業務フローのテスト設計における状態遷移の抽出による業務シナリオ補完手法

A Business Process Complementation Method using Abstraction of State Transition in Test Design

株式会社 NTT データ 技術革新統括本部 システム技術本部
プロジェクトマネジメント・ソリューションセンタ
NTT Data Corporation Project Management Solution Center Production Engineering Department
System Engineering Headquarters
〇井ノ口　伸人　　熊川　一平　　町田　欣史
〇Nobuto Inoguchi　　Ippei Kumagawa　　Yoshinobu Machida

Abstract A business process diagram is drawn in requirement analysis phase of business application development projects. Consensus between a customer and a supplier is built based on the business process diagram. The business process diagram is also used as input of design phase and test phase. In test phase, business process diagram is insufficient because some of business scenarios are not described on it and it is difficult to achieve high coverage of business scenarios. This paper proposes a business process complementation method using abstraction of state transition in test design.

1. はじめに

業務システムの受託開発プロジェクトにおいて、要件定義工程の成果物として、業務フロー図が作成されることが多い。業務フロー図に記述されるのは、業務全体の流れにおけるシステム化範囲と作業概要である。その記述内容に基づき、発注者と開発者との合意形成が行われ、設計工程やテスト工程の成果物が作成される。

業務フロー図をテスト工程の入力成果物（以降テストベース）として利用するとき、実務で実行される業務シナリオがすべて記述されておらず、十分なテストを設計できないことが多い。業務目的を達成できる典型的なシナリオである基本シナリオは記述されるが、基本シナリオではないが業務目的を達成できる代替シナリオや、業務目的を達成できない例外シナリオの記述が一部に留まりがちである。その原因として、すべてのシナリオを記述してしまうと、業務フロー図が煩雑になり、合意形成に時間を要してしまうことが挙げられる。また、要件定義工程で詳細に記述したとしても、後工程でさらに分岐が必要だと判明することもある。例えば、業務フロー図では、1 本の業務シナリオとして記述されていても、後工程で判明した代替/例外シナリオへの分岐が実装される場合がある。

業務フロー図に記述されない代替/例外シナリオを表現するため、状態遷移図を利用する方法がある[1]。状態遷移図ならば、業務フロー図として記述すると煩雑になりがちな代替/例外フローを

株式会社 NTT データ 技術革新統括本部 システム技術本部 プロジェクトマネジメント・ソリューションセンタ
NTT Data Corporation Project Management Solution Center Production Engineering Department System Engineering Headquarters

東京都江東区豊洲 3-3-3 豊洲センタービルアネックス Tel: 050-5546-8388 e-mail:inoguchin@nttdata.co.jp
Toyosu Center Bldg. Annex, 3-9 Toyosu 3-chome, Koto-ku, Tokyo

【キーワード：】ソフトウェアテスト、テスト設計、業務フロー、ユースケーステスト、状態遷移テスト

簡潔に表現できる。

しかし、本論文で想定する業務システムの受託開発においては、上流工程では状態遷移図が作成されない事例が多い。状態遷移図を使った合意形成が難しいためだと考えられる。状態遷移図が、『機能要件の合意形成ガイド』[3]において、合意形成に使う主な図表に含まれていないことからも、業務システムの受託開発においては浸透していないと推測できる。また、要件定義工程以後に判明した分岐が、業務フロー図に反映されないことも多い。

そこで、本論文では、業務フロー図および各業務を詳細化した設計書から状態遷移を抽出し、業務フロー図に記載されていない代替/例外シナリオを補完したテストを設計する手法を提案する。「2 業務フロー図と状態遷移図」で、提案手法の入出力となる、業務フロー図、状態遷移図について記載する。続く「3 提案手法」で、状態遷移を抽出して代替/例外シナリオを補完する手法を説明し、その適用結果を「4 ケーススタディ」で報告する。最後に「5 結論」で、本手法の効果と制約および今後の課題について述べる。

