

RECENZJA ROZPRAWY DOKTORSKIEJ DLA RADY WYDZIAŁU ELEKTRONIKI I TECHNIK INFORMACYJNYCH POLITECHNIKI WARSZAWSKIEJ

Tytuł rozprawy: Automatic generation of robotic system controllers based on a specification

Autor rozprawy: mgr inż. Maksym Figat

1. Analiza strony merytorycznej rozprawy

1.1. Obszar problemowy

Recenzowana rozprawa doktorska dotyczy zagadnień związanych z automatycznym generowaniem sterowników systemów robotycznych na podstawie ich formalnej specyfikacji. Główną osią rozprawy jest poszukiwanie efektywnych metod generowania kodu sterowników robotów (także systemów składających się z wielu robotów) w odniesieniu do sparametryzowanego meta-modelu opisującego ogólny system robotyczny i uwzględniającego jego strukturę oraz sposób działania.

Za główny cel rozprawy Autor obrał opracowanie uniwersalnej metody specyfikowania systemów robotycznych wykorzystującej koncepcję agenta upostaciowionego oraz paradygmat *Model Driven Engineering* (MDE), zakładając, że powstały model systemu może być użyty na różnych etapach rozwoju tego systemu. Istotnym celem praktycznym było uzyskanie metodyki pozwalającej specyfikować systemy poddające się formalnej weryfikacji poprawności, a jednocześnie umożliwiającą automatyczne generowanie szkieletu kodu sterowników.

Autor rozprawy podjął się rozwiązania problemów poruszanych w literaturze z zakresu robotyki od dawna, które jednak jak dotąd nie zostały rozwiązane w sposób powszechnie uznany za satysfakcjonujący, a co za tym idzie, rozwiązania takie nie zostały wdrożone w praktyce. Warto zaznaczyć, że podobne zagadnienia rozważane są w literaturze z zakresu inżynierii oprogramowania w obszarze informatyki, ale dotyczą systemów zazwyczaj mniej złożonych i nie poddanych tak licznym ograniczeniom jak agent upostaciowiony (*embodied agent*), którym jest robot. Uważam więc, że problematyka rozprawy jest istotna i daje możliwość uzyskania nowych wyników naukowych w kontekście aktualnych kierunków badań w zakresie robotyki oraz informatyki.

W rozprawie sformułowano trzy odrębne tezy (str. 29–30). Pierwsza z ich mówi o możliwości opracowania formalnego, sparametryzowanego meta-modelu systemu robotycznego, który umożliwi wygenerowanie modelu opisującego działanie tego systemu. Teza druga dotyczy konieczności podziału specyfikacji systemu na dwie części: ogólną specyfikację struktury systemu i jego działań oraz szczegółową specyfikację funkcji niskopoziomowych elementów systemu w konkretnym języku implementacji. Teza trzecia to stwierdzenie, że możliwe jest opracowanie meta-modelu systemu robotycznego, który poddaje się analizie formalnej.

Tezy te określają cele badań przedstawionych w rozprawie, są jednak na tyle ogólne, że trudno jest zweryfikować osiągnięcie danego celu, odnosząc się do tezy. Autor nie odnosi się bowiem w tezach do jakości uzyskiwanych rozwiązań projektowych, która jest mierzalna poprzez parametry działającego systemu robotycznego. W przypadku tezy trzeciej nie został określony zakres lub rodzaj postulowanej

analizy formalnej. Natomiast teza druga jest raczej pytaniem o właściwy poziom abstrakcji modelu, nie zaś klasyczną tezę naukową, której prawdziwości można dowieść. Wydaje się, że bardziej wartościowe dla czytelnika byłoby przeformułowanie tezy w precyzyjnie opisane cele badawcze, pamiętając, że w obszarze nauk technicznych tezy w klasycznej postaci nie są obowiązkowym elementem rozprawy.

Zdefiniowane cele i zakres podejmowanych badań uzasadniają twierdzenie, że obszar problemowy rozprawy mieści się w dziedzinie robotyki oraz takich działach informatyki jak inżynieria oprogramowania i modelowanie symulacyjne.

