

RÓWNOWAŻNOŚĆ PŁASKICH I HIERARCHICZNYCH SIECI PETRIEGO

Grzegorz Andrzejewski

**Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50**

e-mail: g.andrzejewski@iie.uz.zgora.pl

STRESZCZENIE

W artykule przedstawiono ogólną charakterystykę najistotniejszych własności hierarchicznych sieci Petriego HPN w kontekście możliwości ich modelowania z wykorzystaniem konstrukcji charakterystycznych dla interpretowanych sieci płaskich.

1. WPROWADZENIE

Wprowadzenie hierarchii do sieci Petriego, umożliwiło uproszczenie projektowania nie tylko na poziomie zapisu modelu, ale również jego analizy. Znanych jest wiele rozszerzeń sieci wspierających hierarchię: HCPN (*Hierarchical Coloured PN*) [8], OOPN (*Object Oriented PN*) [4], HOONets (*Hierarchical Object-Oriented PN*) [7], THORN (*Timed Hierarchical Object-Related Nets*) [10], MacroNets [5], PetriCharts [6], GrafChart (hierarchiczne rozszerzenie modelu Grafcet [3]). Ze względu na przydatność do opisywania systemów sterowania cyfrowego na uwagę zasługują HCPN, MacroNets oraz PetriCharts. Hierarchia jest w nich wprowadzana na różne sposoby. W sieciach HCPN oraz MacroNets osiągnęta jest przez zwykle podstawienie pewnego fragmentu sieci w miejsce makromiejsca lub makrotranszycji (hierarchia strukturalna), natomiast w sieciach PetriCharts zauważyć można elementy hierarchii behawioralnej, w której pewna podsieć jest aktywowana w momencie oznaczenia związanego z nią makrowęzła. Prezentowane modele niestety nie uwzględniają opisu zależności czasowych oraz bardzo ważnej (z praktycznego punktu widzenia) własności - historii makrostanów.

Wobec powyższego zdefiniowano takie rozszerzenie sieci Petriego, które przy zachowaniu spójności semantycznej na najniższym poziomie abstrakcji z sieciami interpretowanych, uwzględnia w szerszym stopniu takie elementy opisu jak hierarchia, zależności czasowe, historia czy wyłączenie makromiejsca. W artykule niniejszym przedstawione zostaną najważniejsze konstrukcje sieci hierarchicznych HPN, w kontekście ich realizacji z wykorzystaniem interpretowanych sieci płaskich. Problem równoważności zachowawczej obu modeli jest o tyle istotny, iż z jednej strony pokazuje korzyści z zastosowania modelu HPN do opisywania złożonych systemów, a z drugiej strony pozwala podjąć próby prowadzenia analizy formalnej tych sieci z wykorzystaniem metod opracowanych dla sieci płaskich.

2. HIERARCHICZNA SIEĆ PETRIEGO

Interpretowaną hierarchiczną siecią Petriego HPN nazywamy n-tkę:

$$\text{HPN} = \{P, T, F, S, \mathbb{T}, \chi, \psi, \lambda, \alpha, \varepsilon, \tau\} \quad (1)$$

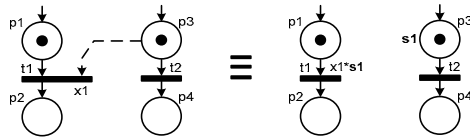
gdzie: P – jest skończonym niepustym zbiorem miejsc; T – jest skończonym niepustym zbiorem tranzycji; F – jest skończonym niepustym zbiorem łuków; S – jest skończonym niepustym zbiorem zmiennych; \mathbb{T} – jest dyskretną skalą czasu; $\chi: P \rightarrow 2^N$ jest funkcją hierarchii, określającą zbiór bezpośrednich podwęzłów określonego miejsca sieci; $\psi: P \rightarrow \{true, false\}$ jest dwuwartościową funkcją historii; $\lambda: N \rightarrow (S \cup E)$ jest funkcją etykietującą, przypisującą do węzłów wyrażenia budowane z elementów zbiorów S i E , oraz wskazanych operatorów; $\alpha: P \rightarrow \{true, false\}$ jest funkcją znakowania początkowego; $\varepsilon: P \rightarrow \{true, false\}$ jest funkcją znakowania końcowego; $\tau: N \rightarrow \mathbb{T}$ jest funkcją czasu, przypisującą liczbę z dyskretnej skali czasu do zbioru miejsc P .

