

## RECENZJA

rozprawy doktorskiej mgr inż. Maksyma Figata pt.

Automatic generation of robotic system controllers based on a specification

### 1 Temat i zakres rozprawy

Recenzowana praca ma objętość 212 stron i składa się z dziewięciu rozdziałów, wykazu cytowanej literatury obejmującego 164 pozycje oraz dodatku.

Problem rozprawy związany jest z automatyzacją konstrukcji sterowników systemów robotycznych. Proponowane podejście zakłada opracowanie ogólnego, opisanego w sposób parametryczny modelu takich systemów, co pozwala na automatyczną syntezę modeli o konkretnych parametrach, a następnie ich wykorzystanie w generacji kodu sterownika.

Przedstawiony dokument stanowi kompletną i logiczną całość. W oparciu o charakterystykę literatury tematu, Autor przedstawił we Wstępie motywację do podjęcia tej pracy oraz określił jej cel i zakres. Rozdział 2 opisuje zaczerpnięty z literatury model agentowy systemu robotycznego, który definiując ogólną architekturę i zasady działania takich systemów stanowi punkt wyjścia dla dalszych rozważań. Rozdział 3 poświęcony jest konstrukcji i parametryzacji klasy sieci Petriego - meta-modelu systemu robotycznego implementującego postulat systemu agentowego. Rozdział 4 skupia się na analizie wybranych własności meta-modelu, wykazując poprawność jego działania. Rozdział 5, uzupełniony dodatkiem, wprowadza język specyfikacji systemów robotycznych, ułatwiający użytkownikowi zdefiniowanie projektowanego systemu, a jednocześnie określenie danych dla automatycznej syntezy jego modelu sieciowego. Rozdziały 6 i 7 omawiają software przeznaczony do, odpowiednio, generacji powyższego modelu oraz (wspomaganej przez projektanta) generacji kodu sterownika. Opracowana w pracy metodologia jest zilustrowana przykładami w Rozdziale 8. Podsumowanie rozprawy zawiera wnioski z przeprowadzonych badań, listę osiągnięć, które Autor uważa za istotne, oraz propozycje dalszych badań.

### 2 Ocena ogólna

Wspomaganie konstrukcji oprogramowania przy użyciu sieci Petriego jest od kilku dziesiątek lat intensywnie eksploatowaną dziedziną badań w zastosowaniu do systemów operacyjnych komputerów, systemów komunikacji, systemów zarządzania, elastycznych systemów produkcyjnych, systemów transportowych i wielu innych komputerowo sterowanych systemów automatyki. Celem tych badań jest uzyskanie modeli podatnych na analizę formalną i testy symulacyjne oraz opracowanie struktur sieciowych i algorytmów zapewniających określone cechy jakościowe i/lub ilościowe działania modelowanych obiektów. Uzyskane wyniki określają pożądaną logikę programów sterowania takimi obiektami i stanowią podstawę syntezy ich kodu.

W powyższą dziedzinę bardzo dobrze wpisuje się tematyka recenzowanej pracy, różniąc się jednocześnie od większości tego typu badań pod względem złożoności modelowanych systemów oraz charakteru projektowanego sterowania. W kontekście tworzenia meta-modeli, w literaturze rozważane są na ogół znacznie prostsze strukturalnie systemy i reprezentujące je sieci Petriego, a problem badań wynika z asynchronicznego działania obiektów takich systemów i poszukiwania odpowiedniego sterowania nadrzędnego o charakterze zdarzeniowym. Np. rozpatrywane modele elastycznych systemów produkcyjnych mają stosunkowo prostą strukturę, która odzwierciedla współbieżny przepływ produkowanych detali przez stacje robocze w kolejności zadanej marszrutami technologicznymi poszczególnych typów detali. Rozważane tu zagadnienia dotyczą zapewnienia poprawnej i efektywnej pracy takich systemów poprzez modyfikację pierwotnej struktury modeli i/lub uzupełnienie ich o algorytmy ograniczające dopuszczalną kolejność palenia tranzycji. Ze względu na asynchronizm modelowanych procesów problemy poprawności wymagają użycia sieci Petriego w ich podstawowej formie, tj. sieci bezczasowych. Natomiast kryteria efektywności mają na ogół charakter czasowy i w poszukiwaniu tego typu rozwiązań konieczne jest uwzględnienie danych czasowych i wykorzystywanie jako modeli czasowych lub stochastycznych sieci Petriego.

