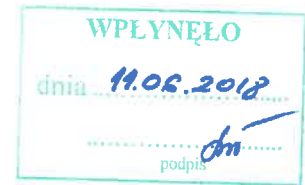


dr hab. inż. Andrzej Karatkiewicz, prof. UZ
Uniwersytet Zielonogórski
Wydział Informatyki, Elektrotechniki i Automatyki
Instytut Inżynierii Elektrycznej
A.Karatkevich@iee.uz.zgora.pl

Zielona Góra, 2018-06-01



**RECENZJA ROZPRAWY DOKTORSKIEJ
DLA RADY WYDZIAŁU
ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ
AKADEMII GÓRNICZO-HUTNICZEJ
IM. STANISŁAWA STASZICA W KRAKOWIE**

Przedmiotem niniejszej recenzji jest rozprawa doktorska zatytułowana „Metody formalnej analizy systemów wbudowanych czasu rzeczywistego”, której autorem jest mgr inż. Jarosław Baniewicz.

Promotorem pracy jest prof. dr hab. inż. Marcin Szpyrka. Niniejsza opinia została przygotowana na zlecenie dra hab. inż. Ryszarda Sroki, prof. nadzw., Dziekana Wydziału Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej AGH, zawarte w piśmie z dnia 27 kwietnia 2018 roku, w związku z decyzją Rady Wydziału EAIIB AGH z dnia 26 kwietnia 2018 roku.

1. Zagadnienia naukowe i naukowo-techniczne rozprawy

Tematyka recenzowanej rozprawy doktorskiej dotyczy modelowania formalnego systemów wbudowanych czasu rzeczywistego za pomocą języka modelowania Alvis, rozwijanego w Katedrze Informatyki Stosowanej AGH. W rozprawie rozpatrywane jest zagadnienie opracowania warstwy systemowej wymienionego języka, zakładającej realizację systemów na platformie jednoprocessorowej (standardowa warstwa systemowa języka Alvis zakłada realizację na systemie o nielimitowanej liczbie dostępnych procesorów). Motywacją pracy jest potrzeba uzupełnienia wspomnianego języka modelowania o możliwość uwzględnienia często występującej w praktyce sytuacji, związanej z ograniczeniem liczby procesorów (w rozpatrywanej wersji – do jednego).

Realizacja tego celu wymagała przede wszystkim opracowania mechanizmów szeregowania zadań, przeznaczonych do wykonania przez różne agenty systemu współbieżnego. Należy również wziąć pod uwagę fakt wykonania zadań przez pojedynczy procesor przy generowaniu grafu osiągalnych stanów modelu. Znaczna część pracy jest poświęcona tym tematom.

Teza rozprawy została sformułowana następująco: *Język Alvis, wsparty odpowiednimi narzędziami komputerowymi, może być efektywnie użyty do modelowania systemów wbudowanych, umożliwiając jednocześnie formalną analizę modelu z zastosowaniem metod i narzędzi typowych dla technik weryfikacji modelowej.*

Teza pracy i cele pracy zostały sformułowane przez Autora w podrozdziale 1.2. Cele zostały określone wystarczająco precyzyjnie, natomiast uważam, że teza jest sformułowana w sposób zbyt ogólny i niefortunny w odniesieniu do danej pracy. Można powiedzieć, że tak sformułowana teza została już uzasadniona we wcześniejszych pracach prof. Szpyrki i jego zespołu.

