

Modelowanie przepływu informacji w rozproszonych systemach pomiarowych z wykorzystaniem sieci Petriego

Robert Łukaszewski

Artykuł dotyczy modelowania systemów pomiarowo-kontrolnych (SPK), uwzględniającego czasowe aspekty przepływu informacji w SPK. Zaproponowano nową metodę opisu modelu SPK z wykorzystaniem sieci Petriego i narzędzia CPN/Tools. Dodatkowo przedstawiono ogólny przegląd metod nieformalnych i formalnych stosowanych w projektowaniu systemów.

Information data flow in distributed measuring system modelling using Petri nets

The paper presents research on a modelling of design Measuring Systems (MS) with time limited data flow. A new method of describing MS model based on Petri Nets and CPN/Tools environment is proposed in the article. The article provides an overall description of informal and formal methods, which can be used while designing systems.

1. Wstęp

Zasady projektowania systemów pomiarowo-kontrolnych (SPK) obecnie mają charakter heurystyczny, zaś proces projektowania systemów pomiarowych zwykle bywa przeprowadzany za pomocą prostych metod inżynierskich, w których nacisk jest kładziony na aspekty związane z dokładnością pomiaru. Wraz z pojawieniem się potrzeby kontrolowania czasu trwania poszczególnych operacji pojawiła się konieczność sprawdzenia możliwości synchronizacji wykonywanych zadań oraz zbadania możliwości dotrzymania wymagań czasowych już na etapie projektu. Z tego też powodu celowe wydaje się tworzenie modelu projektowanego systemu, pozwalającego z dużą dokładnością określić cechy dynamiczne gotowego rozwiązania.

Celem artykułu jest przedstawienie metody tworzenia modelu SPK za pomocą sieci Petriego na przykładzie środowiska CPN/Tools.

2. Modelowanie SPK

Dotychczas podjęto wiele prób modelowania SPK, czy też ich elementów. W zastosowaniach inżynierskich modelowanie systemów pomiarowych często spro-

wadza się do wprowadzenia uproszczonych modeli przyrządów pomiarowych, złożonych z elementów skupionych. Modele te nie zawierają analizy czasowej, związanej z opóźnieniami wprowadzanymi przez poszczególne bloki. Z drugiej strony istnieją metody modelowania komunikacyjnego systemów przemysłowych, gdzie można ściśle określić czasy przepływu komunikatów w systemie. Metody te opierają się na teorii szeregowania zadań [1], metodzie najgorszego przypadku czy też na szeregowaniu GRMS [2]. Niestety w rozproszonych SPK (RSPK) metody te nie są wystarczające. Przyczyną jest niedeterministyczny charakter elementów typowych RSPK takich jak: system operacyjny, sieci komputerowe, protokoły komunikacyjne.

Przeгляд metod modelowania wykazuje, iż nie ma ogólnego sposobu opisu RSPK. Tworzone są modele znanych elementów składowych takiego systemu, takich jak czujniki czy sieci przemysłowe. W systemach o niskim poziomie rozproszenia, np. opartych na sieci dedykowanej, istnieją modele czasowe systemu zarówno dla warstwy akwizycji, jak i transportowej. Dla systemów o dużym stopniu rozproszenia, gdzie nie znamy dokładnie opóźnień wnoszonych przez sieć, takiego modelu nie ma. Z tego powodu konieczne jest opracowanie modelu łączącego znane sposoby opisu systemów opartych na sieci przemysłowej z modelami sieci rozległych, analizowanymi w telekomunikacji. Do stworzenia takiego modelu konieczny jest także pewien usystematyzowany język opisu systemu.

