



不完全な業務フロー図から 網羅的にテスト設計するための状態遷移図の活用事例

2018年11月30日
株式会社NTTデータ 町田欣史

発表の流れ

1. **業務フロー図を用いたテスト設計の問題と対策**
2. **状態遷移図を用いたテスト設計の問題と対策**
3. **適用事例**
4. **まとめ**

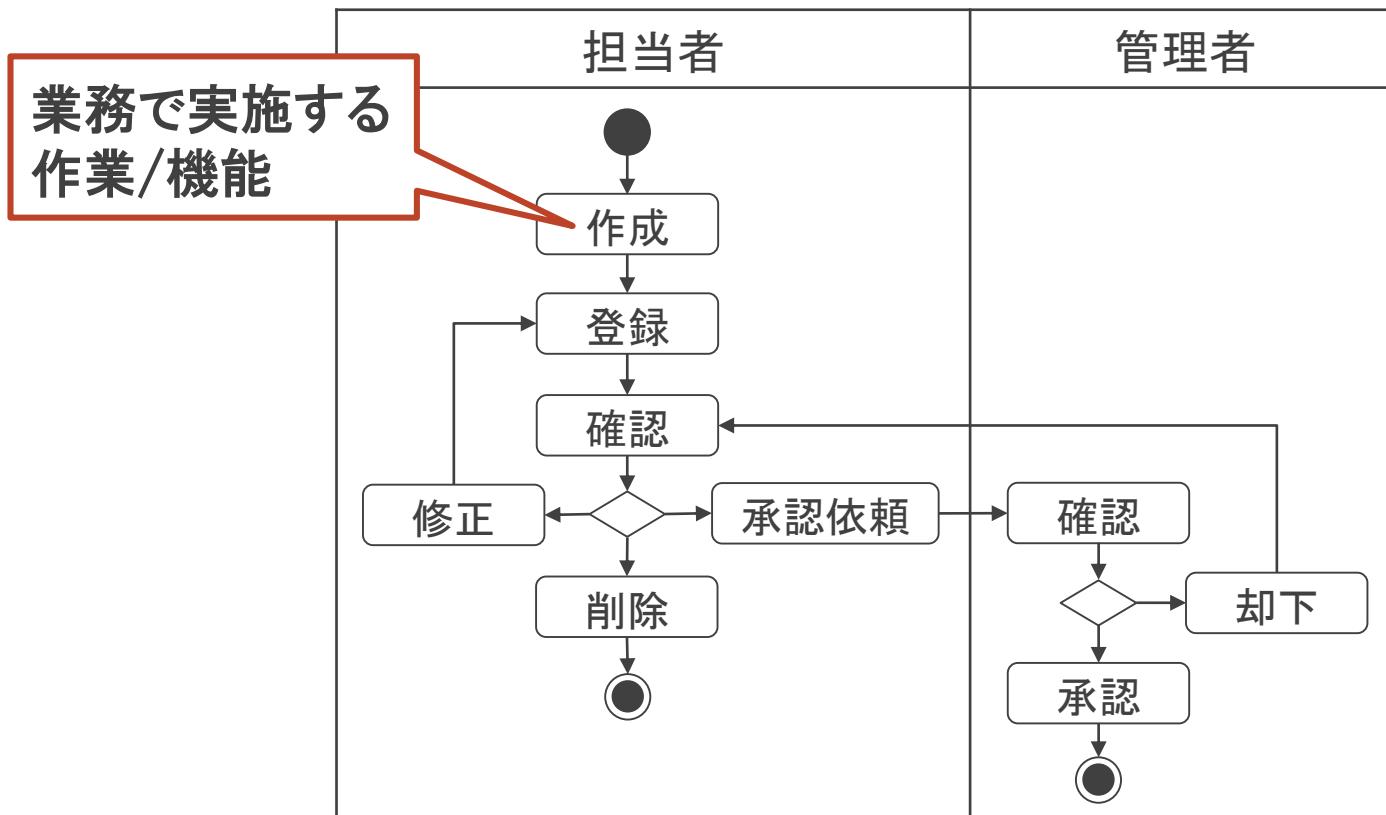
1. 業務フロー図を用いたテスト設計の問題と対策

1.1 業務フロー図とは

業務システム開発では業務フロー図を作成することが多い

