

# 要求分析と基本設計間のトレーサビリティ確保 のためのユースケース記述変換ツール

吉野 魁人<sup>†</sup> 松浦 佐江子<sup>‡</sup>

芝浦工業大学 理工学研究科 システム理工学専攻<sup>†‡</sup>

## 1. はじめに

ソフトウェア開発ではシステムの保守性や品質を高めるために、トレーサビリティ（追跡可能性）と呼ばれる概念が重要である。情報システム開発においては各開発工程における成果物間の関連と定義されている[1]。人々や社会の要求を満たすシステムを実現するためには、要求分析・基本設計・詳細設計・実装・テストといった開発工程における成果物に対して、トレーサビリティを維持管理することが必要となる。しかし、それぞれの成果物は異なる言語（自然言語、プログラミング言語）や表現方法で表されている場合が多く、人手による解釈では成果物の関連付けに漏れや誤りが発生する可能性がある。本研究では、Unified Modeling Language (UML) を用いて要求分析で定義したモデルの要素を基本設計モデルの要素へ自動変換し、それらの対応付けを保存することによりトレーサビリティを確保する手法を提案する。

## 2. トレーサビリティ

トレーサビリティの確保は品質管理の基本的な手段の一つである。Gotel ら[2]は、要求のライフサイクルを記述して前工程と後工程をともに探索できる能力のことを要求トレーサビリティと定義している。トレーサビリティを確保することにより、要求定義から設計、プログラムまで一貫した管理ができるようになり、上流の成果物から下流の成果物への対応を漏れなく確認することができる。また、開発中のシステムに変更が生じた場合に影響を受ける範囲が明確となる。しかし、実際のシステム開発の現場ではシステムが複雑化・大規模化するにつれてトレーサビリティの管理が煩雑になり、作業負荷が高くなる。

効率的に要求定義と設計間のトレーサビリティを管理する方法としてこれまでにさまざまな支援のツールが開発されている[3][4]。これらは成果物である要求定義と設計書を項目ごとに分解し、それぞれの対応付けをトレーサビリティマトリックスやツリーによって表示を行う。しかし、この対応付け自

体はツールの利用者が行うため、成果物の文書の解釈次第によっては漏れや誤りが発生する可能性がある。本研究では、文書形式ではなくUMLのモデル図を用いて要求分析、設計を行うことにより、モデルの要素間の対応としてトレーサビリティを確保することによって漏れ・誤りを無くす手法を提案する。対象となるモデル要素間のトレーサビリティを図1に示す。

## 3. 提案手法

### 3.1. ユースケース記述の記法

要求分析段階におけるユースケース分析では、一般的にテンプレート形式の自然言語によってユースケースを記述する。ここで設計間とのトレーサビリティを確保するために困難な点は、ユースケース記述での各フローや条件が自然言語で書かれるため、そこから正しい意図をくみ取り設計を行わなければならないことである。

そこで我々はユースケースのフローをより可視化・形式化してデータに不整合が起こらないか検証するために、システムのユースケースをアクティビティ図とクラス図を用いて分析を行なっている[5]。ユースケース内の振舞いをアクティビティ図のパーティションを用いてシステム内部とアクターとのやり取りに分離することによって、ユースケースにおける必要な入出力を見極め、その実現可能性をシステムの振舞いフローとして定義できる。アクターが外部システムである場合には、アクションノードの一種であるシグナル送受信ノード（図1アクティビティ図内上部の緑色の四角形で囲んだノード）を用いて、境界における情報の入出力の振舞いをシステム内部の振舞いと分離している。

また、クラス図を使ってシステムにおけるデータをクラスとその属性として分析すると、アクティビティ図内のアクションと緩い結びつきを持たせることができる。ここで緩い結びつきとは、アクション内の記述における名詞と動詞や、オブジェクトノードとアクションの位置関係といったことを指す。例えば、「\*\*を生成する」といったアクションの直後にその対象のデータを表したオブジェクトノードを記述することによって、オブジェクトの生成というアクションの明確な役割を表すことができる。本研究ではこのように書かれたアクティビティ図とク

UML Requirements Analysis Model Conversion Tool  
to Ensure Traceability of the Basic Design Model  
<sup>†</sup>Kaito YOSHINO <sup>‡</sup>Saeko MATSUURA  
<sup>†‡</sup>Systems Engineering and Science, Graduate School of  
Engineering and Science, Shibaura Institute of Technology

