

Michał DOLIGALSKI
 UNIwersYTET ZIELONOGÓRSKI

Konwersja wybranych elementów maszyny stanów UML w ramach dualnej specyfikacji

mgr inż. Michał Doligalski

Absolwent Uniwersytetu Zielonogórskiego (2006). Ukończył studia o specjalności Inżynieria Komputerowa. Asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. Prace badawcze prowadzone w Instytucie Informatyki i Elektroniki skupiają się przede wszystkim na nowoczesnych metodach projektowania sterowników logicznych. Członek Polskiego Towarzystwa Informatycznego.

e-mail: M.Doligalski@iie.uz.zgora.pl



Streszczenie

Powszechnie stosowanym rozwiązaniem w dziedzinie sterowania jest zastosowanie sterowników rekonfigurowalnych i programowalnych. Najczęściej stosowane są układy FPGA bądź przemysłowe sterowniki PLC. Rozwój techniki pociąga za sobą gwałtowny wzrost pojemności jak i funkcjonalności (częściowa rekonfiguracja, wbudowane mikroprocesory) układów reprogramowalnych FPGA. Rozwojowi temu nie towarzyszy w zadawalającym stopniu rozwój technik modelowania, pozwalających w pełni wykorzystać zalety aktualnie produkowanych układów FPGA.

W artykule przedstawiono metodę dywersyfikacji specyfikacji behawioralnej jako odpowiedź na zapotrzebowanie rynku na nowe techniki modelowania. Przedstawiono model dualny w postaci maszyny stanów UML oraz sieci Petriego sterowania. Zastosowanie modelu dualnego pozwala na zwiększenie jakości oraz niezawodności sterownika logicznego, skraca czas projektowania. Maszyna stanów UML pełni rolę interfejsu użytkownika, izomorficzna do niej hierarchiczna sieć Petriego bazę do weryfikacji formalnej i implementacji układowej. Specyfikacja behawioralna w postaci modelu dualnego została zaprezentowana w kontekście projektowania sterowników logicznych.

Słowa kluczowe: UML, maszyna stanów, hierarchiczna sieć Petriego, specyfikacja dualna

UML state machine conversion in field of dual specification

Abstract

The application of logic controllers is a usual solution used in the field of discrete development. FPGA systems or PLCs are the most frequently used solution. In the paper method of the specification diversification was proposed. We can observe rapid capacity (CLB) and functionality grow (partial reconfiguration, microprocessor included) of FPGA devices. That grow not entail developing of modeling techniques, that take advantage of FPGA devices. Also dual model based on hierarchical Petri net and UML state machine diagram was presented as a response for market demand for new modeling techniques. The application of dual model make possible quality grow and controller developing time short. The UML state machine diagram act as front-end and according to UML SM diagram Petri net as back-end – base for formal verification and implementation. The dual specification was presented in context of logical controller developing process.

Keywords: UML, state machine, Petri net, model, dual specification

1. Wstęp

Język UML jest powszechnie stosowanym narzędziem do specyfikacji różnorodnych systemów – zarówno programowych jak i sprzętowych [4]. Powszechność stosowania wynika z uniwersalności standardu oraz nieustannej ewolucji standardu. Język ten ewoluje od początku lat dziewięćdziesiątych XX wieku. W chwili obecnej obowiązującym standardem jest wersja UML 2.1. W zbiorze trzynastu diagramów UML można dokonać podziału na dwa podzbiory: diagramów modelujących strukturę oraz diagramów modelujących dynamikę. Do tej grupy zalicza się między innymi diagramy maszyny stanów, które z powodzeniem można wykorzystać w procesie projektowania i specyfikacji sterowników