3. RÓWNOWAŻNOŚĆ PŁASKICH I HIERARCHICZNYCH SIECI PETRIEGO

Wprowadzenie tylu cech do modelu sieci Petriego może spowodować wrażenie, iż ich weryfikacja formalna jest niezwykle trudna, a nawet wręcz niemożliwa. W punkcie niniejszym zostanie pokazane, że wszystkie wprowadzone elementy opisu, a więc: łuki zezwalające i zabraniające, parametry dynamiczne, hierarchia, historia, oraz wyłączenie – można przedstawić z wykorzystaniem tylko i wyłącznie elementów opisu interpretowanych sieci płaskich (to znaczy tak, aby były one równoważne pod względem funkcjonalnym). Jest to istotne, ze względu na możliwość wykorzystania narzędzi umożliwiających formalną analizę płaskich sieci Petriego (np. DesignCPN) do analizy sieci hierarchicznych.

3.1. Łuki zezwalające i zabraniające

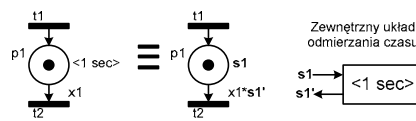
Łuki zezwalające można w prosty sposób zastąpić sygnałami wewnętrznymi. W tym celu należy wszystkie miejsca, z których biorą początek łuki zezwalające zaetykietować nazwami sygnałów wewnętrznych, przy czym dla każdego miejsca przypisuje się tylko jeden sygnał wewnętrzny niezależnie od liczby łuków z niego wychodzących; oraz każde takie miejsce etykietowane jest inną nazwą sygnału wewnętrznego. Następnie tranzycję, do której łuk dochodzi etykietuje się nazwą sygnału wewnętrznego, takiego jak przy odpowiednim miejscu zezwalającym, przy czym: etykieta ta stanowi warunek tranzycji nałożony poprzez operator iloczynu logicznego na warunek istniejący (rys. 1). Łuki zabraniające można również zastąpić sygnałami wewnętrznymi. Procedura zamiany jest bardzo podobna, z tym, że tranzycję, do której łuk dochodzi, etykietuje się negacją sygnału wewnętrznego, takiego jak przy odpowiednim miejscu zabraniającym.



Rys. 1. Zastąpienie łuku zezwalającego sygnałem wewnętrznym

3.2. Parametr czasowy przypisany do miejsca

Parametry czasowe związane z miejscami mogą być postrzegane jako czasy przebywania znacznika w tych miejscach, co ilustruje rysunek 3.

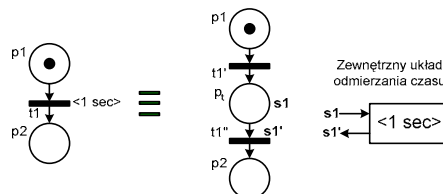


Rys. 3. Interpretacja parametru czasowego przypisanego do miejsca

Warunkiem zezwolenia tranzycji t_2 jest minięcie określonego czasu (1 sec), licząc od momentu aktywacji miejsca p_1 . Sytuację taką można zamodelować z wykorzystaniem interpretowanych sieci płaskich i zewnętrznego układu odmierzania czasu (startowanego sygnałem s i sygnalizującego koniec akcji sygnałem s'). W tym celu należy wszystkie miejsca, do których przypisano parametry czasowe zaetykietować nazwami sygnałów wewnętrznych, przy czym każde takie miejsce etykietowane jest inną nazwą sygnału wewnętrznego. Następnie wszystkie tranzycje, dla których miejsca te są miejscami wejściowymi, etykietuje się odpowiednimi nazwami sygnałów wewnętrznych, takich jak przy odpowiednich miejscach wejściowych rozszerzonych o znak „,”; przy czym: etykiety te stanowią warunki tranzycji nałożone poprzez operator iloczynu logicznego na warunki istniejące.

3.3. Parametr czasowy przypisany do tranzycji

Parametry czasowe związane z tranzycjami mogą być postrzegane jako czasy mijające od momentu usunięcia znaczników z miejsc wejściowych danych tranzycji, do momentu umieszczenia znaczników w ich miejscach wyjściowych (rys. 4).