2. 業務フロー図と状態遷移図の特徴

2.1 業務フロー図の特徴

業務フロー図には、業務の流れに沿ってシステム化される業務の位置づけが記述される。複数の記法が存在するが、本論文ではUnified Modeling Language(以降UML)[2]のアクティビティ図の記法を用いる。ただし、本手法は特定の記法へ依存するものではなく、利用するのは表 1に示す最小限のモデル要素である。記述の凡例を図 1 アクティビティ図の記述凡例に示す。

表 1 利用するアクティビティ図のモデル要素

モデル要素名	モデル要素	説明
アクション	アクション	業務フロー上で実施する個々の業務。
フロー	↓	業務の流れ。
判断、マージ	◇	業務フローの分岐と合流。
レーン	役割区分	利用者の役割区分。
開始	●	業務の開始点。
終了	◉	業務の終了点。

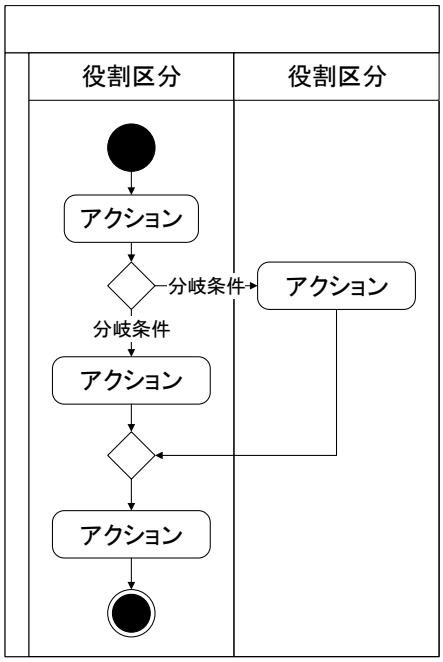

図 1 アクティビティ図の記述凡例

業務フロー図をテストベースとして利用する際、二つの問題が発生する。一つは十分なテストを設計できないリスクである。このリスクが存在するのは、多くのプロジェクトでは、業務フロー図にすべての業務シナリオは記述されないためである。記述されるのは、基本シナリオと一部の代替/例外シナリオに留まる。その主な原因を以下に挙げる。

- 業務フロー図が複雑化するのを避け、理解が容易な状態に保ち、合意形成を円滑に進めるため。例えば、多数の業務から分岐する例外シナリオをすべて記述すると、複雑な分岐が生まれ理解が困難となる。この場合、業務フロー図を後工程で詳細化することが前提となる。
- 要件定義で業務フロー図を作成した後、後工程で代替シナリオや例外シナリオへの分岐が判明することがある。

もう一つの課題は、記述されている業務フローに対してどこまで網羅してテストするか、客観的な基準が設けにくい点である。業務フローをプログラムコードの制御フローと見なしてコード網羅基準を利用できるが、業務フローをテストベースとしたテストで主な関心となるパスカバレッジを網羅しようとすると組合せが爆発する。そのため、テストする組合せを選ぶ必要があるが、客観的な選択条件がないため、設計担当者やテスト担当者の主観に基づき選択することになる[4][6]。このようなテストは、網羅した範囲を明確に識別できず、コスト制約の範囲でテストしているに過ぎないと、Shimon は指摘している[5]。

2.2 状態遷移図の特徴

状態遷移図には、テスト対象の状態の遷移に沿って、遷移を引き起こすイベントや遷移条件が記述される。複数の記法が存在し、また表形式で記述されることも多い。本論文では UML のステートマシン図の記法を用いる。ただし、本手法は特定の記法に依存するものではなく、利用するのは表 2 に示す最小限のモデル要素である。また、記述の凡例を図 2 に示す。

表 2 利用するステートマシン図のモデル要素

モデル要素名	モデル要素	説明
状態	状態	テスト対象の状態。入れ子にできる。
遷移	↓	遷移の流れ。
イベント	イベント	状態遷移を引き起こす業務などのトリガ。
条件	イベント[条件]	イベント発生時に遷移するための条件。
アクション	イベント[条件]/アクション	遷移するときに、同時に実行される動作。
開始	●	状態遷移の開始点。
終了	◉	状態遷移の終了点。