cyfrowych. Pozwalają one na przyjazną dla użytkownika specyfikację hierarchicznej współbieżnej maszyny stanów. Model ten jest transparentny, pozwala na przyjazne dla użytkownika wprowadzanie poprawek. Nie jest on jednak pozbawiony wad. Metody formalnej weryfikacji diagramów maszyny stanów są słabo rozwinięte. Uniwersalność modelu jest niewątpliwie jego zaletą, jednak wpływa na dużą liczbę elementów składowych które powodują pewną nadmiarowość. Może to doprowadzić do sytuacji, w której stworzony model zamiast transparentności cechuje duża złożoność utrudniająca analizę. Należy więc określić podzbiór elementów diagramu maszyny stanów, który byłby wystarczający do specyfikacji sterowników logicznych. Należy również doprecyzować semantykę modelu, gdyż w kontekście projektowania sterowników jest nieprecyzyjna.

Popularność diagramów UML w środowisku inżynierskim, uzasadnia upowszechnienie stosowania diagramów maszyny stanów UML jako modelu systemów sterowania binarnego. Obecnie prowadzone są badania nad możliwością wykorzystania diagramów maszyny stanów UML w procesie projektowania sterowników logicznych, brak jest jednak dostatecznie dobrych rozwiązań, które pozwoliłyby na wykorzystanie modelu w procesie projektowania sterowników logicznych.

Modelem bardziej ukierunkowanym na specyfikację systemów sterowania są sieci Petriego [1,2,3]. Pozwalają one na modelowanie zachowań współbieżnych. Udostępniają również mechanizmy pozwalające na tworzenie hierarchicznych modeli.

Zaletą sieci Petriego jest bogaty aparat matematyczny pozwalający na analizę formalną (badanie żywotności, ograniczoności). Powstało również wiele narzędzi umożliwiających modelowanie i analizę formalną – sieci Petriego są więc doskonałym narzędziem do modelowania systemów o podwyższonym bezpieczeństwie. Sieci Petriego są powszechnie znane w środowisku akademickim [12], jednak nie cieszą się już tak dużą popularnością w grupie inżynierów. Powstało wiele klas oraz rozszerzeń sieci Petriego. Poszczególne odmiany sieci zostały opracowywane pod kątem konkretnych zastosowań. Powstają również nowe typy modeli będących połączeniem sieci Petriego i maszyny stanów UML [6,10], standardy te cieszą się jednak jeszcze mniejszą popularnością.

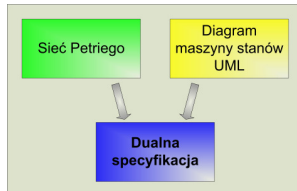
W chwili obecnej dynamicznie rozwija się technologia półprzewodnikowa – rosnące rozmiary i możliwości układów FPGA – pociąga za sobą konieczność opracowania nowych metod modelowania i specyfikacji, które pozwalałyby na pełne wykorzystanie możliwości, które dają obecnie produkowane układy FPGA. Przykładem jest rodzina układów FPGA Virtex-5 firmy Xilinx posiadająca wbudowane mikrokontrolery PowerPC. Język UML może być zastosowany w procesie inżynierii oprogramowania dla tych procesorów. Korzystnym wydaje się zastosowanie jednego modelu do opisu zarówno sterownika logicznego implementowanego w układzie FPGA jak i oprogramowania przeznaczonego dla procesora wbudowanego w ten sam układ.

W chwili obecnej pojedynczy model zarówno w postaci sieci Petriego jak i maszyny stanów UML nie zaspokaja w pełni potrzeb projektowania sterowników logicznych. Rozwiązaniem jest dywersyfikacja specyfikacji behawioralnej sterownika logicznego – model dualny będący połączeniem sieci Petriego i maszyny stanów UML.

