

SEKWENCJA, WSPÓLBIEŻNOŚĆ I HIERARCHIA W MODELOWANIU ZACHOWANIA DYSKRETNÝCH STEROWNIKÓW REAKTYWNYCH *

Grzegorz Łabiak

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

e-mail: G.Łabiak@iie.uz.zgora.pl

STRESZCZENIE

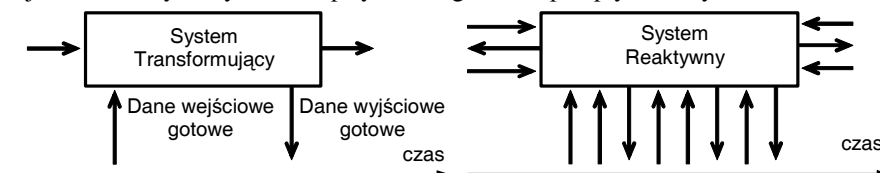
Artykuł w sposób poglądowy przedstawia zagadnienie modelowania zachowania systemów reaktywnych, które znajdują zastosowania jako sterowniki binarne. W pierwszej części artykułu scharakteryzowano systemy reaktywne zestawiając je z systemami transformującymi. Następnie przedstawiono najistotniejsze problemy modelowania takich systemów. W dalszej części opisano najpopularniejsze metodyki: maszyna o skończonej liczbie stanów, sieć Petriego, hierarchiczna sieć Petriego, diagramy statechart oraz dokonano porównania diagramów statechart z hierarchiczną siecią Petriego.

1. SYSTEMY TRANSFORMUJĄCE I SYSTEMY REAKTYWNE

Projektowane systemy, ze względu na zachowanie, można podzielić następująco [1]:

- transformujące,
- reaktywne.

Systemy transformujące (rys. 3) są to systemy sterowane danymi, co oznacza, że zachowanie takiego systemu zależy od przepływu danych pomiędzy jego składowymi. Systemy te charakteryzują się tym, że każdy z jego elementów składowych (a w szczególności cały system) czeka na pojawienie się gotowych danych na jego wejściu, co jest odpowiednio sygnalizowane, po czym następuje przetwarzanie tych danych i dane te są przekazywane na wyjście, co również jest sygnalizowane. Po takim przetwarzaniu system transformujący (lub jego składowe) przechodzą w stan oczekiwania na pojawienie się kolejnego kompletu danych na wejściu. Takie systemy można opisywać diagramami przepływu danych.



Rys. 3. Model systemu transformującego i systemu reaktywnego [1]

* Prace są finansowane ze środków KBN, jako projekt badawczy nr 4 T11C 006 24.

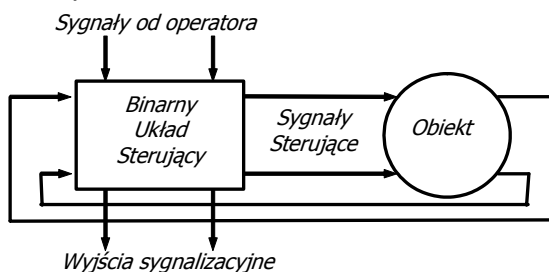
Systemy typu transformującego można modelować metodą od ogółu do szczegółu (*ang.* top-down). Cechą typową tych systemów jest to, że złożone zależności między danymi wejściowymi a danymi wyjściowymi, mogą zostać funkcjonalnie dekomponowane na prostsze zależności składowe.

Systemy reaktywne (rys. 3) nie funkcjonują w tak przewidywalnym scenariuszu czasowym jak systemy transformujące. Ich główną cechą charakterystyczną jest to, że dane wejściowe takiego systemu mogą pojawiać się zupełnie dowolnie, (co na rysunku oznaczone jest poprzez strzałki umieszczone przy systemie), tzn. nie przyjmuje się żadnych założeń, co do tego, na których wejściach i kiedy pojawią się dane, a w przypadku systemów czasu rzeczywistego oczekuje się, że odpowiedź układu będzie natychmiastowa, tzn. czas odpowiedzi jest niezauważalny dla otoczenia (*ang.* perfect synchrony hypothesis).