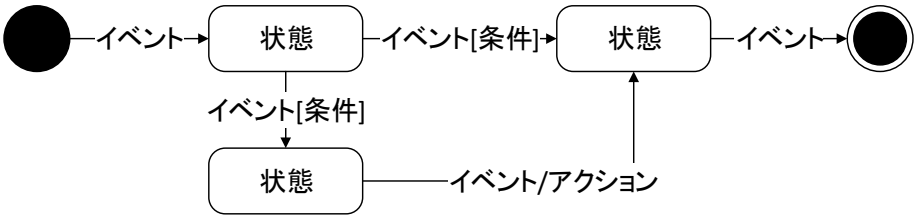

図 2 ステートマシン図の記述凡例

状態遷移図をテストベースとして利用することで、業務フロー図をテストベースとして利用するときの問題のうち、図の複雑化と主観的な網羅基準についての問題を解消できる。

業務フロー図が複雑化する主要因となる代替/例外シナリオへの分岐を、状態遷移図では平易に表現できる。例として、下記仕様の状態遷移図を図 3 に示す。この仕様を業務フロー図に記述すると、登録、訂正、却下後の各タイミングで削除へと分岐し、状態遷移図に比べ複雑となる。

- 申請者は申請を新規登録できる。
- 申請者は、承認依頼前なら申請を訂正または削除できる。
- 承認者は、承認依頼された申請を承認または却下できる。

- 申請者も承認者も、承認された申請を訂正あるいは削除できない。
- 申請者は、却下された申請を、訂正して再び承認依頼できる。
- 申請者は、却下された申請を削除できる。

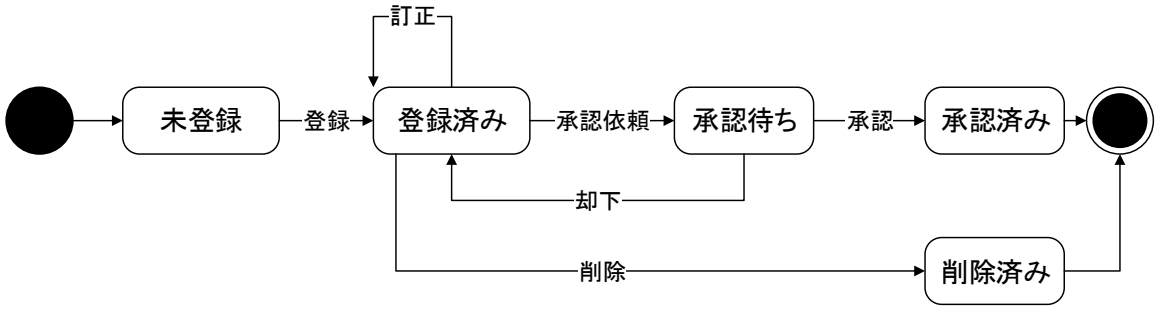

図 3 申請承認業務の状態遷移図

状態遷移図をテストベースに用いると、客観的な網羅基準として Chow の網羅基準[7]を利用できる。Chow の網羅基準では、N 状態間の遷移を網羅することを「N-2 スイッチ網羅」と呼ぶ。例えば、1 スイッチ網羅でテストすれば、3 状態間の遷移が網羅される。

しかしながら、状態遷移図では業務フロー図を代替できない。下記の原因により、状態遷移図によって、システム化する業務の位置づけについて合意形成するのは困難なためである。

- 業務の流れと完全には対応しないため、発注者が理解するのは容易でない。
- 業務システムにおいて、状態遷移図が必要となるほど複雑な状態遷移をする仕様は限られている[8]。業務全体を状態遷移図で記述するのは非効率である。