2. Model dualny

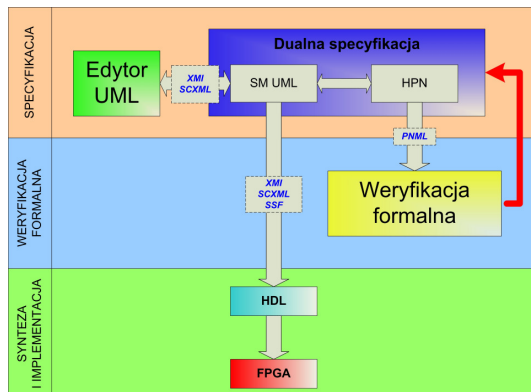
Dywersyfikacja specyfikacji polega na zróżnicowaniu opisu sterownika poprzez zastosowanie połączonych, komplementarnych modeli PN i UML-SM.

Dualna specyfikacja jest abstrakcyjną formą opisu hierarchicznego współbieżnego automatu stanu [6]. Modele wchodzące w skład dualnej specyfikacji przyjmują formę diagramów maszyn stanów UML i sieci Petriego, bądź plików opartych na standardzie XML. Są to odpowiednio pliki w formacie SCXML (ang. *State Chart XML*), XMI (ang. *XML Metadata Interchange*) dla maszyny stanów UML oraz PNML (ang. *Petri Net Markup Language*) dla sieci Petriego. Nadrzędną ideą dualnej specyfikacji jest rygorystyczne przestrzeganie standardów wymiany danych. Pozwala to na wykorzystanie istniejących już narzędzi modelowania jak również syntezy układowej.



Rys. 1. Koncepcja dualnej specyfikacji
Fig. 1. Conception of dual specification

Zalety obu modeli wykorzystane są w pełni w specyfikacji dualnej, kompensując wady poszczególnych modeli. Model maszyny stanów UML pełni funkcję front-endu, natomiast hierarchiczna sieć Petriego back-endu. Wady diagramów stanów, takie jak słabo rozwinięte metody weryfikacji formalnej, kompensowane są poprzez bogaty zbiór metod i narzędzi weryfikacji sieci Petriego. Słabą popularność sieci Petriego w środowisku inżynierskim, niweluje popularność języka UML. Zalety modelu dualnego będzie można wykorzystać w pełni tylko wtedy, gdy będzie zapewniona możliwość wzajemnej i jednoznacznej konwersji pomiędzy modelami składowymi. Warunkami koniecznymi takiej transformacji jest zapewnienie przez model składowy możliwości modelowania na różnych poziomach abstrakcji oraz przedstawienie współbieżności. Warunki te są spełnione zarówno przez maszynę stanów UML jak i hierarchiczną sieć Petriego. Jeśli warunki nie zostałyby spełnione, spłaszczenie HCFSM spowodowało by pogorszenie czytelności modelu, a brak możliwości przedstawienia współbieżności wykładniczy wzrost stanów. W chwili obecnej, złożona syntaktyka i mało precyzyjna syntaktyka diagramów maszyny stanów UML utrudnia znacząco konwersję do sieci Petriego. Istnieje wiele elementów nadmiarowych (punkty: rozwidlenia, scalenia, złączenia). Należy więc tak dobrać podzbiór elementów maszyny stanów UML, aby był wystarczający do modelowania sterowników logicznych, pozwalał na jednoznaczne przekształcenie do sieci Petriego.



Rys. 2. Ścieżka projektowa oparta na dualnej specyfikacji
Fig. 2. Dual-specification based developing process

Rysunek 2 przedstawia proces wytworzenia sterownika logicznego oparty o dualną specyfikację. Model dualny wykorzystany jest na etapie specyfikacji. Projektant korzystając z front-endu w postaci diagramów maszyny stanów UML określa działania sterownika logicznego – pierwszy z modeli składowych dualnej specyfikacji. Na podstawie maszyny stanów tworzony jest ekwi-

walenty model w postaci sieci Petriego. Aby w pełni wykorzystać zalety płynące z zastosowania dualnej specyfikacji w procesie projektowania sterowników logicznych, konieczna jest implementacja narzędzia pozwalającego na automatyczną konwersję pomiędzy modelem w postaci hierarchicznej sieci Petriego a maszyną stanów języka UML.

