5N-07

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　システム理工学部　電子情報システム学科†‡

1. はじめに

要求分析では，システムを使用するユーザやサブシステムの相互関係に着目し，システムの機能要求を含む振る舞いをユースケースとして分析することで要求分析モデルを作成する．本研究では，要求分析モデルを UML（Unified Modeling Language）のユースケース図，アクティビティ図，クラス図を用いて定義する．設計・実装では，要求分析モデルに記述された内容に沿って開発を行うことでシステムの処理手順を実装する．しかし，このようなシステム開発では，システムの構成要素が要求仕様に沿った状態遷移をしているか確認することは困難である．そこで，システムの振る舞いをシステムの構成要素の状態遷移という別の視点から評価するために，要求分析モデルから読み取れる状態遷移を確認できることが好ましい．本稿では，クラス図の要素を基に状態を定義し，アクティビティ図から状態に関する記述を抽出することで状態遷移を抽出する手法を提案する．

2. 要求分析段階におけるモデルの構成要素と関連性

2.1. ユースケース図

ユースケース図は，システムの持つ機能（ユースケース）とその機能を使用する外部環境（アクター，サブシステム等）を整理して表現した図である．[1][2]要求分析では，各サブシステムのユースケースを分析することにより，そのサブシステムの振る舞いを確認できる．本研究では，アクティビティ図で定義されたシステム全体の処理手順であるワークフローとユースケースの振る舞いであるユースケース記述を関連付ける役割を持つ．（図 1）

2.2. クラス図

クラス図はシステムの構成要素である「クラス」とそのクラスの構造的関係を「関連」として表す．クラスの持つデータ構造は「属性」と関連先の要素を示す「関連端」として記述する．属性と関連端を総称して「プロパティ」と呼ぶ．[1][2]本研究ではこのクラス図に記載されているクラスを基に状態を作成し，そのクラスがアクティビティ図でどのように操作されているか特定することにより，状態と状態遷移を抽出する．（図 2, 3, 4）

2.3. アクティビティ図

アクティビティ図は，ワークフローやユースケース記述をシステム利用者やサブシステム毎に「パーティション」で区切り，主体と役割を明確にしたアクション系列で表す．アクションに関係するシステムのオブジェクトはオブジェクトノードに記載され，クラス図で定義したクラスとその属性として整理できる．[1][2]このため，アクションとデータの関係性によりオブジェクトに対する操作が明確となるため，そのオブジェクトの状態や状態遷移を取得する．

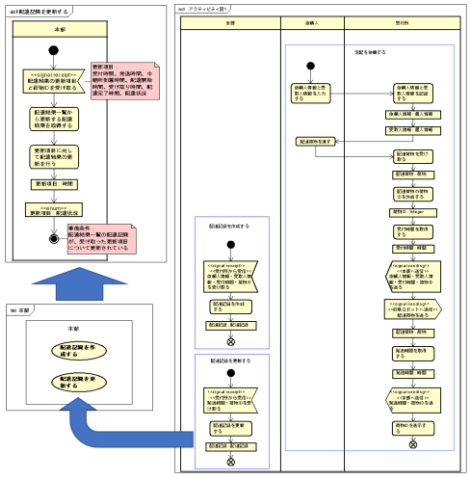

図 1　ワークフロー・ユースケース図・ユースケース記述の関連性

3. 状態遷移図によるモデルの評価

要求分析モデルを評価する際，システムの処理手順であるワークフローやユースケース記述を評価する．これらのアクティビティ図の処理手順が要求仕様に沿って作られているならば，妥当なモデルであると判断し，設計・実装工程に移動する．しかし，システムの処理手順に対する評価という 1 つの観点だけでは，要求分析モデルに要求仕様に対する漏れがないか確認することは困難である．さらに，システムのオブジェクトはワークフローやユースケース記述内の様々な箇所に存在しており，システム規模が拡大するほどこれらのオブジェクトの状態遷移が見えづらくなるため，システムを構成するオブジェクトが要求分析モデルの処理手順に応じた状態遷移をしているか読み取ることは困難である．ここで要求分析モデルに沿った状態・状態遷移を抽出し状態遷移図を用いて表示することにより“想定している状態の欠如”や“状態遷移の誤り”等の矛盾点を発見することが容易になる．システムの処理手順に対する評価とオブジェクトの状態遷移に対する評価両方を満たすモデルを妥当なモデルと判断することで後の工程での支障の減少につながる．そこで，本研究では，要求分析モデルに沿った状態遷移図を作成するツールを開発し、それを用いることで，モデルの妥当性を確認できることを目的とする．

4. 状態遷移抽出の手法

4.1. 概要

本研究では，要求分析によって作成されたクラス図の中からある 1 つのクラスオブジェクトに着目する．着目する要素は着目したクラスとそのクラスが保持するプロパティである．アクティビティ図の処理手順をすべて網羅するために，分岐点である条件分岐ノードとオブジェクトを送信するシグナル送信アクションが記載されてい

Validation Support of Behavioral Flows on Requirements Analysis Model by Extracting the State Transitions
†Hikaru MORITA ‡Saeko MATSUURA
†‡Department of Electronic Information System, Collage of System Engineering and Science, Shibaura Institute of Technology