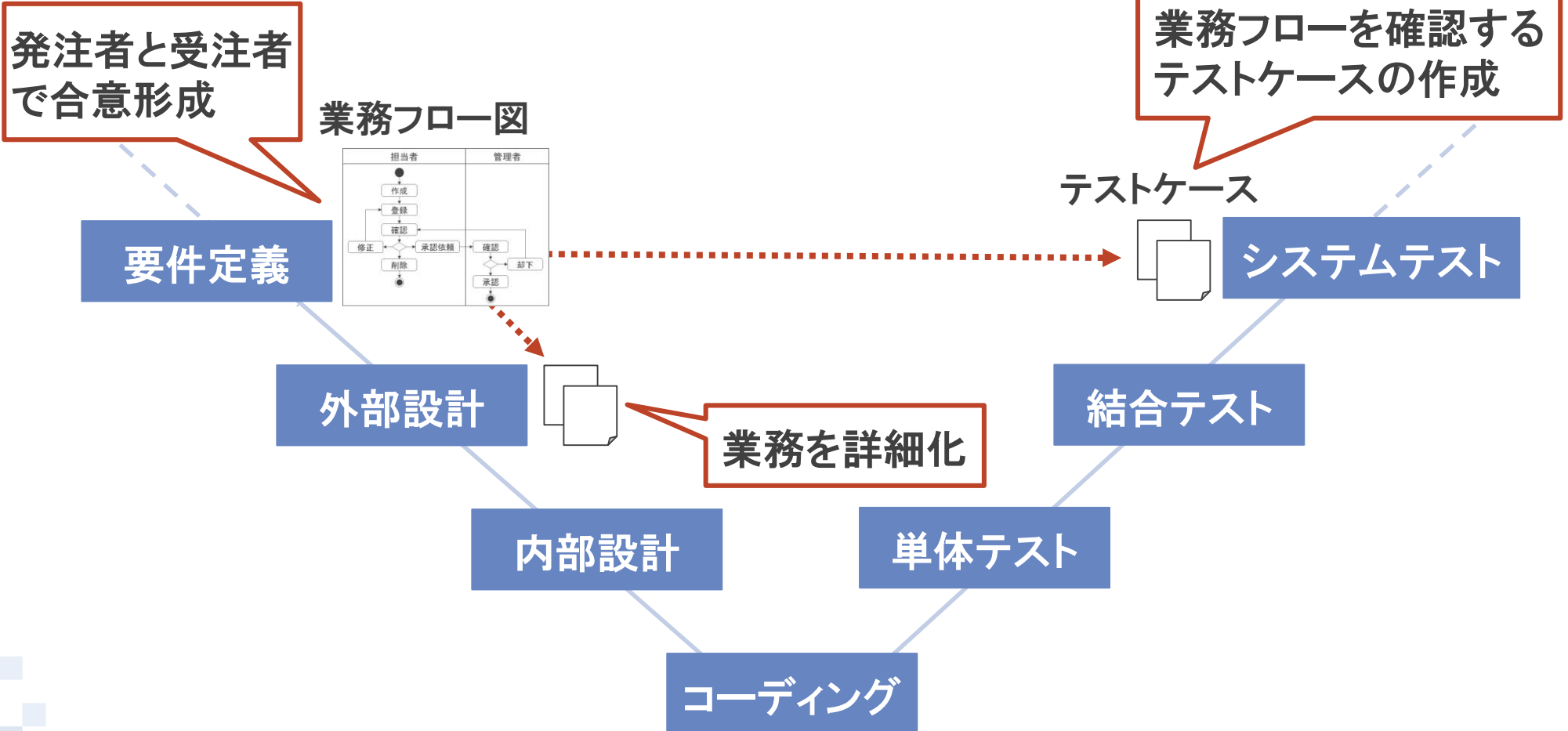
業務における作業の流れ(業務フロー)とシステム化する作業(機能)を明確にする

UMLのアクティビティ図で書いた例



1.2 業務フロー図の位置づけ

業務フロー図に基づき、設計やテストが行われる



1.3 業務フロー図の問題

要件定義工程で業務フローが網羅的に書かれないことが多い

業務フロー図に書かれる業務フローの種類

基本フロー	利用者が業務の目的を達成するときのごく典型的なフロー
代替フロー	基本フローとは別の手段で業務を完了させるフロー
例外フロー	成功を妨げる条件下でのフローや特定の条件下でのフロー



