

Zielona Góra, 18 maja 2022

dr hab. inż. Justyna Patalas-Maliszewska, prof. UZ
Instytut Inżynierii Mechanicznej
Uniwersytet Zielonogórski
E-mail: J.Patalas-Maliszewska@iim.uz.zgora.pl

Recenzja

rozprawy doktorskiej Pana mgr inż. Damiana Giebas
pt.: „Metoda wykrywania konfliktów zasobowych w aplikacjach wielowątkowych”.

Promotor: dr hab. inż. Grzegorz Bocewicz, prof. PK
Promotor pomocniczy: dr inż. Rafał Wojszczyk

Recenzję wykonano na zlecenie Prorektora ds. Nauki dr hab. inż. Błażeja Bałasz, prof. PK, na podstawie uchwały nr 39/2022 Senatu Politechniki Koszalińskiej z dnia 27 kwietnia 2022 w sprawie wyznaczenia na recenzenta rozprawy doktorskiej przygotowanej przez mgr inż. Damiana Giebasę oraz na podstawie pisma nr L.dz.PK/UEiJ/10/04/2022 z dnia 29 kwietnia 2022.

1. Ogólny opis recenzowanej rozprawy doktorskiej

Przedstawiona do recenzji rozprawa doktorska pt.: „Metoda wykrywania konfliktów zasobowych w aplikacjach wielowątkowych” obejmuje 130 stron, składa się ze wstępu, sześciu rozdziałów i podsumowania, streszczenia w języku polskim i języku angielskim, spisu rysunków, spisu tablic, wykazu skrótów, spisu symboli i oznaczeń. Wprowadzenie w pracy spisu skrótów oraz spisu symboli i oznaczeń ułatwiło zapoznanie się z treścią pracy.

W rozdziale pierwszym przedstawiono motywację podjęcia tematu pracy. Poruszana tematyka związana jest z szeroko rozumianym zabezpieczaniem (zapobieganie/identyfikacja zagrożeń) aplikacji wielowątkowych przed skutkami niepożądanych interakcji między realizowanymi współbieżnie wątkami. Przedmiotem badań są aplikacje pisane w języku programowania C, który dzięki włączonej bibliotece *pthread* pozwala na budowanie aplikacji wielowątkowych. Język ten jest wykorzystywany powszechnie w rozwiązaniach przeznaczonych dla urządzeń mobilnych (tzw. „urządzenia ubieralne”) jak i branży medycznej, motoryzacyjnej czy energetycznej. Jednakże w aspekcie dynamicznego trendu rozwoju mobilnych aplikacji wielowątkowych powstaje pytanie, w jaki sposób podejmowany problem w dysertacji można rozważać w kontekście systemów operacyjnych Android czy iOS. W tej części pracy sformułowano problem oraz tezę badawczą a także przedstawiono strukturę i zakres pracy. Niedosyt budzi jednak brak opisu celu głównego oraz szczegółowych celów badawczych.

W rozdziale drugim omówiono istniejące rozwiązania pozwalające na lokalizowanie błędów prowadzących do konfliktów zasobowych typu: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności i naruszenie porządku. Kolejna część pracy to szczegółowy opis zbudowanego modelu

stanowiącego podstawę do opracowania warunków lokalizowania konfliktów zasobowych w aplikacjach wielowątkowych. Rozdział czwarty przedstawia rezultaty prac badawczych, tj. zdefiniowanie warunków koniecznych lokalizowania konfliktów zasobowych, jak również zbudowane twierdzenia i ich dowody. Kolejny rozdział to prezentacja zbudowanej metody lokalizowania konfliktów zasobowych (ang. Races and Deadlocks - Locating, (RD-L)). Rozdział szósty to opis wyników eksperymentalnej oceny metody RD-L implementowanej w aplikacji rdao detector. W podsumowaniu pracy przedawniono przebieg prac, osiągnięte rezultaty prac badawczych oraz kierunki dalszych prac. Układ dysertacji oraz sposób omówienia poszczególnych obszarów badawczych nie budzi zastrzeżeń.