W odróżnieniu od powyższych prac, główny wysiłek badawczy recenzowanej rozprawy skupia się na opracowaniu metodologii syntezy sieciowych modeli systemów robotycznych, a relatywnie mniejszym problemem jest nadzór nad ich zachowaniem. Taka proporcja jest zgodna z założeniem konstrukcji modeli sieciowych w oparciu o modele agentowe, które z jednej strony cechuje wysoka złożoność struktury, a z drugiej dobrze określone działanie. Także przyjęty sposób rozwiązywania sytuacji konfliktowych poprzez priorytetowanie tranzycji ogranicza przestrzeń możliwych zachowań systemu, przez co zapewnienie poprawności sterowania nie wymaga dodatkowych algorytmów nadzoru. Natomiast można by się zastanowić, czy i jak na efektywność działania projektowanych systemów robotycznych wpływa przydział tranzycjom konkretnych wartości priorytetów.

Wynikiem pracy są oryginalne, będące dziełem ich Autora, metodologia i narzędzia syntezy oprogramowania sterowników systemów robotycznych, które w zasadniczym stopniu wspomagają i automatyzują cały proces konstrukcyjny. Model agentowy i oparty na nim meta-model sieciowy dostarczają wzorca, a język RSLL środka jego specyfikacji, co sprowadza koncepcyjną część projektu do określenia parametrów projektowanego systemu. Dla zadanych parametrów automatycznie generowany jest model sieciowy, który określa główne elementy działania sterownika i jest podstawą większości automatycznie generowanego programu sterownika. Pełny kod jest wynikiem połączenia powyższego kodu wraz z kodem określającym specyficzne działanie rozważanego systemu, dostarczonym przez projektanta. Przydatność powyższego podejścia została bardzo dobrze zademonstrowana na czterech przykładach w Rozdziale 8, Eksperymenty.

Wyżej omówiony aspekt rozprawy oceniam bardzo wysoko. Uważam, że osiągnięte tu wyniki zawierają wystarczającą liczbę elementów oryginalnych, aby mogły stanowić podstawę do nadania tytułu doktora nauk technicznych. Natomiast widzę pewne niedociągnięcia związane z teoretyczną częścią rozprawy, formalnym zdefiniowaniem modeli sieciowych i ich analizą, które przedstawiam w następnej części recenzji. Należy jednak podkreślić, że zastrzeżenia te nie umniejszają ogólnej jakości i przydatności praktycznej opracowanej koncepcji.

## 3 Uwagi krytyczne

### 3.1 Podejście ogólne

Pierwsza uwaga dotyczy ogólnej metodologii pracy, która zakłada konstrukcję meta-modelu systemu robotycznego RSHPN przy użyciu koncepcji agenta upostaciowionego EAA i hierarchicznych sieci Petriego HPN, a następnie parametryzację RSHPN. W pracy brakuje bardziej precyzyjnego określenia związku tych trzech modeli, przy czym wydaje się, że każdy system robotyczny, który można rozważać w ramach modelu EAA, może być jednoznacznie przedstawiony w postaci systemu RSHPN. A zatem, z formalnego punktu widzenia postawione zadanie należałoby sprowadzić do określenia odwzorowania modeli agentowych w hierarchiczne sieci Petriego  $map : EAA \rightarrow HPN$ , czyli zdefiniowania transformacji określonej dla dowolnego modelu  $EAA_p \in EAA$  do konkretnej sieci Petriego  $HPN_p = map(EAA_p) \in HPN$ . Wtedy klasa modeli RSHPN jest przeciwdziedzina odwzorowania  $map$ ,  $RSHPN = \{map(EAA_p) : EAA_p \in EAA\}$ . Rozdział 3 opisuje (nieformalnie) taką transformację, jednak nie jest dla mnie jasne, dlaczego następnie przeprowadza się parametryzację RSHPN. Koncepcyjnie pierwotnym źródłem dla syntezy sterownika są modele EAA, a nie modele RSHPN. Co więcej, są już one sparametryzowane przez swoją notację i, moim zdaniem, to na nich powinien się opierać język RSSL i specyfikacja systemów robotycznych. Parametry EAA przenoszą się w trakcie transformacji na modele RSHPN i nie bardzo widzę cel wprowadzania odrębnej parametryzacji.

