

Michał DOLIGALSKI, Marian ADAMSKI

UNIwersYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

Projektowanie strukturalnych programów dla sterowników PLC na podstawie modelu maszyny stanowej UML

Mgr inż. Michał DOLIGALSKI

Absolwent Uniwersytetu Zielonogórskiego (2006). Ukończył studia o specjalności Inżynieria Komputerowa. Od roku 2006 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



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

Prof. dr hab. inż. Marian ADAMSKI

Dyrektor Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują projektowanie mikrosystemów cyfrowych oraz formalne metody programowania sterowników logicznych. Członek IEEE, IEE, ACM, Polskiego Towarzystwa Elektrotechniki Teoretycznej i Stosowanej oraz Polskiego Towarzystwa Informatycznego.



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

Streszczenie

Język UML (ang. Unified Modeling Language) jest efektywnym narzędziem w modelowaniu różnego rodzaju systemów. Jego podzbiorem są diagramy statechart oparte na mapach stanów Davida Harela. Umożliwiają modelowanie procesów współbieżnych i przedstawianie specyfikacji na różnych poziomach abstrakcji. Diagramy statechart stanowią doskonałe narzędzie do specyfikacji systemów reaktywnych, a w szczególności programów dla przemysłowych sterowników logicznych. Zastosowanie ich w miejsce klasycznych metod takich jak sieci SFC czy FSM pozwala na uproszczenie procesu projektowania i tworzenie czytelnej dokumentacji. W artykule przedstawiono sposób wykorzystania języka UML, jako narzędzia do specyfikacji. Zaproponowano metodę implementacji programów sterowania z wykorzystaniem języka SFC oraz ST. Ilustracją omawianej tematyki jest przykładowy system sterowania.

Słowa kluczowe: UML, PLC, statechart.

Design of PLC structured programs from UML state machine model

Abstract

The Unified Modeling Language (UML) is the effective tool in the modeling of different kinds of systems. The statechart diagrams based on the David Harel's state maps represent its subset. The statechart diagrams enable the concurrent modeling processes and the presenting of specification on different abstraction levels. Statechart diagrams are the ideal tools for reactive systems specifications, tools for programmable logic controllers in particular. The use of the controller instead of such classic methods as SFC or FSM networks allows to simplify the design and specification process. The advantages of the UML language, as a specification tool, have been presented in the article. The article also presents the history state conversion method. The following exemplary controlling system is the illustration of the reviewed subject area.

Keywords: UML, PLC, statechart.

1. Wstęp

Programowane sterowniki logiczne (PLC) znajdują zastosowanie w sterowaniu różnorodnych systemów przemysłowych. Główną zaletą sterowników PLC jest możliwość ich swobodnego programowania.

Do specyfikacji i implementacji programów dla sterowników logicznych wykorzystuje się język SFC [9]. Język ten wypiera stosowany dość często schemat drabinkowy (LD). Posiada on swoje korzenie w standardzie GRAFCET [6]. Standard ten został oparty na sieciach Petriego sterowania [1].

Aparat matematyczny, który jest skojarzony z sieciami Petriego, dostępny jest również dla języka SFC. Jest to język graficzny – działanie programu opisuje się za pomocą kroków oraz tranzycji (czyli przejść między krokami). Jeśli spojrzymy na ten język jako narzędzie do specyfikacji programu, to okaże się, że dla dużych systemów taka specyfikacja może być mało czytelna, a wręcz trudna do analizy.

W 1987 roku David Harel [8] przedstawił idee hierarchicznych map stanów. Są one często określane w literaturze angielskiej jako diagram statechart, lub zwyczajnie statechart. Terminologia ta została również zapożyczona do języka polskiego.

