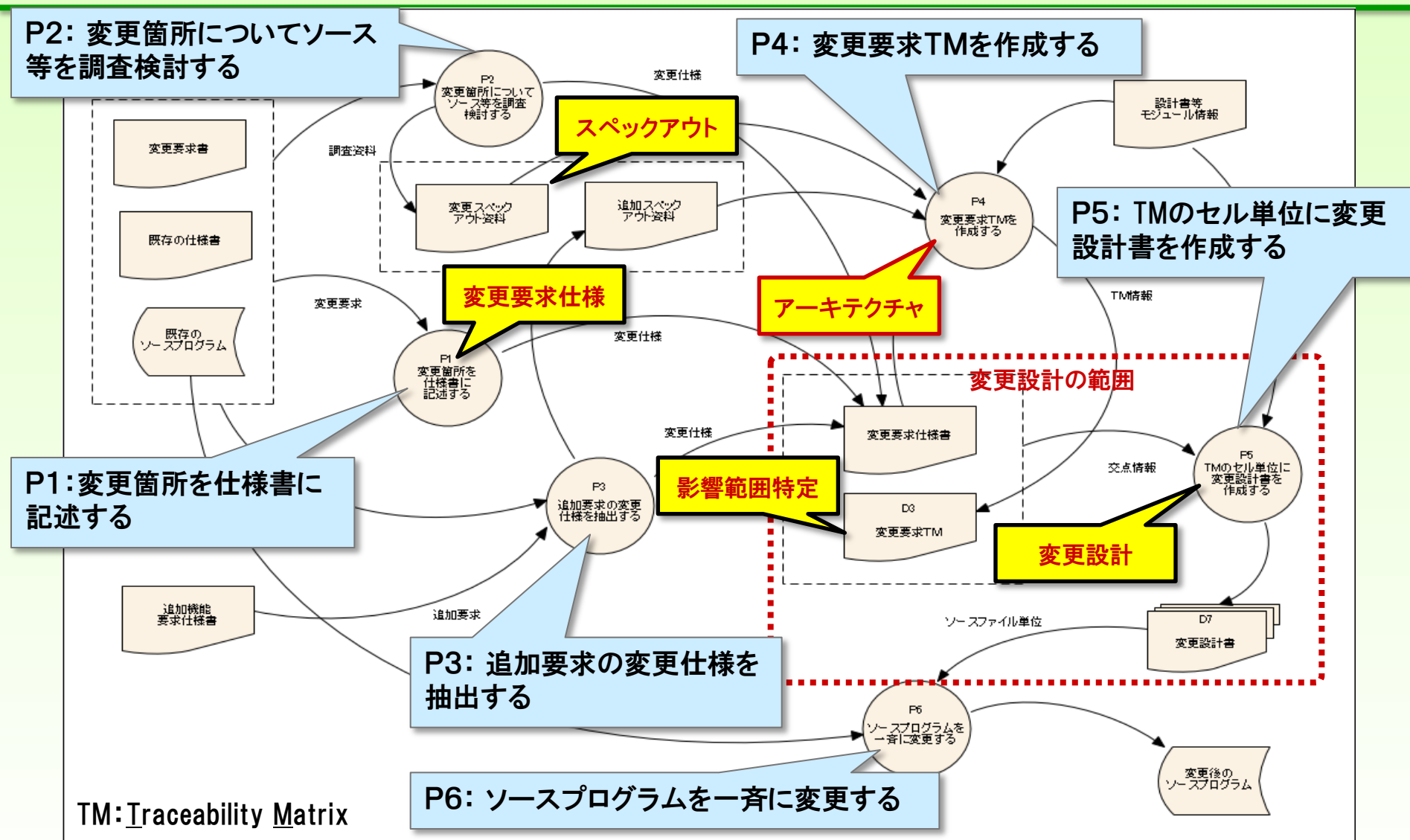
2-2. T20研究会における課題一覧

No.	課題項目	課題内容
1	開発プロセス	派生開発に特化したモデル駆動開発プロセスが確立されていない モデル主体による派生開発を行う場合、プロセスをどのように考えたらよいか
2	変更要求仕様	変更要求仕様の文章表現では曖昧さや個人差があり分かりにくい 変更要求仕様を、変更要求書の文章だけではなく、視覚化したモデルで補足できないか
3	スペックアウト	スペックアウトの成果物作成について有効なガイドがない 対象となるソースコードから、どのようにスペックアウトして成果物を作成するか
4	変更設計	変更設計仕様の記述がプログラム内の視点になりやすく変更漏れが発生しやすい 変更設計には、インタフェース、構造、振る舞いなど、複数の視点が必要になる
5	影響範囲特定	変更の影響範囲の特定や局所化について具体的な方法がない 影響範囲の特定や局所化は、解析ツールだけでは無理で具体的な方法が必要である
6	アーキテクチャ	機能の変更・追加の中ではソフトウェアアーキテクチャの劣化防止や改善が難しい アーキテクチャ全体を見れないので、部分的な範囲の開発では、かなり困難になる
7	開発支援ツール	変更の分析、設計やトレーサビリティ管理を支援する強力なモデリングツール*がない 派生開発に有効な支援機能を持つモデリングツールが提供されていない

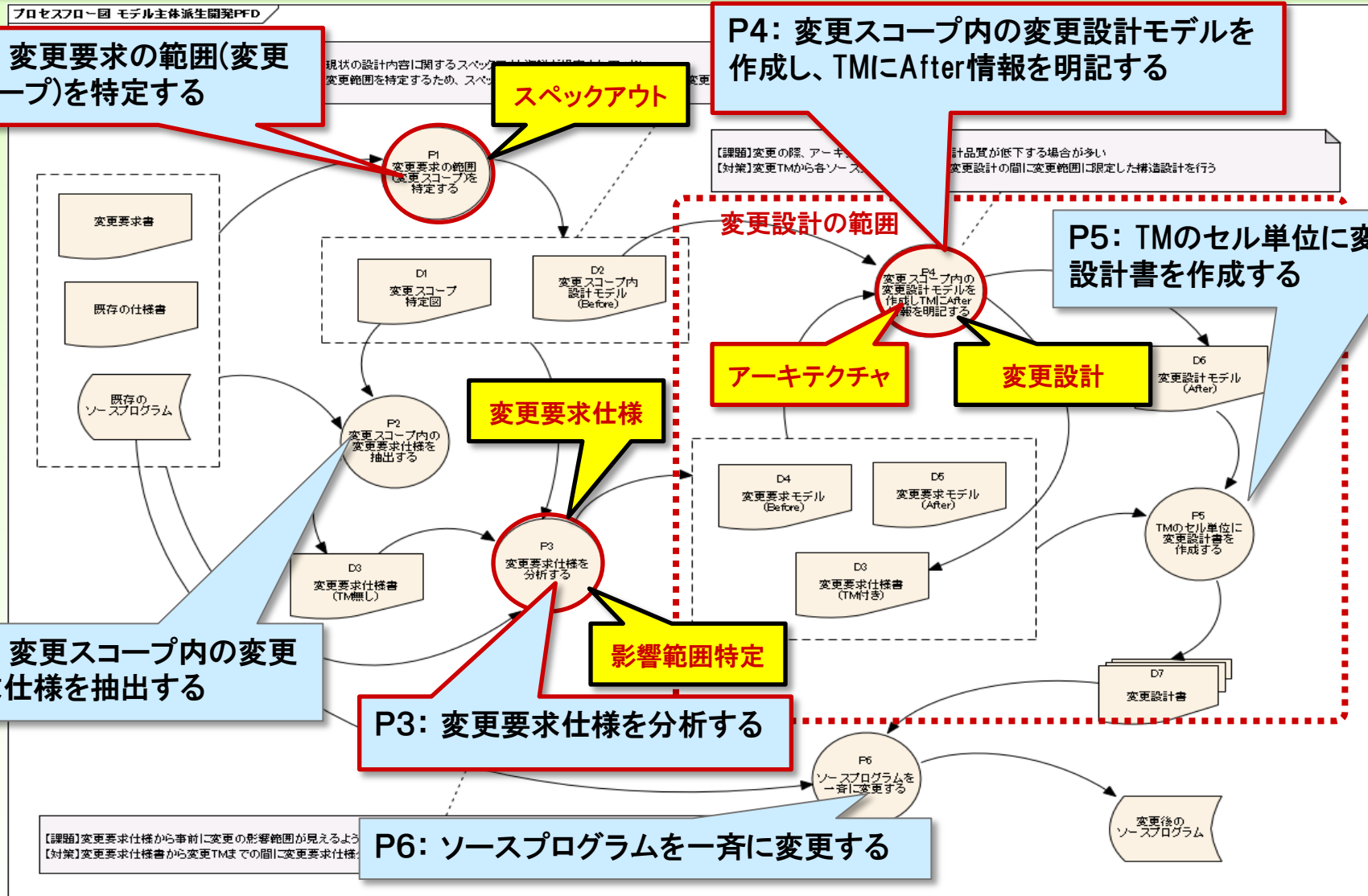
2-3. 課題分析(課題項目間の関係)



