6K-05

要求分析モデルからの状態遷移抽出による振る舞いフローの妥当性確認支援

森田　光†　　松浦　佐江子‡

芝浦工業大学大学院　理工学研究科　システム理工学専攻†‡

1. はじめに

近年のシステムは IoT に見られるように，複数のシステムやハードウェアの連携でサービスを提供することが多い．このようなシステムの要求分析では，システムの連携状況やハードウェア構成を考慮して，初期要求の手順を明らかにすることが望ましい．我々は，このサービス手順を各サブシステムの境界であるユーザやサブシステム間の情報のやり取りに着目したユースケース分析に基づき，これらのユースケースの連携がサービスのゴールを満たす手順をワークフローとしてモデル化する[1]．ワークフローによって，初期要求およびゴールを満たすことを確認した後，各サブシステムのユースケースを定義し，要求仕様としての要求分析モデルを確定する．本稿では，UML を用いてワークフロー，データモデル，ユースケースを定義し，これを要求分析モデルと呼ぶ．初期段階の要求分析モデルの品質が全体の開発に影響を及ぼすことから，要求分析モデルは初期要求を全て満たす仕様書として妥当である必要がある．

しかし，サービス手順としての観点では，定義したワークフローがサブシステムの取りうる状態に応じて適切な振る舞いをしているかの確認はできていない．そこで，テストケースとして各サブシステムの状態遷移図を定義し，要求分析モデルから自動抽出した状態遷移モデルと比較するツールを開発する．これにより要求分析モデルにおけるユースケースの連携とユースケースの振る舞い手順に対し，特定のオブジェクトが要求されていることを満たすための振る舞いとシステムが識別すべき状態の観点から手順が十分妥当であるかを確認する支援を目的とする．

2. 関連研究

形式的検証技術の 1 つであるモデル検査技術を用いて要求仕様の妥当性を確認する研究がある[2][3]．これらは，要求分析で作成したモデルをモデル検査ツールに適する有限状態モデルに変換することで網羅的検証を行い，到達可能性や安全性等を保証する．しかし，これらの検証方法では記述されている要求の過不足までは検証できない為，要求分析自体の妥当性を確認できていない．本研究では，開発するツールを用いることで記述内容の過不足を検証できる．

3. アクティビティ図によるワークフロー記述

本研究では，ワークフローをアクティビティ図を用いて作成する．図 1 に示すように，初期要求に現れる各サブシステムのユースケースの境界をパーティションによって明確にし，ユーザやサブシステム間の情報のやり取りをシグナル送受信やタイマーを使って明らかにする．また，システムのデータ構造はクラス図として定義し，アクティビティ図ではオブジェクトノードとして記述できる．アクションノードから続けて対象データのオブジェクトノードを記述することにより，データに対するアクションであることを明確にする．

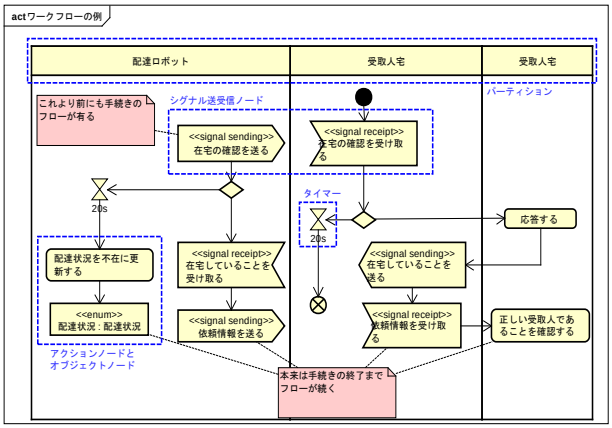

図 1　ワークフローの例

4. 提案手法

本研究の提案手法を図 2 に示す．

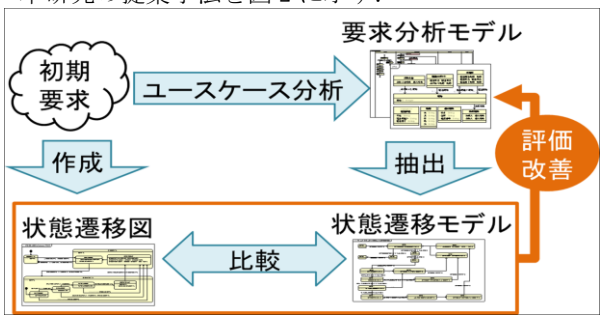

図 2　提案手法

システムの初期要求に対してユースケース分析を行い，要求分析モデルを作成する．その後，要求分析を行った開発者の観点を引きずらない為に別の開発者が同じ初期要求から状態遷移図を作成する．この状態遷移図を要求分析モデルから抽出した状態遷移モデルと比較することで要求分析モデルの評価・改善を行う．

4. 1. 状態遷移モデルの抽出

状態を区別する観点として，初めに，ワークフローの記述構造に着目する．デシジョンマージノードは遷移の制御ノードであり分岐条件によって遷移が異なるため，その前後で状態が分かれていると識別できる．

次の観点として，ワークフローのアクションに着目する．アクションは明示的に状態を表さないが，アク

Validation Support of Behavioral Flows on Requirements Analysis Model by Extracting the State Transitions
†Hikaru MORITA ‡Saeko MATSUURA
†‡Systems Engineering and Science, Graduate School of Engineering and Science, Shibaura Institute of Technology

