Kraków, 13.06.2018

Prof. dr hab. inż. Tomasz Szmuc
Katedra Informatyki Stosowanej
Akademii Górniczo-Hutniczej
Al. Mickiewicza 30, 30-052 Kraków
e-mail: tsz@agh.edu.pl

RECENZJA ROZPRAWY DOKTORSKIEJ
mgr. inż. Jarosława Baniewicza:

METODY FORMALNEJ ANALIZY SYSTEMÓW WBUDOWANYCH CZASU RZECZYWISTEGO

Niniejsza recenzja została przygotowana na zlecenie Dziekana Wydziału Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej dr hab. inż. Ryszarda Sroki, prof. n. – pismo z dnia 27.04.2018. Zlecenie dotyczy oceny spełnienia przez rozprawę doktorską mgr. inż. Jarosława Baniewicza: *Metody formalnej analizy systemów wbudowanych czasu rzeczywistego* warunków określonych w art. 13 ust. 1 Ustawy o stopniach i tytule naukowym.

1. Charakterystyka wyboru tematu i przedmiot rozprawy

Zastosowanie metod formalnych do modelowania i analizy systemów czasu rzeczywistego ma już swoją historię. Początki sięgają lat 70 wieku, kiedy metody te były raczej rozwijane przez teoretyków. Zwiększone zainteresowanie nastąpiło w latach 80 po wypadkach spowodowanych błędnym działaniem systemów czasu rzeczywistego. Zwrócono uwagę na możliwości modelowania i weryfikacji (dowodzenia) poprawności konstruowanego oprogramowania. Spowodowało to rozwój metod formalnych i związanych z nimi narzędzi (modelowanie, analiza, weryfikacja), jak również sformułowanie norm dotyczących użycia tych metod w procesie konstrukcji oprogramowania. Mimo stosunkowo długiego okresu i badań w wielu ośrodkach ciągle brak jest odpowiednich metod i narzędzi umożliwiających efektywne użycie metod formalnych. Rozwijanie siły ekspresji formalnego opisu, narzędzi ułatwiających ich stosowanie jest ciągle istotnym wątkiem badawczym w Informatyce. Jest to szczególnie ważne w dobie ekspotencjalnego wzrostu zastosowań systemów wbudowanych związanych z bezpieczeństwem (inteligentne samochody), jak również w kontekście złożonych systemów czasu rzeczywistego (sterowanie inteligentnymi sieciami).

Recenzowana rozprawa dotyczy rozszerzenia języka Alvis i skojarzonego z nim narzędzia przez zdefiniowanie warstwy systemowej umożliwiającej modelowanie algorytmu szeregującego (*Scheduler*) w systemie jednoprocesorowym. Zakłada się, że jest to szeregowanie z wywłaszczaniem i dotyczy procesów (agentów) posiadających stałe priorytety. Bazujący na algebrze procesów język Alvis jest istotnym krokiem w stronę poprawy czytelności formalnego modelu, a wyposażenie tego języka w dodatkowe możliwości analizy szeregowania jest ważnym jego rozszerzeniem.

2. Konstrukcja rozprawy

Rozprawa składa się z 8 rozdziałów (łącznie z *Wprowadzeniem* oraz *Podsumowaniem*), Aneksu zawierającego pełny kod (w języku Alvis) prezentowanych przypadków studialnych oraz bibliografii zawierającej 59 pozycji związanych z tematyką badań. Całość jest prezentowana łącznie na 142 stronach.

Zawartość poszczególnych rozdziałów opisano poniżej.

1. Rozdział 1 zawiera wprowadzenie do problematyki pracy i składa się z dwóch części. W pierwszej z nich opisano podstawowe pojęcia: systemy czasu rzeczywistego, systemy wbudowane, model systemu i metody formalne. W drugiej przedstawiono określono problem naukowy, sformułowano tezę rozprawy oraz omówiono jej zwartość.

2. Rozdział 2 zawiera krótkie wprowadzenie do języka modelowania Alvis. Prezentacja tego języka została ograniczona do elementów niezbędnych z punktu widzenia dalszej części rozprawy.