Ponadto, o systemach reaktywnych można powiedzieć, że [5]:

- są sterowane zdarzeniami,
- prowadzą stałą interakcję ze swoim otoczeniem, używając do tego sygnałów wejściowych i wyjściowych, które mogą być zarówno ciągłe, jak i dyskretne,
- systemy te powinny odpowiadać na przerwania, tzn. reagować na zdarzenia o odpowiednio wysokim priorytecie, nawet jeżeli są zajęte innymi obliczeniami,
- ich działanie i reakcje na sygnały wejściowe często muszą odpowiadać surowym wymaganiom czasowym,
- zmieniają swój stan w zależności od bieżącego trybu działania i wartości danych oraz od przeszłego zachowania,
- nie muszą być współbieżne, aczkolwiek najczęściej ich zachowanie może być przyrównane do komunikujących się procesów działających współbieżnie.

Przykładami systemów reaktywnych są system rezerwacji miejsc w samolocie, systemy osadzone takie jak układy awioniki, czy zaawansowane elektronicznie artykuły gospodarstwa domowego. Do systemów reaktywnych również należą układy sterujące, a w szczególności cyfrowe układy sterowania binarnego (rys. 4), czyli takie, które na swoim wejściu i wyjściu operują wartościami binarnymi.



Rys. 4. Układ sterowania binarnego [10]

Innym możliwym zastosowaniem reaktywnego systemu sterującego jest jednostka sterująca w systemie przetwarzającym dane (*ang.* Finite-State Machine with Datapath). Taki model układu znajduje zastosowanie w cyfrowym przetwarzaniu sygnałów (*ang.* Digital Signal Processing) oraz stanowi podstawową architekturę mikroprocesorów ogólnego przeznaczenia.

2. MODELOWANIE DYSKRETNYCH SYSTEMÓW REAKTYWNYCH

Do opisów systemów reaktywnych mogą stosować się diagramy przepływu danych oraz dekompozycja funkcjonalna, lecz te metody, skuteczne w przypadku systemów transformujących, są niewystarczające do opisu ich pełnego działania. Do kompletnego przedstawienia zachowania można by użyć automatu skończonego. Utworzenie modelu wówczas polegałoby na wyodrębnieniu stanów lub trybów działania, określeniu zdarzeń i warunków, które powodują przejścia między stanami oraz na wyspecyfikowaniu akcji wykonywanych w poszczególnych stanach i przy rozpatrywanych przejściach. Taki model jednak posiada pewne wady. Brak hierarchii powoduje, że projektant nie może specyfikować projektu takiego systemu na odpowiednim dla swoich potrzeb poziomie szczegółowości, a jako że jest to model „płaski” zmuszony jest operować najdrobniejszymi szczegółami specyfikacji. Z kolei brak wsparcia dla współbieżności często jest przyczyną wykładniczej eksplozji stanów globalnych, co powoduje, że taki model przestaje być zupełnie zrozumiały. Zatem, właściwym modelem do opisu zachowania sterujących systemów reaktywnych powinien być model wspierający trzy wymienione cechy, czyli: operować pojęciem stanu, wspierać hierarchię, opisywać współbieżność. Takim modelem są diagramy statechart zaproponowane przez D. Harel [5], czyli hierarchiczny model automatu współbieżnego [4][8].

3. MASZYNY O SKOŃCZONEJ LICZBIE STANÓW

Maszyna o skończonej liczbie stanów (*ang.* Finite-State Machine lub w skrócie *FSM*), zwana również maszyną stanową lub automatem cyfrowym [1][9], jest najbardziej popularnym modelem opisującym zachowanie systemów sterowania, w którym chwilowe działanie systemu jest w sposób naturalny reprezentowane w formie stanów i przejść między nimi. Zasadniczo *FSM* składa się ze zbioru stanów [9], zbioru przejść między stanami oraz zbioru akcji przypisanych stanom lub przejściom. Ze względu na powiązanie akcji wyjściowych z przejściami lub stanami wyróżnia się dwa rodzaje automatów: Mealy’ego i Moore’a.