また、設計書として作成する際は、詳細化が進んで判明する業務シナリオが存在するため、要件定義工程ですべての遷移を記述することが難しいという問題は解消されない。

3. 提案手法

3.1 業務シナリオの補完アプローチ

業務フロー図は、代替シナリオや例外シナリオが記述されておらず、テストベースとして不十分なことが多い。一方で、状態遷移図は、代替/例外シナリオに対応する状態遷移を簡潔に記述できるため、テストベースとすることで、業務フロー図からは設計できないテストを補完できる。また、状態遷移図をテストベースに使うと、業務フロー図にはない客観的な網羅基準を使用できる。

一方で、状態遷移図は業務フロー図の代替とならず、上流工程で作成されるのは業務フロー図のみであることが多い。

そこで、本論文では業務フロー図とおよび各業務を詳細化した設計書から、代替/例外シナリオを補完するために状態遷移を抽出する手法提案する。

3.2 業務シナリオの補完に必要な設計情報とテストベース

業務フロー図に記載されていない業務シナリオを抽出するため、状態遷移図相当の設計情報を利用する。業務フロー図は、特殊な状態遷移図と見なすことができる[9]ため、状態遷移図から業務シナリオを抽出できる。抽出にあたっては、表 3 に示すステートマシン図とアクティビティ図のモデル要素の対応関係を利用する。

業務システムの受託開発プロジェクトにおいては、状態遷移図が作成されず、テストベースとして利用できないことも多い。しかし、そのようなシステムにおいても状態遷移は発生しており、状態遷移図相当の設計情報は、業務フロー図から詳細化された複数の設計書に分散して記載されている。状態遷移図相当の設計情報として、ステートマシン図のモデル要素に対応する設計書の名前とその記載内容を表 4 に示す。プロジェクトにより設計書名は異なるが、記述内容は受託開発に必須となる設計情報のため、こられの設計書をテストベースとして利用できることを前提と

する

表 3 ステートマシン図とアクティビティ図との対応

ｽﾃｰﾄﾏｼﾝ図のﾓﾃﾞﾙ要素名	ｱｸﾃｨﾋﾞﾃｨ図のﾓﾃﾞﾙ要素名	備考
状態	アクション	状態がアクションの実施結果に対応する。
遷移	フロー	―
イベント	アクション	イベントが状態遷移を引き起こすアクションの実施に対応する。
条件	分岐、マージ	―
アクション	アクション	アクティビティ図のアクションのうち、状態遷移を引き起こさないものが、ステートマシン図のアクションに対応する。
条件	レーン	状態遷移図の条件に役割区分を記載できる。
開始	開始	―
終了	終了	―

表 4 ステートマシン図相当の設計情報が記載される設計書名と想定記述内容

ｽﾃｰﾄﾏｼﾝ図のﾓﾃﾞﾙ要素名	設計書名	記述内容
状態	オンライン処理設計書	データベース更新内容。
	バッチ処理設計書	データベース更新内容。
遷移	オンライン処理設計書	データベース更新内容。
	バッチ処理設計書	データベース更新内容。
イベント	画面定義書	画面上で発生するイベントと実行される処理。
	画面遷移図	画面遷移が発生するイベントと実行される処理。
	バッチ処理設計書	バッチ処理を起動するイベント。
条件	オンライン処理設計書	処理対象の検索条件、入力チェック条件。
	バッチ処理設計書	処理対象の検索条件、入力チェック条件。
	画面定義書	ボタンの有効無効などの画面状態を決める条件。
	画面遷移図	遷移条件。
アクション	オンライン処理設計書	処理内容（データベース更新を除く）。
	バッチ処理設計書	処理内容（データベース更新を除く）。