特に代替フロー・例外フローの一部が書かれないことがある



不完全な業務フロー図

1.4 業務フローが網羅的に書かれない理由

業務フロー図が不完全になる理由はいくつか考えられる

① 要件が固まっていない

- 要件定義工程ですべての業務フローを抽出するのが難しい。

② ステークホルダーが理解できない

- システム開発の知識に乏しい人は厳密すぎる業務フロー図を見ても理解できない。

③ 業務が複雑すぎる

- システム化対象の業務が複雑すぎると、業務フロー図も必然的に複雑になるため、省略せざるを得ない。

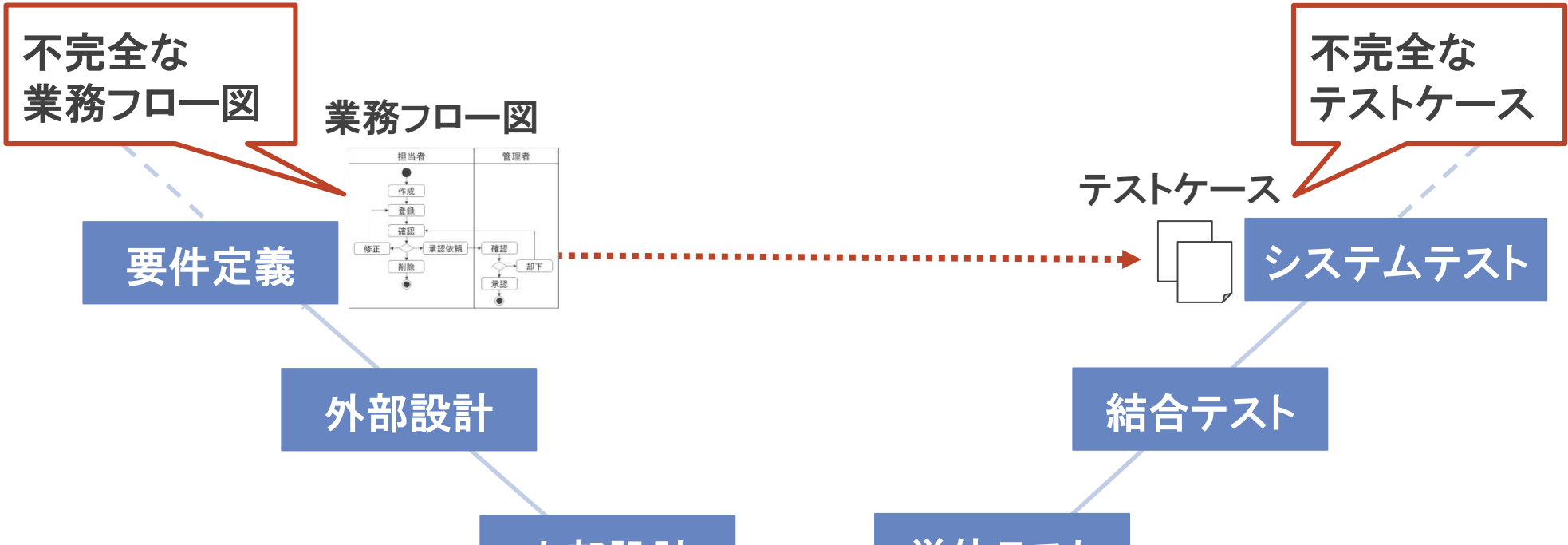
④ 「網羅」の基準が高すぎる

- 要件定義工程で書くべき以上のことを求めている。

これは最後の「まとめ」で触れる

1.5 不完全な業務フロー図を基にしたテストの問題

不完全な業務フロー図では十分なテスト設計ができない



本発表における課題：
この状況でどうやって網羅的にテスト設計をするか

1.6 対策案(1) 完全な業務フロー図を書く

本来あるべき対策だが、複雑な業務には適用しづらい

2つの問題をクリアする必要がある。

① 要件が固まっていない

→ 要件が固まってから業務フロー図を詳細する

② ステークホルダーが理解できない

→ 発注者向けと開発・テスト担当者向けで異なる業務フロー図を書く

対策案 要件定義工程の後に、テストのために業務フロー図を完全化する

さらにもう1つの問題がネックになる。

③ 業務が複雑すぎる

→ 業務フロー図が複雑で、テスト設計が困難である

不採用

1.6 対策案(2) 不完全な業務フロー図のままではなんとかする

状態遷移図を利用することで状態遷移テスト技法を使える

業務フロー図に書かれていない情報を補う必要がある



この情報が書かれるドキュメントはないか？

業務フロー図は、ある作業を行っている状態とその遷移を表現しているため、**状態遷移図の一種**と考えることができます

竹内正治、“業務システム分析のためのUMLモデリング演習 第7回 情報の流れと状態、”
日経ITプロフェッショナル、10月号、130-135、2005



対策案

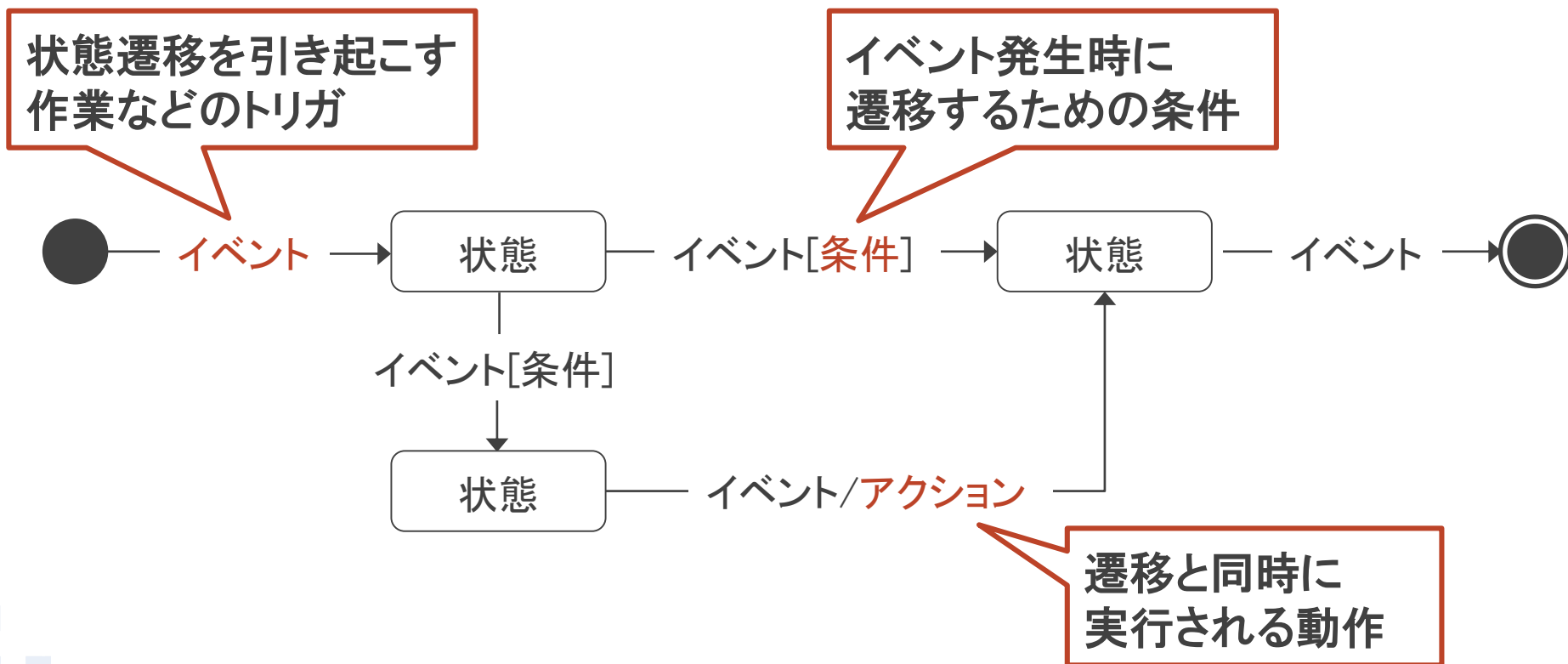
業務フロー図に記述されない例外/代替シナリオを表現するため、**状態遷移図**を利用する **採用**