Robert Łukaszewski
– e-mail: r.lukaszewski@ire.pw.edu.pl
Politechnika Warszawska, Wydział Elektroniki
i Techniki Informatycznych Instytut Radioelektroniki

3. Aspekt czasu w SPK i RSPK

Źródłami opóźnień w SPK mogą być w zasadzie wszystkie bloki toru przetwarzania danych pomiarowych od bloku akwizycji danych po bloki prezentacji wyników. Dodatkowo źródłami opóźnień jest tor przesyłania sygnałów kontrolno-sterujących w SPK, co jest szczególnie widoczne i uciążliwe w RSPK. Sprzętowe przyczyny powstawania opóźnień są związane z bezwładnością elementów fizycznych oraz skończonym czasem wykonywania swoich działań. Opóźnienia sprzętowe powstają szeregowo i mogą być z reguły wyznaczone na podstawie danych katalogowych użytych elementów. Istotniejsze są opóźnienia o charakterze programowym i transmisyjnym.

Rozproszenie terytorialne SPK powoduje wzrost nakładów czasowych związanych z transmisją informacji. Zintegrowanie w jednym RSPK wielu zadań pomiarowych wiąże się z problemem synchronizacji poszczególnych operacji. W systemach synchronicznych podstawowym problemem jest zapewnienie jednolitości czasu w różnych węzłach systemu. Wiąże się to wprost z problemem zapewnienia jednoczesności wykonania pomiarów w wielu węzłach RSPK. Rozproszenie pozwala na współbieżne wykonywanie wielu operacji, co również powoduje, że czas wykonywania poszczególnych zadań jest ograniczony czasem wykonania innych wątków równoległych.

W systemach asynchronicznych główną trudnością związaną z istnieniem opóźnień w transmisji komunikatów jest zapewnienie założonej kolejności wykonywania zadań. Jest to szczególnie uciążliwe w RSPK wykorzystujących niedeterministyczne czasowo łącza komunikacyjne jak np. Internet. Skończony czas przesyłania wyników pomiarów oraz czas przesyłania komunikatów kontrolnych ogranicza szybkość wykonywania pomiarów. Szybkość ta często jest również uzależniona od wydajności kanałów transmisyjnych w szczególności od przepustowości samej magistrali pomiarowej w węzłach wykonawczych oraz od rzeczywistych przepustowości łączy telekomunikacyjnych. Zbyt mała przepustowość kanałów transmisyjnych w systemach może być w krytycznych przypadkach źródłem błędów wykonania oraz rzadziej błędów pomiaru spowodowanych utratą danych, a niekiedy także awarii systemu. Istnienie sprzętowych, programowych oraz transmisyjnych opóźnień w systemie jest źródłem powstawania błędów pomiaru wielkości zmiennych w czasie [3, 4].

4. Metody nieformalne i formalne specyfikacji i projektowania systemów z wymaganiami czasowymi

Istnieje wiele metod formalnych [5, 6] i nieformalnych [7, 8] stosowanych do opisu, projektowania i walidacji systemów z wymaganiami czasowymi. W większości metody te opierają się na pewnych modelach systemu

na różnych etapach jego wytwarzania. Nieformalne podejście do zagadnień wytwarzania oprogramowania sprowadza się na ogół do zastosowania pewnych heurystycznych wzorców do systematycznej analizy, specyfikacji i projektowania.

Większość stosowanych w przemyśle metod tradycyjnych jest odmianą koncepcji ogólnie zwanej analizą strukturalną. Analizy strukturalne skupiają się na analizie wymagań, zwanej popularnie perspektywą funkcjonalną wymagań. Analiza taka opiera się na diagramach przepływu danych (*Data Flow Diagram* – DFD) oraz diagramach relacji danych (*Entity-Relationship Diagrams* – ERD). Takie funkcjonalne modelowanie systemów jest pomocne jedynie w dowodzeniu użyteczności projektowanego systemu. W systemach krytycznych czasowo takie podejście jest niewystarczające. Projektanci takich systemów potrzebują dodatkowo metod opisu dynamicznych właściwości systemu. W odpowiedzi na te potrzeby opracowano wiele rozszerzeń klasycznej analizy strukturalnej. Efekty tych prac można znaleźć w metodykach projektowania systemów czasu rzeczywistego Warda-Mellora [7, 8], Hatley'a-Pirbhai'ego, ESML, MASCOT [8].