Utworzony model składowy w postaci sieci Petriego poddawany jest następnie weryfikacji formalnej. Metody formalnej weryfikacji sieci Petriego pozwalają na sprawdzenie żywotności oraz ograniczonej [7,13]. Model może również zostać zweryfikowany pod kątem spełnienia wymagań użytkownika [8] z wykorzystaniem metod wnioskowania symbolicznego. Jeśli na etapie weryfikacji zostaną wykryte błędy, następuje powrót na etap specyfikacji, gdzie dokonuje się korekty specyfikacji. Kolejnym etapem jest przeprowadzenie procesu syntezy i implementacji układu. Hierarchiczna sieć Petriego będąca jednym z dwóch modeli składowych specyfikacji dualnej jest również bazą do przeprowadzenia syntezy układowej. Zapis sieci z wykorzystaniem standardów PNML PNSF3 pozwala na przeprowadzenie procesu syntezy z wykorzystaniem takich jak system PeNCAD.

3. Przekształcenie maszyny stanów UML do hierarchicznej sieci Petriego

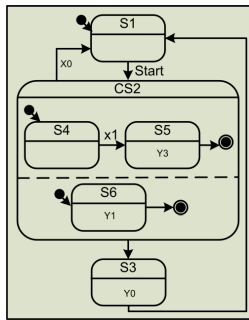
Konieczne jest określenie zasad konwersji pomiędzy modelami. Wstępnie określony podzbiór diagramu maszyny stanów UML zawiera następujące elementy: stan prosty, stan złożony, punkt wejścia, punkt wyjścia, stan początkowy oraz końcowy, przejście, stany głębokiego i płytkiego wznowienia. Graficzne odpowiedniki elementów maszyny stanów oraz sieci Petriego przedstawiono w Tabeli 1. Stany: złożone, płytkiego i głębokiego wznowienia oraz punkty wejścia i wyjścia nie posiadają bezpośrednich odpowiedników w zbiorze elementów sieci Petriego.

Tab. 1. Elementy SM UML i odpowiedniki w sieci Petriego
Tab. 1. SM UML elements and equivalent elements in Petri net subset

	Maszyna stanów UML		Hierarchiczna sieć Petriego
Stan prosty		Miejsce	
Przejście z określonym wyzwalaczem		Tranzycja z określonym warunkiem realizacji	
Stan oznaczony stanem początkowym		Miejsce oznakowane znacznikiem	
Stan końcowy		Miejsce bez określonego oraz przejścia wyjściowego	

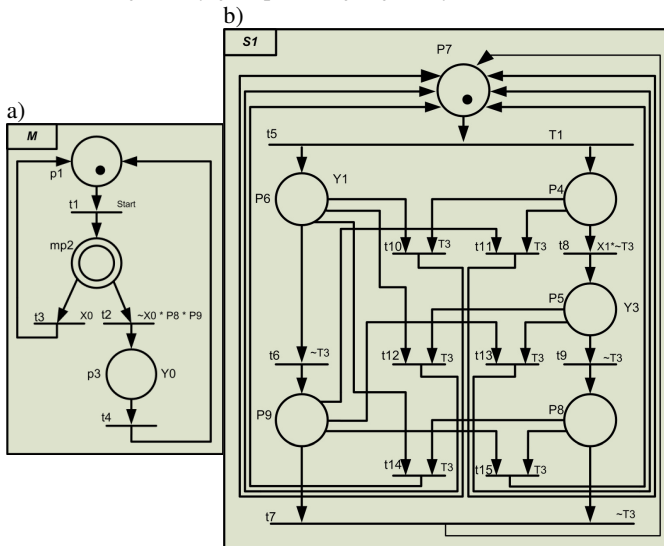
Stan złożony maszyny stanów pozwala na reprezentację hierarchii oraz procesów współbieżnych. Hierarchiczna sieć Petriego posiada również podobny element – makromiejsce oraz makrotranzycję. Niestety, nie da się bezpośrednio przekształcić stanu złożonego na makromiejsce. Konieczne jest zapewnienie możliwości realizacji wyłączenia stanu złożonego. W literaturze przedstawiono możliwość realizacji wyłączenia poprzez wprowadzenie miejsc pomocniczych służących do deponowania znaczników [8] oraz tranzycji synchronizujących procesy współbieżne (jeśli stan złożony posiadał obszary współbieżne). Rozwiązanie to miało wadę w postaci dodatkowych cykli zegarowych koniecznych do realizacji wyłączenia. Dodatkowy cykl pojawiał się na każdym poziomie hierarchii wyłączonego stanu złożonego [5]. Proponuje się wprowadzenie nowej metody odwzorowania stanów złożonych w sieci Petriego. W nowej metodzie nie ma dodatkowych miejsc do deponowania znaczników. Realizację wyłączenia oraz synchronizację procesów osiągnięto poprzez wprowadzenie tranzycji synchronizujących dla każdego z miejsc we

