Piotr KAWALEC [1], Marcin RŻYSKO [2]
[1] POLITECHNIKA WARSZAWSKA, WYDZIAŁ TRANSPORTU, Koszykowa 75, 00-662 Warszawa
[2] BOMBARDIER TRANSPORTATION (RAIL ENGINEERING) POLSKA Sp. z o.o., Ogrodowa 58, 00-876 Warszawa

Algorytmiczne podejście do projektowania logiki zależnościowej w systemach sterowania ruchem kolejowym

Dr hab. inż. Piotr KAWALEC

Ukończył studia na kierunku elektronika, doktoryzował się w specjalności elementy i urządzenia techniki obliczeniowej i systemów sterowania, habilitował się na Politechnice Warszawskiej w specjalności sterowanie ruchem w transporcie. Pracuje na stanowisku profesora nadzwyczajnego na Wydziale Transportu Politechniki Warszawskiej, jego zainteresowania naukowe dotyczą metod modelowania i sterowania ruchem w transporcie z wykorzystaniem specjalizowanych systemów cyfrowych.

e-mail: pka@wt.pw.edu.pl

Mgr inż. Marcin RŻYSKO

W 2013 roku ukończył studia na Wydziale Transportu Politechniki Warszawskiej w specjalności sterowanie ruchem kolejowym. Obecnie pracuje w biurze projektowym Bombardier Transportation (Rail Engineering) Polska. W działalności zawodowej i naukowej zajmuje się projektowaniem i weryfikacją logiki zależnościowej w systemach sterowania ruchem kolejowym.

e-mail: m.rzysko@gmail.com

Streszczenie

W artykule przedstawiono propozycję formalnej metody specyfikacji złożonych funkcji zależnościowych w systemach srk. Opisana metoda opiera się na podejściu geograficznym, umożliwiającym wykorzystanie zbudowanych elementów dla dowolnego układu torowego. Zastosowanie algorytmów do opisu działania pozwala na intuicyjne przejście od elementarnych warunków, opisanych w języku naturalnym, do zapisu formalnego oraz kodu języka VHDL. Zaprezentowane zostały również możliwości weryfikacji otrzymanych algorytmów.

Słowa kluczowe: algorytmy, srk, systemy zależnościowe.

The algorithmic approach to railway interlocking logic design

Abstract

The paper presents several problems concerning the interlocking logic design process in modern railway traffic control systems. This issue is increasingly difficult in implementing new functions and adapting systems to new market requirements. For easy and clear transition from the verbal operation description to the formal notation in HDL language, the algorithmic approach was introduced. Universal input and output alphabets proposed in the paper allow convenient variable identification. The rules for decomposing the system into objects and algorithms are described (Fig. 1). Since the method is based on the geographical approach, the set of designed elements is universal and can be used for any track layout. Using the logical algorithm scheme (2) to describe the functionalities allows one to intuitively transform the elementary conditions in a natural language to a formal notation and finally to the VHDL code (Fig. 4). Verification possibilities of the described method are also presented. Using the Active-HDL integrated design environment it was possible to analyze the created algorithms on many hierarchical levels. Beginning from the algorithm level (proving the correctness of the transition to the finite-state machine graph), through the object level (Fig. 5a, showing the complex interlocking functions) to the whole interlocking logic (Fig. 5b, allowing verification of the design using the interlocking table), all of the performed simulations proved correctness of the specification.

Keywords: algorithms, interlocking, railway traffic control.

1. Wprowadzenie

Coraz istotniejszym problemem projektowania logiki zależnościowej w systemach sterowania ruchem kolejowym jest zagadnienie formalnego opisu działania tego typu układów. Pewne elementy formalizacji stosowane są w istniejących urządzeniach, prowadzone są również badania w tym kierunku [2, 7]. Nie istnieje jednak metoda, która w prosty sposób pozwalałaby uzyskać formalny opis skomplikowanych funkcjonalności urządzeń srk, będący jednocześnie bezpośrednią podstawą do fizycznej realizacji układu.

W ramach prac badawczych prowadzonych na Politechnice Warszawskiej opracowany został macierzowy zapis elementarnych funkcji zależnościowych, niezależny od techniki realizacji urządzeń [6]. W niniejszym artykule zaprezentowane zostaną zagadnienia związane z projektowanie złożonych funkcjonalności systemów sterowania ruchem kolejowym.

