

Dr hab. inż. Ilona Bluemke
Instytut Informatyki, PW
Warszawa, Nowowiejska 15/19

Warszawa 5 lipca 2021

Recenzja rozprawy doktorskiej

Tytuł rozprawy: **Testowanie mutacyjne – optymalizacja procesu i praktyczne zastosowania**

Autor rozprawy: **mgr Michał Mnich**

Zakres i cel rozprawy

Testowanie mutacyjne zaproponowano w latach siedemdziesiątych poprzedniego stulecia. Polega ono na celowym wprowadzeniu do programu drobnej zmiany np. zmiana operatora czy np. zmiennej. Taki zmieniony program nazywa się mutantem. Mutanty wykonuje się podając przygotowane dane testowe. Przypadki testowe powinny skutkować innym stanem dla oryginalnego programu i mutantu. Jeśli test wykryje wprowadzoną zmianę oznacza to że mutant został wykryty – „zabity” (ang. killed). Niektóre mutanty dla dowolnego zestawu danych wejściowych dają takie same wyniki działania jak program oryginalny i nie mogą zostać wykryte, nazywane są mutantami równoważnymi (ang. equivalent mutant). Stosunek liczby wykrytych mutantów do ogólnej liczby wygenerowanych mutantów (eng. mutation score indicator) może być miarą jakości zbioru przypadków testowych.

Mimo że testowanie mutacyjne wprowadzono ponad czterdzieści lat temu to nie jest ono powszechnie stosowane. Przyczyną jest bardzo duża pracochłonność testowania ze względu na ogromną liczbę wygenerowanych mutantów, problemy z detekcją mutantów równoważnych (często trzeba to robić „ręcznie”) oraz błędy sztucznie wprowadzane przez operatory mutacyjne są mało realistyczne. Recenzowana rozprawa próbuje osłabić niektóre z tych problemów.

Praca dotyczy zagadnień optymalizacji procesu testowania mutacyjnego oraz jego zastosowania w praktycznych procesach inżynierii oprogramowania. W pracy zaproponowano

WPLYNĘŁO

05-07-2021

mechanizmy optymalizacyjne, mające na celu zmniejszenie czasu trwania procesu testowania mutacyjnego. Pierwszy mechanizm dotyczy redukcji liczby mutantów na podstawie analizy zmian w kodzie pomiędzy różnymi wersjami oprogramowania. Drugi oparty jest na podejściu bayesowskim do ustalania prawdopodobieństwa generacji mutantów z określonej grupy operatorów mutacyjnych.

W pracy przedstawiono generowanie wielu mutantów w jednej kompilacji oraz opisano eksperymenty weryfikujące poprawę wydajności procesu testowania mutacyjnego. Przebadano także zastosowanie testowania mutacyjnego w procesie wytwarzania oprogramowania typu Test Driven Development.

Celem pracy jest optymalizacja procesu testowania mutacyjnego, możliwa do zastosowania w praktyce.

Autor postawił sobie następujące cele (tezy) badawczych:

1. Efektywność testowania mutacyjnego nie zmniejsza się znacząco, jeśli w nowej wersji oprogramowania mutanty generowane są wyłącznie dla kodu nowego bądź zmodyfikowanego.
2. Zmiana rozkładu prawdopodobieństwa losowania operatorów mutacyjnych, dokonana przy użyciu podejścia bayesowskiego, może być wykorzystana do istotnego zmniejszenia liczby mutantów przy dopuszczalnej niewielkiej utracie efektywności procesu.
3. Wzbogacenie podejścia Test-Driven Development o testowanie mutacyjne znacząco podnosi charakterystyki jakościowe kodu i zwiększa niezależność testowania poprzez obniżenie zjawiska stronniczości (ang. bias) autora testowanego kodu.
4. Podejście polegające na wprowadzeniu wielu mutacji do jednej kompilacji ma swoje ograniczenia i nie każdy operator mutacyjny może współistnieć z wszystkimi innymi.
5. Podejście polegające na wprowadzeniu wielu mutacji do jednej kompilacji dla języków, w których modyfikacja kodu pośredniego nie jest możliwa, może znacząco przyspieszyć proces testowania mutacyjnego.