współbieżnych ścieżkach. Tranzycje pokrywają wszystkie możliwości znakowania w jakich może znaleźć się podsieć w trakcie realizacji stanu złożonego z współbieżnymi obszarami. Proponuje się zastosowanie nowej metody aktywacji podsieci (Rys. 4b) poprzez wprowadzenie miejsca pomocniczego $P7$ posiadającego w znakowaniu początkowym żeton. Tranzycja $T5$ (aktywator podsieci) odpowiada za uruchomienie podsieci $S1$ odpowiadającej stanowi złożonemu $CS2$. Warunek realizacji tranzycji $T5$ ma postać sumy logicznej iloczynów wartości aktywności stanów oraz warunków realizacji przejść skierowanych na stan złożony.



Rys. 3. Maszyna stanów UML
Fig. 3. UML State machine

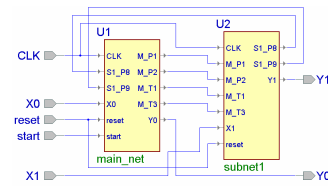
Diagram maszyny stanów UML będący modelem behawioralnym sterownika (Rys. 3), może być przekształcony do modelu w postaci hierarchicznej sieci Petriego M i SI . Stan złożony posiada dwa przejścia wyjściowe: bezwarunkowe – realizowane w momencie osiągnięcia stanów końcowych przez wszystkie jego podmaszyny, oraz przejście zwrotne z wyłączeniem, realizowane w chwili spełnienia warunku $X0$. Przejście bezwarunkowe realizowane jest poprzez tranzycję $T7$. Wyłączenie zrealizowano poprzez wprowadzenie tranzycji wyłączających $T10 - T15$, które odpowiadają za deaktywację podsieci oraz połączenie procesów współbieżnych. Tranzycje wyłączające opisane są warunkiem $T3$ – realizacji tranzycji odpowiadającej za wyłączenie.



Rys. 4. Sieć Petriego a) sieć główna b) podsieć skojarzona z makromiejscem mp2
Fig. 4. Petri net a) Main net B) subnet connected with mp2 macroplace

Realizacja przejścia bezwarunkowego odbywa się gdy zostaną osiągnięte miejsca $P8$ oraz $P9$ odpowiadające stanom końcowym maszyny stanów. Tranzycja $T2$ opisana jest warunkiem w postaci iloczynu wartości aktywności miejsc $P8$ i $P9$ oraz zanegowanym warunkiem realizacji tranzycji wyłączających.

Sterownik realizowany jest układowo w postaci dwóch bloków odpowiadających sieci głównej oraz podsieci. Sygnały M_P1 , M_P2 , SI_P8 , SI_P9 sygnalizujące aktywność danego stanu oraz M_T1 , M_T3 odpowiadające warunkom realizacji tranzycji są wyprowadzone na zewnątrz modułów, pozwalają one na realizację aktywowania podsieci oraz wyłączeń, czy też przejść bezwarunkowych.