### 3.2 Modele sieciowe

Przedstawiona w rozprawie konstrukcja modeli sieciowych polega na konstrukcji nowej klasy sieci Petriego nazwanej Hierarchicznymi Sieciami Petriego, HPN, oraz jej użycia do budowy modeli sieciowych RSHPN - Hierarchicznych Sieci Petriego Systemów Robotycznych. Poniżej przedstawiam moje zastrzeżenia dotyczące przedstawionego podejścia.

Formalne określenie klasy sieci Petriego zastosowanych do konstrukcji modeli rozważanych systemów robotycznych jest bardzo istotne. Jest to konieczne aby precyzyjnie określić odwzorowanie systemu robotycznego w postaci EAA w jego model sieciowy i umożliwić w ten sposób jego automatyczną syntezę. Co więcej, formalny charakter takiego odwzorowania pozwala wykazać poprawność modeli dla całej dziedziny modeli EAA, tzn. formalnie zapewnić, że każdy konstruowany w ten sposób model  $HPN_p = map(EAA_p) \in HPN$  będzie posiadał oczekiwane własności. W tym sensie zaproponowane podejście polegające na określeniu klasy HPN jest bardzo cenne, ale niestety nie do końca udane.

Po pierwsze, klasa sieci HPN przedstawiona jest w sposób opisowy, a brakuje jej kompletnej i precyzyjnej definicji. Sieci Petriego są obiektami abstrakcyjnymi matematyki dyskretnej i wprowadzając nową ich klasę należy w sposób formalny zdefiniować ich strukturę i działanie, tak jak to ma miejsce w literaturze dotyczącej tej tematyki, np. w klasycznych już pozycjach [3] czy [2]. Definicja struktury, najczęściej w postaci n-tki, zawiera graf będący podstawą sieci (zbiory miejsc, tranzycji i łączącą je relację) oraz wszystkie dodatkowe elementy potrzebne do opisu dynamiki sieci. Np. w przypadku sieci P/T (miejsc i tranzycji) dodatkowym elementem jest funkcja wagi łuków, a w przypadku sieci kolorowanych wprowadza się zbiór kolorów oraz funkcje incydencji Pre i Post określone dla trojek (miejsce, tranzycja, kolor). Po zdefiniowaniu struktury systemu sieciowego na-

leży formalnie określić: 1) sposób reprezentacji stanu sieci, 2) regułę (funkcję) pozwalającą określić przygotowanie danej tranzycji (możliwość jej odpalenia) w danym stanie sieci), 3) regułę (funkcję) pozwalającą określić zmianę stanu sieci spowodowaną odpaleniem danej tranzycji w danym stanie. W pracy brakuje takiego formalnego wprowadzenia HPN.