ションによって状態が何らかの変化をすると考えられる．このようなアクションは他のシステムからデータを受け取るアクション，時間経過を示すアクション，システムのもつ属性を変化させるアクション，サブシステム独自のアクションから構成される．他のシステムからデータを受け取るアクションはシグナル受信ノードとして記述されており，時間経過を示すアクションはタイマーとして記述されている．これらの前後で状態を区別し，区別した状態間の遷移のトリガーとして抽出する．システムのもつ属性を変化させるアクションも状態を前後で区別し，区別した先の状態の入場動作として抽出する．サブシステム独自のアクションは，次の状態への遷移のアクションとして抽出する．シグナル送信アクションは，他のシステムとの連携を示すアクションであるが，自身から行うアクションであるため区別された状態の入場動作として抽出する．

また，ワークフローに記述されている事前条件・事後条件は，クラスに定義したデータによる条件であり，システムの状態を明確に表している．これらはノートとしてイニシャルノードとファイナルノードに紐づいているため，イニシャルノードとファイナルノードを初期状態と終了状態として抽出する．

上記の抽出方法を用いて，状態遷移モデルの抽出を自動的に行う．また，抽出した状態遷移モデルには無条件遷移の前後に動作がない状態が存在する可能性がある為，このような状態・状態遷移は縮退する．抽出した状態遷移モデルは比較までツールで行うため，図としてではなくデータとして保持する．

4.2. 比較による状態遷移モデルの妥当性確認

抽出した状態遷移モデルとテストケースとして作成した状態遷移図の比較をツールによって行う．作成した状態遷移図が開発者の言葉で記述されると記述内容が同等であるかの判定が困難になるため，ワークフローのアクションを基に使用する語句の記述形式をそろえて作成する．下記に比較方法を示す．

1. 作成した状態遷移図が階層で分かれている場合，遷移や状態の動作を基に 1 段目の状態数と同じ数に分けられるか判定する．
2. 状態数が同じならば状態間の遷移数と遷移方向が同じであるか判定する．
3. 抽出した状態遷移モデルと作成した状態遷移図に記述されている語句が全て対応付くか判定する．
4. 2 段目以降の階層でも 1～3 の工程を階層がなくなるまで繰り返す．

上記の比較方法を用いてツールによる比較を行い，同等であると判定されなかった状態・状態遷移や対応付かなかった語句をワークフローと作成した状態遷移図の該当箇所に表示する．表示された箇所は，要求分析モデルがシステムの初期仕様を満たさない可能性があるため開発者間で議論を行うことで適切であるかないかの判断を行う．例えば，表示された箇所がガードやトリガー等の記述要素ならばその条件や動作が考慮されていないのではないか，状態や状態遷移ならば本来考慮すべき手順が分析できていないのではないかといった議論ができる．

5. 適用事例

事例として芝浦工業大学の組み込みシステムの開発を目的とした PBL 課題である荷物自動搬送システムを用いる．この課題で作成した要求分析モデルに対してツールを適用した．比較結果の一例を図 3 に示す．

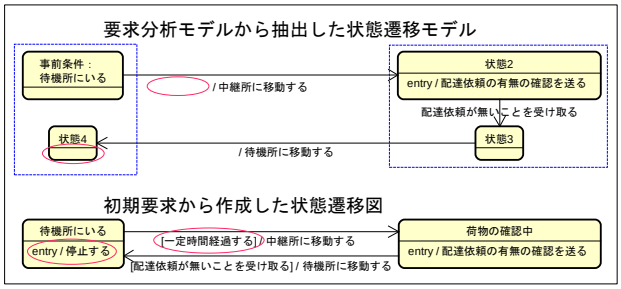

図 3　比較結果の一例

比較の結果，作成した状態遷移図では待機所からの遷移に“一定時間経過する”というハードウェアの動作条件や“待機所にいる”という状態の entry として“停止する”という動作を考慮しているが，抽出した状態遷移モデルでは考慮されていないことが分かる．状態遷移図を作成する開発者は，システムが要求を満たす振る舞いをするうえでこれらの条件や動作が必要であると判断している．その為，これらの条件や動作が必要かどうか再度検討し要求分析モデルの修正を行う．事例では，必要であると判断されたため要求分析モデルの該当箇所を修正した．また他の判明点として，“事前条件の有無”や“状態や遷移の差異による処理手順の欠如”等が発見できた．

6. まとめ・今後の方針

本稿では要求分析段階で得られたワークフローから状態遷移モデルを抽出し，テストケースとして作成した状態遷移図と比較することによる要求分析の妥当性確認の支援方法を提案した．これにより，システムの処理手順だけでなく状態遷移という観点から評価できるようなり，要求分析モデルで考慮されていなかった仕様の発見が容易になると考える．今後は開発したツールの適用実験を様々なワークフローで行い，現在定義している抽出規則が適切であるかの確認を課題とする．また，本ツールは astah* Professional[4]内のプラグインとして実装する予定のため，ツールの未完成部分の完成を目指す．

7. 参考文献

[1] 松浦佐江子，ソフトウェア設計論 ―役に立つ UML モデリングへ向けて―，コロナ社，2016.

[2] Omar Tariq, Jun Sang, Kanza Gulzar, Hong Xiang, “Automated analysis of UML activity diagram using CPNs”, 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)

[3] Vahid Rafe, Reza Rafeh, Somayeh Azizi, Mohamad Reza Zand Miralvand, “Verification and Validation of Activity Diagrams Using Graph Transformation”, 2009 International Conference on Computer Technology and Development

[4] “astah* professional”, http://astah.change-vision.com/ja/product/astah-professional.html. [アクセス日: 2010/1/10].