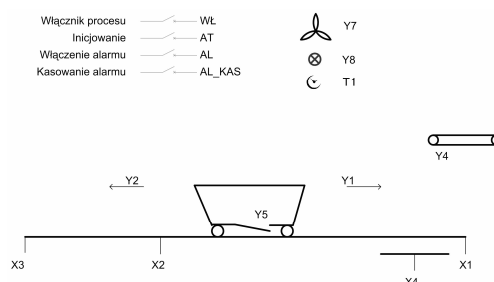
Mapy stanów Davida Harela zostały włączone do standardu UML (Unified Modeling Language) [5, 11]. Język UML jest stale rozwijany. Aktualnie obowiązujący jest standard 2.0 [11,13]. Należy tutaj nadmienić, że wraz z rozwojem języka nastąpiły zmiany w nomenklaturze. Początkowo określając diagramy maszyny stanów, stosowano termin „statechart”. Począwszy od wersji 2.0 używana jest nazwa „diagram maszyn stanowych” bądź „diagram maszyn stanów”. Język UML jest powszechnie znanym i stosowanym w wielu dziedzinach standardem (w informatyce, biznesie, automatyce, mechanice, itp.). Pozwala m.in. na modelowanie struktury, dynamiki, architektury.

Mapy stanów, posiadają możliwość modelowania procesów równoległych (współbieżnych) i hierarchii. Te dwa elementy są bardzo ważne w przypadku modelowania systemów reaktywnych. Są one realizowane poprzez stany złożone – w poszczególnych obszarach umieszcza się podmaszyny stanów.

W pracy proponuje się użycie diagramów statechart jako specyfikacji programów dla sterowników PLC. Stworzone diagramy można następnie bezpośrednio przekształcić do języka SFC lub ST. Brak jest jednak na rynku narzędzi inżynierskich oferujących możliwość bezpośredniej konwersji. Przedstawiono metodę konwersji diagramów statechart na program w języku SFC. Opracowanie narzędzi wspierających konwersję pozwoli projektantowi systemu na programowanie za pomocą diagramów statechart. Diagramy tworzone na potrzeby specyfikacji, mogą być wykorzystane jako dokumentacja projektu. Dokumentacja taka jest czytelna i zrozumiała dla nabywcy systemu sterowania.

2. Przykładowy system sterowania

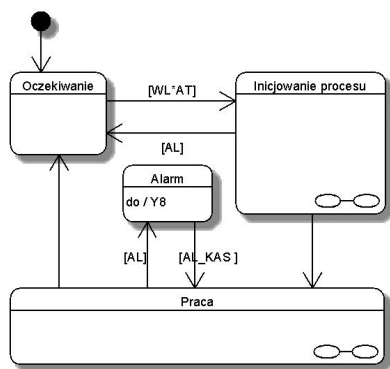
Przykład ten ilustruje sposób modelowania zarówno systemów z elementami współbieżnymi jak i sekwencyjnymi. Poprzez wprowadzenie sygnalizacji alarmowej przedstawiono możliwości wykorzystania stanów płytkowy wznowienia.



Rys. 1. Schemat procesu sterowania
Fig. 1. Diagram of control system

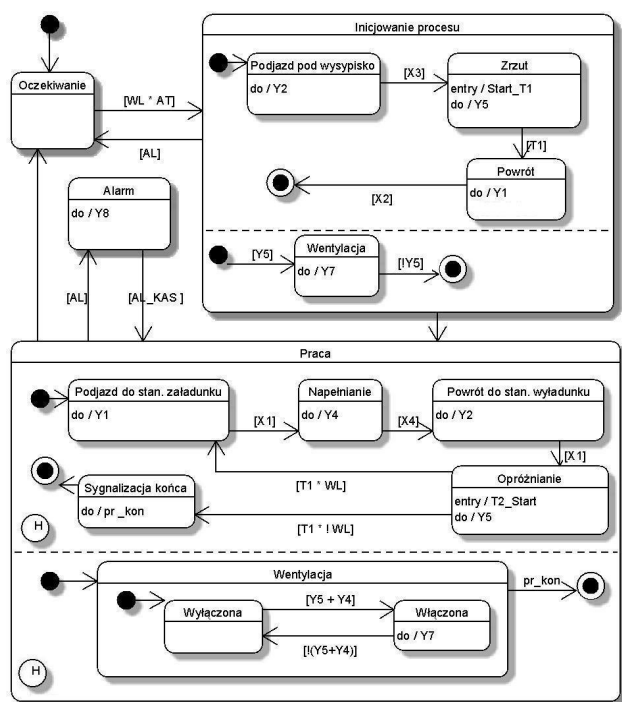
Po włączeniu systemu (WL), i rozpoczęciu procesu (AT) następuje przygotowanie wózka, poprzez skierowanie do stanowiska zrzutu odpadów i otworzeniu klapy. Podczas opróżniania i napełniania transportera, konieczna jest wentylacja pomieszczenia (Y7). Po zakończeniu opróżniania, wózek podjeżdża do stanowiska napełniania (X1). Stanowisko to wyposażone jest w wagę, która generuje sygnał X4, gdy zostanie pobrana właściwa ilość substancji. Napełniony wózek zostaje przekierowany do stanowiska wyładunku (X2) gdzie następuje jego opróżnienie.