Po drugie, podchodząc do konstrukcji nowej klasy sieci Petriego należy najpierw określić jej wyróżniki i umiejscowić ją pod względem pożądanych cech w istniejącej klasyfikacji, co zupełnie nie zostało zrobione. Podstawową cechą odróżniającą od siebie typy sieci Petriego jest abstrakcja ich działania i wynikający z niej podział na trzy typy - bezczasowe sieci Petriego, deterministyczne czasowe sieci Petriego i stochastyczne sieci Petriego. Z Rozdziału 3.1.1 wynika, że możliwość odpalenia tranzycji w HPN zależy nie tylko od markowania sieci, ale również od czasu. Wskazuje to na konieczność użycia czasowych sieci Petriego (ang. Timed Petri nets lub Time Petri nets) omawianych np. w [5] i określenia w sposób formalny reprezentacji ich stanu i reguł rządzących jego zmianami. Dodatkowo, jak założono w pracy, oprócz markowania i czasu o możliwości odpalenia tranzycji decydują pewne warunki. Formalne zdefiniowanie struktury i działania takich sieci wymaga określenia klasy używanych predykatów i powiązania ich wartości ze stanem sieci. W pracy to zagadnienie nie jest poruszane i nie jest jasne, jakie warunki są uwzględniane, ani jak są one wartościowane. W literaturze rozważane są różne klasy predykatowych sieci Petriego, np. [4] czy [1], przy czym są to sieci bezczasowe i pozwalają na określanie skomplikowanych predykatów. Interesujące byłoby ograniczenie tej ostatniej cechy do potrzeb modelowania rozważanych systemów robotycznych, a jednocześnie integracja w tym modelu elementu czasu.

Kolejnym problemem jest sama koncepcja klasy HPN. W pierwszym zdaniu Rozdziału 3 Autor rozprawy słusznie zakłada dwudzielność takich sieci, ponieważ jest to immanentna cecha wszystkich sieci Petriego, właściwa z ich natury. Jednak nie mogę zgodzić się z kolejnym zdaniem - założeniem o trzech typach wierzchołków sieci HPN - tranzycji, miejsc i stron. Równie generyczną cechą sieci Petriego jak poprzednia jest posiadanie dwóch typów wierzchołków - miejsc, których markowanie określa stan sieci, i tranzycji, których odpalenie powoduje zmiany tego markowania. Wprowadzenie trzeciego typu wierzchołków jest niezgodne z pojęciem ogólnym sieci Petriego. Możliwe jest natomiast oczywiście założenie, że zbiór miejsc składa się z dwóch podzbiorów, np. miejsc rozwijalnych (stron) i miejsc nierozwijalnych. Nie jest to jednak żadnym wyróżnikiem klasy sieci, bo zarówno podział zbioru miejsc na dwa podzbiory, jak i operację rozwijania miejsc można określić dla dowolnej sieci i dowolnych jej miejsc. W literaturze znane są operacje zwijania i rozwijania sieci Petriego (ang. folding/unfolding), tj. zastępowania fragmentów sieci pojedynczym miejscem lub zastępowania pojedynczego miejsca fragmentem sieci. Operacje te można zastosować do dowolnej klasy sieci Petriego, a sieci będące wynikiem takiej operacji mogą mieć taką samą strukturę ogólną i funkcjonować dokładnie wg tych samych reguł co sieci pierwotne. Ponieważ w ogólności zwinąć/rozwinąć można dowolny fragment każdej sieci i powtarzać operacje zwijania/rozwijania wielokrotnie, na kolejno otrzymywanych sieciach, nie ma uzasadnienia wprowadzenie nowej klasy (Hierarchical Petri Nets) w oparciu o to, że występują w niej miejsca, które można rozwinąć. Wyróżnienie takich miejsc wynika z semantyki modelu, a nie decyduje o klasie sieci i powinno być rozważane na etapie definicji sieci RSHPN, a nie klasy sieci użytych do ich modelowania (tu HPN).

Idąc dalej, zaskakująca jest uwaga dotycząca własności bezpieczeństwa sieci HPN "It

*should be noted that here only safe nets are considered, thus in each place at the most a single token may reside*". Jaka jest tego podstawa? Bezpieczeństwo określonej klasy sieci Petriego nie może być arbitralnym założeniem, ale musi wynikać z postulowanej struktury i zasad działania sieci należących do tej klasy. Taką własność należy formalnie wykazać, podać dowód, że dla dowolnej sieci tej klasy, w każdym markowaniu osiągalnym z markowania początkowego liczba znaczników w dowolnym miejscu nie przekracza 1. Rozprawa nie dostarcza wystarczającej informacji o klasie HPN, z której można by wysnuć taki wniosek, a tym bardziej dowodu własności bezpieczeństwa.