Tezy rozprawy dobrze określają zakres pracy i są uzasadnione. Właściwe jest także ich wzajemne powiązanie.

Zawartość rozprawy

Rozprawa zawiera dziesięć rozdziałów, sto osiemdziesiąt trzy strony (łącznie z dodatkami) i dziewięćdziesiąt siedem pozycji literatury.

Rozdział pierwszy jest wprowadzeniem w tematykę rozprawy. Przedstawiono w nim cel oraz tezy pracy.

W rozdziale drugim Autor zamieścił rys historyczny i króciutki przegląd literatury dotyczącej testowania oprogramowania.

W rozdziale trzecim Autor opisuje testowanie mutacyjne, przedstawia różne operatory mutacyjne. Zawarto tu także rys historyczny i krótki przegląd literatury dotyczącej testowania mutacyjnego.

W rozdziale czwartym Autor opisuje znane w literaturze podejścia do optymalizacji procesu testowania mutacyjnego, a także omawia modele optymalizacyjne, które zostały dokładniej przedstawione w rozdziałach następujących.

W rozdziale piątym opisano warunki, w jakich niektóre operatory mutacyjne mogą lub nie mogą współistnieć w jednym kodzie. Na tej podstawie zaimplementowano proces testowania mutacyjnego oparty o generacje wielu mutantów w jednej kompilacji.

W rozdziale 6 wprowadzono model optymalizacyjny (tzw. 'Mutation Churn Model'), którego celem jest redukcja liczby mutantów w kodzie, nie powodująca znaczącego spadku efektywności procesu testowania mutacyjnego. Omówione zostały również wyniki badań empirycznych weryfikujących skuteczność zdefiniowanego modelu.

W rozdziale siódmym wprowadzono drugi model optymalizacyjny, model redukcji liczby mutantów oparty na podejściu bayesowskim. Przedstawiono także wyniki badań empirycznych.

Rozdział ósmy zawiera opis systemu S.A.M. (Symultanic Automatic Mutation System) — aplikacji stworzonej przez Autora.

Rozdział dziewiąty zawiera propozycję metody wytwarzania oprogramowania opartą na mutacyjnym testowaniu oprogramowania. Metoda ta jest modyfikacją podejścia TDD poprzez wzbogacenie jej o krok analizy mutacyjnej. W rozdziale tym przedstawiono wyniki eksperymentów.

Rozdział dziesiąty zawiera wnioski oraz podsumowanie.

Uwagi szczegółowe

Mimo poprawnej edycji rozprawy Autor nie ustrzegł się pewnych drobnych usterek i błędów. Poniżej omawiam niektóre z nich:

- Kilka drobnych błędów językowych np. str. 64 – „oraz zestaw katalogów zawierających każdorazowo *skompilowanych plik* z pojedynczą mutacją”, czy sporo zdań typu : (str.15) *”Rozdział 7 wprowadza drugi model ..”*- nie rozdział wprowadza model ale „w rozdziale wprowadzono /opisano model .
- Np. str. 40-41 – brak odwołania do listingu 3.12
- Niejasności stylistyczne np. str. 41 „W pracy Kintis et al. [27] uzyskano wyniki wahające się od 65.5% do 86.8%. W tej pracy *badania były przeprowadzane w języku Java.*”- mogę się domyślać że mutacje dotyczyły programów napisanych w języku Java
- Str. 54 : „W powyższym kodzie MNO ” - w kodzie listingu 5.1 jest zmienna M_NO
- Str. 62: „Listing 5.11 prezentuje kod przed mutacją, natomiast listing 5.11 prezentuje zmutowany kod.” , powinno być - natomiast listing 5.12 prezentuje zmutowany kod.
- Str. 121: „Transpozycja: Brak *wyplenionego* testu” – co to jest wypleniony test ?
- powtórzenia słów np. str. 122: „W trzeciej iteracji poprawione zostało *wypisywanie wypisywanie* wartości na konsolę”.
- Niejednorodny format pozycji bibliograficznych

Wyżej wymienione drobne usterki nie wpływają na moją pozytywną ocenę rozprawy.