2-4. 課題分析(XDDP変更プロセスでの課題確認)



2-5. 解決策検討(モデル主体派生開発プロセス)



2-6. 解決策検討(モデル主体派生開発プロセス: P1)

P1: 変更要求の範囲(変更スコープ)を特定する

P1
変更要求の範囲
(変更スコープ)を
特定する

変更要求書

既存の仕様書

既存の
ソースプログラム

D1
変更スコープ
特定図

D2
変更スコープ内
設計モデル
(Before)

P2
変更スコープ内の
変更要求仕様を
抽出する

解決できる課題項目

スペックアウト

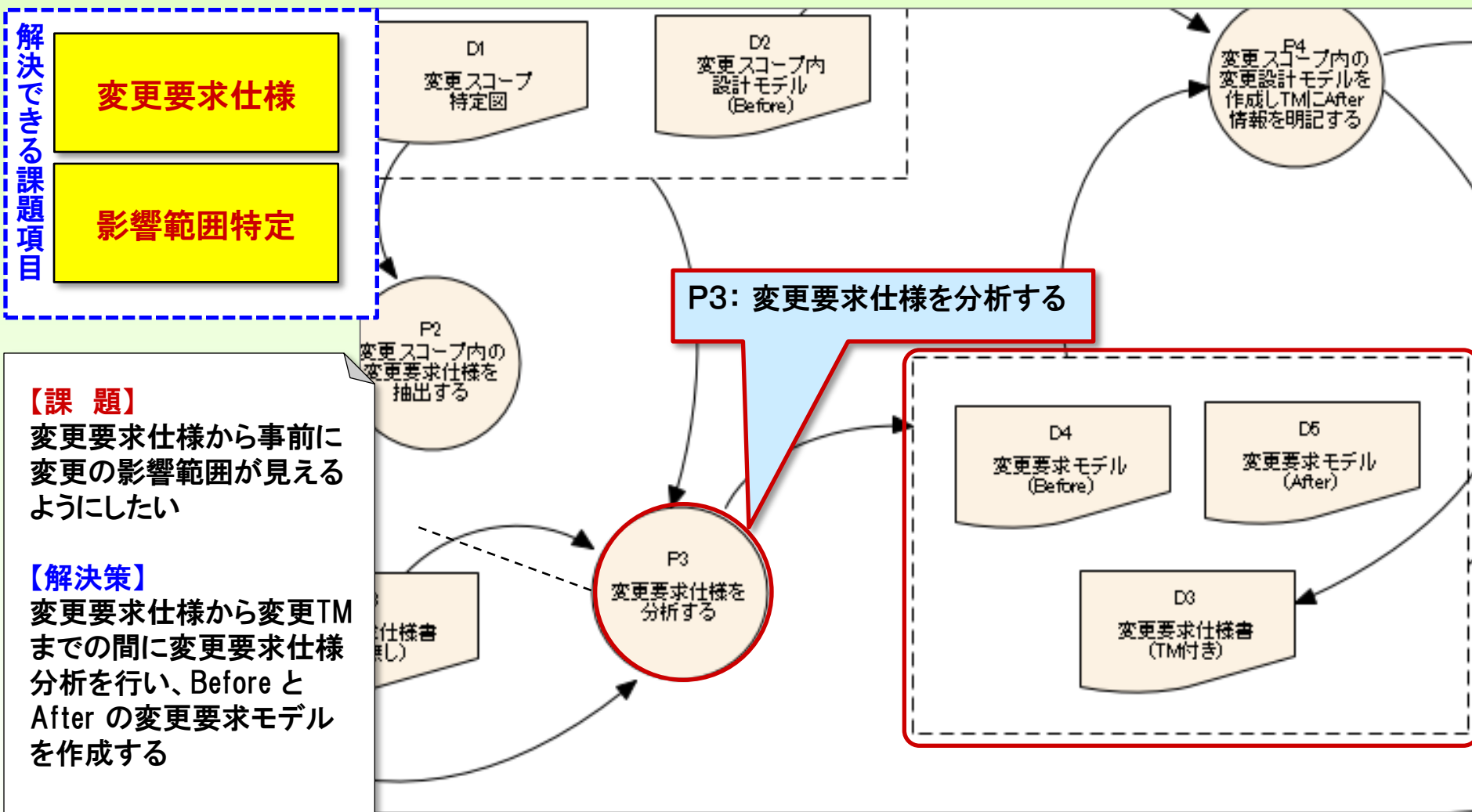
【課題】

現状の設計内容に関するスペックアウト資料が規定されていない

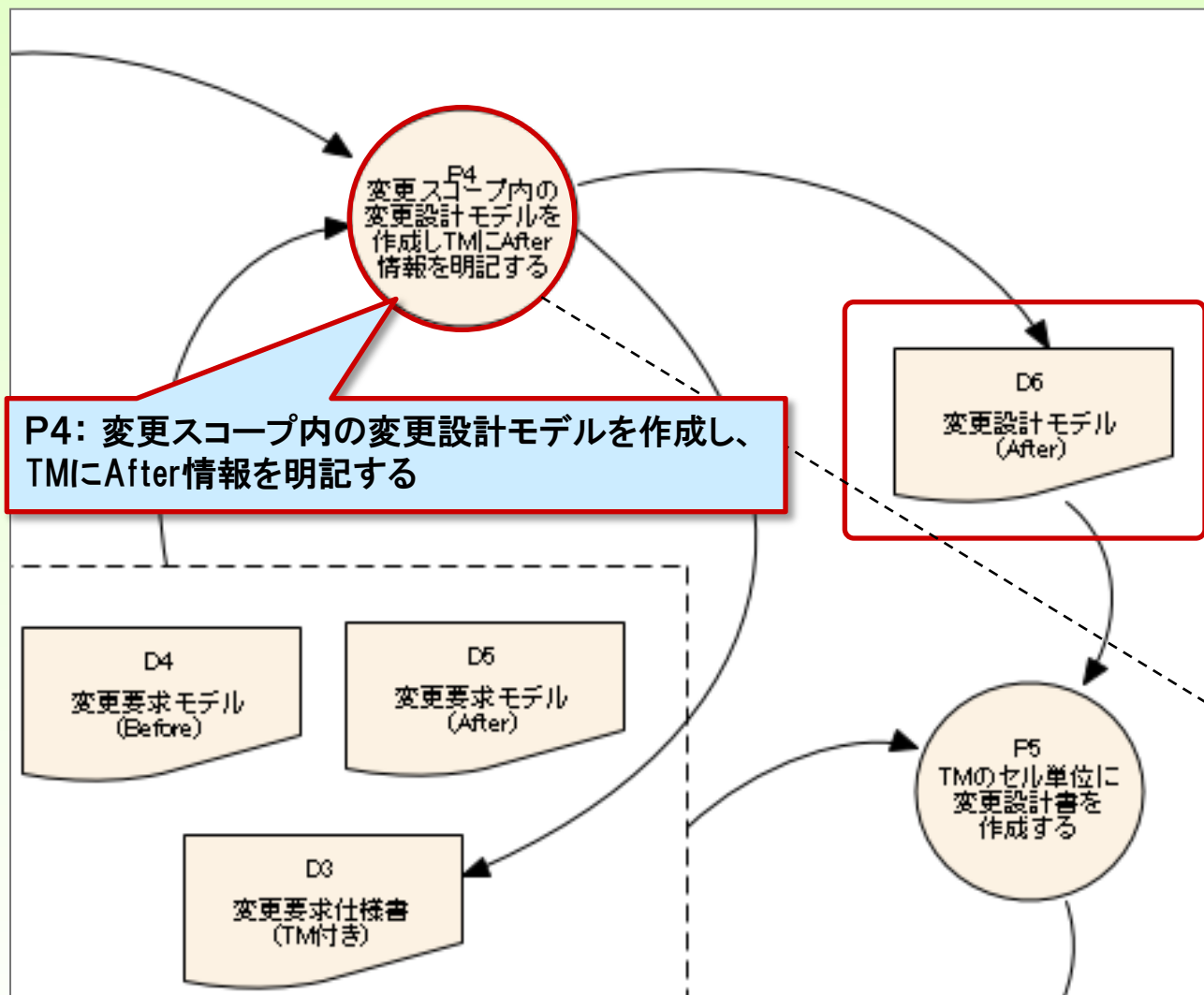
【解決策】

変更範囲を特定するため、スペックアウト資料としてモデル図を作成し、変更対象箇所と変更に伴う影響箇所を明確にする

2-7. 解決策検討(モデル主体派生開発プロセス: P3)



2-8. 解決策検討(モデル主体派生開発プロセス: P4)