Pierwszy rozdział rozprawy stanowi wstęp, zawierający zwięzłe, lecz wystarczające uzasadnienie dla podjęcia tematu badawczego oraz opis stanu wiedzy w zakresie metod projektowania systemów robotycznych. Przedstawiono cztery najpowszechniej stosowane podejścia do tego problemu: bezpośrednią implementację, podejście oparte na komponentach, podejście oparte na modelu i podejście oparte na modelu formalnym, a następnie dość obszernie wnioski jakie Autor wyciągnął na podstawie analizy literatury. Rozdział pierwszy zawiera też tezy pracy wraz z uwagami ułatwiającymi ich interpretację. Rozdział drugi przedstawia opis proponowanej metody tworzenia sterowników systemów robotycznych opartej na koncepcji agentowej, metodyce MDE i zastosowaniu sieci Petriego. W rozdziale tym zdefiniowano podstawowe koncepcje i pojęcia użyte w rozprawie oraz przedstawiono skomplikowaną notację symboliczną elementów proponowanych modeli. Kolejne trzy rozdziały (3, 4 i 5) stanowią zasadniczą część rozprawy i prezentują najważniejsze nowe koncepcje i metody stanowiące oryginalny wkład doktoranta w rozwój metodyki projektowania systemów robotycznych. Dotyczą one koncepcji hierarchicznej sieci Petriego jako meta-modelu systemu robotycznego (*Robotic System Hierarchical Petri Net – RSHPN*, rozdział 3), metod analizy formalnej modeli RSHPN jako sieci Petriego pozwalających określić niektóre ich własności: bezblokadowość, żywotność, konserwatywność (rozdział 4), oraz języka formalnego proponowanego do określania parametrów sparametryzowanego meta-modelu RSHPN (*Robotic System Specification Language – RSSL*, rozdział 5). Rozdział 6 opisuje narzędzia opracowane w ramach prowadzonych badań: kompilator języka RSSL oraz interaktywne narzędzie do edycji modeli RSHPN, a rozdział 7 przedstawia metodę generowania kodu sterownika systemu robotycznego na podstawie modelu RSHPN. Eksperymenty zmierzające do wykazania cech proponowanych metod i modeli oraz użyteczności opracowanych narzędzi zaprezentowane zostały w rozdziale 8, który przedstawia cztery różne scenariusze, dla różnych klas systemów robotycznych. Eksperymenty dobrano tak, aby ich wyniki pozwalały wykazać trafność postawionych w rozprawie tez badawczych. Rozprawę zamyka rozdział 9, prezentujący wnioski z przeprowadzonych badań, podsumowujący oryginalne dokonania Autora i sugerujący kierunki możliwego rozwoju prezentowanej koncepcji.

Rozprawa zawiera reprezentatywną dla poruszanej problematyki bibliografię, która w większości przypadków właściwie ilustruje omawiane zagadnienia oraz dobrze dokumentuje wkład własny doktoranta.

1.2. Ocena wyników oraz stopnia ich oryginalności

Recenzowana rozprawa dotyczy efektywnego projektowania systemów robotycznych, w tym systemów wielu robotów, w sposób zapewniający weryfikację pewnych własności formalnych modelu systemu, takich jak bezblokadowość wymiany informacji i sterowania. Zaprezentowane w rozprawie rozwiązanie tego zagadnienia oparte jest na modelu agentowym dekompozycji systemu oraz zastosowaniu sieci Petriego jako formalizmu opisu logiki działania tego systemu. Zastosowano prostą, podstawową wersję sieci Petriego typu pozycja/przejście (P/T), jednak formalizm opisu został rozbudowany poprzez mapowanie koncepcji z zakresu robotyki na hierarchiczną strukturę sieci. Pozwala to połączyć na poziomie koncepcyjnym i poziomie modelowania systemu robotycznego pewne cechy opisu agentowego oraz modelowania sieciowego, poddającego się analizie formalnej. W świetle znanej literatury dotyczącej architektur systemów robotycznych oraz sposobów ich projektowania i analizy jest to koncepcja oryginalna i dobrze osadzona w analizie stanu wiedzy przedstawionej w rozprawie.

Wśród prezentowanych w rozprawie nowych koncepcji oraz rozwiązań należy wyróżnić.

- Koncepcję meta-modelu systemu robotycznego w postaci RSHPN, który umożliwia kreowanie

sterowników dla systemu robotycznych w sposób niezależny od warstwy sprzętowej i warstwy *middleware*, do pewnego stopnia niezależnie od języka implementacji, oraz dla szerokiej gamy zadań. Cechą wyróżniającą zaproponowanego rozwiązania jest definiowanie działania poszczególnych komponentów systemu poprzez parametryzację ogólnego meta-modelu, a nie poprzez kompozycję operacji składowych.