2. 状態遷移図を用いたテスト設計の問題と対策

2.1 状態遷移図とは

システムの状態に着目して、システムの振る舞いをモデル化する

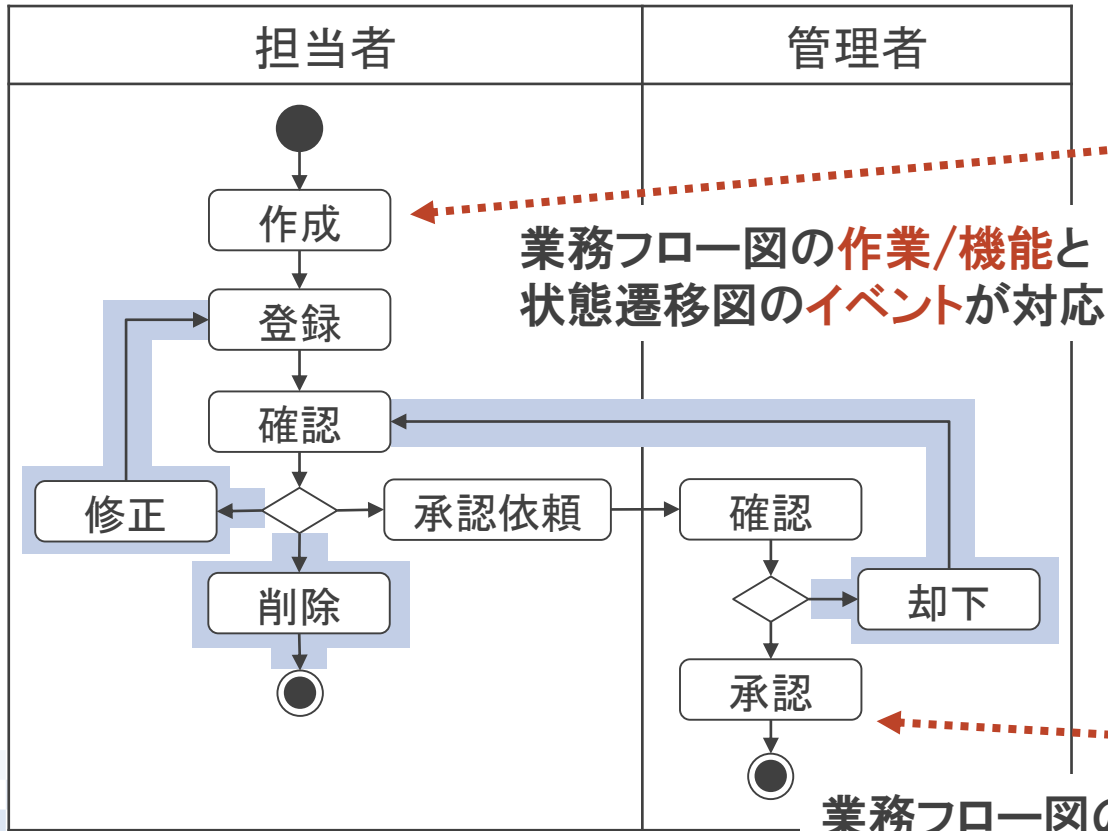
対象とするシステムの状態の遷移に沿って、遷移を起こすイベントや遷移条件が記述される。