解決できる課題項目

アーキテクチャ

変更設計

【課題】

変更の際、アーキテクチャが劣化し、設計品質が低下する機会が多い

【解決策】

変更TMから各ソースプログラム単位の変更設計の間に変更範囲に限定したアーキテクチャ劣化防止のための構造設計を行う

2-9. 解決策検討(成果物一覧)

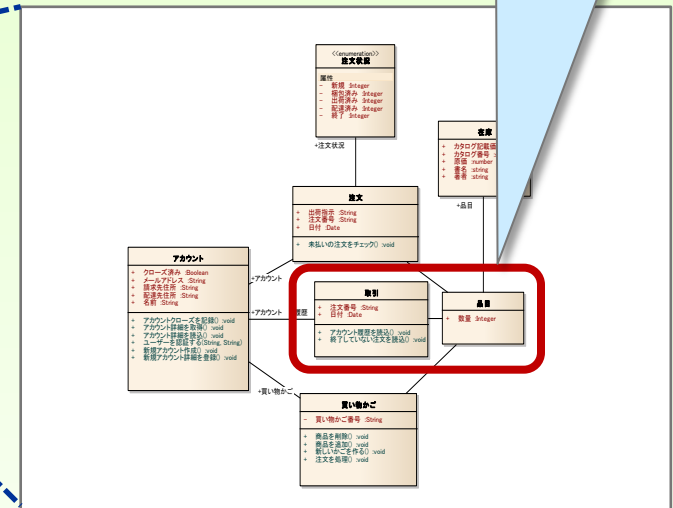
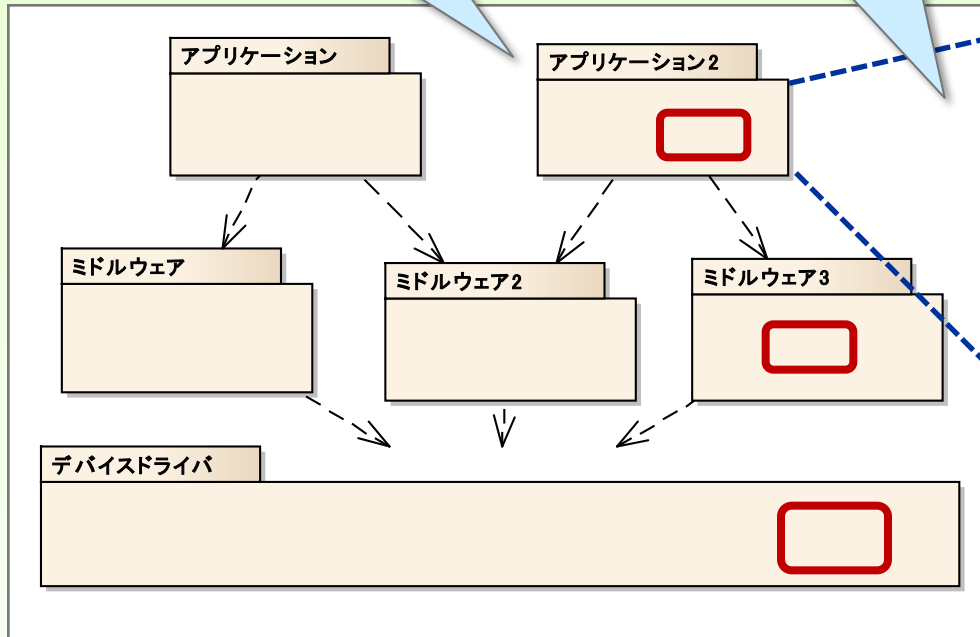
プロセス	成果物名	具体的成果物
P1: 変更要求の範囲(変更スコープ)を特定する	D1: 変更スコープ特定図	(1)アーキテクチャレベル(パッケージ図) パッケージの単位が異なるので注意 (2)コンポーネントレベル コンポーネントの単位が異なるので注意
	D2: 変更スコープ内設計モデル(Before)	ソースコードからリバース可能なUML図 構造図は可能だが振る舞い図は難しい
P3: 変更要求仕様を分析する	D3: 変更要求仕様書(TM付き) (1)変更TMIに下記列を追加: ・変更スコープ特定 ・変更要求モデル(Before/Afterの2行に分割) ・変更設計モデル(") (2)上記以外の変更対象もBefore/Afterの2行に分割	モデル主体派生開発向けの変更要求仕様書(TM付き)
	D4: 変更要求モデル(Before)	UMLアクティビティ図(データと処理の流れを表記可能)
	D5: 変更要求モデル(After)	UMLアクティビティ図(データと処理の流れを表記可能)
P4: 変更スコープ内の変更設計モデルを作成しTMにAfter情報を明記する	D6: 変更設計モデル(After)	必要に応じたUMLやMATLAB/Simulinkの構造図、振る舞い図

2-10. 成果物(変更スコープ特定図)

変更範囲(スコープ)を特定
するため、変更箇所の当
たりを付ける

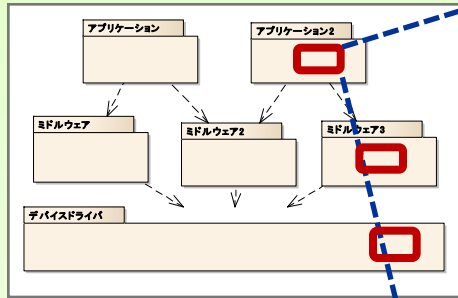
ツール等により、ソースコード
からモデル図にリバースして
構造を可視化する

変更箇所を括り出し、変更箇所
と、それ以外のインターフェース
を明らかにする



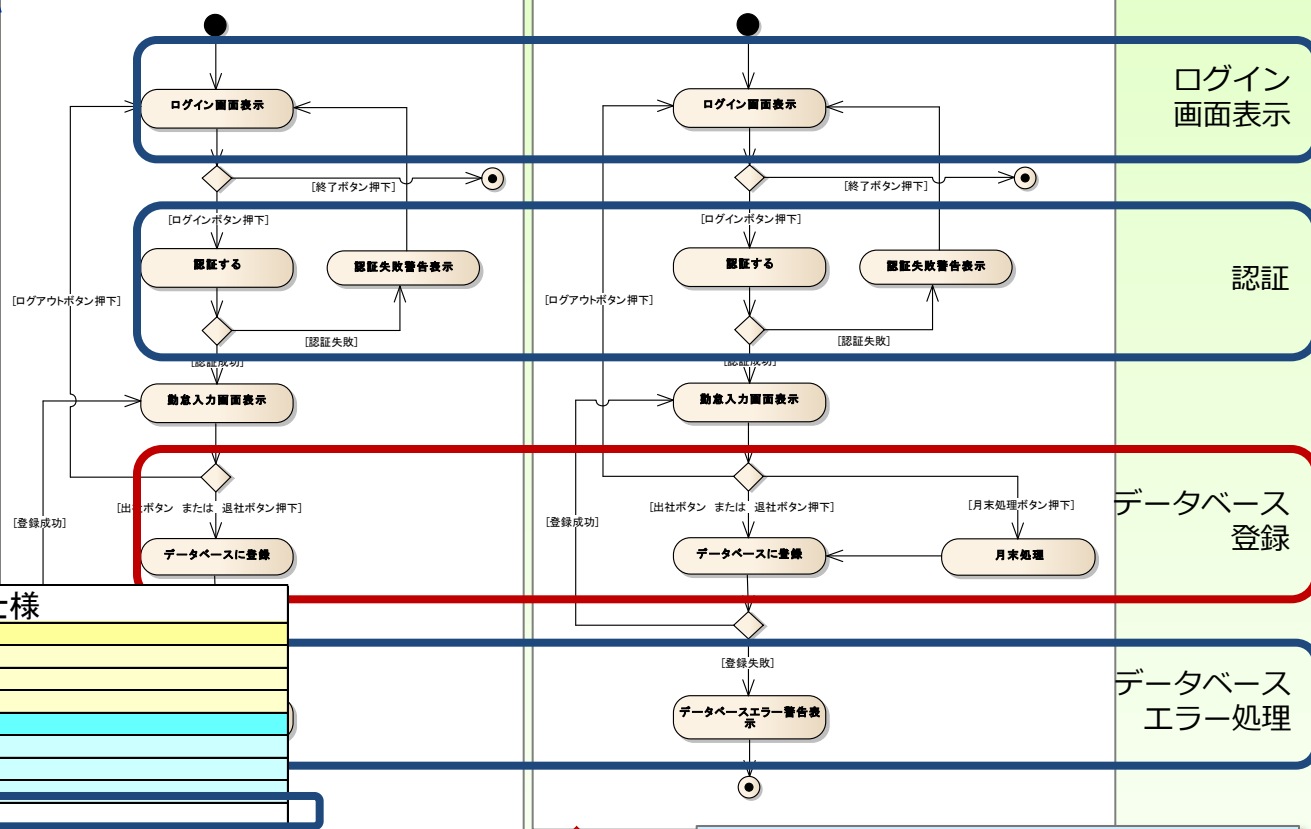
2-11. 成果物(変更要求モデル: Before と After)