2. Metoda budowy algorytmów dla stacyjnych urządzeń srk

Specyfikacja kompletnych funkcji zależnościowych dla geograficznego systemu srk może być wykonana z wykorzystaniem formalnego opisu algorytmów [3, 5]. Takie podejście pozwala na łatwą konstrukcję złożonych funkcjonalności, takich jak realizacja przebiegów czy dobór sygnałów świetlnych. Wykorzystanie narzędzi wspomagania projektowania pozwala na łatwe przekształcenie zbudowanych algorytmów w kod języka opisu sprzętu.

Z uwagi na sekwencyjny charakter pracy urządzeń automatyki kolejowej, naturalnym sposobem opisu jest zastosowanie algorytmów. Rozpatrując stacyjny system zależnościowy jako całość wyróżnić można alfabety:

- stanów wejść: $\boldsymbol{X} = \{X_P, X_K\}$,
- stanów wyjść: $\boldsymbol{Y} = \{Y_M, Y_S\}$.

Składnikami wyżej wymienionych zbiorów są następujące wektory:

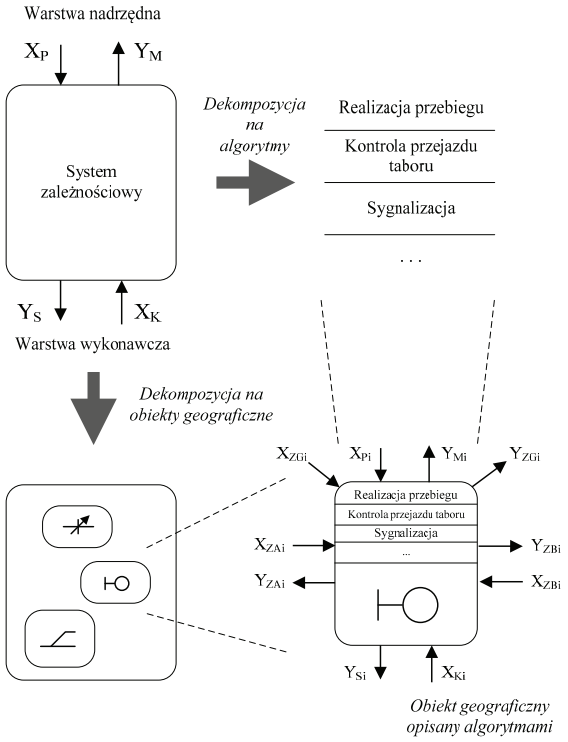
- w. poleceń: $\boldsymbol{X_P} = \{X_{P1}, X_{P2},...,X_{PN}\}$, $X_{P...} = (x_{p0}, x_{p1}, ..., x_{pn})$;
- w. kontroli: $\boldsymbol{X_K} = \{X_{K1}, X_{K2},...,X_{KN}\}$, $X_{K...} = (x_{k0}, x_{k1}, ..., x_{kn})$;
- w. meldunków: $\boldsymbol{Y_M} = \{Y_{M1}, Y_{M2},...,Y_{MN}\}$, $Y_{M...} = (y_{m0}, y_{m1}, ..., y_{mn})$;
- w. sterowań: $\boldsymbol{Y_S} = \{Y_{S1}, Y_{S2},...,Y_{SN}\}$, $Y_{S...} = (y_{s0}, y_{s1}, ..., y_{sn})$.

W przypadku opisu pojedynczego elementu systemu wprowadzony został dodatkowo wektor opisujący wzajemne relacje pomiędzy opisywanym elementem, a elementami sąsiadującymi. W odniesieniu do jednego elementu alfabety stanów wejść i stanów wyjść uzupełnione zostały o wektor zależności:

$$\boldsymbol{X_Z} = \{X_{ZA}, X_{ZB},...,X_{ZG}\}, \qquad X_{Z...} = (x_{z0}, x_{z1}, ..., x_{zn}) ;$$
$$\boldsymbol{Y_Z} = \{Y_{ZA}, Y_{ZB},...,Y_{ZG}\}, \qquad Y_{Z...} = (y_{z0}, y_{z1}, ..., y_{zn}) ;$$

gdzie: *A*, *B*, ... - połączenia geograficzne do sąsiednich obiektów logicznych, *G* - połączenie niegeograficzne.