る箇所で経路を分解し，すべての処理手順を取得する．取得した処理手順を 1 つずつ探索することにより，アクションとオブジェクトノードに着目する要素に関する記述が存在するかしないかを判断する．アクションと続けて書かれているオブジェクトノードに着目する要素が存在していれば状態遷移箇所であると特定する．（図 2）

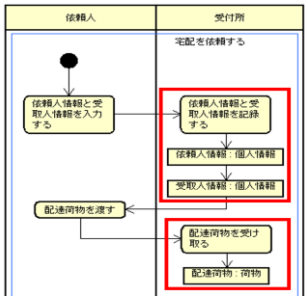

図 2　アクティビティ図内の状態遷移箇所

4.2. 状態の作成

本研究では，状態はクラスの保持するプロパティによって定義するためクラス名とそのプロパティを文字列として連結し状態名に記述することで状態を作成する．（図 3,4）プロパティの状態は，関連端の多重度，クラスの型等のクラス図より読み取れる情報を利用する．図 3,4 を例とすると配達荷物は多重度が 0,1 なので有・無で表せる．enum 型のクラスの状態は属性として記述されているのでそのクラスの属性を引用する．

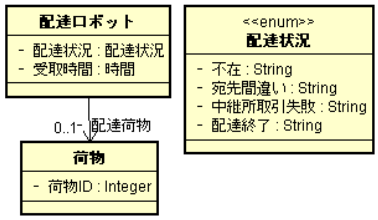

図 3　クラスの関係図

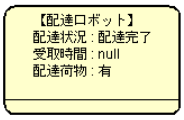

図 4　作成される状態の形式

4.3. 状態遷移の作成

状態遷移図における遷移は”遷移元”“遷移先”“イベント”“アクション”“ガード”を要素に持つ．遷移元は前の状態を記憶しておくことで特定できる．遷移先は状態遷移箇所を抽出した際に特定できる．イベントは，状態遷移を抽出した際のアクションとし，状態遷移のアクションは着目するオブジェクトが初めて更新された時のアクションを取得する．ガードは条件分岐に記述されているガード条件を取得する．記載がなければ条件分岐先のアクションをガードとして取得する．

4.4. ユースケースの切り替わり

ユースケース分析を行っているのでシステムの機能ごとに振る舞いを表すユースケース記述が作成される．そのため，アクティビティ図の処理手順を探索中にユースケースが切り替わることがあれば切り替わった先のユースケース記述の処理手順を探索することでシステムの振る舞いを漏れなく探索する．

4.5. 着目するオブジェクトの役割

本研究では，着目するオブジェクトに対して 2 つの分類に分け状態遷移を抽出する．

①着目するオブジェクトがシステム内のサブシステムとして存在している場合
②着目するオブジェクトがサブシステム間を移動するオブジェクトとして存在している場合
上記の場合は，サブシステムはアクティビティ図内においてパーティションとして記述されているため，着目するオブジェクトに該当するパーティションを取得する．そのパーティションに記述されている処理手順からユースケースと発火する順序を特定し，発火した順序に沿ってユースケース記述を探索することで状態遷移の特定を行う．
下記の場合は，ワークフロー全体の処理手順に対して状態遷移の特定を行う．

5. 状態遷移抽出ツール

本研究で開発するツールは UML モデリングツールの 1 つである astah*[3]内のプラグインとして開発する．ツールには以下の機能がある．

- 抽出可能なオブジェクト一覧の表示
- 選択されたオブジェクトの状態遷移抽出
- 抽出した状態遷移を状態遷移図として描写

基本的にツールの使用者は，表示されたオブジェクト一覧から状態遷移を抽出したいオブジェクトを選択するだけでよい．

6. 適用実験と考察

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，様々なオブジェクトに対して状態遷移の抽出を行った．その結果，サブシステムとして機能しているオブジェクトに関しては，要求分析モデルに沿った状態遷移を取得できていることが確認できた．状態遷移図を確認することで更新されていない属性に気付き，要求分析モデルの不備を修正することができた．一方，サブシステム間を移動するオブジェクトとして存在しているオブジェクトに関しては，状態の種類は正しく取得できていたが状態遷移が正しいものと正しくないものに分かれた．これは，システムの全体の振る舞いを取得するためワークフローの処理手順を全パターン取得しているが，複数のシグナル送信アクションによって分岐した処理手順間では時系列の整合性が取れていないことが原因であった．組み込みシステムでは，サブシステムの処理手順は独立して機能するため，ある処理手順が別の処理手順に対してオブジェクトの操作をしている等の影響を考慮しなければならない．現在は事前条件を用いて処理手順間の時系列を特定する方法を検討中である．

7. 今後の方針

今後の方針としては，処理手順間の時系列の特定方法を確立し，影響を及ぼす処理手順においても要求分析モデルに沿った状態遷移図を出力できるようにすることである．次に，本ツールは，まだ astah*のプラグインとして実装できていないのでその実装を行う．

参考文献

[1]松浦，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，2017
[2]株式会社オージス総研，オブジェクトの広場編集部，その場でつかえるしっかり学べる UML2.0，2006
[3]astah*，http://astah.change-vision.com/ja/（2019/01/11 参照）