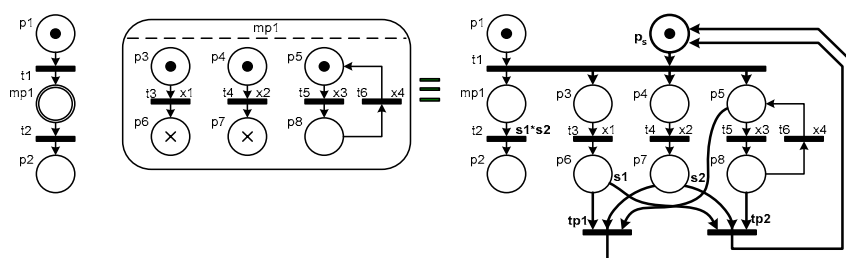
Rys. 4. Interpretacja parametru czasowego przypisanego do tranzycji

Sytuację taką można zamodelować z wykorzystaniem mechanizmu przedstawionego w poprzednim podpunkcie. W tym celu wystarczy w miejsce tranzycji, do których przypisano

parametry czasowe wprowadzić konstrukcje (t', p_i, t''), takie że t' etykietowana jest tak jak t , do miejsca p_i przypisany jest sygnał $s1$ startujący akcję czasową realizowaną przez układ zewnętrzny, a tranzycja t'' etykietowana jest sygnałem $s1'$, sygnalizującym koniec akcji czasowej.

3.4. Makromiejsce bez historii

W sieciach HPN wprowadza się założenie, iż w momencie deaktywacji makromiejsca, wszystkie znaczniki wewnątrz jego rozwinięcia są usuwane. Jest to założenie upraszczające zapis, ponieważ użytkownik nie musi zastanawiać się, w jaki sposób to osiągnąć. Jednakże implikuje to niestety niespełnienie własności bezpośredniego podstawienia podsieci w miejsce odpowiedniego makromiejsca. Sprowadzając zapis do interpretowanej sieci płaskiej, należy wprowadzić jedno dodatkowe miejsce spoczynkowe p_s , oraz szereg tranzycji pomocniczych, których zadaniem jest właśnie usunięcie wszystkich znaczników z makromiejsca i aktywacja miejsca spoczynkowego. Ilustruje to rysunek 5.

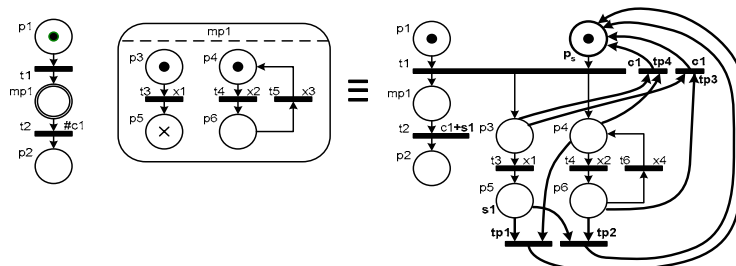


Rys. 5. Rozwinięcie hierarchii do sieci płaskiej

Biorąc pod uwagę fakt, iż warunkiem opuszczenia znacznika z makromiejsca jest osiągnięcie wszystkich miejsc końcowych wewnątrz podsieci związanej z tym makromiejscem, wówczas łatwo zauważyć, iż tranzycje pomocnicze będą potrzebne do łączenia wszystkich tych miejsc, które są równocześnie oznakowane wraz z miejscami końcowymi. Tranzycje pomocnicze muszą posiadać wyższy priorytet od pozostałych, w celu likwidacji ewentualnych niedeterminizmów. Miejsca końcowe można zaetykietować sygnałami wewnętrznymi, które są następnie użyte do warunkowania tranzycji wyjściowych odpowiedniego makromiejsca (na rysunku 5 sygnały $s1$ oraz $s2$, aktywowane w miejscach $p6$ i $p7$, warunkują tranzycję $t2$).

3.5. Etykieta warunku silnego (wyłączenie)

Makromiejsce można pozbawić aktywności m.in. poprzez nałożenie na jego tranzycję wyjściową warunku silnego *abort* i spełnienie *go*. Oznacza to usunięcie wszystkich znaczników z odpowiedniej podsieci. Aby sytuację taką można było przedstawić z wykorzystaniem sieci płaskiej należy wprowadzić tyle tranzycji pomocniczych ile wierzchołków posiada graf znakowań rozpatrywanej podsieci (rys. 6).