D1. 変更スコープ特定図



D4. 変更要求モデル(Before)

D5. 変更要求モデル(After)



変更要求仕様書

要求仕様

要求仕様			
<機能要求>			
要求	ACC-01		
理由			
説明			
<>			
要求	ACC-01.01		
理由			
説明			
<ログイン表示画面>			
□□□	ACC-01.01.01	変更なし	
□□□	ACC-01.01.02		
<認証>			
□□□	ACC-01.01.01	変更なし	
□□□	ACC-01.01.02		
<データベース登録の変更>			
□□□	ACC-01.01.01	これまで、△△△だった部分を、○○○に変更する。	
□□□	ACC-01.01.02		
<データベースエラー処理>			
□□□	ACC-01.01.01	変更なし	
□□□	ACC-01.01.02		

アクティビティ図とUSDMの関係は、アクティビティとUSDMの<グループ>で対応付ける方法がある

USDM: Universal Specification Describing Manner

2-12. 成果物(変更要求仕様書: TM要素)

変更要求仕様書	パッケージ1	パッケージ2	パッケージ3	...	変更要求モデル	パッケージ1	パッケージ2	パッケージ3	...	変更設計モデル	パッケージ1	パッケージ2	パッケージ3	...	パッケージ1	パッケージ2	パッケージ3	...						
<p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p> <p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p> <p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p> <p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p> <p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p> <p><変更名物></p> <p>ID</p> <p>理由</p> <p>説明</p>	コンポーネント1	コンポーネント2	コンポーネント3	コンポーネント4	コンポーネント5	...	コンポーネント1	コンポーネント2	コンポーネント3	コンポーネント4	コンポーネント5	...	コンポーネント1	コンポーネント2	コンポーネント3	コンポーネント4	コンポーネント5	...	コンポーネント1	コンポーネント2	コンポーネント3	コンポーネント4	コンポーネント5	...

モデル主体派生開発の要素を追加して表記する:

- 変更スコープ特定図(アーキテクチャ・コンポーネントのレベル)
- 変更要求モデル(Before/After)
- 変更設計モデル(Before/After)

2-13. 成果物(変更要求仕様書: 変更仕様)

変更要求仕様書																																																																			
変更要求		変更仕様																																																																	
変更要求ID	理由	説明	変更仕様																																																																
<table border="1"> <thead> <tr> <th colspan="4"><変更名称></th> </tr> <tr> <th>変更要求</th> <th>ID</th> <th>Before</th> <th>After</th> </tr> </thead> <tbody> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td colspan="4"><変更名称></td> </tr> <tr> <th>変更要求</th> <th>ID</th> <th>Before</th> <th>After</th> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td colspan="4"><変更名称></td> </tr> <tr> <th>変更要求</th> <th>ID</th> <th>Before</th> <th>After</th> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td colspan="4"><変更名称></td> </tr> <tr> <th>変更要求</th> <th>ID</th> <th>Before</th> <th>After</th> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> <tr> <td>□□□</td> <td>ID</td> <td>Before</td> <td>After</td> </tr> </tbody> </table>				<変更名称>				変更要求	ID	Before	After	□□□	ID	Before	After	□□□	ID	Before	After	<変更名称>				変更要求	ID	Before	After	□□□	ID	Before	After	□□□	ID	Before	After	<変更名称>				変更要求	ID	Before	After	□□□	ID	Before	After	□□□	ID	Before	After	<変更名称>				変更要求	ID	Before	After	□□□	ID	Before	After	□□□	ID	Before	After
<変更名称>																																																																			
変更要求	ID	Before	After																																																																
□□□	ID	Before	After																																																																
□□□	ID	Before	After																																																																
<変更名称>																																																																			
変更要求	ID	Before	After																																																																
□□□	ID	Before	After																																																																
□□□	ID	Before	After																																																																
<変更名称>																																																																			
変更要求	ID	Before	After																																																																
□□□	ID	Before	After																																																																
□□□	ID	Before	After																																																																
<変更名称>																																																																			
変更要求	ID	Before	After																																																																
□□□	ID	Before	After																																																																
□□□	ID	Before	After																																																																

変更要求仕様をBefore、Afterの2行に分割し、TMの横軸に表記されるモデル要素からプログラム要素までの対応付けを行う

派生開発カンファレンス2013

ソースコード主体の派生開発からモデル主体の派生開発へ

～ 設計の見える化による設計品質の維持と改善 ～

Contents

1. T20研究会のメンバと活動
2. 課題分析と解決策検討
- ★ 3. サンプル事例による効果検証
4. 今後の課題と方針

3-1. サンプル事例の概要

■ 液晶パネル加工制御システムの組み込みソフト

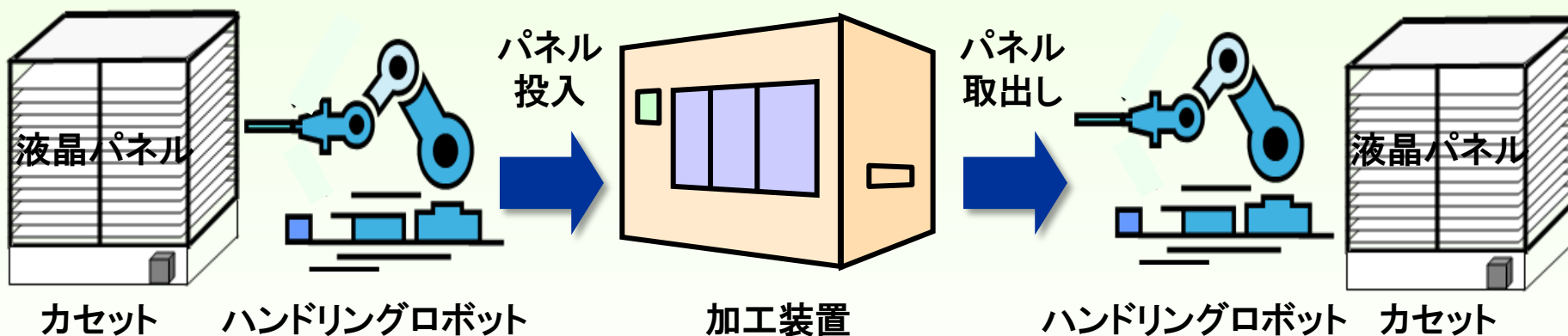
- ・ 液晶パネル製造工程において、カセットに格納されたパネルを加工装置に投入、取出しを行う。
- ・ カセットに格納されているパネルは、カセット毎に異なり、常に満杯とは限らない。

変更要求仕様【Before】

- ・ パネルを加工装置に投入時、ハンドリングロボットは、カセットの最下部からパネルを取出す。
- ・ 取出し時にパネルの有無をセンサで判断して、最上段までパネルを取得する動作を行う。
→ カセット内のパネルをすべて加工装置に投入後も、無駄な搬送動作が発生している。

変更要求仕様【After】

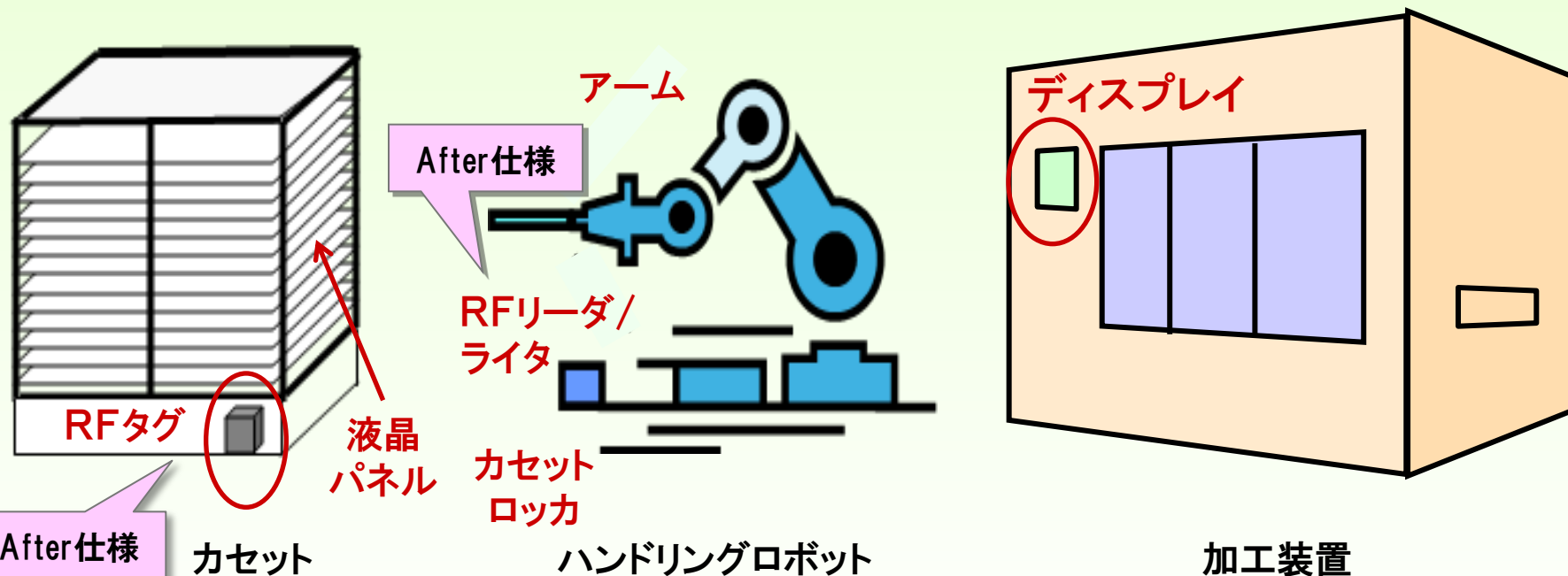
- ・ カセット毎にRF(Radio Frequency)タグを取り付け、カセット固定位置に設置されたRFリーダー/ライタより読出しと書込みを行う。
- ・ RFタグにはカセット内のパネル数が記憶されており、パネル数を取得できるようにする。