Inne problematyczne stwierdzenie to reguła przygotowania tranzycji w sieciach HPN *"If the number of tokens residing in input places of a transition is equal to or greater than the number of directed arcs connecting those places with the transition then the transition is called enabled*. Warunek ten mówi, że tranzycja jest przygotowana, jeżeli suma znaczników w jej miejscach wejściowych jest większa/równa od liczby/liczbie łuków łączących te miejsca z tranzycją. Takie sformułowanie jest kontrowersyjne, bo może się zdarzyć, że tranzycja jest przygotowana w sytuacji, gdy jakieś miejsce ma dwa znaczniki, a inne nie jest w ogóle markowane. Jest to sprzeczne z duchem sieci Petriego i wydaje się też, że nie o to Autorowi pracy chodziło. Sytuacja taka nie wystąpi w sieciach bezpiecznych, ale postulowany wyróżnik klasy HPN - hierarchiczność sieci - nie implikuje takiej własności.

Ponadto definicja powyższa jest niepełna. Zgodnie z założeniem pracy, sieci HPN oprócz miejsc posiadają jeszcze odrębny typ wierzchołków nazwanych 'stronami' ('pages'), które też mogą wystąpić na wejściu danej tranzycji. W rozprawie brakuje informacji o tym, jak określa się markowanie stron, jak wpływa ono na przygotowanie tranzycji i jak zmienia się to markowanie pod wpływem palenia tranzycji. Autor pracy prawdopodobnie założył, że strony są markowane tak jak miejsca, a markowanie miejsc i jego zmiany są takie jak w najczęściej rozważanych sieciach P/T (miejsc i przejść), w związku z czym pominął to milczeniem. Tymczasem wprowadzając nową klasę sieci należy kompletnie określić zachowanie jej komponentów.

### 3.3 Analiza własności modelu RSHPN

Analiza własności modelu RSHPN jest przedstawiona w Rozdziale 4. Jego pierwszy akapit sugeruje, że zamierzeniem Autora było tu pokazanie szczególnej przydatności modeli RSHPN dla projektantów sterowników ze względu na możliwość weryfikacji ich projektów jeszcze na etapie konstrukcji modelu przy użyciu typowych metod analizy sieci Petriego. Tymczasem, w sytuacji automatycznej syntezy sterowników, jaka jest celem tej pracy, generowane modele nie będą poddawane analizie, ale to sposób ich generacji ma zapewnić automatycznie ich poprawność. Czyli celem tego rozdziału powinno być nie tyle pokazanie, że konkretne modele RSHPN można analizować metodami sieci Petriego, co wykazanie, że opracowana transformacja modeli EAA w modele RSHPN zachowuje istotne założenia dotyczące działania EAA. Gdyby odwzorowanie  $EAA \Rightarrow HPN$  było przedstawione bardziej formalnie, to można by nawet sformułować takie twierdzenie jako centralne twierdzenie pracy i przedstawić jego formalny dowód.

Dobrane przez Doktoranta własności bezpieczeństwa i konserwatywności sieci oraz braku zakleszczeń (ang. deadlock) dobrze charakteryzują poprawność zachowania przez modele RSHPN wybranych założeń modeli EAA. Natomiast omawiane metody analizy są mało użyteczne dla celów weryfikacji posiadania wymienionych własności, ponieważ mogą być

zastosowane jedynie do analizy pojedynczych modeli, a nie całej ich klasy. Wykazanie, że każdy model RSHPN posiada te własności może być tylko dokonane drogą logicznego rozumowania. De facto, właśnie w ten sposób poprawnie pokazuje Doktorant posiadanie rozważanych własności przez poszczególne warstwy ogólnego modelu RSHPN oprócz warstwy akcji (Rozdział 4.1.6). W tym ostatnim przypadku zastosowana jest analiza oparta na inwariantach, która obarczona jest tu podwójnym błędem metodologicznym. Po pierwsze, sieci reprezentujące akcje mogą mieć bardzo skomplikowaną strukturę, a analizowany jest tylko bardzo prosty przypadek. Nie jest pokazane, jak można by uogólnić uzyskany wynik dla dowolnych struktur akcji, zwłaszcza że nie jest rozważana żadna ich struktura ogólna. Po drugie, stosowana algebra jest określona dla sieci Petriego typu P/T (miejsc i tranzycji), do której nie należą sieci RHSPN ze względu na swój charakter czasowy i użycie dodatkowych warunków palenia tranzycji. Być może analizę inwariantową można by dostosować do sieci RSHPN, ale to należałoby wcześniej pokazać i formalnie uzasadnić.