Dla inżyniera projektanta modelowanie zachowania realizowanego jako *FSM* odbywa się głównie poprzez narysowanie równoważnego mu grafu stanu lub, co już jest mniej czytelne, poprzez tabelaryczne zdefiniowanie funkcji przejść. Wadą *FSM* są trudności modelowania zachowania składającego się z dużej liczby stanów, a tą cechą charakteryzuje się wiele systemów sterujących. Rozwiązaniem może być inne spojrzenie na zagadnienie sterowania

polegające na opisie zachowania w sposób współbieżny lub też operowanie pojęciami uogólnionymi (abstrakcyjnymi), co już nie jest wspierane przez metodykę *FSM*.

4. SIECI PETRIEGO

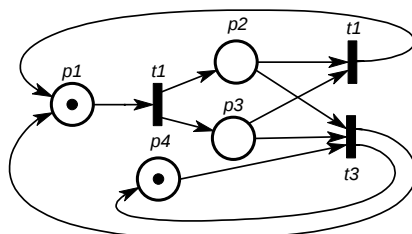
Sieć Petriego (w skrócie oznaczana PN), jako model matematyczny [1][4][11], jest dwudzielnym skierowanym grafem, w którym dwa rodzaje węzłów zwane są miejscami i tranzycjami. W graficznej postaci (rys. 5) miejsca reprezentowane są przez kółka a tranzycje przez prostokąty połączone łukami skierowanymi. Każdemu miejscu w sieci może być przyporządkowany co najwyżej jeden żeton (marker, znacznik).

Działanie sieci Petriego określone jest następującymi regułami:

- tranzycja jest gotowa do realizacji, jeżeli wszystkie jej miejsca wejściowe posiadają żeton,
- realizacja tranzycji pociąga za sobą usunięcie żetonów (markerów) ze wszystkich jej miejsc wejściowych i umieszczenie żetonów we wszystkich jej miejscach wyjściowych.

W oparciu na tak przyjętym modelu matematycznym sieci Petriego można zrealizować współbieżny automat cyfrowy (multiautomat cyfrowy), realizujący funkcje systemu reaktywnego [1][2][4][11][12]. W tym celu należy wprowadzić pojęcie interpretowanej sieci Petriego, tzn. takiej sieci, która reaguje na sygnały wejściowe, generując sygnały do otoczenia. W interpretowanej sieci Petriego generowanie sygnałów wyjściowych można związać z miejscem (sieć typu Moore'a) lub z tranzycją (sieć typu Mealy'ego). Sygnały wejściowe mogą występować w wyrażeniach logicznych nałożonych na tranzycje, zwanych predykatami lub etykietami, których spełnienie jest koniecznym warunkiem ich realizacji. Dodatkowo, w celu rozszerzenia możliwości opisu, stosuje się łuki zabraniające i zezwalające.

Główną zaletą stosowania sieci Petriego, w porównaniu z tradycyjnym modelem automatu sekwencyjnego, jest zapobieżenie wykładniczemu wzrostowi liczby stanów modelowanego zachowania w równoważnym automacie sekwencyjnym. Wynika to z możliwości opisu uwzględniającego współbieżność.