3-2. サンプル事例のシステム構成

■ ハードウェア構成

- ・ カセット：RFタグ付(追加)、加工前と加工後の液晶パネル収納用の2種類がある
複数のカセットがベルトコンベアで移動してくる
- ・ ハンドリングロボット：RFリーダ/ライタ(追加)、アーム、カセットロックを装備する
- ・ 加工装置：生産データ表示用ディスプレイ付、液晶パネルを加工する装置

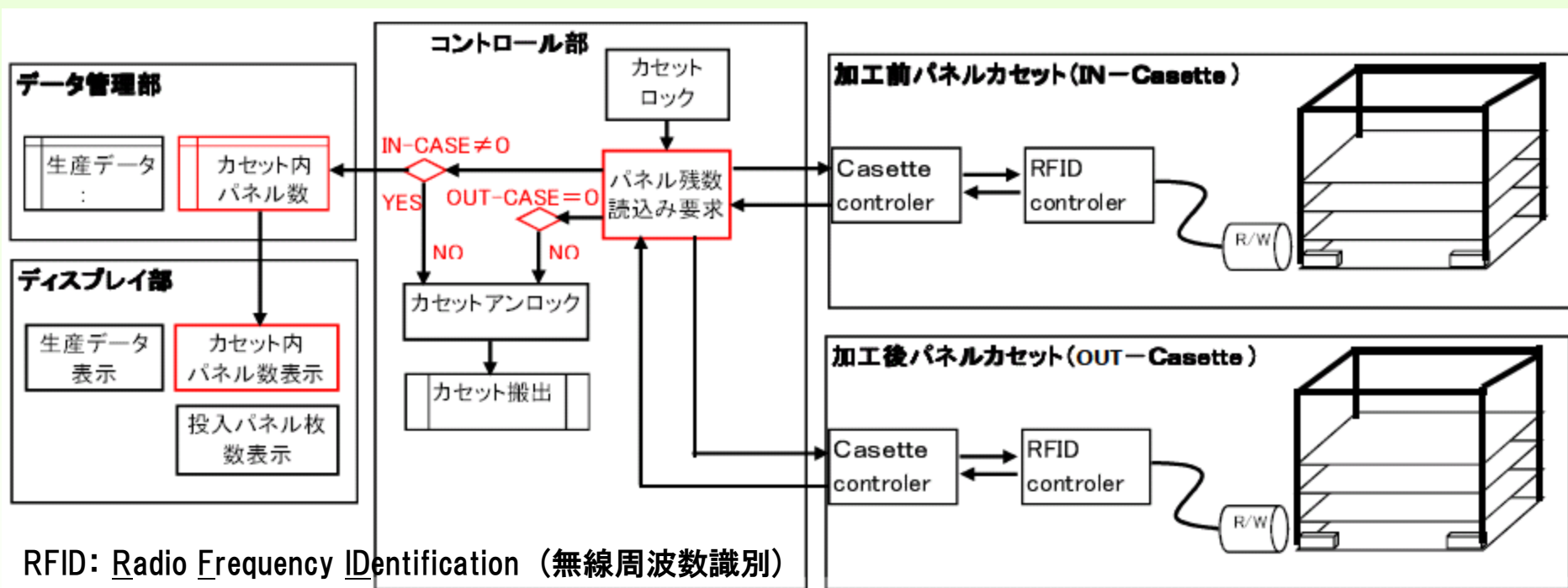


3-3. サンプル事例の変更要求：RFタグ読み込み

■ RFIDを用いたカセット内パネル数管理機能の追加

(1) RFタグからパネル数読み込み

- ・ カセット搬入時にパネル数を読み込む
- ・ 加工前パネルカセットでは、読込んだパネル数から投入パネルの最終位置を割出す
- ・ 加工後パネルカセットでは、格納パネル数が“0”であることを確認する