2. Organizacja i redakcja rozprawy, odniesienia do literatury

Rozprawa składa się z 8 rozdziałów (wliczając wprowadzenie i podsumowanie), dodatku oraz wykazu literatury. Wykaz literatury obejmuje 58 pozycji (pozycje 8 i 9 są identyczne). Jest to liczba raczej niska dla pracy doktorskiej – z reguły liczba cytowanych w dysertacji doktorskiej źródeł jest powyżej 100. Mimo to, wykaz ten obejmuje wiele publikacji związanych z tematyką rozprawy. Chronologicznie spis literatury rozpoczyna się od klasycznej dysertacji C. A. Petriego z 1962 roku, w której wprowadzono pojęcie sieci Petriego, i kończy się najnowszymi publikacjami z lat 2017-2018. Obejmuje on zarówno szereg istotnych monografii na tematy modelowania i formalnej analizy systemów jak i publikacje omawiające szczegółowe zagadnienia z tego zakresu. Można więc stwierdzić, że pozycje bibliograficzne pokrywają zakres tematyczny rozprawy. W skład bibliografii wchodzi 2 autorskie publikacje – rozdział w monografii oraz artykuł w materiałach konferencyjnych.

Praca została napisana starannie z zachowaniem zasad edytorskich, choć Autor nie ustrzegł się szeregu drobnych błędów edytorskich, a w niektórych miejscach (zwłaszcza w podrozdziale 5.3) używa potocznego języka i nieprzemyślanych sformułowań. Poza tym, można odnieść wrażenie że Autor w niektórych rozdziałach zbyt zagłębia się w opisy szczegółów formalnych. Struktura pracy jest jasna i przejrzysta, podział na rozdziały i podrozdziały jest logiczny i sensowny.

3. Osiągnięcia naukowe i uwagi pozytywne

Język Alvis niewątpliwie jest dobrym środkiem modelowania systemów współbieżnych. Niezbędnym uzupełnieniem tego języka jest tak zwana warstwa systemowa, która określa sposób wykonywania operacji przez agenty. Najbardziej ogólną i najlepiej

rozwiniętą warstwą systemową języka jest „bezczasowa” warstwa α^0 , która zakłada, że każdy aktywny agent ma dostęp do własnego procesora. Jest to uproszczenie, wystarczające w wielu praktycznych przypadkach do analizy modelu systemu, lecz są sytuacje, w których trzeba wziąć pod uwagę konkretną liczbę procesorów i rywalizację agentów o procesory. Dlatego w założeniu język Alvis ma być rozwijany o kolejne warstwy. Oczywiście, w pierwszej kolejności powinna zostać opracowana warstwa, zakładająca istnienie jednego procesora. Właśnie to jest główną treścią recenzowanej rozprawy.

Realizacja tego celu wymagała niewątpliwie dużego nakładu pracy, ponieważ pojawiają się tu problemy nie występujące w przypadku nieograniczonej liczby dostępnych procesorów. Do najistotniejszych autorskich *osiągnięć naukowych* w wymienionym kierunku należy zaliczyć:

- precyzyjne określenie algorytmu kolejkowania i szeregowania zadań, wykorzystującego dwuwymiarową kolejkę priorytetową;
- uzupełnienie zestawu tranzycji o dodatkową tranzycję systemową, niezbędną dla realizacji algorytmu szeregowania, oraz dopasowanie funkcjonowania innych tranzycji do potrzeb warstwy systemowej α^1 ;
- opracowanie reprezentacji stanów modelu oraz wersji algorytmu generowania grafu osiągalności, uwzględniającej istnienie kolejki zadań oraz okresowe wywołanie algorytmu szeregującego;
- adaptację najważniejszych funkcji, opisujących dynamikę modelu (mianowicie, funkcji *enable* i *fire*), do potrzeb warstwy, będącej przedmiotem rozprawy.

To wszystko składa się na kompletne, spójne i przemyślane opracowanie warstwy systemowej α^1 , stanowiąc ważne uzupełnienie dotychczasowych prac w zakresie języka Alvis.

Autor ilustruje (w rozdziale 6) funkcjonowanie opracowanej warstwy na przykładach wystarczająco prostych dla zrozumienia i jednocześnie dość dobrze demonstrujących szczegóły autorskich rozwiązań, podając w aneksie pełny kod dla tych przykładów. W rozdziale 7 Autor porównuje język Alvis, wykorzystujący opracowaną warstwę z innymi formalizmami, stosowanymi do modelowania współbieżnych systemów wbudowanych. Natomiast rozdział 2 zawiera zwięzły opis języka Alvis, wystarczający dla przyswojenia jego głównych cech i sposobu modelowania systemów za jego pomocą.