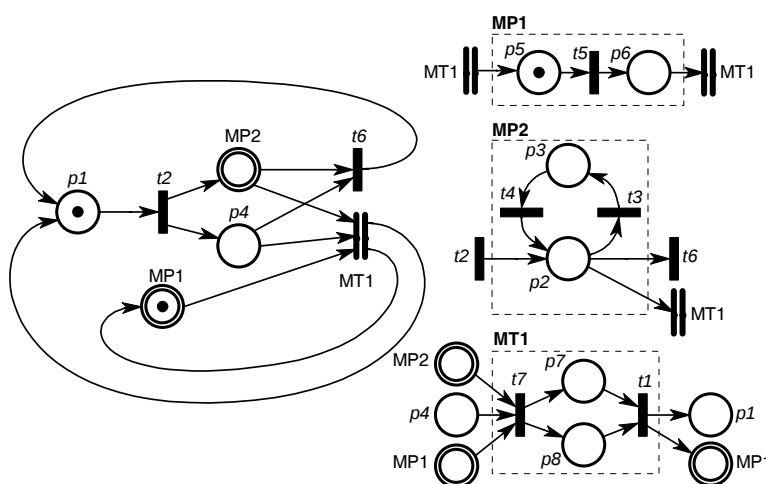


Rys. 5. Przykład sieci Petriego

5. HIERARCHICZNE SIECI PETRIEGO

Hierarchiczną siecią Petriego (w skrócie oznaczaną HPN), czasami zwaną makrosiecią [12][13][14], jest sieć, w której występują makroelementy typu makromiejsca i makrotranzycje. Rys. 6 przedstawia przykładową sieć hierarchiczną (zaczepniętą z pracy [14]), gdzie makromiejsca zaznaczone są podwójnymi kółkami, makrotranzycje podwójnymi belkami, a obok na rysunku znajdują się ich rozwinięcia. Sieć hierarchiczna jest siecią powstałą z tradycyjnej „płaskiej” sieci Petriego, w której z wybranych jej fragmentów wydodrębniono podsieci. Takie podsieci, w zależności od elementów, poprzez które są dołączane do sieci, zaznaczane są jako makromiejsca (elementami łączącymi są miejsca) lub makrotranzycje (elementami łączącymi są tranzycje). W rozwinięciach makroelementów można wydodrębniać kolejne podsieci, tworząc z nich makroelementy niższych poziomów hierarchii. Uzyskany w ten sposób model charakteryzuje się hierarchią strukturalną.

Modelowanie zachowania przy użyciu hierarchicznych sieci Petriego zasadniczo polega na operowaniu pojęciami najniższego poziomu szczegółowości (podejście od szczegółu do ogółu, *ang.* bottom-up). Dopiero mając zachowanie opisane tradycyjną siecią Petriego w sposób naturalny można w niej wydodrębniać makroelementy. Główna zaleta tak otrzymanego modelu ujawnia się, gdy jest on wykorzystywany do dokumentowania zachowania. Przy postępowaniu odwrotnym – od ogółu do szczegółu (*ang.* top-down) – projektant, co prawda może operować abstrakcyjnymi pojęciami makroelementów (abstrakcyjnymi w sensie oderwania od szczegółów najniższego poziomu), jednakże w przypadku łuków dochodzących do makroelementów zawsze trzeba pamiętać o końcowym (lub początkowym) miejscu lub tranzycji, będących elementami najniższego poziomu szczegółowości, co jest wbrew zasadzie abstrahowania.



Rys. 6. Przykład hierarchicznej sieci Petriego [13]

6. DIAGRAMY STATECHARTS

Diagramy Statecharts zostały zaproponowane przez Davida Harela jako wizualny formalizm do opisu złożonych systemów reaktywnych [5]. Formalizm ten jest notacją wykorzystującą pojęcie stanu, która jest rozszerzeniem pojęcia automatu skończonego – jego graficznej reprezentacji jako diagramu stanów i przejść. Diagramy statecharts obrazowo można opisać równaniem [5]:

$$\text{diagram statechart} = \text{diagram stanów} + \text{głębia} + \text{ortogonalność} + \text{mechanizm rozgłaszania}$$