3.3 業務フロー図および各種設計書からの状態遷移の抽出

業務フロー図と表 4 に示した設計書から、状態遷移を抽出し、業務シナリオを補完する手順を以下に示す。

(1) 業務フロー図から業務シナリオの補完範囲を決定する。
(2) 基本シナリオの流れに沿って、各業務を詳細化したオンライン処理またはバッチ処理を特定する。その後、それらの業務で更新しているデータベースから、状態を管理しているテスト対象となる実体を特定し、基本シナリオに沿った状態遷移を抽出する。
(3) 各業務を詳細化したオンライン処理設計書、バッチ処理設計書、画面定義書、画面遷移図から、業務シナリオに沿って業務を進めるための条件と、進められない時の条件を抽出する。
(4) 業務シナリオの流れをさかのぼり、業務を進められないときの条件を設定している業務を特定し、設定後の状態とその状態に遷移するための条件を抽出する。該当する状態が存在しない場合は、状態を追加する。複数の状態から当該状態に遷移できるときは、必要に応じて状態を入れ子にする。

(5) 代替/例外シナリオに対しても、基本シナリオと合流するか業務を終了するまで、手順(3)～(5)を繰り返す。
(6) 手順(1)で決定した範囲で、新たな代替/例外シナリオへの派生がなくなったら、手順を終了する。

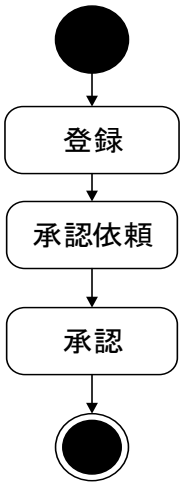

図 4 案件情報管理の業務フロー図

4. ケーススタディ

4.1 業務シナリオの補完例

金融商品を扱う大規模システムの開発プロジェクト 1 件において、業務フロー図の記述が粗くかつ状態遷移図も作成しなかったため、提案手法を試行適用した。試行適用したテストの期間は、テスト設計および実施を含めて約 9 ヶ月である。適用例を以下に示す。なお、説明上の理由と機密情報保持により、業務フローを単純化し図 4 に示す案件情報管理の業務フローを対象とし、手順(4)までの例示とする。

(1) 案件情報の登録から承認までを、補完範囲とする。
(2) 業務フローの基本シナリオに沿って、案件情報の状態遷移図を抽出する。抽出結果を図 5 に示す。
(3) 抽出した状態遷移に沿って、オンライン処理設計書、画面定義書、画面遷移図から、次の状態へと遷移できる条件と遷移できない条件を抽出する。本例では、登録済みへの遷移条件はなく、承認待ちへの遷移条件として下記を抽出できた。承認できない条件は、下記条件の否定なので省略する。
   - 案件種別がその他でないこと。
   - 案件種別がその他のときは、案件詳細情報が登録されていること。
(4) 業務シナリオをさかのぼり、承認できない条件を設定している業務を特定する。この例では、登録業務のオンライン処理設計書のデータベース更新内容で案件種別を設定していることと、画面遷移図から案件種別がその他のときは詳細情報登録画面に遷移できることが分かった。これらの設計書から、状態への遷移とその遷移条件を抽出する。抽出結果を図 6 に示す。

図 5 基本シナリオに対応する状態遷移図

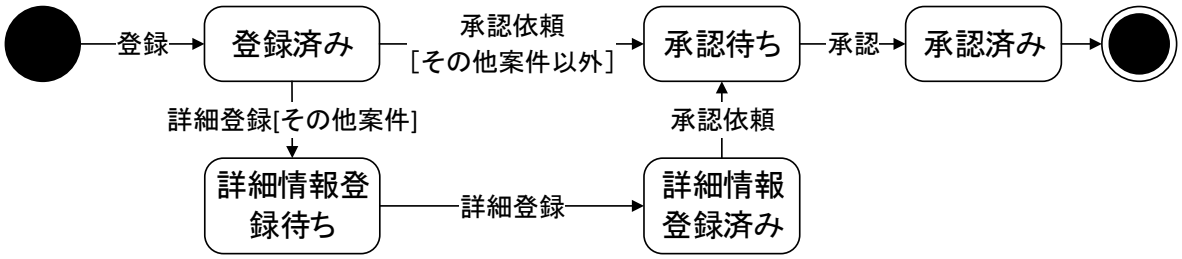

図 6 代替シナリオを補完した状態遷移図

4.2 考察

本手法の適用により、業務フロー図に記述されていなかった代替/例外シナリオを補完することができた。「4.1 業務シナリオの補完例」で挙げた例を含め、適用事例全体を通して補完できたシナリオのパターンを以下に示す。