Rys. 5. Implementacja na poziomie blokowym
Fig. 5. Implementation on block level

Rysunek 5 przedstawia schemat blokowy, dodatkowe porty wejściowe i wyjściowe sygnalizujące aktywność stanu bądź wyłączenie tranzycji zostały dodatkowo oznaczone nazwą funkcjonalną w postaci prefiksów $M_$ oraz $SI_$.

Weryfikacja poprawności funkcjonowania projektu sterownika binarnego odbyła się w środowisku Active - HDL.

4. Podsumowanie

Zastosowanie dualnej specyfikacji pozwala na zwiększenie transparentności modelu. Weryfikacja formalna sieci Petriego w połączeniu z przyjaznym interfejsem użytkownika pozwala na zwiększenie jakości specyfikacji [11] sterownika poprzez dokładniejsze odzwierciedlenie założeń projektowych oraz weryfikację funkcjonowania sterownika. Przekłada się to bezpośrednio na zmniejszenie kosztów wytworzenia sterownika – koszt wprowadzenia zmian rośnie w funkcji potęgi w czasie. Zastosowanie hierarchii na etapie specyfikacji i implementacji ułatwia przeprowadzenie procesu częściowej modułowej rekonfiguracji sterownika [9]. Zastosowanie metody transformacji stanu złożonego oraz przejść z wyłączeniem do modelu sieci Petriego powoduje zmniejszenie liczby cykli zegara, koniecznych do przeprowadzenia procesu wyłączenia stanu złożonego.

5. Literatura

- [1] M. Adamski, A. Karatkevich, M. Węgrzyn (red): Design of embedded control systems, Springer, New York 2005
- [2] D. Andreu, G. Souquet, T. Gil.: Petri Net Based Rapid Prototyping of Digital Complex System, IEEE Computer Society Annual Symposium on VLSI, IEEE, 2008
- [3] F. Basile, P. Chiacchio, D. Del Grosso: Modelling automation systems by UML and Petri Nets, Proceedings of the 9th International Workshop on Discrete Event Systems, Goeteborg, IEEE, 2008
- [4] Bazydło G.: From UML state machine diagrams to FPGA, International Workshop Control and Information Technology - IWCIT 2007. Ostrava, Czechy, 2007. - Ostrava: VSB - Technical University of Ostrava, 2007, s. 183-186, ISBN: 978-80-248-1567-1
- [5] M. Doligalski, Specyfikacja programów sterowania oparta na hierarchicznym modelu maszyny stanów UML, IX International PHD Workshop - OWD 2007. Wisła, Polska, 2007. - Warszawa, 2007. - Conference Archives PTETiS, Vol. 23, s. 299-304
- [6] D. D. Gajski, F. Vahid, S. Narayan, J. Gong, Specification and Design of Embedded Systems, P T R Prentice Hall, New Jersey 1994
- [7] Grobelna I., Formal verification of logic controller specification using NuSMV model checker, X International PHD Workshop - OWD 2008. Wisła, Polska, 2008, Conference Archives PTETiS ; Vol. 25, s. 459-464. - ISBN: 83-922242-4-8
- [8] G. Łabiak: Wykorzystanie hierarchicznego modelu współbieżnego automatu w projektowaniu sterowników cyfrowych, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona góra 2005
- [9] C. Kao: Benefits of Partial Reconfiguration, Xilinx, Inc. 2005
- [10] T. Holvoet, P. Verbaeten: Petri Charts: an Alternative Technique for Hierarchical Net Construction
- [11] L. Józwiak: Quality-driven System on Chip Design (REASON Tutorial), Kraków, 2005
- [12] M. Szyrka: Sieci Petriego w modelowaniu i analizie systemów współbieżnych, WNT, Warszawa 2008
- [13] A. Węgrzyn: Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego, Oficyna Wydawnicza UZ, Zielona Góra 2003

Michał DOLIGALSKI
UNIwersytet Zielonogórski