3. Rozdział poświęcono tematyce określania stanu modelu w języku Alvis. Punktem wyjścia jest definicja modelu zaczerpnięta z monografii autorstwa Promotora rozprawy. W dalszej części definicja ta jest analizowana z punktu widzenia przyjętej przez Doktoranta tzw. warstwy systemowej α^1_{FPPS}, specyfikującej przyjęte założenia odnośnie architektury sprzętowej, w której następnie implementowane są model oraz algorytm szeregowania agentów (procesów). W porównaniu do dostępnego w literaturze formalnego opisu języka Alvis (tzw. warstwa α^0), wprowadzono nowy tryb/fazę *Ready* (R) dla agentów, które mogą wykonać bieżącą instrukcję, ale nie mają dostępu do procesora. W podrozdziale 3.3 opisano przyjęty w rozprawie algorytm szeregujący, bazujący na kolejkowaniu agentów w fazie *R* zgodnie z ich priorytetem i kolejnością zgłoszenia gotowości. Informacja o gotowości agentów zapisana jest opisana w dwuwymiarowej kolejce FIFO. Uzupełnieniem opisu tej kolejki jest przedstawienie zasad współdzielenia procesora przez agenty. Przyjęto zasadę o stałym okresie, po którym następuje wywołanie algorytmu szeregującego. Rozdział kończy formalna definicja stanu modelu, uwzględniająca dwuwymiarową kolejkę agentów oraz zasady określania stanu początkowego.

4. Rozdział 4 zawiera opis przejść (tranzycji) powodujących zmiany stanów w języku Alvis. Tranzycje szczegółowo określają kiedy agent może wykonać instrukcje i jaki będzie wpływ ich wykonania na stan modelu. Przedstawiony opis bazuje na materiałach źródłowych, ale został zaadoptowany do potrzeb nowej warstwy systemowej. W efekcie tych zmian nie tylko zmodyfikowano istniejące reguły aktywności (*enable*) i wykonania tranzycji (*fire*), ale również wprowadzono nową tranzycję systemową *SysTick* reprezentującą przerwanie, związane z wywołaniem algorytmu szeregującego oraz reprezentującą nowo wprowadzoną instrukcję *critical*.

5. Rozdział 5 zawiera opis algorytmu generowania grafu LTS dla systemów modelowanych z wykorzystaniem warstwy systemowej α^1_{FPPS}. Rozdział ten jest kluczowym z punktu widzenia użyteczności rozwiązań zaprezentowanych w rozprawie doktorskiej. Przedstawiono w nim rozszerzenie pośredniej reprezentacji modelu w języku Haskell (IHR – *Intermediate Haskell Representation*), które umożliwia generowanie drzewa LTS zgodnie z założeniami wprowadzonej warstwy systemowej. Warto podkreślić, że zaimplementowane rozwiązanie bazuje na dostępnym kompilatorze języka Alvis, tj. wprowadzenie nowych możliwości nie wymagało zmian w implementacji tego kompilatora. Obsługa nowych elementów: dwuwymiarowa kolejka FIFO, współdzielenie procesora, przerwania systemowe i szeregowanie agentów, zostały zaimplementowane w nowym module.

6. W rozdziale 6 przedstawiono dwa opracowane przed Doktoranta przykłady modeli w języku Alvis, stosujące warstwę systemową α^1_{FPPS}. Pierwszy z nich jest modelem wzorca *Publikator i subskrybent*, a drugi wzorca *Obserwator*. W obu przypadkach opisano najważniejsze elementy modelu oraz omówiono wybrane fragmenty wygenerowanych drzew LTS.

7. W rozdziale 7 podjęto próbę porównania języka Alvis z dwoma najpopularniejszymi formalizmami do modelowania systemów z czasem tj. kolorowanymi sieciami Petriego i automatami czasowymi.

8. Rozdział 8 zawiera podsumowanie osiągnięć Doktoranta oraz propozycję dalszych praz nad rozwojem języka Alvis.

Prezentacja materiału przedstawionego w pracy dokonana jest w sposób relatywnie czytelny i zrozumiały. Stosowana terminologia i symbole nie budzą w zasadzie zastrzeżeń i są zgodne z terminologią stosowaną w innych pracach na temat języka Alvis. Praca charakteryzuje się dobrze przemyślanym układem rozdziałów.

3. Ocena merytoryczna rozprawy

Przedstawiona do oceny rozprawa doktorska zawiera wartościowe wyniki, które stanowią oryginalny wkład Autora w obszarze metod formalnych. Oryginalne wyniki zawarto w rozdziałach od 3 do 6, najważniejsze wymieniono poniżej.