Ocena rozprawy

Rozprawa jest poprawnie napisana, jej układ jest logiczny jednak jest to **bardzo słaba** praca. Bardziej przypomina dobrą pracę magisterską a nie rozprawę doktorską. Mam zastrzeżenia co do zawartości poszczególnych rozdziałów. Moim zdaniem zupełnie niepotrzebny jest rozdział drugi z bardzo podstawowymi informacjami o testowaniu, może przydatny w pracy inżynierskiej czy magisterskiej ale nie jest to poziom rozprawy doktorskiej i nie jest potrzebny w pracy doktorskiej. Rozdział trzeci jest za długi, niepotrzebne są opisy wraz z listingami prostych operatorów mutacyjnych. Zupełnie niepotrzebne są „rysy historyczne” w doktoracie technicznym (w rozdziale drugim i trzecim). Łącznie 47 stron ze 159 (pozostałe strony to dodatki) są to rzeczy „ogólne”, nie będące wkładem Doktoranta.

Literatura zawiera 97 pozycji ale wiele z nich są to prace dotyczące podstaw testowania. Jedynie 35 pozycji dotyczy testowania mutacyjnego (nie licząc stron internetowych) a aż 19 Test Driven Development dotyczącego jednego tylko rozdziału (ósmego). Proporcje pozycji bibliografii są moim zdaniem niewłaściwe. Literatura jest nieaktualna. Autor zakończył poszukiwania literaturowe na roku 2017 i z 2020 dodał tylko jedną własną publikację. Ustawienia klucza wyszukiwania „mutation testing” tylko w jednej bazie - IEEE w okresie 2017-2021 zwróciło

ponad 900 trafień (na pewno nie wszystkie będą pasowały ale wiele na pewno tak). Przegląd literatury jest bardzo słaby, pomijając nieaktualną literaturę i niespójny format. W pracy doktorskiej spodziewałabym się systematycznego przeglądu literatury (ang. Systematic Literature Review – SLR) dotyczącego tematyki pracy czyli optymalizacji testowania mutacyjnego oraz testowania mutacyjnego w procesach wytwórczych) i na tle takiego przeglądu lepiej widać by było wkład Doktoranta w dziedzinę. W przypadku stron internetowych Autor nie podał kiedy następował dostęp do stron a wiadomo że ich zawartość szybko się zmienia.

Dla mnie praca ma charakter czysto eksperymentalny. Co prawda Autor pisze że praca „zawiera także teoretyczne uzasadnienie tego, które mutanty mogą, a które nie mogą istnieć w jednej kompilacji dla programów w paradygmacie obiektowym” oraz że „Praca opisuje również teoretyczny model, wraz z doświadczalną jego weryfikacją w praktyce, zastosowania podejścia mutacyjnego w procesach zwinnego wytwarzania oprogramowania. Opisana metodyka dotyczy włączenia procesu mutacyjnego w model TDD”. Niestety ja nie mogę się dopatrzeć modeli teoretycznych a jedynie widzę opisowe wyjaśnienia, uzasadnienia.

Eksperymenty polegały na przebadaniu :

- Efektywności procesu testowania mutacyjnego z użyciem metody generacji wielu mutantów w jednej kompilacji
- Procesu testowania mutacyjnego stosującego mutacje tylko i wyłącznie na zmienionym lub nowym kodzie między wersjami danego oprogramowania.
- Zastosowania podejścia bayesowskiego w testowaniu mutacyjnym.
- Zastosowania testowania mutacyjnego w Test Driven Development.