W celu ilustracji modelowania na różnych poziomach hierarchii zostały stworzone dwa diagramy: Diagram pierwszy (rys. 2) ilustruje działanie systemu na najwyższym poziomie hierarchii. Diagram ten zawiera jedynie stany sekwencyjne. Nie oznacza to jednak, że proces sterowania nie ma ukrytych elementów współbieżnych.



Rys. 2. Program sterowania – diagram najwyższego poziomu
Fig. 2. Control program – top level diagram

Rozwinięciem jest diagram drugi rys.3. Stany złożone zostały na tym diagramie uzupełnione o podmaszyny stanów. Dodatkowo w stan złożony „Praca” został oznaczony symbolem płytkiego wznowienia. Takie rozwiązanie pozwala na kontynuowanie pracy w przypadku usunięcia drobnej awarii. Wznowienie procesu sterowania (produkcyjnego) od momentu, w którym się zakończył, minimalizuje straty materiałowe i czasowe.



Rys. 3. Program sterowania – szczegółowy diagram
Fig. 3. Control program – detail chart

3. Implementacja systemu sterowania

Diagramy statechart mogą być bezpośrednio przekonwertowane na program, dla sterownika PLC.

Języki dedykowane sterownikom logicznym określa norma IEC 1131-3. Proponuje się, aby wykorzystywać w procesie implementacji język SFC lub ST. Są to rekomendowane, algorytmiczne formy specyfikacji funkcjonowania sterowników logicznych. Warto zwrócić uwagę na fakt, że nowe pokolenie informatyków i automatyków preferuje specyfikacje formalne zamiast tradycyjnych schematów drabinkowych lub szczegółowych schematów blokowych. Z drugiej strony podobieństwo między diagramami SFC i sieciami Petriego sterowania [6] gwarantuje, że do analizy poprawności formalnej programów zostanie wykorzystana metodologia mająca oparcie w licznych pracach naukowych Instytutu Informatyki i Elektroniki UZ [2, 3, 10, 12].

Jednoznaczne i sprawne przekształcenie diagramu statechart na program w języku SFC jest możliwe tylko wtedy, jeśli projektant zachowa odpowiednią dyscyplinę programowania strukturalnego. W najprostszym przypadku odwzorowania stanów maszyny przyporządkowuje się kroki, a przejściom – tranzycje. W przypadku stanów złożonych (współbieżnych lub sekwencyjnych), poszczególne podmaszyny stanów przekształca się na podprogramy SFC. Połączenie otrzymanych w ten sposób podprogramów z programem nadrzędnym jest uzależnione od wykorzystywanego oprogramowania.

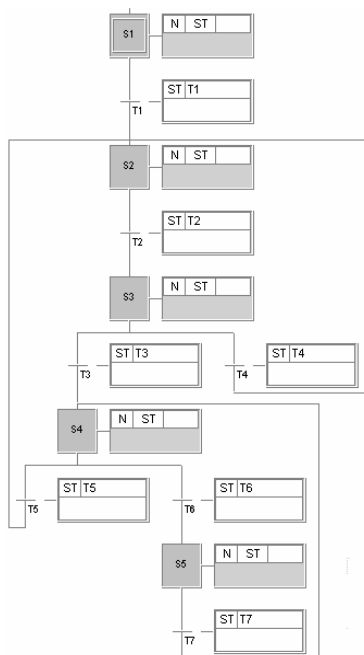
Niestety mimo prób całkowitej standaryzacji [9] oprogramowania wspierające programowanie sterowników PLC dedykowane jest częściowo do urządzeń konkretnej firmy. Producenci sprzętu nie zachowali całkowitej zgodności z normą IEC 1131. W przypadku oprogramowania STEP 7 firmy SIEMENS doraźnym rozwiązaniem może być „spłaszczenie” programu SFC do jednego poziomu hierarchii.