1. Opracowanie koncepcji warstwy systemowej α^1_{FPPS}, w tym głównych założeń algorytmu szeregującego, sposobu kolejkowania i wywłaszczania agentów przy stałym okresie zgłaszania przerwania systemowego i dwuwymiarowej kolejki FIFO
2. Adaptacja stosowanego w języku Alvis podejścia *enable-fire* do potrzeb nowej warstwy systemowej, uwzględniająca m.in. określenie sposobu wyznaczania upływu czasu (tranzycja systemowa *STTime*) i zarządzanie algorytmem szeregowania agentów (tranzycja *STSysTick*).
3. Projekt i implementacja algorytmu wyznaczania etykietowanego systemu przejść (LTS) dla modeli z warstwą systemową α^1_{FPPS}. Implementacja polega na rozszerzeniu istniejącego kompilatora języka Alvis, bez konieczności modyfikacji pozostałych jego części.
4. Opracowanie dwóch modeli testowych ilustrujących przydatność rozwiązań opisanych w rozprawie doktorskiej.

Autor wykazał dobrą znajomość prezentowanej tematyki. Wykazał dojrzałość w formułowaniu zagadnień teoretycznych dotyczących w szczególności języka Alvis, jak również umiejętności praktycznej implementacji. Zaproponowane rozwiązanie jest oryginalne na polu metod formalnych, gdyż umożliwia dodatkową specyfikację architektury sprzętowej i analizę zagadnień szeregowania.

Uwagi polemiczne

1. W rozprawie wprowadzono warstwę systemową określoną skrótem FPPS (*Fixed Priority Preemptive Scheduling*). Jest to najprostszy algorytm szeregowania. Jak bardzo należałoby zmienić warstwę systemową, aby wprowadzić rozszerzenia, np. priorytety dynamiczne? Przyjęta zasada niedokończenie wykonania bieżącej instrukcji wskutek wywłaszczenia danego agenta nie została uzasadniona. Wskazana byłaby szersza dyskusja względem opcji dopuszczającej dokończenie wykonania instrukcji. Jakie konsekwencje ma założenie o stałym okresie zgłaszania przerwania systemowego?
2. Zaproponowane rozwiązanie dla architektury jednoprocesorowej należy traktować jako pierwszy krok. Powstaje pytanie w jakim zakresie konieczne są zmiany tego rozwiązania w celu rozszerzenia funkcjonalności o dowolnej liczbie procesorów?
3. Zaproponowane rozwiązania dotyczą są opisane w rozprawie do momentu wygenerowania struktury LTS, opisującej odpowiedni model. Wyraźnie odczuwa się brak wyjaśnienia dalszych etapów, tzn. w jaki sposób można przeprowadzić analizę/weryfikację i z jakich narzędzi skorzystać.
4. Rozpatrywany przypadek studialny *publikator-subskrybenty* jest ograniczony do jednego subskrybenta, co wydaje się zbyt radykalnym uproszczeniem. Należało uwzględnić przynajmniej dwóch subskrybentów, co mogło również skutkować pewnymi regularnościami w LTS.

5. Wybór sieci Petriego jako alternatywnego formalizmu do porównania (rozdział 7) może być dyskutowany. Wydaje się, że bardziej odpowiedni byłby język LOTOS, szczególnie wersja NT (http://cadp.inria.fr/tutorial/).

4. Wniosek końcowy

Przedstawiona w pracy problematyka dotyczy aktualnych i ważnych zagadnień naukowych współczesnej informatyki, związanych z wykorzystaniem metod formalnych do modelowania i analizy systemów wbudowanych/czasu rzeczywistego. Celem pracy było opracowanie koncepcji, adaptacja formalnego opisu modeli w języku Alvis i implementacja modułów w języku Haskell, które umożliwiłyby weryfikację modeli przeznaczonych do implementacji z środowisku jednoprocesorowym, a projektowanych z użyciem języka Alvis. Postawiony cel został zrealizowany, a użyteczność zaproponowanego rozwiązania zilustrowano przykładami studialnymi. Rozprawa doktorska potwierdza również obszerną wiedzę Autora w zakresie stosowania metod formalnych w inżynierii oprogramowania.

W związku z powyższym stwierdzam, że rozprawa doktorska pt.: *Metody formalnej analizy systemów wbudowanych czasu rzeczywistego* **spełnia wymagania stawiane rozprawom doktorskim** przez odnośne przepisy i wnoszę o **dopuszczenie mgr. inż. Jarosława Baniewicza do dalszych etapów przewodu doktorskiego w dyscyplinie Informatyka**.

Tomasz Szmuc