Doktorant podjął się rozwiązania tematu badawczego związanego z zagadnieniem lokalizowania błędów prowadzących do konfliktów zasobowych w procesie wytwarzania aplikacji wielowątkowych. Temat nie jest nowy, jednak w kontekście dynamicznego rozwoju technologii informacyjnych oczekuje się obecnie narzędzi wspierających pracę programistów rozwijających programy komputerowe. Zdefiniowany problem badawczy w postaci poszukiwania metody wykrywania błędów prowadzących do konfliktów zasobowych w danym programie pokazuje aktualność podejmowanej tematyki w aspekcie konieczności wdrażania narzędzi wspierających pracę programistów w obszarze identyfikacji błędów aplikacji wielowątkowych.

2. Ocena formalna poszczególnych części recenzowanej rozprawy doktorskiej oraz uwagi ogólne

We wstępie pracy przedstawiono definicje najważniejszych pojęć związanych z podejmowanym tematem dysertacji, m.in. język programowania, program, aplikacja wielowątkowa. Wskazano, że na podstawie analizy literatury przedmiotu przeprowadzonej w rozdziale 2, nie można wskazać metod opartych na statycznej analizie kodu źródłowego aplikacji wielowątkowych, których zastosowanie pozwala na lokalizowanie konfliktów zasobowych oraz pokazano, że metody dynamiczne są nieskuteczne w przypadku programów o dużej liczbie scenariuszy zachowań. Niedosyt budzi brak zdefiniowania potrzeby badawczej dotyczącej budowania rozwiązań mających docelowo za zadanie wspieranie pracy programistów w obszarze identyfikacji błędów aplikacji wielowątkowych. Dodanie takiej analizy pozwoliłoby na lepsze zrozumienie ważności podejmowanego tematu. Postawioną tezę uważam za dającą podstawę do przeprowadzenia prac badawczych.

Rozdział drugi został zatytułowany: „przegląd literatury”, w którym dokonano charakterystyki metod lokalizowania konfliktów zasobowych. Pozytywnie oceniam zdefiniowanie siedmiu kryteriów jakościowych i ilościowych, których zastosowanie pozwala na ocenę skuteczności znanych z literatury przedmiotu metod i narzędzi. Analizę przeprowadzono według podziału konfliktów zasobowych, tj.: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności, naruszenie porządku. Każdy podrozdział zamyka lista metod i narzędzi stosowanych do lokalizowania konfliktów zasobowych. Powstaje pytanie, jak można rozumieć nadane oceny w tabeli 2.6? Zrozumiałe jest, że odpowiadają one zdefiniowanym kryteriom oceny, ale w jakim stopniu? Takie wyjaśnienie pozwoliłoby na lepsze zdefiniowanie potrzeby badawczej. W tej części pracy brakuje krytycznej analizy przedmiotu w obszarze potrzeby budowania nowego narzędzia wspierającego pracę programistów.

W rozdziale trzecim podjęto badania dotyczące sformułowania modelu umożliwiającego reprezentację struktury kodu źródłowego aplikacji wielowątkowej w dwóch wymiarach: wymiarze wątków i wymiarze czasu. Oczekuje się dyskusji, dlaczego skoncentrowano badania na modelu

i w konsekwencji metody dedykowanej do kodu programu napisanego w języku C? Zbudowany model przedstawiono również za pomocą grafu, co uważam za słuszne w kontekście, iż opracowana w dalszej części rozwiązanie może być traktowane jako narzędzie wspomagające podejmowanie decyzji przez programistów.