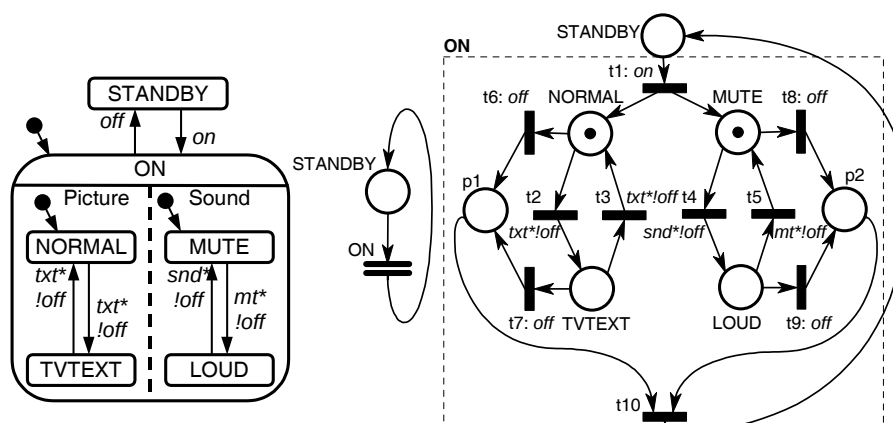
Rys. 7 przedstawia prosty diagram statechart. Stany na diagramie są przedstawione jako kragłokąty, przyjscia jako strzałki z etykietami. Pojęcie głębi w równaniu odnosi się do możliwości przyporządkowywania stanom złożonym podległych diagramów. Na przykład na rys. 7 stanowi *ON* przyporządkowano dwa automaty sekwencyjne *Picture* i *Sound*, które są w relacji współbieżności. Ortogonalność oznacza, że dwa lub więcej automatów sekwencyjnych mogą modelować zjawiska współbieżne (np. *Picture* i *Sound*). Z kolei mechanizm rozgłaszania realizowany jest przez generowanie zdarzeń, które zwrotnie oddziałują na funkcjonowanie diagramu.

7. DIAGRAMY STATECHART A HIERARCHICZNE SIECI PETRIEGO

Prezentując diagramy statechart bardzo często zadawane jest pytanie, jaka jest zasadnicza różnica między diagramami a hierarchiczną siecią Petriego. Aby na to pytanie odpowiedzieć należy najpierw uściślić z jakiego rodzaju hierarchiczną siecią Petriego diagramy są porównywane, gdyż istnieje wiele odmian sieci, co do których stosowany jest przymiotnik hierarchiczne. Kolejną kwestią jest kryterium porównawcze. Można zestawiać ze sobą przeróżne cechy obu modeli, takie jak na przykład właściwość historii czy istnienie stanu końcowego, lecz wydaje się, że cechą o pierwszorzędym znaczeniu przy takim porównaniu jest reprezentacja hierarchii w modelowaniu. W diagramach statechart obok stanów wyższych poziomów hierarchii zwanych abstrakcyjnymi, występują również tranzycje wyższych poziomów związane z takimi stanami, zwane tranzycjami abstrakcyjnymi. Realizacja takiej tranzycji ma charakter wyłączeniowy, co w opinii autora stanowi istotę pojęcia hierarchii behawioralnej. W przypadku sieci Petriego typu makro, różnice w modelowaniu zachowania są zasadnicze. W sieciach tego rodzaju brak jest tranzycji abstrakcyjnych, co znacznie utrudnia modelowanie polegające na podejściu od ogółu do szczegółu.

Przykładowo, realizacja równoważnej sieci Petriego diagramowi statecharts z rys. 7 polega na przyporządkowaniu każdemu nieabstrakcyjnemu (podstawowemu) stanowi z diagramu odpowiadającego mu miejsca w sieci Petriego [7]. Ilustruje to rys. 7. Następnie do każdego miejsca odpowiadającego stanowi, z którego tranzycja abstrakcyjna może wywłaszczyć sterowanie (w przykładzie stany: *NORMAL*, *TVTEXT*, *MUTE*, *LOUD*) należy dołączyć

tranzycje (na rysunku oznaczone kolejno jako t_6, t_7, t_8, t_9), których rolą jest zabranie żetonu na wypadek realizacji tranzycji abstrakcyjnej. Odebrane żetony są następnie deponowane w pomocniczych miejscach p_1 i p_2 – po jednym dla każdego z podległych automatów sekwencyjnych (odpowiednio *Picture* i *Sound*). Dalej poprzez tranzycję pomocniczą t_{10} , żeton jest kierowany do miejsca *STANDBY*, co kończy wyłączeniowe przekazanie sterowania. Na realizację abstrakcyjnej tranzycji ze stanu *STANDBY* do stanu *ON* składają się tranzycje t_6, t_7, t_8, t_9 i t_{10} oraz miejsca p_1 i p_2 .