Rozwiązanie polegają na przekształceniu programu hierarchicznego na program płaski nie jest korzystne, ponieważ pociąga za sobą konieczność umieszczenia dodatkowych kroków i tranzycji, które powodują „rozrost” programu. Dodatkowym niekorzystnym zjawiskiem jest wprowadzenie nadmiarowych elementów służących jedynie do skoordynowania pracy poszczególnych podsieci, znajdujących się na różnych poziomach hierarchii. Sprawa komplikuje się w przypadku gdy program będzie wykorzystywany jednocześnie na sterownikach różnych producentów. Jeśli nie ma możliwości importowania programów, należy wprowadzić całą sieć sterowania ręcznie. W przypadku dużych systemów takie rozwiązanie jest utrudnione, a czasami wręcz niemożliwe.

Niedogodności płynące z zastosowania graficznego języka programowania można wyeliminować poprzez zastosowanie języka tekstowego. W takim przypadku wygenerowany kod programu może być skopiowany i wykorzystany do programowania sterownika innego producenta. Do dyspozycji są dwa języki tekstowe: IL oraz ST. Bardziej korzystnym rozwiązaniem wydaje się zastosowanie języka ST. W porównaniu do języka IL jest to język wyższego poziomu. Język ten umożliwia również za pomocą konstrukcji STEP.. END_STEP oraz TRANSITION tekstową reprezentację grafu sekwencji. Bezpośrednio przekształcić bezpośrednio model w postaci diagramu statechart na program w języku ST. Możliwe jest połączenie obu języków (SFC i ST). Z każdym krokiem sieci SFC można skojarzyć jedno lub więcej działań. Działanie może być określone poprzez zmienną boolowską, zestaw instrukcji IL, szczebli języka LD czy też poleceń w języku ST. Za pomocą języka ST można również specyfikować warunki realizacji przejść.

W proponowanym rozwiązaniu zastosowanie języka SFC lub ST zależy od poziomu hierarchii danej podmaszyny stanów. Proponuje się aby podmaszynę najwyższego poziomu przedstawić za pomocą sieci SFC. Graf sekwencji dla omawianego przypadku systemu sterowania został przedstawiony na rysunku. Ponieważ oprogramowanie STEP7 nie umożliwia przypisywania blokom działań instrukcji w języku ST, do badań wykorzystano bardziej uniwersalne oprogramowanie ISaGRAF.

Rysunek 4 przedstawia diagram sekwencji dla maszyny stanów przykładowego systemu sterowania. (rys. 2).



Rys. 4. Diagram SFC systemu sterowania
Fig. 4. SFC diagram of the system

Do bloków operacji wpisuje się opis funkcjonowania podsieci w języku ST. Alternatywnym rozwiązaniem może być odwołanie się do podsieci opisanej również w postaci graficznej diagramów SFC. Projektant w tym przypadku powinien przestrzegać zasad dotyczących transformacji hierarchicznej sieci Petriego na współpracujące ze sobą moduły [2, 3, 6]. Wskazana jest również rozpoznawanie podobieństw między SFC i diagramami maszyny stanów [4].

W celu uproszczenia schematu pominięto specyfikację warunków realizacji przejść. W tabeli 2 zestawiono poszczególne składowe diagramu sekwencji oraz odpowiadające im elementy diagramu statechart.

Przejścia T1, T3, T5 nie posiadają określonych warunków. Ich realizacja nastąpi w momencie zakończenia wykonywania operacji w bloku.

Wykorzystanie połączonych diagramów sekwencji i programu w języku ST pozwala na zachowanie hierarchicznej struktury programu sterowania. Ma to duże znaczenie w przypadku złożonych systemów, ponieważ umożliwia przejrzystą symulację projektu. Możemy spojrzeć na ogólny zarys systemu, bez wdawania się w szczegóły poszczególnych podprogramów.