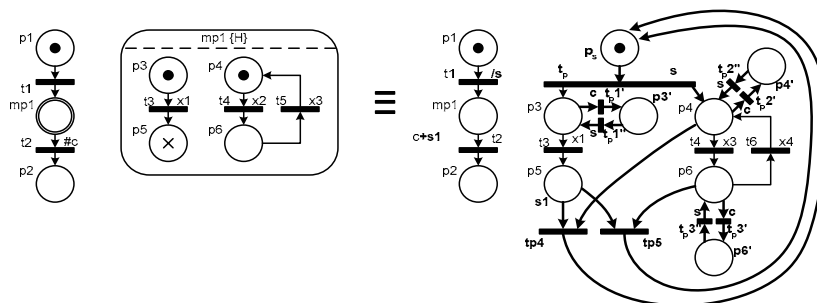


Rys. 6. Interpretacja warunku wyłuszczającego

Każdemu wierzchołkowi odpowiada odrębna tranzycja a jej miejscami wejściowymi są wszystkie miejsca należące do tego wierzchołka. Miejscem wyjściowym każdej tranzycji pomocniczej jest pomocnicze miejsce spoczynkowe p_s . Warunkiem zezwolenia tranzycji pomocniczych jest warunek wyłuszczający (*abort*), przy czym po osiągnięciu znakowania końcowego podsieć zachowuje się tak jak w pkt. poprzednim (bez wyłuszczenia). Warunek wyłuszczenia w sieci nadrzędnej dopisuje się poprzez operator sumy logicznej do pozostałych warunków zezwolenia tranzycji.

3.6. Makromiejsce z historią

W celu sprowadzenia do sieci płaskich makromiejsc z historią konieczne jest dodatkowe (w stosunku do sytuacji przedstawionej w poprzednim podpunkcie) wprowadzenie dla każdego miejsca podsieci dodatkowego miejsca spoczynkowego, do którego może zostać wprowadzony znacznik w momencie wyłuszczenia podsieci (rys. 7).



Rys. 7. Interpretacja parametru historii

Wyprowadzenie z niego znacznika do odpowiadającego mu miejscu głównemu, następuje po ponownej aktywacji makromiejsc w sieci nadrzędnej. Warto zauważyć, że tak jak w podpunkcie poprzednim, osiągnięcie znakowania końcowego, oznacza możliwość deaktywacji odpowiedniego makromiejsc, a jego ponowne aktywowanie spowoduje wprowadzenie znaczników do miejsc początkowych podsieci.

Użycie w jednym modelu wielu elementów opisu, w celu sprowadzenia do sieci płaskiej, wymaga superpozycji odpowiednich metod opisanych powyżej.

4. PODSUMOWANIE

W artykule pokazane zostało, w jaki sposób można sieć hierarchiczną sprowadzić do interpretowanej sieci płaskiej. Wobec powyższego można stwierdzić, iż do analizy i weryfikacji formalnej sieci hierarchicznych można użyć znanych metod opracowanych dla sieci płaskich. Jednakże sama transformacja jest zadaniem bardzo skomplikowanym, a sieć wynikowa może mieć duże rozmiary, wydaje się więc raczej koniecznym opracowanie metod alternatywnych analizy dla sieci hierarchicznych [9].

Praca naukowa finansowana ze środków Komitetu Badań Naukowych w latach 2003-2006 jako projekt badawczy nr 4 T11C 006 24.

LITERATURA

- [1] G. Andrzejewski: *Hierarchical Petri Net as a Representation of Reactive Behaviors*, Proc. Int. Conf. Advanced Computer Systems: ACS'2001, Szczecin, Part 2, pp. 145-154
- [2] G. Andrzejewski: *Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych*. Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra, 2003
- [3] R. David, H. Alla: *Petri Nets & Grafcet. Tools for modeling discrete event systems*. Prentice Hall, New York, 1992
- [4] R. Esser: *An object oriented Petri net approach to embedded systems design*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1996
- [5] J.M. Fernandes, M. Adamski, A.J. Proenca: *VHDL Generation from Hierarchical Petri Net Specifications of Parallel Controllers*. IEE Proc.: Computers and Digital Techniques, no. 144, vol. 2, Mar., 1997, pp. 127-137
- [6] T. Holvoet, P. Verbaeten: *Petri Charts: an Alternative Technique For Hierarchical Net Construction*. In IEEE Conference on Systems, Man and Cybernetics, 1995
- [7] J.E. Hong, D.H.Bae: *HOONets: Hierarchical Object-Oriented Petri Nets for System Modeling and Analysis*, KAIST Technical Report CS/TR-98-132, November, 1998
- [8] K. Jensen: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, Monographs in Theoretical Computer Science, Springer-Verlag, 1997
- [9] A. Karatkievich, G. Andrzejewski: *Analiza wybranych własności interpretowanych sieci Petriego metodą optymalnej symulacji*, Mat. Kraj. Konf. Krajowa Konferencja Elektroniki: KKE'2002, Kołobrzeg, t. II, ss. 685-690
- [10] S. Schof, M. Sonnenschein, R. Wieting: *High-level Modeling with THORNs*, Proceedings of the 14th International Congress on Cybernetics, Namur, Belgium, 1995