- Koncepcję rozszerzenia formalizmu sieci P/T do postaci hierarchicznej, odwzorowującej system robotyczny na poziomie struktury i funkcjonalności, co umożliwi zdekomponowanie złożonego modelu systemu na wiele warstw i uzyskanie modularności oraz możliwości parametryzacji meta-modelu. Meta-model jest traktowany jako model zastępczy, a model projektowanego systemu robotycznego uzyskuje się z meta-modelu w wyniku odpowiedniej parametryzacji.
- Opracowanie dziedzinowego języka modelowania RSSL, który pozwala zdefiniować formalny model określający ogólną specyfikację systemu i jego działań, oraz specyfikację implementacji – szczegółową specyfikację przeciążonych funkcji agentów w postaci kodu źródłowego. Elementem nowym jest wskazanie (poprzez odpowiednie uzasadnienie) granicy oddzielającej specyfikację formalną od specyfikacji funkcji niskopoziomowych, co sprzyja redukcji wysiłku projektanta systemu.
- Wykazanie możliwości analizy wybranych własności modelu RSHPN poprzez analizę poszczególnych sieci zawartych w panelach, oraz analizę sieci zawierających wspólne miejsca. Zaproponowana struktura całego modelu, w której istnieją warstwy o trywialnej strukturze oraz strukturze stałej, pozwala ograniczyć analizę własności do sieci definiujących działanie podsystemów oraz komunikację pomiędzy tymi podsystemami.
- Implementacja narzędzi wspomagających proces projektowania systemów robotycznych i generowania kodu: (i) kompilatora RSSL realizującego analizę specyfikacji systemu i meta-modelu RSHPN; (ii) narzędzia służącego do definiowania działań systemu robotycznego w postaci RSHPN, które pozwala projektować sieci Petriego, parametryzować model RSHPN, oraz generować kod C++ interpretujący sieć Petriego albo generować kod sterownika w języku C++ lub Python.

Zauważone niedoskonałości dysertacji dotyczą przede wszystkim powierzchownego opisu niektórych zagadnień, głównie w części opisującej eksperymenty (rozdział 8). Pewne wątpliwości można mieć także co do zakresu uzasadnienia wyciąganych wniosków – w opinii recenzenta niektóre z wniosków są zbyt ogólne względem przedstawionych w rozprawie mierzalnych rezultatów. W poniższym zestawieniu zwracam uwagę na dostrzeżone niedostatki merytoryczne i zadaję pytania, na które oczekuję odpowiedzi doktoranta.

- Wprawdzie rozdział 1 zawiera obszerny przegląd literatury dotyczącej metod projektowania oprogramowania robotów, jednak przegląd ten w niewielkim stopniu porusza dwa ważne aspekty: systemy (wielo)agentowe w robotyce oraz podstawowe paradygmaty architektur oprogramowania robotów (klasyczny/hierarchiczny, behawioralny, hybrydowy). Konsekwencją takiego potraktowania analizy stanu wiedzy wydają się być pewne niedostatki dostrzegalne w dalszej części pracy, na przykład brak jednoznacznej definicji pojęcia agenta (na str. 35 znaleźć można jedynie definicję agenta upostaciowionego) oraz brak porównania struktury kreowanych za pomocą proponowanego formalizmu architektur z powszechnie stosowanymi rozwiązaniami. Jakie wobec tego cechy danego modułu programowego lub programowo-sprzętowego decydują o wyodrębnieniu tego bytu jako agenta w proponowanym systemie i jak mają się one do rozwiązań wcześniej proponowanych w literaturze, np. J. Müller, *The Design of Intelligent Agents: A Layered Approach*, Berlin, Springer 1996? Czy za pomocą proponowanego rozwiązania (język RSSL, model RSHPN) można odtworzyć architekturę typu *sense-plan-act* oraz architekturę uogólniającą (*subsumption*) i czy użycie tegoż rozwiązania będzie miało jakiś wpływ na efektywność finalnego rozwiązania (np. poprzez ograniczenie możliwych sposobów komunikacji modułów)?