Należy zwrócić uwagę na to, że Autor opisuje swoje osiągnięcia i pokrewne zagadnienia w sposób systematyczny, precyzyjny i, za pewnymi wyjątkami, wyczerpujący.

4. Wątpliwości i uwagi krytyczne

1. Zdaniem recenzenta, tytuł rozprawy nie do końca odpowiada jej treści. Po pierwsze, jest zbyt ogólny – taki tytuł raczej by pasował do obszernej monografii, opisującej rozmaite metody analizy formalnej układów czasu rzeczywistego, i nic nie mówi o konkretnej tematyce rozprawy. Po drugie, praca dotyczy formalnego modelowania, które jest niezbędne dla formalnej analizy, natomiast kwestie formalnej analizy jako takiej nie są w niej rozpatrywane.
2. W pracy brakuje uzasadnienia, czemu wybrano właśnie taki, spośród wielu możliwych, algorytm szeregowania. Nie uzasadniono również, czemu został wykorzystany mechanizm wyłączenia, które może nastąpić nawet przed zakończeniem wykonania bieżącej instrukcji, co komplikuje zarządzanie wykonaniem zadań i konstruowanie grafu osiągalności. Autor co prawda zaznacza, że możliwe są inne wersje warstwy jednoprocessorowej, i dlatego oznacza swoją wersję jako α^1_{FPPS} (od *Fixed Priority Preemptive Scheduling*), jednak pytanie o uzasadnienie wyboru takiej właśnie wersji pozostaje bez odpowiedzi.
3. Zaproponowana przez Autora dla algorytmu szeregującego "dwuwymiarowa kolejka" bardzo przypomina znaną od kilkudziesięciu lat "wielopoziomową kolejkę" (*multilevel queue*), opisaną m. in. w książce A. Silberschatza, P. B. Galvina i G. Gagne'a *Operating system concepts* (John Wiley & Sons, Inc., różne wydania). Na czym polega różnica między tymi strukturami? Jeśli zaś różnicy nie ma, czym jest uzasadnione wykorzystanie innej niż tradycyjna nazwy?
4. Ciekawe mogłoby być porównanie modelowania tych samych systemów, na przykład opisanych w rozdziale 6, w warstwie α^1 i α^0 . Na przykład, jak by się różniła dla takich przypadków liczba osiągalnych stanów?
5. W rozdziale 7 Autor dokonuje porównania języka Alvis z automatami czasowymi oraz z czasowymi interpretowanymi sieciami Petriego. W ramach porównania języka Alvis z automatami czasowymi Autor demonstruje i analizuje modelowanie tych samych systemów za pomocą dwóch wymienionych formalizmów, co pozwala ocenić ich wady i zalety. Natomiast inaczej skonstruowana ta część rozdziału 7, w której Alvis jest

porównywany z czasowymi interpretowanymi sieciami Petriego. Autor w tym przypadku przedstawia pewien przykład, opisany za pomocą sieci Petriego należącej do wspomnianej klasy, i dość szczegółowo pokazuje kroki jego symulacji. Autor nie przedstawia modelu tego samego systemu za pomocą języka Alvis, natomiast w dalszej części tekstu, dotyczącej porównania, po raz kolejny wspomina o możliwości szeregowania zadań i innych cechach opracowanej przez siebie warstwy. W takim przypadku opis czasowych kolorowanych sieci Petriego oraz opis przykładu modelowanego za pomocą takiej sieci systemu mijają się z celem. Brakuje opisu tego samego przykładu w języku Alvis.