- データの種別により派生する業務シナリオ
  「4.1 業務シナリオの補完例」のように、業務フロー図上では1本のフローで表現されていたが、データの種別によって追加業務が必要なシナリオを補完できた。
- 取消に派生する業務シナリオ
  本事例では、取消に掛かる業務フローが取消通知受領から始まっており、取消に派生できる事前条件が曖昧だった。本手法の適用により、業務フロー図上の取消条件を満たす範囲を明確にし、テストが必要な業務シナリオを具体化できた。
- 入れ子の状態に対応する業務シナリオ
  入れ子の状態に対応する業務シナリオの記載が省略されていた。分岐が複雑となるため、意図的に単純化されたものと思われる。こうしたシナリオも補完できた。

また、本手法適用時の副次的な効果として、設計書の品質改善とテスト設計担当者の業務知識向上が見られた。設計書の品質改善については、業務シナリオに沿って各種条件を確認することで、設計の不統一を発見できた。例えば、1, 2, 3いずれかの値をとる項目に対して、条件評価式に「1または2である」と「3でない」が混在していた。担当者の業務知識向上については、本手順に沿って複数の設計書を併せて参照する過程で、業務知識の拡大につながった。

課題として、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点が挙がった。この傾向は、少数の業務で完結する業務シナリオにおいて顕著であった。本事例ではすべての業務フロー図を補完対象としたが、一般に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]。少数の業務で完結する業務シナリオに対しては、提案手法は過剰であり、状態遷移を抽出せずに代替/例外シナリオを補完するような軽量な手順で十分と思われる。

なお、本適用事例では、Chowの網羅基準を適用できなかった。発注者との調整の結果、状態遷移図そのものではなく業務シナリオ一覧が発注者のレビュー対象となり、その業務シナリオを全網羅することを基準としたためである。

5. 結論

本論文では、業務システムの受託開発プロジェクトのテスト工程において、業務フロー図に記述されていない代替/例外シナリオを補完する手法を提案した。提案手法では、状態遷移図を利用できないときに、業務フロー図と各業務を詳細化した設計書（オンライン処理設計書、バッチ処理設計書、画面定義書、画面遷移図）から、状態遷移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ことを示せた。

適用事例を通して、本手法で代替/例外シナリオを補完できることを確認できた。加えて、設計改善、担当者の業務知識向上の効果も得られた。一方で、提案手法の適用が過剰となる業務フロー図も確認できた。

今後は、適用案件を拡大して適用効果の再現性を検証しつつ、各案件において本手法の適用が効率的な網羅性向上につながる業務フロー図とそうでない業務フロー図の差異を調査し、本手法の汎用性を改善するとともに適用条件を明確化したい。

6. 参考文献

[1] 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
[2] Object Management Group、“Unified Modeling Language、” http://www.omg.org/spec/UML/
[3] 情報処理推進機構 ソフトウェア・エンジニアリング・センター 要求・アーキテクチャ領域機能要件の合意形成技法WG、“機能要件の合意形成ガイド、” 2010
[4] Jim Heumann、“Using Use Cases to Create Test Cases、” The Rational Edge、6月、2001
[5] Simons, Anthony JH.、“On the Compositional Properties of UML Statechart Diagrams、” Rigorous Object-Oriented Methods、2000

[6] ボーリス・バイザー著、小野間章／山浦恒央訳、“ソフトウェアテスト技法、”日経BP出版センター、1994
[7] CHOW, Tsun S.、“Testing software design modeled by finite-state machines、”IEEE transactions on software engineering、178-187、1978
[8] Scott W. Ambler、“The Object Primer: Agile Model-Driven Development with UML 2.0 3rd Edition、” Cambridge University Press、2004
[9] Michael Havey、“Essential Business Process Modeling、” O'Reilly Media, 2009