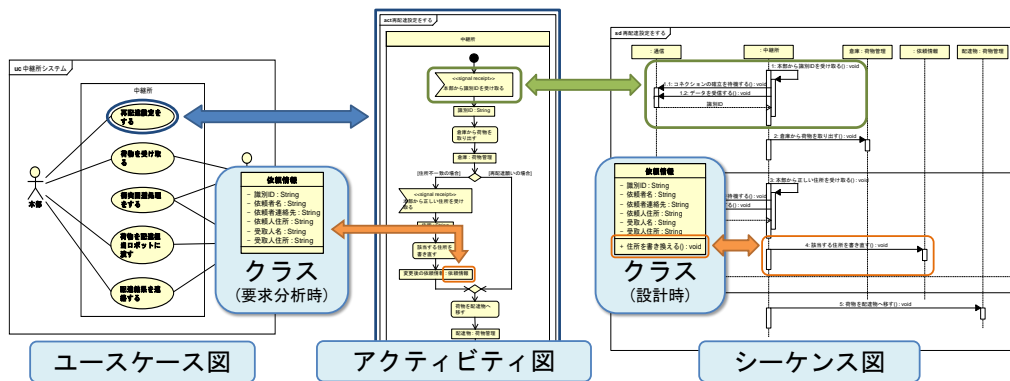


図 1 各モデルの要素間のトレーサビリティ

ラス図によるユースケース記述を対象とする。

### 3.2. ユースケース記述変換ツール

本ツールは UML モデリングツールの一つである astah\* professional[6]内のプラグインとして実装した。ツールの目的は 3.1 で述べた緩い結びつきをもとにし、ユースケース記述の要素を変換してシーケンス図を作成することによってシステムの外側から見た振舞いを分析する基本設計を支援することである。われわれはこれまでにツールを適用する上での各要素の変換の定義を行った[7]。

変換の流れとしてはまず、アクティビティ図のアクションとオブジェクトノードの位置関係を読み取り、対象のクラスの操作を決定するためのシーケンス図の要素であるメッセージを生成する。この位置関係はクラス図によって分析されたデータに対する基本的な処理である CRUD (Create, Read, Update, Delete) を表したものを対象とし、アクションの前後にどのようにオブジェクトノードがあるかによって定まるものとする。また、オブジェクトノードを省略した場合でもアクション内の目的語・動詞とクラス図の情報を照合することで対象のクラスを特定することを可能とする。これにより属性とそれに対する操作をまとめたクラスを設計することができる。

こうしてできたシーケンス図は図 1 のようにアクティビティ図のアクションに対応付いたものになる。プログラムを実装するためにはここからさらに操作名として適切な名前に変更し、シグネチャの制定をする必要があるが、これらの対応付けはツール内に履歴として残して表示するため、変更した後ももとの要求分析のどこにあたるかが分かるようになる。具体的には変換時に対応が付いたトレーサビリティマトリックスを作成するとともに、どのようにモデル要素の内容が遷移していったかの階層構造を作成して表示する。このようにしてトレーサビリティを確保しながら最終的な設計モデルの作成を支援する。

### 4. おわりに

本稿では要求分析段階で得られたユースケース記述の要素からシーケンス図を自動生成することに

よる基本設計の支援方法を提案した。ツールの利用者はその後シグネチャの制定や要求分析の段階では分析されていなかったメソッド内部の構造を追加する作業を追加する詳細設計のみを行えばよいことになる。本ツールの適用により各段階で作られたモデルの要素間のトレーサビリティを確保することが容易となる。今後はツールの実装をさらに進めていくことを課題とする。また、ツールの評価方法として本学の電子情報システム学科内の学生が作成した様々なモデルに対してツールを適用し、問題なく変換が行われるかと、どのようなモデルの場合に要素の対応が正しいものにならないかを考察する。

### 参考文献

- [1] 宇田川佳久, “情報システム開発標準におけるトレーサビリティの事例と今後,” *情報処理 Vol. 51 No. 2*, p. 151, 2010.
- [2] O. C. Z. Gotel and A. C. W. Finkelstein, “An analysis of the requirements traceability problem,” *Proceedings of IEEE International Conference on Requirements Engineering*, 1994.
- [3] “microTRACER,” [https://www.dts-insight.co.jp/product/eps\\_tool/microtracer/index.html](https://www.dts-insight.co.jp/product/eps_tool/microtracer/index.html). [アクセス日: 2019/1/7].
- [4] 山中美穂, 大音真由美, 山元和子, “トレーサビリティを確保してソフトウェア開発を効率化する要求管理ツール,” *東芝レビュー Vol. 73 No. 5*, 2018.
- [5] 松井, 奥田, 野呂, 岡田, 加藤, 渡辺, 松浦, “モデリング能力育成を目的としたユースケース記述の自動評価手法,” *電子情報通信学会論文誌, VOL. J97-D, NO.3*, 2014.
- [6] “astah\* professional,” <http://astah.change-vision.com/ja/product/astah-professional.html>. [アクセス日: 2019/1/7].
- [7] 吉野魁人, 松浦佐江子, “要求分析と設計間のトレーサビリティ確保のための UML 変換ツール,” *情報処理学会第 80 回全国大会*, 2018.