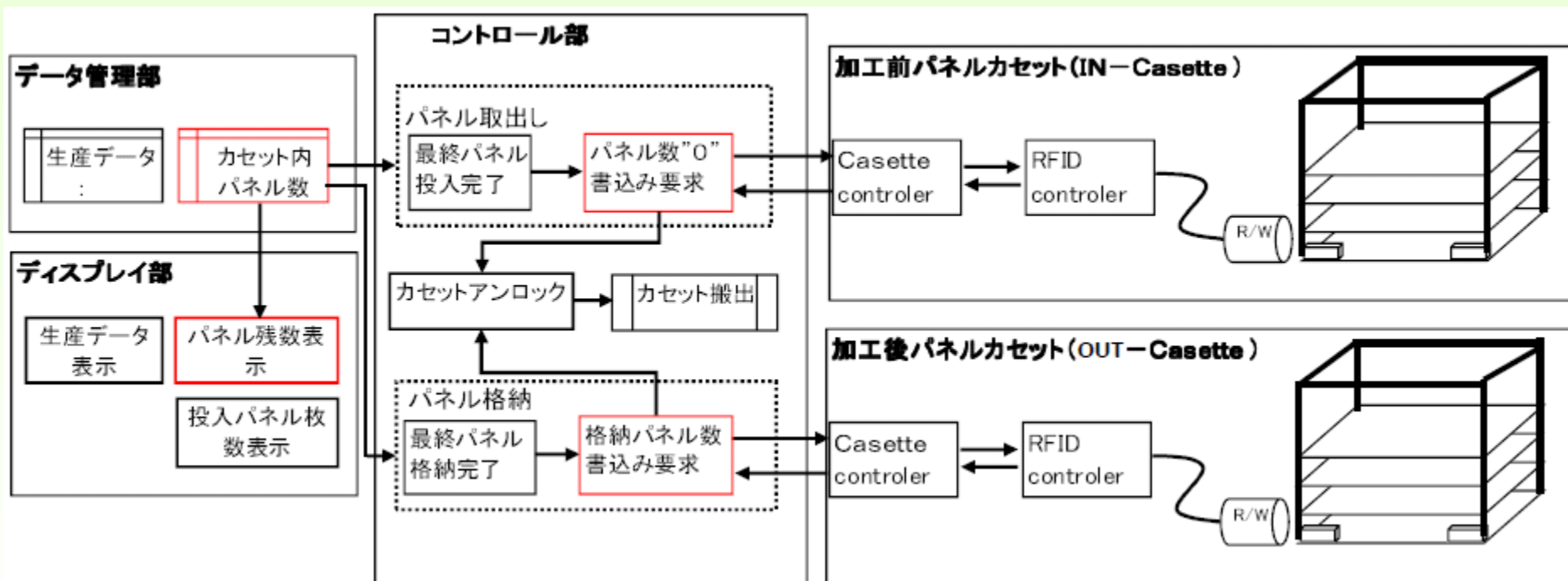


3-4. サンプル事例の変更要求：RFタグ書込み

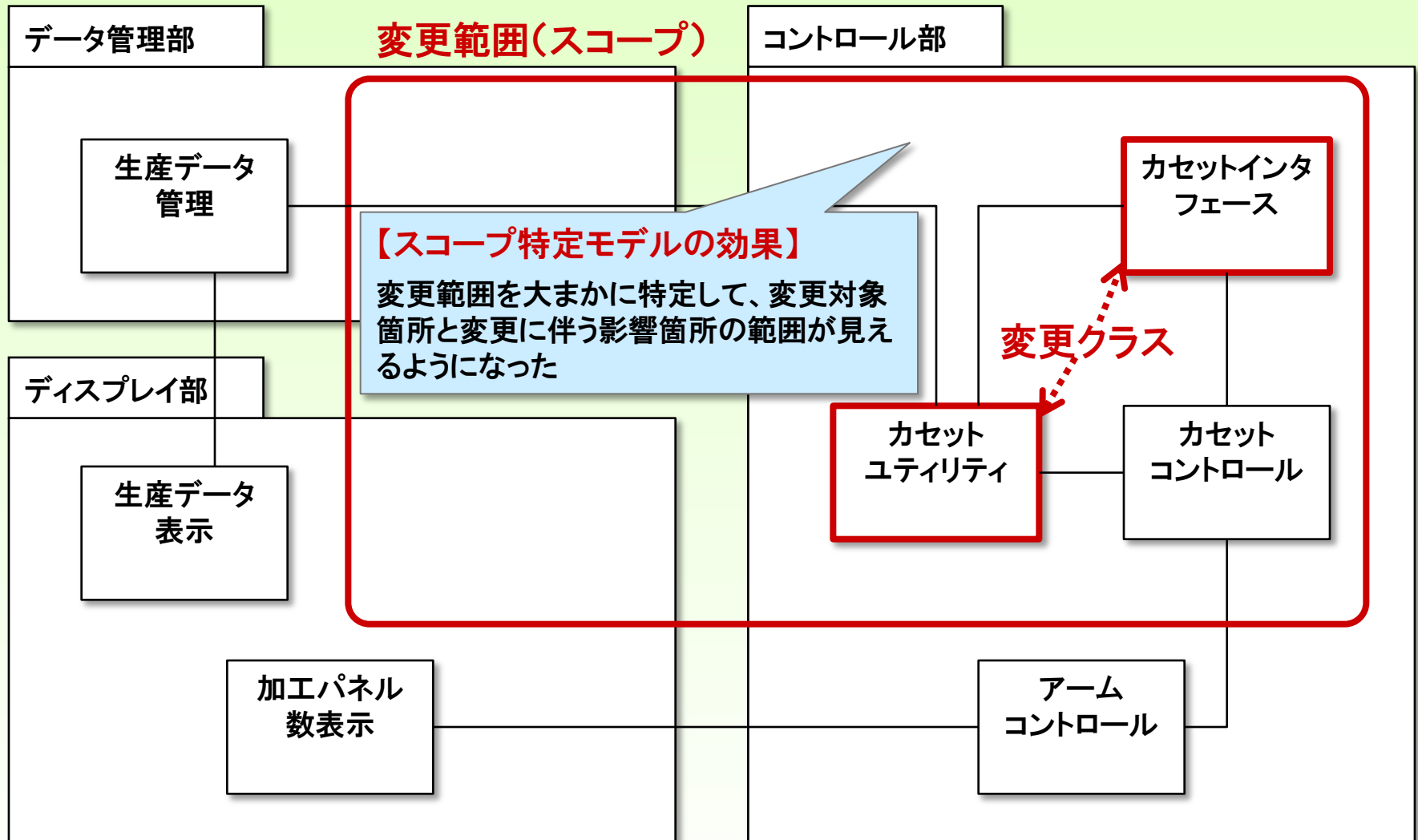
■ RFIDを用いたカセット内パネル数管理機能の追加

(2) RFタグへのパネル数書込み

- ・ カセット搬出時にパネル数書込みを行う
- ・ 加工前パネルカセットには、最終パネル投入完了時にパネル数“0”を書込む
- ・ 加工後パネルカセットには、格納パネル数を書込む



3-5. サンプル事例(変更スコープ特定図の効果)



3-6. サンプル事例(変更要求仕様書)

タイトル	RFIDを用いたカセット内パネル数管理	
要求	RFIDでカセット内のパネルの数を管理したい	
理由	カセット内のパネル搬送の時間を短縮するため。現状は、パネルの有無に拘らず、最終パネル位置まで搬送動作を行っているため、パネルが無い位置での搬送動作が無駄になる。	
説明	カセット搬出時にRFIDへの書き込みを行う。 ・加工前パネルカセットには、搬出時に"0"を書き込む ・加工後パネルカセットには、格納枚数を書き込む カセット搬入時にRFIDの読み込みを行う。 ※読込んだパネル数により、最終パネル位置を割出す	
変更要求	RFID_1	RFIDからパネル数を読み込み、読込んだ値が異常値であれば、カセットを搬出したい。
	理由	加工前パネルカセットのRFIDにパネル数"0"が書き込まれている場合、搬入パネルが無いため、カセットを搬出する必要がある。また、加工後パネルカセットにパネルがある場合、加工装置から取出したパネルがカセット内のパネルと衝突する可能性があるため。
	説明	
変更仕様	加工前パネルカセット搬入処理の変更	
□□□	RFID_1.1	【Before】 カセット搬入時、カセットをロックする。 【After】 カセットをロックし、RFIDからパネル数を読み込む。
□□□	RFID_1.2	【Before】 該当処理なし。 【After】 パネルの部品番号が異なる場合、異なる部品番号を通知する。
□□□	RFID_1.3	【Before】 該当処理なし。 【After】 パネル数が"0"であれば、カセットのロックを解除し、搬出する。
変更仕様	加工後パネルカセット搬入処理の変更	
□□□	RFID_1.4	【Before】 カセット搬入時、カセットをロックする。 【After】 カセットをロックし、RFIDからパネル数を読み込む。
□□□	RFID_1.5	【Before】 該当処理なし。 【After】 パネル数が0以外であれば、カセットのロックを解除し、搬出する。
変更要求	RFID_2	パネル投入時、RFIDから読込んだパネル数分の投入完了後、ハンドリング動作を止めたい。
	理由	存在しないパネルに対してハンドリング動作をすることになり、その時間が無駄になるため。
	説明	
変更仕様	加工前パネルカセットハンドリング動作の変更	
□□□	RFID_2.1	【Before】 カセット格納パネル数と無関係に最終パネル位置までハンドリング動作を行う。 【After】 RFIDから読込んだパネル数分のハンドリング動作を行う。
変更要求	RFID_3	カセット搬出時、RFIDにパネル枚数を書き込むようにしたい。
	理由	カセット搬出後も、カセット内のパネル数をRFIDに記憶しておきたいため。
	説明	
変更仕様	加工前パネルカセット搬出処理の変更	
□□□	RFID_3.1	【Before】 パネルを加工装置に全て投入完了後、カセットのロックを解除する。 【After】 RFIDに"0"を書き込み、ロックを解除する。
変更仕様	加工後パネルカセット搬出処理の変更	
□□□	RFID_3.2	【Before】 加工されたパネルを全てカセットに格納完了後、カセットのロックを解除する。 【After】 RFIDに格納パネル数を書き込み、ロックを解除する。
変更要求	RFID_4	ディスプレイにカセット内のパネル残数をリアルタイムに表示させたい。
	理由	カセット内のパネル残数をリアルタイムに確認できるようにしたいため。
	説明	
変更仕様	加工前パネル投入処理の変更	
□□□	RFID_4.1	【Before】 該当処理なし。 【After】 格納パネル数から投入パネル数を差し引きパネル残数を求めディスプレイに表示する。
変更仕様	加工後パネル取出し処理の変更	
□□□	RFID_4.2	【Before】 該当処理なし。 【After】 格納パネル数に取出しパネル数を加えパネル残数を求めディスプレイに表示する。

【要求】
RFIDでカセット内のパネルの数を管理したい

【理由】
カセット内パネル搬送の時間を短縮するため。現状は、パネルの有無に拘らず、最終パネル位置まで搬送動作を行っているため、パネルが無い位置での搬送動作が無駄になる。