6. Zgodnie z opisem systemu *WatchDog* (podrozdział 7.1), jeden z jego elementów odpowiada za wysyłanie sygnału *hb* raz na 100 jednostek czasu, chyba że element ten otrzyma polecenie wykonania resetu, i wtedy procedura odliczania 100 jednostek czasu rozpoczyna się ponownie. Zostało to zamodelowane za pomocą automatu czasowego pokazanego na Rys. 7.2. Natomiast w modelu odpowiedniego agenta w języku Alvis (listing 7.1) procedura resetowania nie została zamodelowana. Czy to oznacza, że w języku Alvis takiej sytuacji nie można zamodelować? Jeżeli tak, wymagałoby to odpowiedniego komentarza. W przeciwnym przypadku należałoby to zrobić.

Niektóre uwagi szczegółowe

- str. 63: "upływ czasu jest kolejną tranzycją systemową" – jest to mało precyzyjne sformułowanie.
- str. 64: dwukrotnie powtórzone słowo "zmiany".
- str. 73: „Jeżeli kolejka ma długość 1 (czyli tak naprawdę 0)” – czytelnik musi się domyślać, o co chodziło Autorowi.
- str. 74: "Dla każdej tranzycji rozważane są 4 wartości i na podstawie tego, które z nich mają określoną wartość" – bardzo nieprecyzyjne sformułowanie.
- str. 87, 88: zamiast "PublisherA, PublisherB oraz PublisherC" powinno być "ReceiverA, ReceiverB oraz ReceiverC", jak wynika z Rys. 6.7 i dalszego tekstu.
- str. 100: "Proces zerowania zegara wyraża się w następujący sposób: $x := 0$. Zapis ten oznacza wyzerowanie zegara x ." Wyrażna nadmiarowość w sformułowaniu.

- str. 106, listing 7.3, linijka 10: zamiast `log_ok_` powinno być `log_nok_` (błąd w listingu prowokuje pytanie: czy została przeprowadzona symulacja tego modelu?)
- str. 109: "Relacja mniejszości stanowi, że jeżeli rozmiar danego wektora pieczętek czasowych jest mniejszy bądź równy w stosunku do rozmiaru porównywanego wektora oraz jeżeli wartości jego pieczętek są równe lub większe od pieczętek czasowych tegoż porównywanego wektora, to wektor ten uznajemy za mniejszy od porównywanego." – tu nie jest jasne, jak należy porównywać pieczętki dwóch wektorów o różnej długości.

5. Przydatność rozprawy dla nauk technicznych

Analiza i weryfikacja formalna systemów czasu rzeczywistego jest zagadnieniem o bardzo dużym i rosnącym znaczeniu na aktualnym etapie rozwoju techniki cyfrowej i w dającej się przewidzieć przyszłości. Problem formalnej analizy okazuje się skomplikowanym zwłaszcza w przypadku systemów współbieżnych, do których należy znaczna część sterujących systemów wbudowanych. W tym kontekście niewątpliwie ważnym jest rozwój formalizmów pozwalających modelować takie systemy. Praca mgra Baniewicza wpisuje się w tę tematykę, stanowiąc bardzo istotne uzupełnienie dotychczasowych prac nad językiem modelowania Alvis. Uzupełnienie to polega na kompletnym opracowaniu warstwy systemowej, odpowiedniej do modelowania systemów jednoprocessorowych. Wobec powyższego nie mam wątpliwości odnośnie przydatności pracy Autora dla nauk technicznych, a mianowicie dla informatyki.

6. Wniosek końcowy

Zdaniem recenzenta, rozprawa doktorska mgra inż. Jarosława Baniewicza spełnia wymagania stawiane w "Ustawie o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki" z dnia 14 marca 2003r. (Dz. U. Nr 65 poz. 595) stawiane pracom na stopień doktora nauk technicznych. Wnoszę o dopuszczenie rozprawy doktorskiej do publicznej obrony pracy doktorskiej w dyscyplinie *Informatyka*.

dr hab. inż. Andrzej Karatkiewicz, prof. UZ