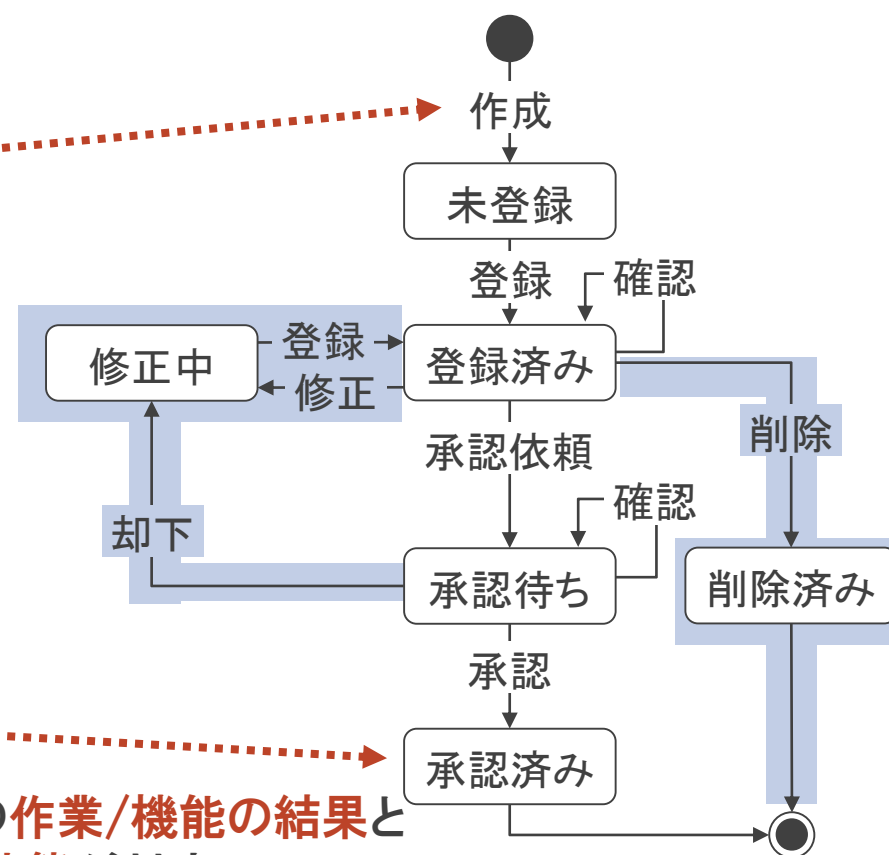
2.2 業務フローと状態遷移図の関係

状態遷移図でも業務フローの代替/例外フローを表現できる

業務フロー図



状態遷移図



2.3 状態遷移図の問題点

業務システム開発では、状態遷移図を作成しないことがある

状態遷移図を作成しない理由

- ①業務全体の流れが分かりにくいので、発注者との合意形成が難しい
➡ システムの挙動について開発者間で認識を合わせるためには必要では？
- ②業務システムでは「状態」を意識することが少ない
➡ 画面遷移図は書くが、システムの状態はあまり気にしない
- ③業務システムの「状態」は、設計が進まないと考慮するのが難しい
(状態の例) ・ データの状態
 ・ 画面の状態

テストのために状態遷移図を後から作成する

状態遷移図がない場合に、どうするか

案1 業務フロー図を完全にする

→ 業務が複雑だと・・・という問題に逆戻りする

不採用

案2 状態遷移図以外に使えるものを探す

→ 何かあるかもしれないが、現時点で妙案がない

不採用

案3 状態遷移図を後から作る

→ 工数の増加が許容できれば実現できる

採用

2.5 状態遷移図の作成

要件定義工程後に作成したドキュメントから状態遷移図を作成する

状態遷移図の作成に必要なドキュメント

- 業務フロー図
- 業務を詳細化した設計書 ← 要件定義後に作成したドキュメントを利用

状態遷移図の作成に必要な記述内容

要素	必要な記述内容
状態	<ul style="list-style-type: none">• 機能実行後の結果 (例) データベースの更新結果
イベント	<ul style="list-style-type: none">• 業務フロー図上の機能/作業• 機能において状態を変えるトリガ (例) 画面上で発生するイベント 画面遷移が発生するイベント バッチ処理を起動するイベント

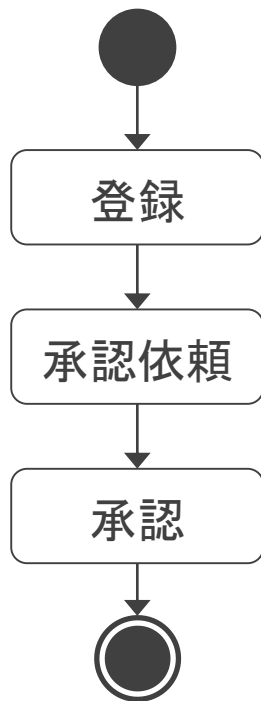
要素	必要な記述内容
条件	<ul style="list-style-type: none">• 機能において状態が変わる条件 (例) 入力チェック条件 画面状態を決める条件 画面遷移条件
アクション	<ul style="list-style-type: none">• 状態が変わるときに行う作業

