

クラス図とソースコード間のトレーサビリティ支援

橋浦研究室

1155453 吉田優介

1. はじめに

ソフトウェア開発や保守において、設計書とソースコード間のトレーサビリティが確保されていないと、ソースコードの理解に時間がかかる等の問題が生じる。近年、ソフトウェアの大規模化・複雑化が進んでおり、トレーサビリティの確保はますます困難になっている。

2. 研究目的

本研究は、設計書通りにコーディングされていないという問題に着目する。開発者はトレーサビリティが確保されているかどうかを確認する機会に乏しいため、知らず知らずのうちにこのような問題が発生していることがある。これを防ぐためには、設計書とソースコードの整合性を自動的に確認し、その差をフィードバックすることによって、開発者自らがトレーサビリティの確保されたソースコードを作成できるようにする必要があると考えた。

3. 提案手法

前述の目的を達成するために、設計書とソースコードそれぞれから要素を抽出し、利用者にその差分をフィードバックすることで修正を促すこととした。

対象となる設計書はクラス図、ソースコードは Java で書かれているものとした。

4. ツールの実装

クラス図は KIfU[1]から、ソースコードは Eclipse から情報の取得を行う。また、フィードバック部分を含めたツールの全体は Eclipse プラグインとして実現することとした。

4. 1. 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。

4. 2. 実装する機能

本研究で実装する機能は以下の通りである。

- ① ソースコードからクラス名、フィールド名、メソッド名を取得
- ② モデルのクラス名、フィールド名、メソッド名の取得
- ③ ①②のデータを比較し、結果を利用者にフィードバック(図 1)

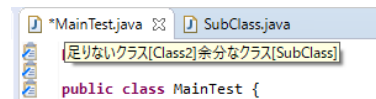


図 1 ツールのフィードバック例

5. 実験

ツールの有効性を確認するために、日本工業大学工学部情報工学科に所属する 13 名を被験者とする実験を行った。

実験内容は、被験者に設計書(クラス図と補足説明)を与え、被験者にその設計書に合致するようなプログラミングを作成させるものである。被験者 1 人につき、ツール支援がある場合となしの場合の計 2 問について解答させ、ツールの支援の有無による差を調べる。

具体的には以下の手順で実験を行った。

- ① 実験の概要を説明する(10分)
- ② 1つ目の課題を仕様書とクラス図を元にプログラミングを行う(開始 45分以降はリタイア可能)
- ③ 2つ目の課題を仕様書とクラス図を元にプログラミングを行う(開始 45分以降はリタイア可能)

プログラミングを行う際、インターネットなどを利用してプログラムに関する内容を調べることは可能とした。

6. 評価方法

実験を行うにあたって評価する項目は、以下の3点である。

1. クラス図と比較して、ソースコード中のクラスに過不足がないか
2. クラス図と比較して、ソースコード中

のフィールドに過不足がないか

3. クラス図と比較して、ソースコード中のメソッドに過不足がないか

本実験における適合率(P), 再現率(R)の定義を(1)(2)に示す.

$$P = \frac{\text{被験者が作成した要素のうち、設計書に含まれる要素の数}}{\text{被験者が記述した要素の数}} \quad (1)$$

$$R = \frac{\text{被験者が作成した要素のうち、設計書に含まれる要素の数}}{\text{設計書に含まれる要素の数}} \quad (2)$$

調和平均(F 値)は適合率(P)と再現率(R)を用いて算出する.

7. 実験結果と考察

実験結果の概要を表 1 に示す. 表 1 を見ると, ツールを使用した際にはクラス名とメソッド名で F 値が 1.00 となった. また表 1 の①-②の部分を見ると, フィールドに関して, ツールを使用した場合と使用していない場合で 0.16 と最も大きな差が出ていることがわかる. これについて, Welch の t 検定を行った. 帰無仮説 H_0 を「ツール未使用時とツール使用時の F 値は等しい」とした. 有意水準 α は 0.05 とし, 自由度 n は 12 であるから, $t_{(0.05)12}=2.17$ と求まる. 検定の結果を表 2 に示す. $t_{(12)}=2.03$, n.s. のため最も差が出たフィールドでもわずかに届かなかった.

表 1 実験結果の平均

		クラス	フィールド	メソッド
ツールを 未使用	適合率	0.96	0.73	0.98
	再現率	0.96	0.95	0.98
	F 値(①)	0.96	0.82	0.98
ツールを 使用	適合率	1.00	0.98	1.00
	再現率	1.00	0.98	1.00
	F 値(②)	1.00	0.98	1.00
F 値の変化 (②-①)		+0.04	+0.16	+0.02

表 2 フィールド名に関する検定表

	ツールを使用	ツールを未使用
平均	0.98077	0.73
分散	0.00481	0.19231
観測数	13	13
自由度	12	
t	2.03026	
P	0.06331	

ツールを使った場合と使わなかった場合のソースコードを詳細に調べたところ, 外部的振る舞いが仕様書通りでなかったり, 動作しなかったりしたものに関しても, 本ツールを使用した場合, クラス名とメソッド名に関してはトレーサビリティが確保されていることが分かった. また, ツールを使用していないかった場合には, 図 2 のようにスペルミスが見られた. 従って, このような間違いはツールを使用することで削減することができることが明らかになった.

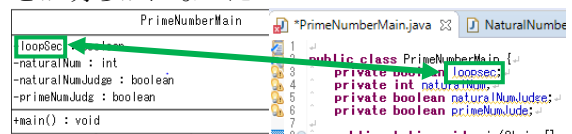


図 2 ツールを使用しなかった場合の結果例

8. まとめと今後の課題

本研究では, クラス図とソースコード間のクラス名, フィールド名, メソッド名に関してトレーサビリティを確保できるように支援する Eclipse のプラグインを開発した. また, このプラグインが有効であるかを実験により検証した. ツールを使用した際にはクラス名とメソッド名で F 値が 1.00 となった. また, ツールを使用していない場合にはスペルミスなどが発生しており, ツールがそれを抑制することが分かった. 今後の課題として, トレーサビリティを確保できる範囲が狭いため範囲を広げる必要があることが挙げられる. 今回, 開発したプラグインではクラス, フィールド, メソッドの名前しか判別することができない. そのため範囲を拡張して, 型や引数などにも対応できるようにすることでより確実なトレーサビリティの確保が期待できる.

参考文献

- [1] 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.