W rozdziale czwartym przedstawiono poszukiwaną metodę lokalizowania konfliktów zasobowych w aplikacji wielowątkowej w kontekście typu: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności i naruszenie porządku. Pozytywnie oceniam w tej części pracy sformułowane twierdzenia i przeprowadzone dowody. Szkoda tylko, iż nie wskazano na ograniczenia opracowanej metody w kontekście jej złożoności obliczeniowej. Jest to niezmiernie ważny element w kontekście zdefiniowanych w rozdziale drugim kryteriów oceny jakościowej i ilościowej dostępnych obecnie metod i narzędzi (np. kryterium VII).

W rozdziale piątym przedstawiono autorską metodę lokalizowania konfliktów zasobowych (ang. Races and Deadlocks - Locating, (RD-L)), której zastosowanie pozwala na identyfikowanie fragmentów kodu źródłowego występowania błędów aplikacji wielowątkowych. Powstaje pytanie, dlaczego autor pracy zaimplementował metodę w języku Python, skoro głównym obszarem zainteresowania jest język C? Napisano również w tym rozdziale, że metoda ta może wspomagać programistów, w szczególności pracujących zgodnie z metodykami zwinnymi zarządzania projektami informatycznymi, co powinno być wyjaśnione i przedstawione zarówno we wstępie pracy, jak i w rozdziale drugim. Ponadto w tej części pracy brakuje wskazania przewagi opracowanej metody nad rozwiązaniami przedstawionymi w tabeli nr 2.6 i jej oceny wedle kryteriów I-VII.

Rozdział szósty prezentuje wyniki eksperymentów badawczych przeprowadzonych dla 19 aplikacji wielowątkowych napisanych w języku C. Według jakich kryteriów dobrano takie aplikacje do testów? Czy jest możliwe wykazanie skuteczności opracowanej metody na przykładzie jej zastosowania w pracy programistów przy wytwarzaniu aplikacji wielowątkowej, pracujących zgodnie np. z metodyka SCRUM?

W podsumowaniu przedstawiono wnioski poznawcze oraz kierunki dalszych prac. Spis pozycji literatury uważam za prawidłowy i wystarczający. Mgr Giebas powołuje się na 114 pozycji literaturowych i źródeł internetowych, w tym w 10 pozycjach jest współautorem.

3. Ocena merytoryczna pracy

Zagadnienie naukowe i ocena oryginalności rozwiązania problemu naukowego

Podejmowany problem obejmuje poszukiwanie metody lokalizowania konfliktów zasobowych w aplikacjach wielowątkowych napisanych w języku C z wykorzystaniem biblioteki pthread oraz jej implementację i weryfikację skuteczności w eksperymentach badawczych. Doktorant w sposób poprawny przedstawił podejmowany problem naukowy.

Za oryginalne osiągnięcia w pracy, które wnoszą wkład w rozwój dyscypliny: „informatyka techniczna i telekomunikacja”, w obszarze teorii języków programowania, przetwarzania współbieżnego i równoległego, systemów współbieżnych procesów, uznaję:



- Wyznaczenie siedmiu jakościowych i ilościowych kryteriów do oceny rozwiązań pozwalających na lokalizowanie błędów prowadzących do konfliktów zasobowych typu: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności i naruszenie porządku.
- Opracowanie modelu kodu źródłowego aplikacji wielowątkowej pisanych w języku C z wykorzystaniem biblioteki pthread.
- Sformułowanie i wykazanie czterech twierdzeń dotyczących warunków powstania konfliktów zasobowych typu: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności i naruszenie porządku.
- Opracowanie metody lokalizowania konfliktów zasobowych w instancji modelu kodu źródłowego aplikacji wielowątkowej pisanych w języku C z wykorzystaniem biblioteki pthread.
- Wykazanie użyteczności wyników badań w postaci narzędzia rdao detector wspierającego programistów przy wytwarzaniu oprogramowania.
- Wykazanie skuteczności metody przy pomocy wykonanych eksperymentów badawczych dla wybranych aplikacji wielowątkowych.