Rys. 7. Diagram statechart i równoważne mu sieci Petriego

Przedstawione przejście skupia się na modelowaniu najistotniejszych elementów diagramów statechart: stanów i tranzycji w tym tranzycji abstrakcyjnych. Pominięto zagadnienia stanów z pamięcią (atrybut historii), stanu końcowego, akcji związanych ze zdarzeniami.

8. PODSUMOWANIE

Dyskretne systemy sterujące posiadają cechy systemów reaktywnych. Projektownie takich systemów przebiega w sposób odmienny niż to ma miejsce w przypadku bardziej tradycyjnych systemów transformujących. Podstawowa różnica między tymi układami jest konsekwencją przyjęcia założenia o natychmiastowej odpowiedzi układu reaktywnego. Szczególną rolę w projektowaniu odgrywa modelowanie zachowania. Tradycyjnie do celów modelowania można wykorzystać metodykę *FSM* lub jej bardziej nowoczesne i wydajne warianty takie jak sieci Petriego czy diagramy statecharts. Bardzo istotna jest możliwość operowania hierarchią strukturalną – w przypadku sieci Petriego typu makro, i behavioralna – w przypadku diagramów statecharts.

Osobną kwestią, jak dotychczas nie rozwiązana w sposób zadowalający, jest równoważność poszczególnych modeli zachowania. Przedstawione porównanie diagramów statechart z sieciami Petriego typu makro ilustruje złożoną naturę tego zagadnienia, a głównym źródłem

trudności są rozbieżności wśród badaczy odnośnie składni i semantyki poszczególnych modeli, zwłaszcza diagramów statecharts.

LITERATURA

- [1] M. Adamski: *Projektowanie układów cyfrowych systematyczną metodą strukturalną*, Monografia Nr 49, Wydawnictwo Wyższej Szkoły Inżynierskiej w Zielonej Górze, Zielona Góra, 1990
- [2] K. Biliński: *Application of Petri Nets in parallel controllers design*, PhD. Thesis, University of Bristol, Electrical and Electronic Engineering Department, Bristol, 1996
- [3] D. Drusinsky, BetterState Pro Tutorial: *An Introduction to Design with StateCharts*, Integrated Systems, Inc., Sunnyvale April 1997
- [4] D. D. Gajski, F. Vahid, S. Narayan, J. Gong: *Specification and Design of Embedded Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1994
- [5] D. Harel: *Statecharts: A visual formalism for complex Systems*, Science of Computer Programming, Vol.8, 1987
- [6] D. Harel, M. Politi: *Modeling Reactive Systems With Statecharts : The StateMate Approach*, McGraw Hill Text; ASIN: 0070262055; October 8, 1998
- [7] G. Łabiak: *Modelling Statecharts Diagram by Means Of Petri Nets*, VI International Conference on Advanced Computer Systems, Szczecin – Poland 1999, pp. 253-259
- [8] G. Łabiak: *Wykorzystanie hierarchicznego modelu współbieżnego automatu w projektowaniu sterowników cyfrowych*, praca doktorska, Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, , Warszawa 2003
- [9] W. Majewski: *Układy Logiczne. Wybrane zagadnienia*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1998
- [10] P. Misiurewicz: *Układy automatyki cyfrowej*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1987
- [11] T. Murata: *Petri Nets: Properties, Analysis and Application*, Proceedings of the IEEE, vol. 77, No. 4, April 1989
- [12] M. Węgrzyn: *Hierarchiczna synteza kontrolerów logicznych z wykorzystaniem FPGA*, I Krajowa Konferencja Naukowa - Reprogramowalne Układy Cyfrowe, RUC'98, Szczecin, 12-13.03.1998, ss.33-40, 1998
- [13] M. Węgrzyn: *Hierarchiczna implementacja współbieżnych kontrolerów cyfrowych z wykorzystaniem FPGA*, Rozprawa doktorska, Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Czerwiec 1998
- [14] M. Węgrzyn, M.Adamski: *Hierarchical Approach for Design of Application Specific Logic Controller*, ISIE'99, Bled, Słowenia, 1999, Vol. 3, s.1389-1394