- W rozdziale 8 przedstawiono cztery eksperymenty ilustrujące własności proponowanej metody projektowania systemów robotycznych. Jednak tylko dla eksperymentu pierwszego – robota mobilnego zbierającego piłki zademonstrowano analizę własności sieci Petriego, przy czym analiza przedstawiona na str. 165–167 ma charakter interpretacji działania sieci podsystemów sterowania ($\mathcal{H}_{\text{auto,cs}}$ i $\mathcal{H}_{\text{super,cs}}$). Przyjmując za słuszny argument braku konieczności analizy własności sieci położonych wyżej w hierachii, które mają stałą lub bardzo prostą strukturę, można postawić pytanie czy sieci $\mathcal{H}_{\text{auto,cs}}$, $\mathcal{H}_{\text{super,cs}}$ poddają się automatycznej analizie opartej na generowaniu drzew osiągalności, np. za pomocą narzędzia TINA, które okazało się niezdolne do analizy całej sieci (przedstawionej na rys. 8.4)? Czy modele systemów wykorzystujących komunikację z blokowaniem mogą być analizowane w taki sam sposób, czy też powstają dodatkowe trudności?
- Na str. 125 podano informację, że generowany kod w języku Python może wykorzystywać tylko komunikację ROS *topics*, a na str. 189 pojawia się komentarz, że w obecnej fazie rozwoju proponowanego rozwiązania dostępna jest tylko komunikacja *non-blocking* implementowana przez ROS *topics*. Jednak w eksperymentach z robotem Velma użyto także pamięci dzielonej (tab. 8.8) oraz mechanizmu ROS *actions* (tab. 8.13). Czy oznacza to, że mechanizmy te są implementowane w przypadku generowania kodu C++, czy też ich użycie wymagało ręcznej modyfikacji wygenerowanego kodu? Jeżeli konieczne były modyfikacje, to czy zagwarantowane jest w takim przypadku zachowanie formalnych własności modelu (głównie bezblokadości)?
- W strukturze oprogramowania robota zbierającego piłki wygenerowanej na podstawie specyfikacji RSSL (rys. 8.2) wyróżniony został agent $a_{\text{img_proc}}$, podczas gdy druga kamera zamontowana na pokładzie robota obsługiwana jest przez procesy wewnątrz agenta a_{bc} (*Ball collector agent*). Jakie czynniki zadecydowały o przyjęciu takiej struktury i czy mają one związek z przetwarzaniem obrazów z kamery Intel RealSense przez zewnętrzny komputer? Należy zauważyć, że sposób przetwarzania obrazów z kamery podłączonej do Raspberry Pi nie został w rozprawie dokładnie opisany. Piłki na obrazach z Intel RealSense wykrywane są przez sieć neuronową YOLOv3. Obraz przedstawiony na rys. 8.6a sugeruje, że w przypadku drugiej kamery dzieje się podobnie. Czy oznacza to, że obrazy są przesyłane poprzez Raspberry Pi do komputera zewnętrznego?
- W opisie zadania wykreślania okręgu przez manipulator LWR4+ (str. 151) pojawia się uwaga dotycząca wyboru sekwencyjnej organizacji komunikacji między modułami (*pages*) zamiast równoległej organizacji przetwarzania z powodu różnic w częstotliwości odświeżania informacji między tymi modułami. Ponieważ jest to czynnik zależny od implementacji, powstaje pytanie, czy możliwe jest jego przewidzenie na etapie specyfikowania systemu (języka RSSL) jeżeli specyfikacja uważana jest za niezależną od implementacji? Czy możliwe jest iteracyjne poprawianie specyfikacji lub modelu RSHPN po zauważeniu tego rodzaju zależności na poziomie implementacji?

Pomimo braku w recenzowanej dysertacji opisu niektórych szczegółów technicznych zrealizowanych eksperymentów, wykonane badania pozwoliły na jakościową ocenę stopnia spełnienia założeń przez proponowane rozwiązanie, a tym samym osiągnięcie wystarczającego stopnia przekonania o prawdziwości tez rozprawy. Analiza oraz wyniki eksperymentów wskazują na to, że główny cel podjętych badań – opracowanie metody automatycznego generowania programów sterujących systemów robotycznych na podstawie formalnej specyfikacji został osiągnięty.

1.3. Zagadnienia dyskusyjne

Poniższe uwagi dotyczą ogólniejszych kwestii poruszonych w rozprawie i nie odnoszą się bezpośrednio do treści manuskryptu. Oczekuję jednak komentarza doktoranta dotyczącego tych uwag.

1. W jakim stopniu zaproponowane rozwiązanie jest niezależne od użytej na poziomie implementacji platformy programowo-sprzętowej (zarówno *middleware* jak i jednostek obliczeniowych)? Czy ograniczenia i specyfika jednostek klasy *edge computing*, szczególnie zawierających GPGPU,

które są coraz częściej stosowane w robotyce wpłyną na architekturę projektowanego systemu i czy mogą być uwzględnione w specyfikacji RSSL?

