

# UML とソースコード間の異なる要素名のトレーサビリティリンク 作成手法の提案

吉田優介<sup>†1</sup> 橋浦弘明<sup>†1</sup> 田中昂文<sup>†2</sup> 櫛山淳雄<sup>†3</sup> 高瀬浩史<sup>†1</sup>

**概要:** ソフトウェアの開発や保守において、トレーサビリティリンクは重要である。そのため、トレーサビリティリンクを回復する手法は多く提案されている。しかしながら日本語で書かれた設計書とソースコード間のトレーサビリティリンクを回復する手法はあまり提案されていない。本研究では日本語で書かれたクラス図とソースコード間のトレーサビリティリンク回復の支援手法を提案する。

**キーワード:** トレーサビリティ, リンク回復, UML

## 1. はじめに

ソフトウェア開発や保守において、トレーサビリティは重要である。変更影響範囲解析などに利用されている。トレーサビリティが失われている場合はリンクを回復する必要がある。一般に日本語で書かれる設計書の要素名と、一般に英語で書かれるソースコードの要素名の表記は一致することがないため、これらの間のトレーサビリティの確保は困難である。このような問題に対してはコサイン類似度などで、候補を導き出すことができないため、自動化することが難しい。また、日本語で書かれた設計書をソースコードの要素名と一致させるために英語で書き直すことは現実的ではない。現状、日本語で書かれた設計書と英語で書かれたソースコードのトレーサビリティリンクを回復する手法はあまり提案されていない。

## 2. 研究目的

本研究は、日本語で書かれた設計書と英語で書かれたソースコード間のトレーサビリティリンク回復の支援を目指す。具体的にはソースコードのアノテーションを用いて、対応する設計書の要素名を書くことによってトレーサビリティリンクを回復する。さらに、設計書とソースコードの整合性をツールが自動的に確認し、その差分をフィードバックすることによって、ユーザーがリンク候補を見つけやすくする。

## 3. 関連研究

設計書とソースコードのトレーサビリティリンク回復を支援する手法は多く提案されている。しかし、日本語で書かれた設計書とソースコードのトレーサビリティリンクを回復する手法は少ない。伊藤ら[1]の研究では UML 記述

の設計モデルとオブジェクト指向言語記述のソースコードを対象としている。クラス名一致やコサイン類似度による対応付けなど、4 つのルールからなるアルゴリズムが自動で抽出したリンク候補から、ユーザーが人手で正しいリンクを選択する半自動的なプロセスをとっている。GQM 法を用いてトレーサビリティ測定手法の提案を行っている。類似点として、対象としている成果物、クラス名が一致しているものを自動で対応付けることなどが挙げられる。本研究との相違点としては、伊藤らの研究ではコサイン類似度などを用いてリンク候補をより多く抽出できるようにしていること、本研究ではアノテーションを用いることで設計書に日本語で書かれた要素名とも対応が取れるようにすることが挙げられる。

森ら[2]の研究では UML のクラス図と Java で書かれたソースコード間でリアルタイムにトレーサビリティを維持するツールを提案している。UML と Java ソースコードそれぞれの記述エリアが存在する。Java ソースコードを記述すると、その内容に対応したクラス図のクラスをツールが UML 記述エリアに記述する。本研究との類似点は対象がクラス図と Java で書かれたソースコードであること、ソースコードを書き換えながら使用できることが挙げられる。相違点としては森らの研究はクラス図のクラスなどを自動で作成できること、本研究はアノテーションを用いることで設計書に日本語で書かれた要素名とも対応が取れるようにすることが挙げられる。

## 4. リンクの作成手法

既存の手法では、日本語で書かれた設計書の要素と英語で書かれたソースコードの要素間のトレーサビリティリンクを明示的に示すことは難しい。そこで、本研究ではこのような要素にトレーサビリティリンクを作成するためにアノテーションを利用する。アノテーションを利用することで書き方や書かれる場所が統一される。さらに、スペルミスなど書き間違いをした際に、コンパイラがチェックを行

<sup>†1</sup> 日本工業大学  
Nippon Institute of Technology

<sup>†2</sup> 玉川大学  
Tamagawa University

<sup>†3</sup> 東京学芸大学  
Tokyo Gakugei University

エラーとして検出できるため、トレーサビリティリンク失われていることにユーザーが気づきやすくなる。アノテーションはクラスやフィールドなど、予めターゲットを決めて用意する。こうすることによって他の場所に間違えて書くことを防ぐことができる。アノテーションの変数に設計書の要素名を入れることで書き換える量を最小限にリンクを作成することができる。さらに、コンパイラが要素名を取得することもできる。図1は設計書で四角形と書いてあるクラスとソースコードで Square と書いてあるクラスに、アノテーションを用いてリンクを作成した場合の例を示している。

```
1 @UMLClass(name="四角形") ←  
2 public class Square { ←  
3  
4 } ←
```