3. 適用事例

3.1 適用事例の概要

- 金融系の大規模システム開発プロジェクト
- 状態遷移図を作成していない → **テスト設計者が状態遷移図を作成**
- 業務フロー図の記述が粗い

適用した業務フロー図の例 … 基本フローしか書かれていない

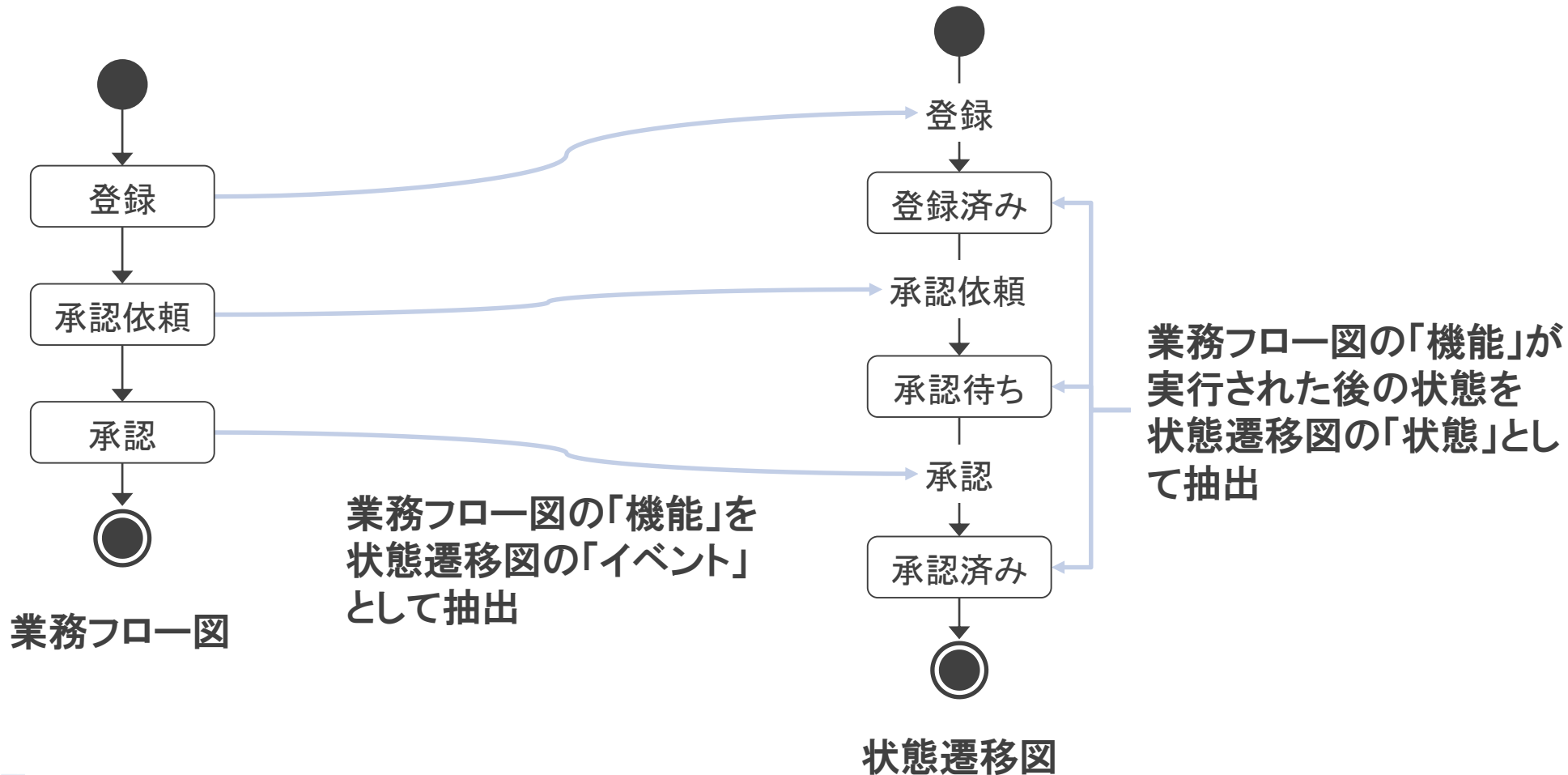


3.2 状態遷移図の作成に使用したドキュメント

抽出する要素	ドキュメントに記述される内容	ドキュメント			
		業務フロー図	処理設計書	画面定義書	画面遷移図
状態	機能実行後の結果	✓	✓		
イベント	業務フロー図上の機能/作業	✓			
	機能において状態を変えるトリガ		✓	✓	✓
条件	機能において状態が変わる条件		✓	✓	✓
アクション	状態が変わるときに行う作業		✓		