2. Proszę o skomentowanie skalowalności mechanizmów komunikacyjnych występujących w proponowanym rozwiązaniu w funkcji liczby agentów systemu wieloagentowego. Czy istnieje możliwość blokady komunikacji na skutek nasycenia fizycznych kanałów komunikacji w systemie o bardzo dużej liczbie agenów (np. system rozproszonych sensorów)? Czy możliwa jest rekonfiguracja logiki przesyłania informacji w takiej sytuacji lub w przypadku utraty danego kanału komunikacyjnego?

2. Analiza strony formalnej rozprawy

2.1. Ocena układu pracy i redakcji manuskryptu

Recenzowana rozprawa doktorska została przygotowana w języku angielskim. Obejmuje ona w kolejności: streszczenie w języku angielskim, streszczenie w języku polskim, spis treści, dziewięć rozdziałów zasadniczych, a także bibliografię i załącznik, zawierający formalny opis języka RSSL w postaci diagramów syntaktyki oraz tabel w formacie parsera LARL. Rozprawa liczy 212 ponumerowanych stron.

Bibliografia składa się ze 164 pozycji uporządkowanych w kolejności cytowania. W rozprawie cytowanych jest 11 współautorskich prac doktoranta, w tym pięć opublikowanych w czasopismach, oraz jeden odnośnik do publicznego repozytorium GitHub z oprogramowaniem jego autorstwa. Spośród cytowanych prac 35 (ponad 20%) ukazało się w okresie ostatnich pięciu lat (po roku 2016), co świadczy o dość dobrym umiejscowieniu tematyki rozprawy w aktualnym nurcie badań. Bibliografia nie budzi zastrzeżeń od strony merytorycznej, a jej redakcja jest staranna. Zauważyć można jedynie pojedyncze błędy redakcyjne, np. [25], [97] lub niekompletne dane cytowanej pozycji [34].

Układ rozprawy jest prawidłowy, jest ona także starannie opracowana pod względem edytorskim i graficznym. Użyta terminologia (w języku angielskim) jest właściwa dla obszaru problemowego rozprawy, zarówno w zakresie robotyki, jak również informatyki (inżynierii oprogramowania, modelowania symulacyjnego). Tekst rozprawy jest poprawny pod względem językowym i stylistycznym. Podczas lektury manuskryptu zauważyłem jedynie pewne niepotrzebne powtórzenia oraz nieliczne błędy redakcyjne np.: "Considering that and the fulfillment..." (str. 166, w. 2 od dołu). Lekturę manuskryptu, w którym rolę kluczową dla zrozumienia opisywanych rozwiązań odgrywają bardzo rozbudowane symbole, utrudnia brak listy tych symboli, na przykład w postaci dodatku (*Appendix*).

3. Konkluzja

Uważam, że recenzowana dysertacja Pana mgr inż. Maksyma Figata spełnia wymogi stawiane rozprawom doktorskim przez art. 187 ustawy z dnia 20 lipca 2018 r. – Prawo o szkolnictwie wyższym i nauce, ponieważ zawiera oryginalną koncepcję rozwiązania istotnych problemów w dziedzinie robotyki oraz informatyki. Rezultaty nawiązują do aktualnego stanu wiedzy i mają potencjalne znaczenie praktyczne. Uzyskane wyniki w wystarczającym stopniu dokumentują poprawność proponowanych koncepcji i opracowanych narzędzi oraz skuteczność działania ich implementacji. Uwagi krytyczne sformułowane w treści recenzji, po części mające charakter dyskusyjny, nie umniejszają mojej pozytywnej oceny oryginalności i wysokiego poziomu merytorycznego rozprawy.

Sformułowanie problemu badawczego, zaproponowane oryginalne metody jego rozwiązania, sposób przeprowadzenia badań oraz zademonstrowana umiejętność formułowania wniosków świadczą o dobrym przygotowaniu doktoranta do pracy naukowej. Na tej podstawie wnioskuję o dopuszczenie mgr inż. Maksyma Figata do publicznej obrony. Biorąc pod uwagę oryginalność zaprezentowanych w rozprawie wyników oraz ich pozytywny odbiór w międzynarodowym środowisku naukowym, potwierdzony

publikacjami Autora, wnoszę o rozważenie przez Radę Dyscypliny wyróżnienia rozprawy doktorskiej M. Figata.


Piotr Skrzypczyński