Wyniki badań eksperymentalnych są interesujące jednak znacząca ich część była już publikowana a prezentacja wyników eksperymentu 2 w rozdziale ósmym jest nudna i bardzo długa. Ogólnie mam duży problem z określeniem, które eksperymenty nie były wcześniej publikowane gdyż informacje o publikowaniu pojawiają się czasem dość późno w tekście np. eksperyment 1 – rozdział ósmy pojawia się na stronie 116 a na stronie 146 znajduje informację „Dane z pierwszego eksperymentu opisanego w tym rozdziale były prezentowane na konferencji TestWarez 2017 [91].”

Autor arbitralnie przyjmuje pewne wartości istotne w przebiegu eksperymentu nie uzasadniając ich np. w eksperymencie „Bayes per klasa” – strona 91, początkowe wartości α i β ustawia na 12 – dlaczego akurat tyle ?.

Mam również zastrzeżenia co do wielu wniosków z eksperymentów. Np. str. 102 : „Wzrost wydajności mierzony za pomocą ogólnej liczby analizowanych mutantów wynosi 55% w stosunku do mutacji klasycznej, a liczba niewykrytych mutantów, które nie były pokryte przez testy, jest mniejsza o 50%. Na podstawie tych danych możemy wnioskować, że przy

odpowiednim ustawieniu parametrów w modelu możemy uzyskać zadowalający wzrost wydajności przy stosunkowo niskim ryzyku utraty istotnych danych.” Dla mnie liczba niewykrytych mutantów na poziomie 50 % jest NIEZADAWALAJĄCA.

W związku z dużą liczbą niewykrytych mutantów w proponowanych optymalizacjach mam wątpliwości czy można te metody stosować w praktyce a celem Doktoranta były praktyczne zastosowania.

Bardzo dużą wadą pracy jest brak związku między eksperymentami a tezami pracy. Tezy są w rozdziale pierwszym a krótkie ich uzasadnienie pojawia się dopiero w rozdziale dziesiątym, w podsumowaniu pracy. Wygląda to tak jakby Doktorant miał zbiór eksperymentów, opisał je i na samym końcu przypomniał sobie że w doktoracie powinny być tezy badawcze więc dodał je ale już nie chciało mu się modyfikować opisu eksperymentów.

Autor jest świadomy pewnych zagrożeń dla wiarygodności uzyskanych przez siebie wyników i w skrócie je opisuje w rozdziałach 5.6, 6.5 i 7.12, 8.7.

Należy podkreślić duży nakład pracy Autora w przeprowadzeniu badań eksperymentalnych.

Końcowe uwagi merytoryczne

Autor uzyskał postawione przez siebie cele. Problematyka testowania mutacyjnego jest bardzo ważna i aktualna dla dziedziny inżynierii oprogramowania. Wkład Autora w problematykę testowania mutacyjnego jest zadowalający. Za najważniejsze osiągnięcia uważam:

- Przeprowadzenie badań eksperymentalnych.
- Implementacja systemu służącego do testowania mutacyjnego z zaimplementowaną metodą optymalizacji .

Podsumowując należy stwierdzić, że przedmiotem rozprawy jest trafnie sformułowany i bardzo przydatny problem badawczy. Tematyka rozprawy, związana z mutacyjnym testowaniem oprogramowania jest jak najbardziej aktualna i praktycznie ważna. Częściowe rezultaty pracy były publikowane w 3 publikacjach ([57, 63, 91]) z lat 2017 i 2020.

Wniosek końcowy

W opiniowanej rozprawie mgr Michał Mních pokazał umiejętność formułowania celów badawczych oraz warsztat badawczy prowadzący do rozwiązań stawianych problemów. Na podkreślenie zasługuje nakład pracy związany z realizacją eksperymentów.

Uważam że rozprawa brzegowo spełnia wymagania stawiane rozprawom doktorskim przez obowiązującą ustawę o stopniach i tytułach naukowych, wnoszę zatem o dopuszczenie jej do publicznej obrony.

A handwritten signature in cursive script, appearing to read 'J. Duda'.