【変更要求1】
RFIDからパネル数を読み込み、読込んだ値が異常値であれば、カセットを搬出したい。

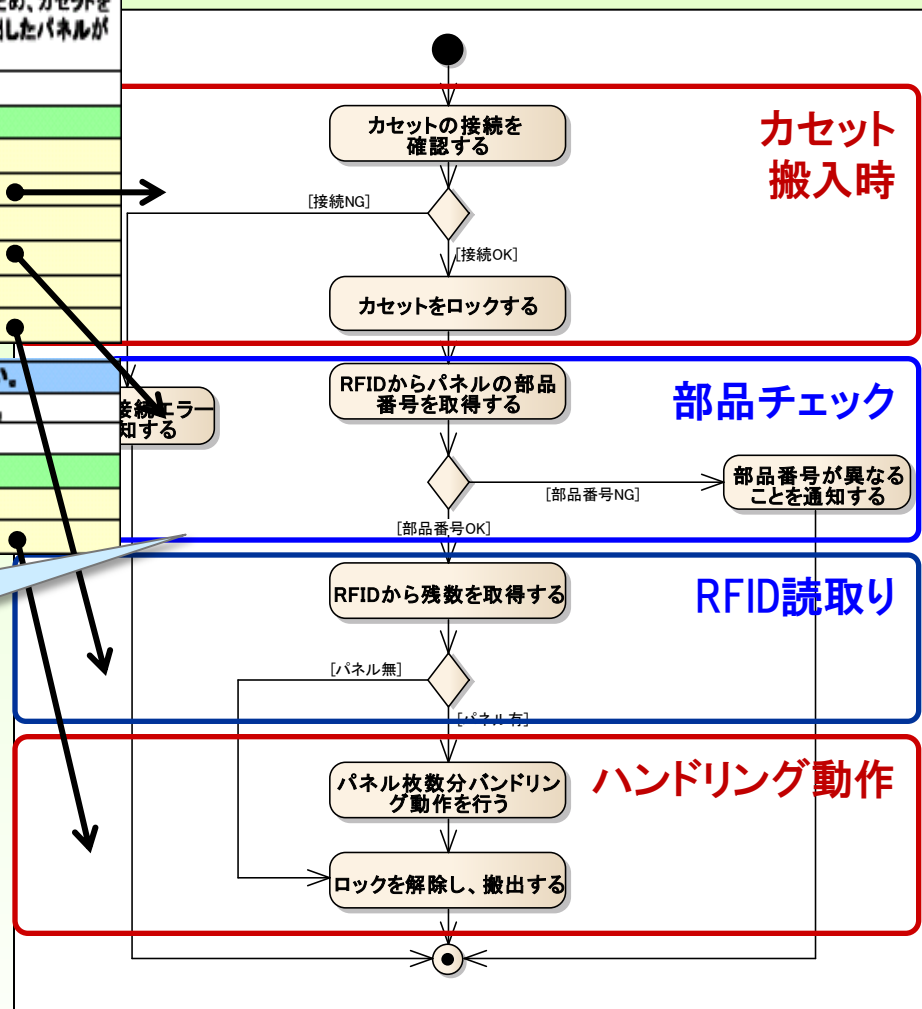
【変更要求2】
パネル投入時、RFIDから読込んだパネル数分投入完了後、ハンドリング動作を止めたい。

【変更要求3】
カセット搬出時、RFIDにパネル枚数を書き込めるようにしたい。

【変更要求4】
ディスプレイにカセット内のパネル残数をリアルタイムに表示させたい。

3-7. サンプル事例(変更要求モデルの効果-1)

変更要求1 RFID_1		RFIDからパネル数を読み込み、読み込んだ値が異常値であれば、カセットを搬出したい。
	理由	加工前パネルカセットのRFIDにパネル数"0"が書込まれている場合、搬入パネルが無い場合、カセットを搬出する必要がある。また、加工後パネルカセットにパネルがある場合、加工装置から取出したパネルがカセット内のパネルと衝突する可能性があるため。
	説明	
変更仕様 < 加工前パネルカセット搬入処理の変更 >		
□□□	RFID_1.1	【Before】 カセット搬入時、カセットをロックする。 【After】 カセットをロックし、RFIDからパネル数を読み込む。
□□□	RFID_1.2	【Before】 該当処理なし。 【After】 パネルの部品番号が異なる場合、異なる部品番号を通知する。
□□□	RFID_1.3	【Before】 該当処理なし。 【After】 パネル数が"0"であれば、カセットのロックを解除し、搬出する。
変更要求2 RFID_2		パネル投入時、RFIDから読み込んだパネル数分の投入完了後、ハンドリング動作を止めたい。
	理由	存在しないパネルに対してハンドリング動作をすることになり、その時間が無駄になるため。
	説明	
変更仕様 < 加工前パネルカセットハンドリング動作の変更 >		
□□□	RFID_2.1	【Before】 カセット格納パネル数と無関係に最終パネル位置までハンドリング動作を行う。 【After】 RFIDから読み込んだパネル数分のハンドリング動作を行う。

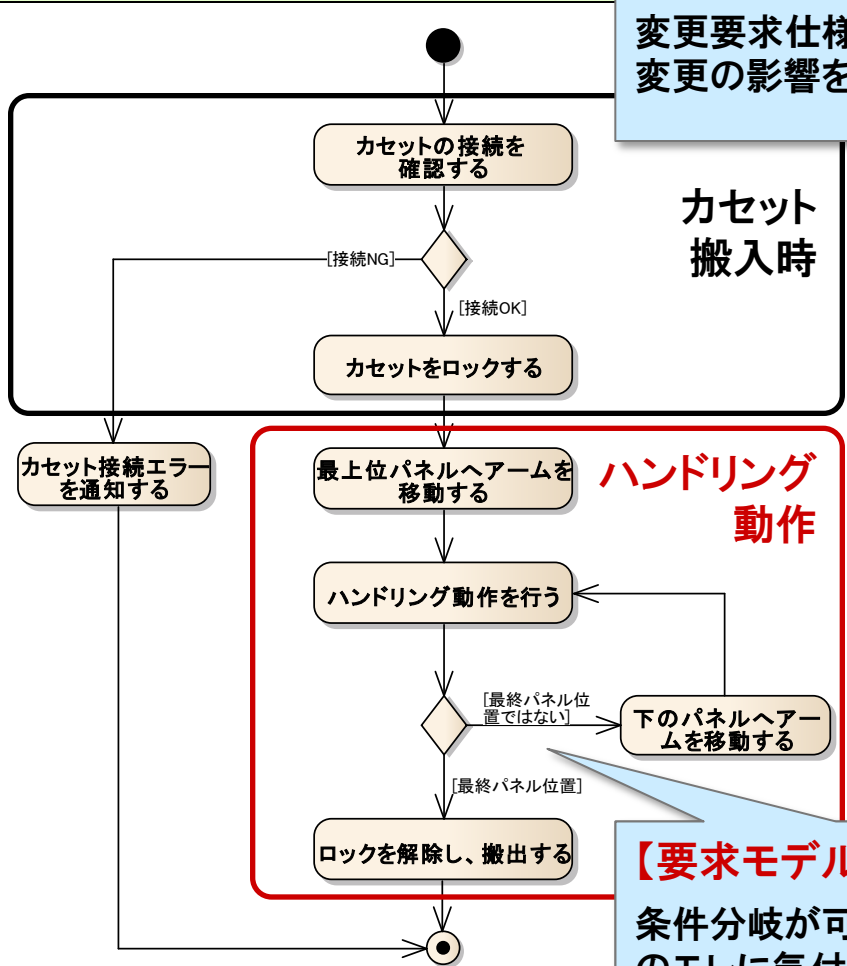


【要求モデルの効果1】
カセットのロックからパネル数を読み込むまでの間に追加機能が入ることを確認できた

【要求モデルの効果2】
各アクションの結果に対する条件のモデルに気付くことができた

3-8. サンプル事例(変更要求モデルの効果-2)