Kolejnym krokiem jest wykorzystanie zdefiniowanych alfabetów do formalnego opisu działania układu. Metoda dekompozycji systemu na obiekty i funkcje przedstawiona została na rys. 1.



Rys. 1. Schemat metody dekompozycji
Fig. 1. Scheme of the decomposition method

Metodą formalnego zapisu algorytmu, który może być łatwo uzyskany z opisu słownego, jest logiczny schemat algorytmu [1, 8]. Jest to formalny zapis działania, mający zwartą, matematyczną formę. W tej metodzie algorytm zapisany jest w postaci ciągu następujących po sobie symboli. Stany (będące elementami zbioru ***Y***) warunkują wartości zmiennych wyjściowych zgodnie z przyjętym kodowaniem. Zmienne wejściowe (zbiór ***X***) decydują o kolejnym wykonywanym kroku. Jeśli sprawdzany warunek ma wartość logiczną '1', wykonywana jest kolejna operacja (po prawej stronie). Jeśli natomiast warunek jest równy '0', kolejny krok wskazywany jest przez strzałkę o odpowiednim numerze, skierowaną w górę. Strzałka skierowana w dół oznacza miejsce, od którego należy kontynuować realizację algorytmu, występuje tylko raz dla każdego numeru. Znak ω oznacza bezwzględny skok do miejsca wskazanego przez strzałkę (warunek logiczny zawsze fałszywy).

Poniżej zestawiono używane symbole:

$\boldsymbol{Y}_k$ - symbol wektora zmiennych wyjściowych,
x_k - symbol zmiennej wejściowej,
$\boldsymbol{X}_k$ - symbol wektora zmiennych wejściowych,
$\overset{i}{\uparrow}$ - symbol sprawdzenia warunku, w przypadku negatywnego wyniku testu logicznego wykonywany jest skok do miejsca oznaczonego numerem i,
$\overset{i}{\downarrow}$ - symbol miejsca skoku o numerze i,
$\omega\overset{i}{\uparrow}$ - symbol bezwzględnego skoku (warunek zawsze fałszywy).

Na potrzeby opracowanej metody przyjęto podział funkcji systemu srk na funkcje proste (realizowane przez jeden obiekt geograficzny - na przykład blokowanie przestawiania zwrotnicy) oraz złożone (realizowane przez więcej niż jeden obiekt geograficzny - na przykład realizacja drogi jazdy przebiegu).

Zakłada się, że budowa każdego algorytmu składać się będzie z następujących kroków:

1. Dekompozycja funkcji na etapy realizacji (na przykład realizacja przebiegu: stan spoczynkowy → wybieranie → nastawianie → utwierdzenie → zwolnienie → stan spoczynkowy),
2. Stworzenie możliwych ścieżek wykonania algorytmu zgodnie z liczbą przebiegów możliwą do realizacji przez dany element.

W odniesieniu do pojedynczego modułu urządzeń geograficznych przebieg rozumiany jest jako możliwość wykorzystania danego elementu w drodze jazdy. Liczba możliwych do zrealizowania przebiegów p została oznaczona jako P i zależy ona od rodzaju elementu. Jeżeli charakter pracy elementu tego wymaga, poszczególne etapy w każdym z przebiegów mogą być rozróżnione na rodzaje (na przykład nastawianie przebiegu manewrowego lub pociągowego, zwolnienie przebiegu przez przejazd taboru lub doraźnie).