Zaproponowana metoda zapobiegania zakleszczeniom jest poprawna, ale należy zwrócić uwagę, że występowanie zakleszczeń jest tu skutkiem przyjęcia bardzo prostego protokołu transmisji, gdzie nadajnik zawiesza działanie do momentu otrzymania potwierdzenia wcześniej wysłanej wiadomości. Wystarczyłoby przyjąć niewiele bardziej skomplikowany protokół, w którym ta sama wiadomość jest powtarzana z tym samym dodatkowym bitem do momentu uzyskania potwierdzenia odbioru (tzw. alternate bit protocol), aby nie tylko uniemożliwić występowanie zakleszczeń, ale i zagwarantować bezstratną transmisję.

Z drugiej strony, ponieważ w rozprawie rozważane są również systemy wielorobotyczne, zakleszczenia mogą powstawać nie tylko na etapie komunikacji robotów, ale wynikać także z innych względów, np. z dzielenia przez roboty wspólnej przestrzeni ruchu. Nie jest jasne, jak ten aspekt jest uwzględniony w konstrukcji modeli RSHPN.

### 3.4 Użyte nazewnictwo

Intencja Autora wprowadzenia dekompozycji funkcji tranzycji (wzór 2.2) jest zrozumiała, ale użyta nazwa 'funkcje częściowe' nie jest formalnie poprawna. W matematyce funkcja częściowa jest to funkcja, która jest zdefiniowana tylko dla części elementów swojej dziedziny, a dla pozostałych elementów nie jest zdefiniowana. Np. w sieciach Petriego funkcja następnego markowania  $M' = f(M, t)$  jest funkcją częściową, ponieważ jest określona tylko dla par  $(M, t)$  takich, że tranzycja  $t$  jest przygotowana w markowaniu  $M$ . Dla par  $(M, t)$  takich, że tranzycja  $t$  nie jest przygotowana w  $M$ , funkcja  $f(M, t)$  nie jest określona. Tymczasem w rozprawie pojęcie 'funkcja częściowa' zostało użyte nie dla wyróżnienia pewnych argumentów, ale dla rozdziału wartości funkcji tranzycji.

## 4 Wniosek końcowy

Uważam, że praca mgr inż. Maksyma Figata wnosi istotny przyczynek do automatyzacji procesu projektowania sterowników systemów robotycznych w oparciu o Inżynierię Opartą na Modelu (ang. 'Model Driven Engineering'), a ogólniej do rozwoju dyscypliny Automatyka, Elektronika i Elektrotechnika. Spełnia ona zatem wymagania stawiane rozprawom doktorskim przez aktualnie obowiązującą Ustawę o stopniach i tytule naukowym. W związku z tym wnioskuje o dopuszczenie tej rozprawy do publicznej obrony.

## Literatura

- [1] H. Genrich. Predicate / transition nets. In *High-level Petri Nets: Theory and Application*. Springer, 1991.
- [2] K. Jensen and G. Rozenberg, editors. *High-level Petri Nets: Theory and Application*. Springer, 1991.
- [3] W. Reisig. *Petri Nets. An Introduction*. Springer, 1985.
- [4] W. Reisig. Predicate/event nets. In *Petri Nets. An Introduction*, chapter 8. Springer, 1985.
- [5] J.R. Silva and P.M. del Foyo. Timed Petri Nets. In *Petri Nets*, chapter 16. IntechOpen, 2012.

Elżbieta Roszkowska