Tab. 1. Opis diagramu sekwencji
Tab. 2. Description of SFC diagram

Diagram sekwencji	Diagram maszyny stanów
S1	Stan początkowy
S2	Oczekiwanie
S3	Inicjowanie
S4	Praca
S5	Alarm
T1	
T2	WL * A5
T3	
T4	AL
T5	
T6	AL
T7	AL_KAS

Proponowaną metodę zweryfikowano w środowisku STEP7 oraz ISaGRAF. Opracowano programy sterowania i przeprowadzono symulację ich działania. Środowisko STEP7 nie umożliwia tworzenia hierarchicznych diagramów sekwencji. W związku z tym opracowano najpierw podprogramy na bazie maszyn stanów

na najniższym poziomie hierarchii, a następnie połączono i uzupełniono o maszyny na wyższych poziomach. Otrzymane diagramy – ze względu na dość dużą wielkość – nie mogły być przedstawione w niniejszej pracy. Zauważono również, że oprogramowanie STEP7 pozwala na określenie warunków realizacji tranzycji jedynie za pomocą języka LD lub FBD. Jest to kolejne odstępstwo od standardu IEC 1131-3.

Alternatywnym rozwiązaniem do proponowanej programowej implementacji systemu sterowania jest zastosowanie syntezy układowej. System sterowania opisany diagramem maszyny stanowej może być zaimplementowany w strukturze reprogramowalnej (np. FPGA). Opracowana specyfikacja przekształcana jest na równoważny opis w języku VHDL [10].

4. Podsumowanie

Diagramy maszyn stanowych UML 2.0 są efektywną metodą modelowania programów dla sterowników PLC. Pozwalają również na tworzenie jednoznacznej i przejrzystej dokumentacji projektu. Dość często można się spotkać w literaturze z przykładami zastosowań diagramów maszyn stanowych zarówno do celów specyfikacji szeroko pojętych systemów reaktywnych [7, 10], jak i programów dla sterowników PLC [4]. Nie ma jednak określonej jednoznacznie formy translacji diagramu statechart na program dla sterownika PLC.

Z wśród dostępnych języków opisanych normą IEC [9], najbardziej popularne języki to SFC i ST. Określenie najważniejszych elementów diagramów maszyn stanowych i sposobu przekształcania na program w języku ST, pozwoliłoby na stworzenie narzędzi inżynierskich automatyzujących ten proces. Konieczne jest również stworzenie reguł konstruowania diagramów statechart: chodzi tu w szczególności o deterministyczne realizacje przejść, sposoby implementacji stanów wznowienia, komunikację wewnętrzną, zasady etykietowania, realizację wyłączeń i przejść bezwarunkowych.

5. Literatura

- [1] Marian Adamski, Małgorzata Chodań: Modelowanie układów sterowania dyskretnego z wykorzystaniem sieci SFC, Zielona Góra 2000
- [2] Marian Adamski, Andrei Karatkevich, Marek Węgrzyn (red) Design of embedded control systems, Springer, New York 2005
- [3] Grzegorz Andrzejewski, Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych, Oficyna Wydawnicza UZ, Zielona Góra 2003
- [4] Nanette Bauer, Sebastian Engell "A Comparison of Sequential Function Charts and Statecharts and an Approach towards Integration", INT 2002
- [5] Grady Booch, James Rumbaugh Ivar Jacobson: UML przewodnik użytkownika, WNT, Warszawa 2000
- [6] R. David, H. Alla: Petri Nets & Grafset. Tools for modelling discrete events systems, Prentice Hall, 1992
- [7] Daniel D. Gajski, Frank Vahid, Sanjiv Narayan, Jie Gong, Specification and Design of Embedded Systems, P T R Prentice Hall, New Jersey 1994
- [8] David Harel, Rzecz o istocie informatyki Algorytmika, WNT, Warszawa 2001
- [9] International Electrotechnical Commission, Draft IEC 1131-1: Programmable Controllers - Part3: Programming languages, IEC 1992
- [10] Grzegorz Łabiak, Wykorzystanie hierarchicznego modelu współbieżnego automatu w projektowaniu sterowników cyfrowych, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona góra 2005
- [11] Object Management Group, UML 2.0 Superstructure Specification, Revised Final Adopted Specification (ptc/04-10-02), www.uml.org
- [12] Agnieszka Węgrzyn, Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego, Oficyna Wydawnicza UZ, Zielona Góra 2003
- [13] Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski, Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice 2005