Logiczny schemat algorytmu opisujący funkcję realizacji przebiegu przedstawiono poniżej. Przyjęto następujące oznaczenia:

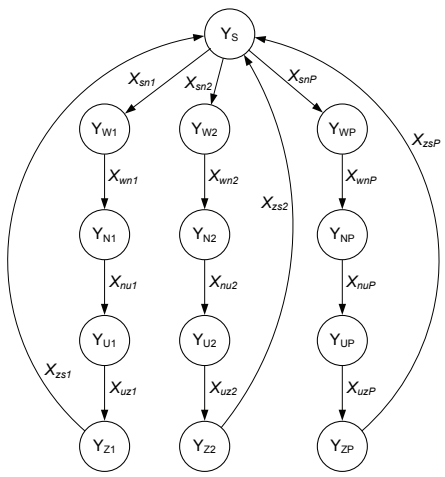
Y_{Mp} - symbol wektora zmiennych wyjściowych w stanie m w przebiegu p,
X_{mnp} -symbol wektora zmiennych wejściowych, będącego warunkiem przejścia ze stanu m do stanu n w przebiegu p,

$$\begin{aligned}
&\overset{0}{\downarrow}Y_S X_{sw1}\overset{11}{\uparrow}X_{sw2}\overset{21}{\uparrow}\ldots X_{swP}\overset{P1}{\uparrow}\omega\overset{0}{\uparrow}\\
&\overset{11}{\downarrow}Y_{W1}X_{wn1}\overset{12}{\uparrow}\omega\overset{11}{\uparrow}\overset{21}{\downarrow}Y_{W2}X_{wn2}\overset{22}{\uparrow}\omega\overset{21}{\uparrow}\ldots\overset{P1}{\downarrow}Y_{WP}X_{wnP}\overset{P2}{\uparrow}\omega\overset{P1}{\uparrow}\\
&\overset{12}{\downarrow}Y_{N1}X_{nu1}\overset{13}{\uparrow}\omega\overset{12}{\uparrow}\overset{22}{\downarrow}Y_{N2}X_{nu2}\overset{23}{\uparrow}\omega\overset{22}{\uparrow}\ldots\overset{P2}{\downarrow}Y_{NP}X_{nuP}\overset{P3}{\uparrow}\omega\overset{P2}{\uparrow}\\
&\overset{13}{\downarrow}Y_{U1}X_{uz1}\overset{14}{\uparrow}\omega\overset{13}{\uparrow}\overset{23}{\downarrow}Y_{U2}X_{uz2}\overset{24}{\uparrow}\omega\overset{23}{\uparrow}\ldots\overset{P3}{\downarrow}Y_{UP}X_{uzP}\overset{P4}{\uparrow}\omega\overset{P3}{\uparrow}\\
&\overset{14}{\downarrow}Y_{Z1}X_{zs1}\overset{0}{\uparrow}\omega\overset{14}{\uparrow}\overset{24}{\downarrow}Y_{Z2}X_{zs2}\overset{0}{\uparrow}\omega\overset{24}{\uparrow}\ldots\overset{P4}{\downarrow}Y_{ZP}X_{zsP}\overset{0}{\uparrow}\omega\overset{P4}{\uparrow}
\end{aligned} \tag{1}$$

Algorytm zapisany w postaci LSA stanowi dogodną podstawę do budowy grafu stanów automatu skończonego. Przekształcając zbiór symboli Y w stany, a zbiór symboli X w tranzycje, otrzymano graf stanów przedstawiony na rys. 2.

Opisana wyżej metoda pozwala zamodelować algorytm działania dowolnego elementu logicznego zblokowanego systemu zależnościowego w postaci grafu stanów. Dla zadanego elementu konieczne jest zidentyfikowanie warunków, które muszą zostać spełnione, aby układ mógł przejść do każdego kolejnego stanu (czyli opisanie poszczególnych wektorów $X...$).

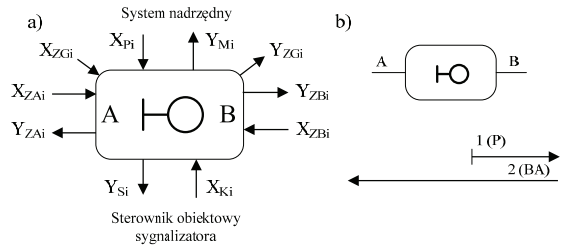
Skonstruowanie grafu przejść o strukturze dostosowanej do pracy zadanego elementu oraz zidentyfikowanie warunków wykonania wszystkich tranzycji umożliwia wykonanie specyfikacji w języku opisu sprzętu. Jest to możliwe dzięki edytorowi grafów przejść FSM, który jest jednym z edytorów pakietu Active-HDL, wykorzystywanego do prac nad zagadnieniem.