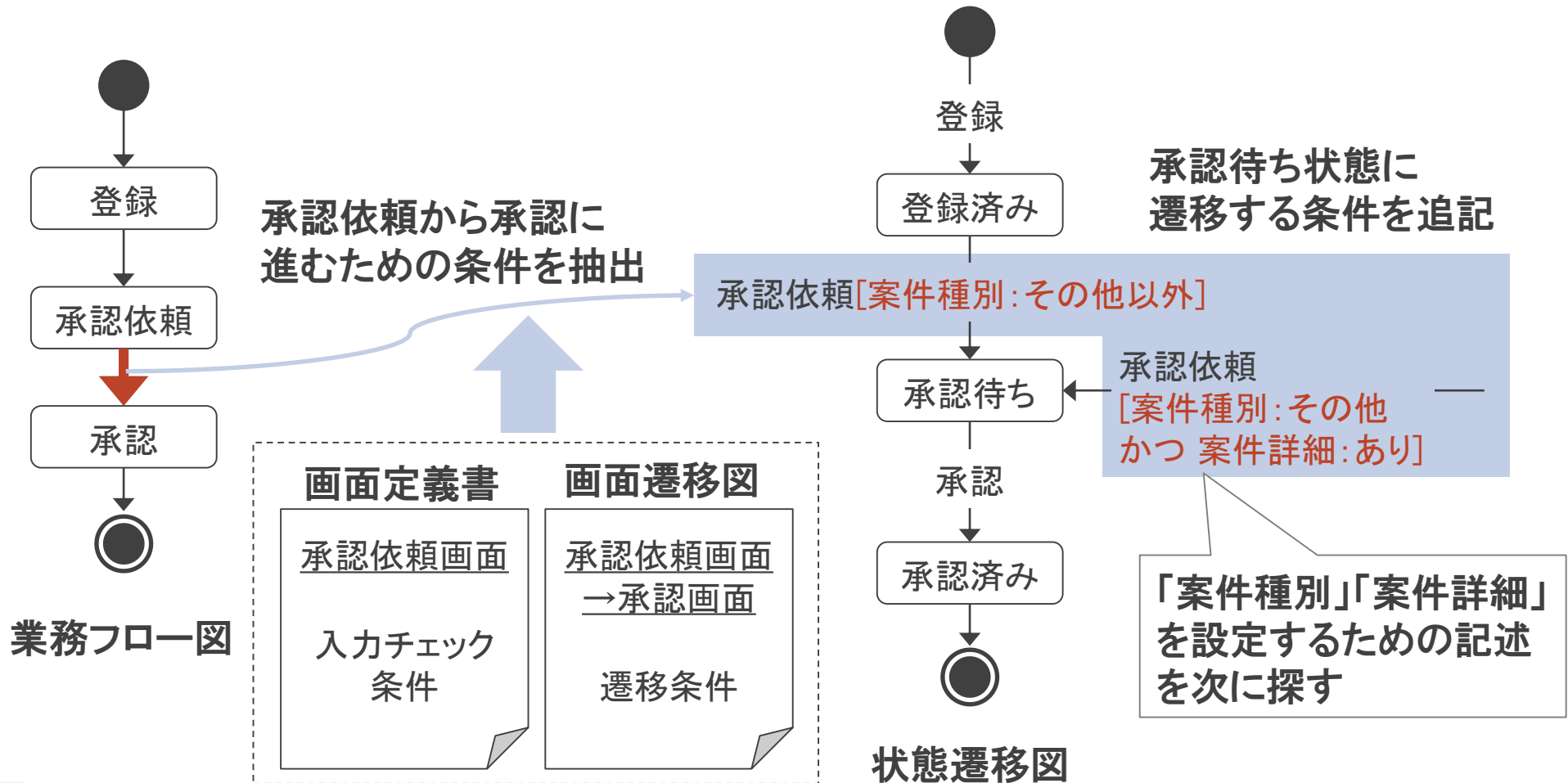
3.3 適用手順 (1)基本フローに対応する状態遷移の作成

業務フローの基本シナリオに対応する状態遷移を作成する



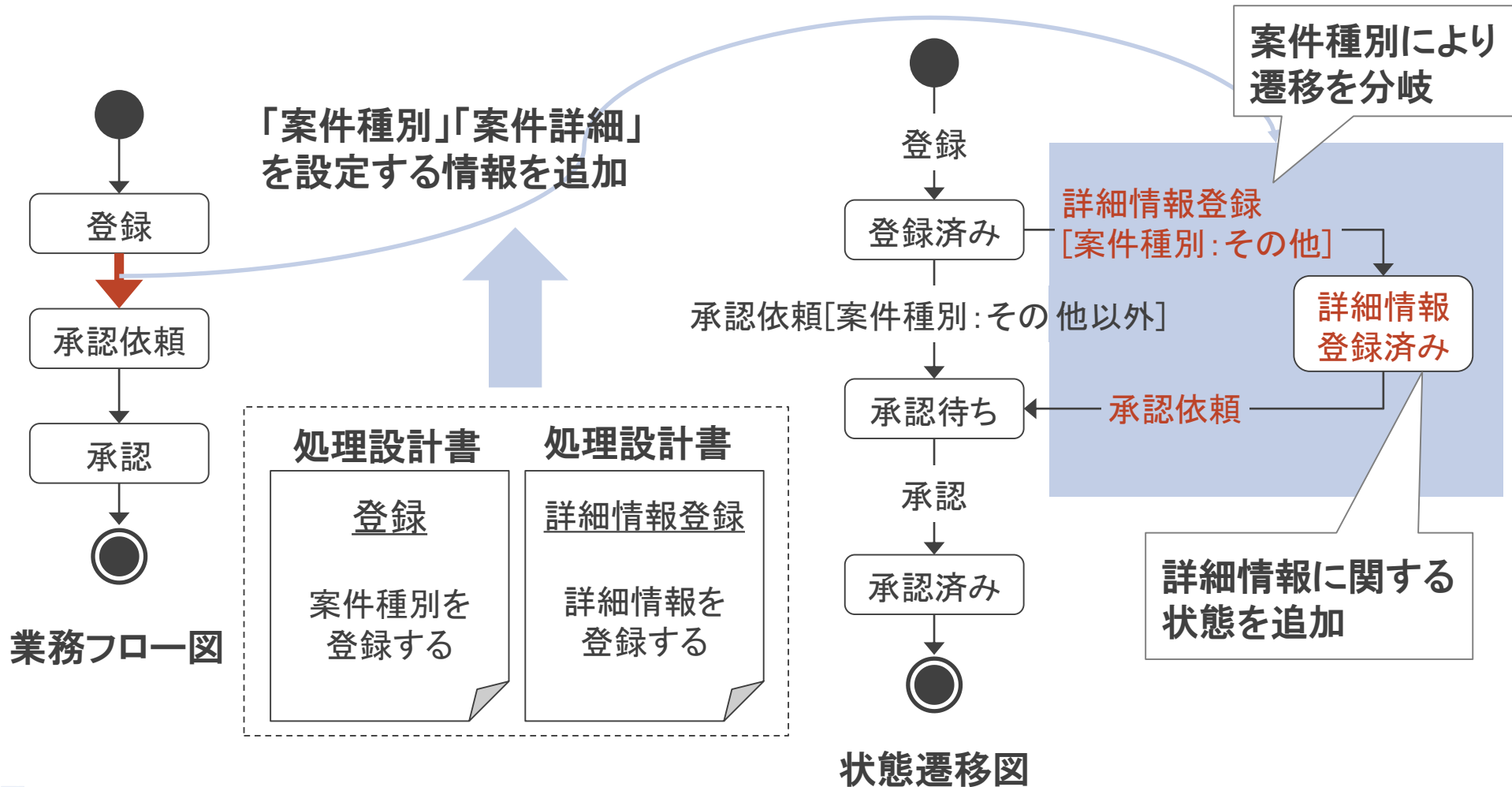
3.3 適用手順 (2)状態遷移する条件の抽出

ある状態に遷移するための詳細な条件を抽出する



3.3 適用手順 (3)代替/例外フローに相当する状態遷移の追加

代替/例外フローに相当する状態遷移を設計書から導出する



3.4 適用結果

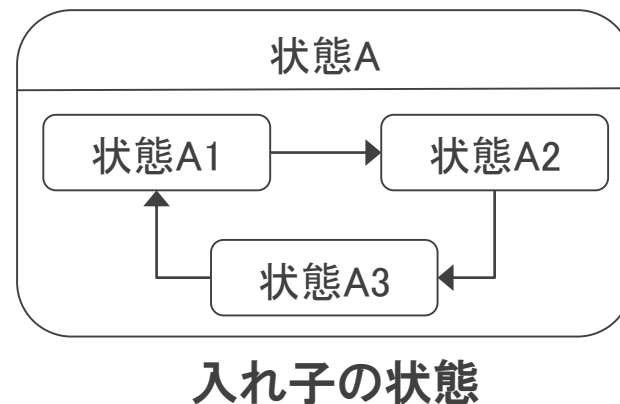
代替/例外フローを状態遷移図で表し、テストケースを抽出できた

紹介した業務フローについて

- 業務フロー図上では1本のフローで表現されていたが、データの種別により追加機能を実行する必要がある代替フローを補完できた。

その他の業務フローについて

- 「取消」に関する業務フローが「取消通知受領」機能から始まっている、すなわち「取消」ができる／できない条件が記載されていなかったが、本手法を適用することで、それらが明確になった。
- 入れ子の状態をもつ複雑な業務フローに対して、状態遷移図を作成することで、代替/例外フローを補完できた。



4. まとめ

4.1 得られた効果

網羅的なテスト設計だけでなく、品質改善やスキル向上も実現した

テスト設計の網羅性向上

状態遷移図を作成することで、代替/例外フローに相当する状態遷移を抽出した。それを基に状態遷移テスト技法を適用することで、業務フロー図からのテスト設計では漏れていたテストケースを追加できた。

設計書の品質改善 ※副次的効果

業務フローの詳細な条件を確認することで、設計の不統一を発見できた。

テスト設計者の業務知識向上 ※副次的効果

業務フロー図だけでなく、様々な種類の設計書を読み込む過程で、業務知識の拡大につながった。

4.2 今後の課題(1)

業務システム開発における状態遷移図について改めて考えるべき

作成するドキュメントについて

- 状態遷移図が有効なら初めから作っておけばよいのでは？