Before



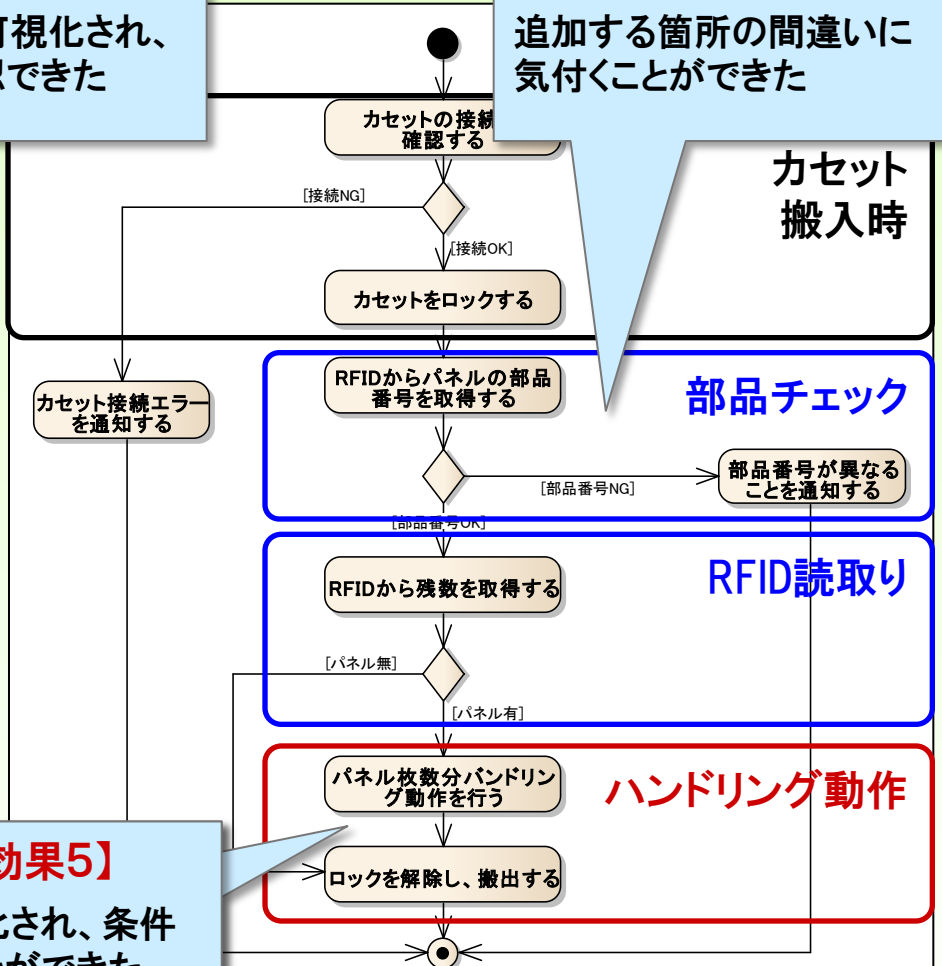
【要求モデルの効果3】
変更要求仕様が可視化され、
変更の影響を確認できた

カセット
搬入時

ハンドリング
動作

【要求モデルの効果5】
条件分岐が可視化され、条件
のモレに気付くことができた

After



【要求モデルの効果4】
追加する箇所間違いに
気付くことができた

カセット
搬入時

部品チェック

RFID読取り

ハンドリング動作

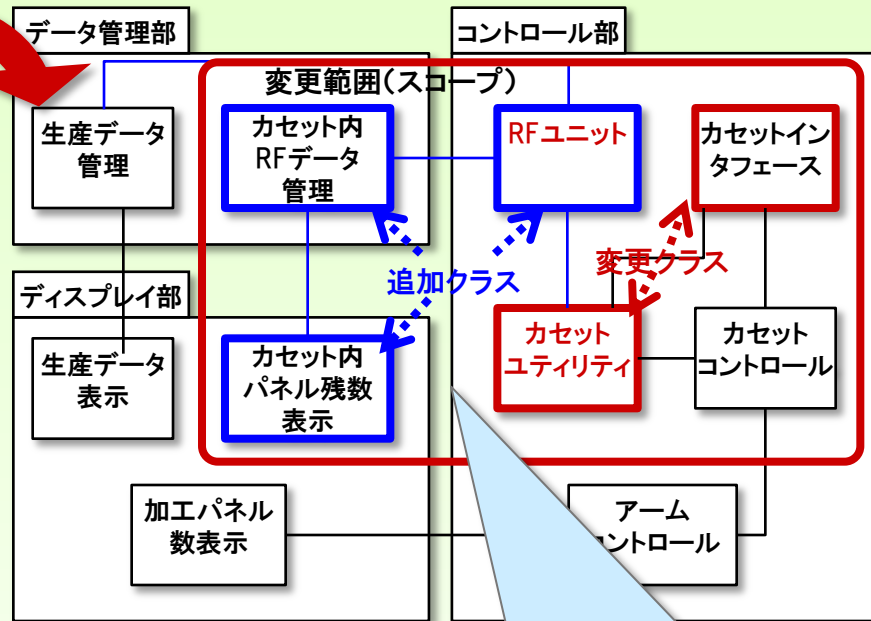
3-9. サンプル事例(変更設計モデルの効果)

これまでの派生開発

構造を見える化

モデル主体の派生開発

変更要求仕様書	生産データ管理 ディスプレイ部	生産データ管理 データ管理部	生産データ管理 コントロール部	RFユニット コントロール部	RFユニット コントロール部
タイトル	RFIDを用いたカセット内パネル数管理				
要求	RFIDでカセット内のパネルの数を管理したい				
理由	カセット内のパネル搬送の時間を短縮するため。現状は、パネルの有無に拘らず、最終パネル位置まで搬送動作を行っているため、パネルが無い位置での搬送動作が無駄になる。				
説明	カセット搬出時にRFIDへの読み込みを行う。 ・加工前パネルカセットには、搬出時に"0"を読み込む ・加工後パネルカセットには、最終搬出後を読み込む カセット搬入時にRFIDの読み込みを行う。 ※搬込んだパネル数により、最終パネル位置を割出す				
変更要求1	RFID_1	RFIDからパネル数を読み込み、搬込んだ値が異常値であれば、カセットを搬出したい。			
理由		加工前パネルカセットのRFIDにパネル数"0"が読み込まれている場合、搬入パネルが無い場合、カセット搬出する必要がある。また、加工後パネルカセットにパネルがある場合、加工搬出から取出したパネルのカセット内のパネルと搬送する可能性があるため。			
説明					
変更仕様	加工前パネルカセット搬入処理の変更				
□□□	RFID_1.1	[Before] カセット搬入時、カセットをロックする。 [After] カセットをロックし、RFIDからパネル数を読み込む。	○	○	○
□□□	RFID_1.2	[Before] 搬出処理なし。 [After] パネルの部品番号が異なる場合、異なる部品番号を通知する。	○	○	○
□□□	RFID_1.3	[Before] 搬出処理なし。 [After] パネル数が"0"であれば、カセットの口を開け、搬出する。	○	○	○
変更仕様	加工後パネルカセット搬入処理の変更				
□□□	RFID_1.4	[Before] カセット搬入時、カセットをロックする。 [After] カセットをロックし、RFIDからパネル数を読み込む。	○	○	○
□□□	RFID_1.5	[Before] 搬出処理なし。 [After] パネル数が0以外であれば、カセットの口を開け、搬出する。	○	○	○
変更要求2	RFID_2	パネル搬入時、RFIDから搬込んだパネル数分の搬入完了存在しないパネルに対してハンドリング動作をすることになり			
理由					
説明					



【予想されたアーキテクチャの品質劣化】
 ソースコードレベルの変更仕様と既存の構成要素だけに着目してしまい、本来あるべきアーキテクチャの観点置き去りになる
 その結果、RFID機能に関する変更がカセットユティリティクラスに集中し、メソッド数が24から33に増加する

【設計モデルの効果】
 モデル化により、RFID機能はカセットユティリティの責務ではないことに気付き、RFユニットクラスを新たに追加して機能分散させた
 その結果、カセットユティリティクラスのメソッド数増加による肥大化を防げた

3-11. サンプル事例による効果検証のまとめ

No.	課題項目	課題内容	検証できた効果
1	開発プロセス	派生開発に特化したモデル駆動開発プロセスが確立されていない	今回該当なし
2	変更要求仕様	変更要求仕様の文章表現では曖昧さや個人差があり分かりにくい	USDMとアクティビティ図の連携により、フローと分岐が可視化され、ミス、モレに気付けた
3	スペックアウト	スペックアウトの成果物作成について有効なガイドがない	変更スコープ特定図、変更設計モデル(Before)、変更要求モデル(Before)を作成するスペックアウトガイドができた
4	変更設計	変更設計仕様の記述がプログラム内の視点になりやすく変更漏れが発生しやすい	変更範囲内の構造を可視化することで、構成要素の責務が明確になり、最適な要素へ変更を行うことで設計品質の劣化を防げた
5	影響範囲特定	変更の影響範囲の特定や局所化について具体的な方法がない	変更スコープ特定図や変更TMのモデル要素により、変更箇所とそれ以外の部分が明確になり、影響範囲の確認が容易になった
6	アーキテクチャ	機能の変更・追加の中ではソフトウェアアーキテクチャの劣化防止や改善が難しい	パッケージレベルから変更箇所をモデルで表現することで、アーキテクチャレベルの品質劣化を防止できた
7	開発支援ツール	派生開発を支援する強力なモデリングツールがない	今回該当なし

派生開発カンファレンス2013

ソースコード主体の派生開発からモデル主体の派生開発へ ～ 設計の見える化による設計品質の維持と改善 ～

Contents

1. T20研究会のメンバと活動
2. 課題分析と解決策検討
3. サンプル事例による効果検証
- ★ 4. 今後の課題と方針

4-1. 今後の課題

■ 今後の活動計画

(1) XDDPへのモデリング手法導入:

【検討予定項目】

- ・ モデルによるリファクタリング
- ・ 機能追加のモデル化

(2) モデル駆動開発へのXDDP導入:

モデル駆動開発における派生開発プロセスについて検討する

(3) 影響範囲のモデル化:

モデル化する上での影響度、影響範囲の分析と変更範囲の判断に至るプロセスを検討する

(4) モデリングツールとの連携:

派生開発でモデリングツールの支援が必要な部分を検討して具体化する

4-2. 今後の方針

■ ガイドラインの作成と公開

「モデル主体派生開発ガイドライン」の構成案:

- プロセス定義
- 成果物と作成手順
- スペックアウト技法
- 事例紹介

■ 他研究会との連携

本研究会テーマと関連している研究会テーマ:

- T5 : 影響箇所の気付き
- T8 : 大規模システムへの効果的対応
- T9 : ビジネス領域での「XDDP」の活用
- T14: SPLと「XDDP」の連携
- T18: USDMと形式言語との接合における曖昧表現の克服
- T19: 派生開発におけるスペックアウトの仕方

派生開発カンファレンス2013

END

ソースコード主体の派生開発からモデル主体の派生開発へ
～ 設計の見える化による設計品質の維持と改善 ～

2013. 5. 24

派生開発推進協議会 T20研究会
(株)日立情報制御ソリューションズ

渡辺 滋