Rys. 2. Graf stanów automatu skończonego dla algorytmu (1)
Fig. 2. Graph of the finished-state machine for algorithm (1)

3. Specyfikacja i weryfikacja wybranego algorytmu

Jako przykład wykorzystania metody wyspecyfikowany i zweryfikowany został algorytm wykorzystania obiektu w drodze jazdy przebiegu dla semafora wjazdowego. Ogólna struktura oraz przebiegi możliwe do realizacji przez ten obiekt przestawione zostały na rys. 3.



Rys. 3. Struktura wejść i wyjść obiektu logicznego semafora wjazdowego (a) oraz przebiegi przewidziane do realizacji przez ten obiekt (b)
Fig. 3. Inputs and outputs of the entry signal logical object (a) and routes available for this type of object (b)

Semafor wjazdowy pełni funkcję elementu początkowego przebiegu wjazdowego oraz elementu końcowego przebiegu wyjazdowego. Liczba możliwych do realizacji przebiegów wynosi więc P=2. Wykorzystując opisaną wyżej metodę budowy algorytmów, wykonany został logiczny schemat algorytmu.

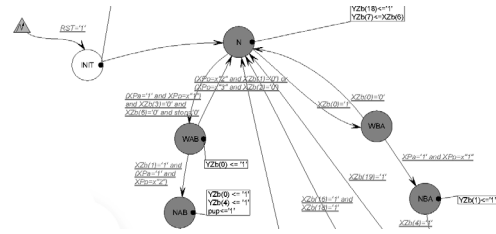
$$
\begin{aligned}
SMWJ = {} & \overset{0}{\downarrow} Y_S X_{sw1} \overset{11}{\uparrow} X_{sw2} \overset{21}{\uparrow} \omega \overset{0}{\uparrow} \\
& \overset{11}{\downarrow} Y_{W1} X_{wn1} \overset{12}{\uparrow} \omega \overset{11\,21}{\uparrow} \downarrow Y_{W2} X_{wn2} \overset{22}{\uparrow} \omega \overset{21}{\uparrow} \\
& \overset{12}{\downarrow} Y_{N1} X_{nu1} \overset{13}{\uparrow} \omega \overset{12\,22}{\uparrow} \downarrow Y_{N2} X_{nu2} \overset{23}{\uparrow} \omega \overset{22}{\uparrow} \\
& \overset{13}{\downarrow} Y_{U1} X_{uz1} \overset{14}{\uparrow} \omega \overset{13\,23}{\uparrow} \downarrow Y_{U2} X_{uz2} \overset{24}{\uparrow} \omega \overset{23}{\uparrow} \\
& \overset{14}{\downarrow} Y_{Z1} X_{zs1} \overset{0}{\uparrow} \omega \overset{14\,24}{\uparrow} \downarrow Y_{Z2} X_{zs2} \overset{0}{\uparrow} \omega \overset{24}{\uparrow}
\end{aligned}
\tag{2}
$$

Kolejnym krokiem prowadzącym do uzyskania kompletnego opisu tak stworzonego algorytmu jest zidentyfikowanie elementarnych warunków wykonania dla tranzycji oraz wartości wektorów wyjściowych dla stanów. Specyfikacja wykonana w edytorze FSM na podstawie zbudowanego powyżej logicznego schematu algorytmu przedstawiona została na rys. 4.