 - ➡ 状態遷移図を作成する意義や作成方法を見直し、開発者に理解させる必要がある。
- 状態遷移図が(元の)業務フロー図を包含しているなら、状態遷移図だけでよいのでは？
 - ➡ 発注者が理解できるかどうか大きな課題だが、業務システムの開発者も理解できない人が多いと思われる。
- 完全な業務フローを書くのと、どちらがよいのか？
もしくは、両者を適切に使い分けるべきなのか？
 - ➡ どの方法が最も効率的か、効果的かを検証してみたい。

モデルの抽象度を適切に決める必要がある

業務フローに対するテストで確認すべきことについて

- 要件定義よりも後に作成された成果物から状態遷移図を作成し、テストケースを作成している。
すなわち、本来は別のテストレベルで確認すべき内容をテストしているのでは？
 - ➡ テストを無駄に増やしすぎている可能性がある。
このテストレベルで求められる詳細度を再考する必要がある。
- 業務フローは本当に不完全だったのか？
要件定義で求められる詳細度を超えていたのでは？
 - ➡ 業務フロー図に書くべき代替/例外フローの詳細度を再考する必要がある。
と同時に、開発プロセス全体で、どの工程(レベル)で何を定義すべきかを決める必要がある。



NTT DATA

Trusted Global Innovator

まとめと今後の課題

■提案手法の特徴のまとめ

【利用シナリオ】

業務システムの受託開発プロジェクトのテストにおける、業務フロー図に記述されていない代替/例外シナリオの補完

【メリット】

業務フロー図と各業務を詳細化した設計書から、状態遷移図に相当する設計情報を抽出し、代替/例外シナリオを補完できる

【制約】

提案手法の適用が過剰となる業務フロー図が存在する

■今後の課題

適用案件を拡大しての適用効果の検証・適用条件の具体化

- 適用効果の定量的な測定
- 適用条件の具体化(適用が過剰となる業務フローの特徴づけ)

【課題】

業務フロー図をテストベースとしたテストとしては粒度が細か過ぎる分岐も抽出してしまう。状態遷移図が必要となるほど複雑な状態遷移をする業務は限られている[8]