図1 アノテーションの利用例

アノテーションの導入だけでは、トレーサビリティリンクを記述するユーザーの負担が大きくなってしまふ。このような問題に対応するために、IDEのコード補完機能を拡張して、トレーサビリティリンクを回復させるのに必要な要素のリストを表示させることとした。

```
2  
3 @UMLClass(name="")  
4 public class Squa  
5  
6 }  
7
```

丸  
四角形

図2 コード補完の例

ツールは、ユーザーが多くの要素の中から、リンク候補を探しやすくするために、リンクのない要素をユーザーに優先的にフィードバックする。表示する要素はクラス名、フィールド名、メソッド名の3つである。これらは Eclipse 上に表示される。また、フィールド名とメソッド名は Eclipse 上でアクティブになっているクラスの要素のみ表示される。表示される要素を少なくすることによって、よりユーザーが対象となる要素を見つけやすくする。フィードバックの内容は、設計書にはあるがソースコードにはない要素、ソースコードにはあるが設計書にはない要素の2種類である。この2つのリストが表示されることでユーザーは容易に設計書とソースコードの要素を比較することができる。

## 5. 実現手法

対象とする設計書はクラス図、ソースコードは Java で書かれたものとした。比較する要素はクラス、フィールド、メソッドの3つとした。ツール全体の流れを図3に示す。ツールは Eclipse のプラグインとして実現した。

ツールの機能は以下の3つである。

1. モデルの要素を取得する機能
2. ソースコードから要素を取得する機能
3. それぞれの要素を比較し、結果を利用者にフィードバックする機能

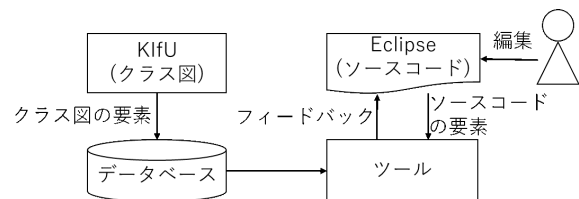


図3 ツール全体の流れ

クラス図は KifU[3]上で編集する。KifU とは概念モデリングの編集過程のデータを細粒度に収集し、編集過程を明らかにすることを実現するために開発されたオンラインのUMLエディタである。クラスの作成、属性、メソッドの追加、関連の作成など、必要な機能が備わっている。記入されたクラス図の要素は KifU のデータベースに格納される。1の機能は、KifUのデータベースからクラス図の要素を取得する機能である。ソースコードは Eclipse 上で編集する。2の機能は、ユーザーがソースコードを変更するたびに、ソースコードから要素を取得する機能である。その際、アノテーションが付けられた要素は、アノテーションに書かれた要素名を取得する。3の機能は、KifUのデータベースから取得したクラス図の要素のリストとソースコードの要素のリストの比較を行う機能である。要素名が完全に一致しているものをリストから削除し、最後まででリストに残った要素がリンクの確保されていない要素となる。

## 6. まとめと今後の課題

本研究では日本語で書かれた設計書とソースコードのトレーサビリティリンクの作成手法の提案を行った。またトレーサビリティリンク回復を支援するツールの提案を行った。今後はツールの有用性を検証していきたい。

**謝辞** 本研究の一部は JSPS 科研費 18K11579 の助成を受けた。

## 参考文献

- [1] 伊藤弘毅, 田邊浩之, 波木理恵子, 鷲崎弘宜, 深澤良彰, “トレーサビリティリンク回復を通じたトレーサビリティ測定と改善支援,” コンピュータソフトウェア, Vol.2013-30, No.3, pp.123-129, Jul. 2013.
- [2] 森敬介, 片山徹郎, “UMLとソースコード間でリアルタイムにトレーサビリティを維持するツール RETUSS について,” 平成29年度電気・情報関係学会九州支部連合大会講演論文集, pp.104-105, Jun. 2019.
- [3] Takafumi TANAKA, Hiroaki HASHIURA, Atsuo HAZEYAMA, Seiichi KOMIYA, Yuki HIRAI, Keiichi KANEKO, “Learners Self Checking and its Effectiveness in Conceptual Data Modeling Exercises,” IEICE TRANSACTIONS on Information and Systems, Vol. E101-D, No.7, pp.1801-1810, Jun. 2018.