Aktualność tematyki i ocena wyników badań

W mojej ocenie podjęty temat badawczy jest aktualny, szczególnie w kontekście wsparcia pracy programistów przy wytwarzaniu oprogramowania, pracujących zgodnie z metodyką zwinną zarządzania projektami informatycznymi. Doktorant opracował właściwy plan badawczy, przeprowadził analizę literatury przedmiotu w zakresie metod i narzędzi wspierających lokalizowanie błędów prowadzących do konfliktów zasobowych, zbudował autorski model, a następnie metodę lokalizowania takich konfliktów zasobowych w aplikacjach wielowątkowych napisanych w języku C z wykorzystaniem biblioteki pthread. Metoda ta stanowi konkurencyjną alternatywę wobec dostępnych obecnie rozwiązań w obszarze procesu statycznej analizy kodu źródłowego aplikacji wielowątkowych.

Krytyczna analiza treści rozprawy i uwagi szczegółowe

Przedstawiona rozprawa powinna zawierać uszczegółowienie przyjętych etapów prac badawczych ze wskazaniem:

- Uzasadnienia podjęcia tematu poszukiwania rozwiązania do lokalizowania konfliktów zasobowych w aplikacjach wielowątkowych napisanych w języku C z wykorzystaniem biblioteki pthread, a nie w innych językach, np. Java, Python, czy językach platformy .NET.
- Wykazania przewagi opracowanej metody nad rozwiązaniami obecnie stosowanymi w obszarze identyfikacji konfliktów zasobowych w aplikacjach wielowątkowych, w szczególności rozwiązań obszarze procesu statycznej analizy kodu źródłowego aplikacji wielowątkowych.
- Wyjaśnienia przyjętych założeń do przeprowadzonych eksperymentów badawczych (np. w kontekście wyboru aplikacji do weryfikacji metody).

Doktorant deklaruje również, że zbudowany model kodu źródłowego aplikacji wielowątkowej pisanych w języku C z wykorzystaniem biblioteki pthread jest dedykowany dla innych języków. W pracy jednak eksperymenty badawcze przeprowadzono wyłącznie dla aplikacji napisanych w języku programowania C z wykorzystaniem biblioteki pthread, bez wskazania aplikacji wielowątkowych napisanych w innych językach, dla których zbudowana metoda mogłaby być skuteczna.

Uwagi szczegółowe:

1. We wstępie pracy przedstawiono potrzebę podjęcia tematu pracy w odniesieniu do metod analizy statycznej, dynamicznej i mieszanej kodów programów. Nie dokonano opisu rozwiązań, w których jak autor podaje, zauważa się ciągłą potrzebę aplikacji napisanych w języku C (branża medyczna, motoryzacyjna, energetyczna). Ponadto w tej części pracy brakuje sformułowania celów badawczych.
2. W rozdziale drugim nie przeprowadzono analizy literatury przedmiotu w obszarze narzędzi wspomagających pracę programistów, którzy wytwarzają oprogramowanie zgodnie z metodykami zwinnymi zarządzania projektami informatycznymi (np. SCRUM). Oczekuje się krytycznej dyskusji dotyczących takich rozwiązań.
3. Oczekuje się szczegółowych wniosków z przeprowadzonej oceny metod i narzędzi stosowanych do lokalizowania konfliktów zasobowych wg zdefiniowanych siedmiu kryteriów (rozdział 2). W jaki sposób można wskazać potrzebę budowania nowej metody?
4. Wyjaśnienia wymaga ograniczenie się do czterech warunków powstania konfliktów zasobowych typu: szkodliwa rywalizacja, zakleszczenie, naruszenie niepodzielności i naruszenie porządku.
5. Rozdziały 3-6 stanowią oryginalny wkład autora w obszarze badań dotyczących języków programowania oraz przetwarzania współbieżnego i systemów współbieżnych procesów.
 - Na jakiej podstawie dokonano reprezentacji kodu źródłowego aplikacji wielowątkowej w postaci siedmiu elementów? Czy można wskazać możliwości rozszerzenia tego modelu?
 - Dlaczego dobrano graf operacji do reprezentacji modelu kodu źródłowego aplikacji wielowątkowej pisanych w języku C z wykorzystaniem biblioteki pthread? Czy sprawdzono możliwości zastosowania innych tego typu rozwiązań?
 - Czy dokonano obliczeń złożoności obliczeniowej opracowanej metody statycznej analizy aplikacji wielowątkowych? Ponadto czy przeprowadzono ocenę efektywności opracowanej metody, w szczególności w zależności od liczby wątków aplikacji?
 - Dlaczego zaimplementowano metodę lokalizowania konfliktów zasobowych w instancji modelu kodu źródłowego aplikacji wielowątkowej w języku Python, skoro głównym obszarem zainteresowania jest język C?
 - Skuteczność opracowanej metody wymaga oceny zgodnie z przyjętymi kryteriami wedle kryteriów I-VII.
 - Brakuje wyjaśnienia doboru 19 aplikacji wielowątkowych eksperymentów badawczych.
 - Oczekuje się dyskusji nad możliwościami wykorzystania proponowanej metody dla aplikacji napisanych w innych językach niż w języku programowania C.
 - W przedstawionym rozwiązaniu metoda wykorzystywania jest w trybie online w trybie interakcji z programistą. Jakie są szanse jej rozszerzenia, które umożliwią automatyczną korektę błędów?

4. Uwagi redakcyjne

W pracy zauważono tylko nieliczne błędy stylistyczno-redakcyjne i niezręczności językowe.

str. 24	Brak podania daty i autora źródła danych: „Zgodnie z „Zgodnie z Concurrency bugs in multithreaded software: Modeling and analysis using Petri nets”
str. 36	Brak podania daty źródła danych: „ (...) pochodzi z książki Encyclopedia of parallel computing Davida Padua i cytowana jest m. in. (...)”
str. 43	Brak podania daty i autora źródła danych: „W pracy pt. „Automatic Detection, Validation and Repair of Race Conditions in Interrupt-Driven Embedded Software”

5. Wnioski końcowe

Merytorycznie dysertację oceniam pozytywnie. Zbudowana metoda lokalizowania konfliktów zasobowych (ang. Races and Deadlocks - Locating, (RD-L)), stanowi poprawne rozwiązanie formalne postawionego problemu. Stwierdzam, że mgr inż. Damian Giebas przedstawił osiągnięcie będące wynikiem przeprowadzonych badań, które wnoszą wkład w rozwój dyscypliny: informatyka techniczna i telekomunikacja. W mojej opinii Doktorant wykazał się dobrą znajomością obszaru budowania rozwiązań wspierających lokalizowanie konfliktów zasobowych w aplikacjach wielowątkowych. Ponadto udowodnił, że potrafi prowadzić samodzielnie odpowiedzialne badania naukowe.

Stwierdzam, działając na podstawie art. 179 ust. 1 Ustawy z dnia 3 lipca 2018 Przepisy wprowadzające ustawę – Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1669), że rozprawa doktorska mgr inż. Damiana Giebas pt.: „Metoda wykrywania konfliktów zasobowych w aplikacjach wielowątkowych” spełnia wymogi określone w ustawie z dnia 14 marca 2003 r. (Dz. U. Nr 65 poz. 595 z późn. zm.) o stopniach naukowych i tytułach naukowych oraz stopniach i tytułach w zakresie sztuki, w dziedzinie nauk inżynieryjno-technicznych i dyscyplinie informatyka techniczna i telekomunikacja.

W związku z tym wnioskuję o przyjęcie rozprawy doktorskiej oraz o dopuszczenie Pana mgr inż. Damiana Giebas do publicznej obrony pracy.