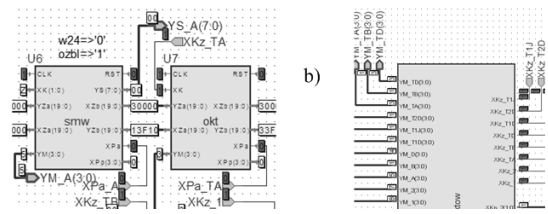
Ponieważ edytor FSM umożliwia automatyczne wygenerowanie kodu języka VHDL dla zadanego grafu, stanowi on ostatni etap przejścia od słownego opisu funkcjonalności do opisu formalnego i przeznaczonego do bezpośredniego wykorzystania w procesie implementacji i prototypowania. Zanim jednak wyspecyfikowany algorytm będzie mógł stać się podstawą do budowy układu, konieczna jest weryfikacja poprawności wykonania specyfikacji. Pierwszym poziomem hierarchicznym, na którym możliwe jest wyeliminowanie błędów powstałych przy tworzeniu grafu na podstawie logicznego schematu algorytmu, jest poziom grafu FSM. Edytor FSM pakietu Active-HDL umożliwia symulowanie działania algorytmu opisanego grafem zarówno w sposób ręczny (podając odpowiednie wymuszenia) jak i automatyczny (przy pomocy tzw. *Testbench'y*, pozwalających na gruntowną weryfikację grafu). Obserwacja przebiegu symulacji możliwa jest bezpośrednio na grafie, jak również na rejestrowanych przebiegach wartości zmiennych.

Na kolejnym poziomie algorytm wyspecyfikowany powyżej posłużył do stworzenia kompletnego obiektu logicznego sygnalizatora wjazdowego, który został przetestowany pod względem integracji algorytmów i współpracy z obiektami sąsiadującymi.

Na tym etapie możliwa jest również obserwacja stanu zmiennych w edytorze schematów blokowych BDE, który wykorzystywany jest do budowy układu z pojedynczych obiektów (rys. 5a).



Rys. 4. Specyfikacja w edytorze FSM
Fig. 4. Specification in the FSM editor



Rys. 5. Poziom obiektów (a) i kompletnej specyfikacji (b)
Fig. 5. Object level (a) and complete specification (b)

Najwyższy poziom hierarchiczny opisuje wszystkie elementy układu torowego w postaci jednego obiektu w edytorze BDE, który stanowi kompletny zapis zależności dla zadanego układu torowego (rys. 5b). Możliwa jest więc weryfikacja kompletnych funkcjonalności z tablicą zależności [4].

4. Wnioski

Wykonane w przebiegu prac nad zagadnieniem specyfikacje pozostałych obiektów geograficznego systemu srk pozwoliły na zaprojektowanie zależności dla przykładowych układów torowych. Badania symulacyjne wykonane przy pomocy opisanych narzędzi wykazały poprawność opracowania na poziomie algorytmów, obiektów oraz kompletnego układu (weryfikacja z tablicą zależności). Stworzony zaprezentowaną metodą opis charakteryzuje się przejrzystością oraz łatwością w modyfikacji i przy implementowaniu nowych funkcji.

5. Literatura

[1] Apuniewicz S.: Układy przekaźnikowe w automatyce zabezpieczenia ruchu kolejowego. WPW Warszawa, 1969.
[2] Hlavatý T., Přeučil L., Štěpán P.: Case Study: Formal Methods in Development and Testing of Safety Critical Systems: Railway Interlocking System. EUROMICRO, 2001.
[3] Kawalec P., Rżysko M.: Metoda tworzenia formalnego zapisu algorytmów działania urządzeń srk. Prace Naukowe - Transport, z. 100, OWPW Warszawa, s. 91-108, 2013.
[4] Kawalec P., Rżysko M.: Weryfikacja równań zależnościowych z wykorzystaniem symulatorów logicznych na przykładzie zastosowania pakietu Active-HDL. Technika Transportu Szynowego 10/2013, s. 1587-1596.
[5] Kawalec P., Rżysko M.: Zastosowanie grafów przejść automatów skończonych do opisu algorytmów działania urządzeń srk. Politechnika Warszawska, Prace Naukowe - Transport, z. 95, OWPW Warszawa, s. 221-230, 2013.
[6] Koliński D.: Formalny opis funkcji zależnościowych systemów srk dla współczesnych posterunków ruchu. Prace Naukowe - Transport, z. 86, OWPW Warszawa, s. 35-52, 2012.
[7] Minkowitz C., Atkiss J.: An object-oriented formal specification of a configuration language for railway interlockings. 3rd Northern Formal Methods Workshop, 1998.
[8] Traczyk W.: Układy cyfrowe. Podstawy teoretyczne i metody syntezy. Wydawnictwa Naukowo-Techniczne, 1982.

otrzymano / received: 08.07.2014
przyjęto do druku / accepted: 02.09.2014 *artykuł recenzowany / revised paper*