Inną grupą są nieformalne metody specyfikacji systemów oparte na językach obiektowych. Właściwości dynamiczne systemów uwzględniają metody projektowania systemów czasu rzeczywistego COMET [9] oraz ROPES [6a] wykorzystujące język UML, metodykę SaMI [8] opracowaną na bazie języka SDL, metodyki HOODTM/HRT-HOODTM [8] i LACATRE [8] używające rodzimych języków specyfikacji.

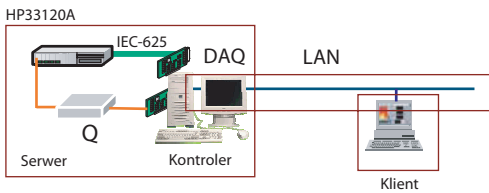
Istnieje klasa metod specyfikacji i projektowania zwanych ogólnie metodami formalnymi [10]. Formalne podejście do zagadnień specyfikacji, analizy i projektowania polega na sformułowaniu modelu konstruowanego systemu z zastosowaniem języka matematycznego, a następnie na systematycznym przekształcaniu uzyskanego opisu na drodze ściśle zdefiniowanych transformacji [11]. Języki formalne umożliwiają wykazanie zgodności opisu systemów (weryfikacja) [11], udowodnienie kluczowych właściwości systemu, jak brak zakleszczeń, czy brak spełnień [12], pozwalają również na **wykazanie spełnialności ograniczeń czasowych** w systemach [13].

Większość formalnych języków umożliwiających definiowanie czasowych właściwości systemów jest rozszerzeniem matematycznych form opisu systemów klasycznych (nieuwzględniających ograniczeń czasowych), np. czasowe algebry procesów [11], czasowe sieci Petriego [14], rozszerzenia czasowe logik temporalnych [5].

5. Modelowanie RSPK z wykorzystaniem Sieci Petriego

Sieci Petriego są postrzegane jako formalizm szczególnie nadający się do specyfikowania i analizy systemów cechujących się asynchronicznością i równoległością. Podstawowym wyróżnikiem sieci jest możliwość

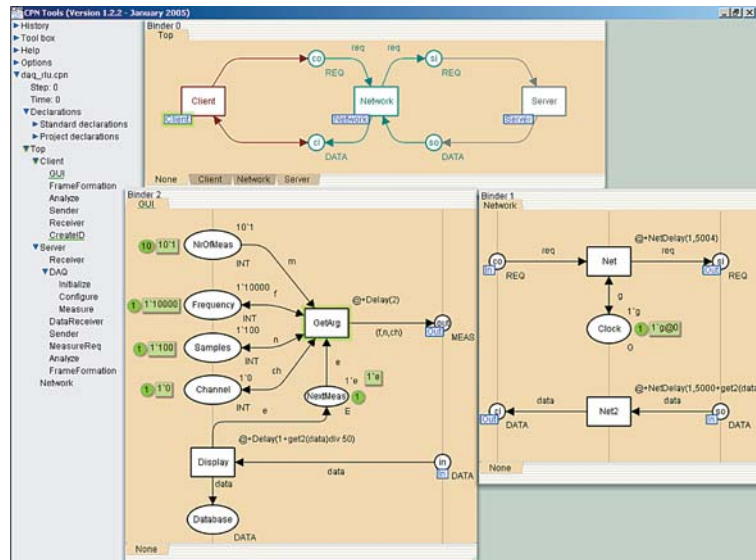
równoczesnej prezentacji przepływu sterowania i przepływu danych w systemach współbieżnych. Graficzną reprezentacją sieci jest dwudzielny graf składający się z dwu rodzajów węzłów zwanych odpowiednio miejscami (*places*) i przejściami (*transitions*). Przemieszczające się znaczniki (*tokens*) między węzłami sieci określają dynamikę systemu. Analiza podstawowych właściwości behawioralnych sieci [8, 16] umożliwia m.in. identyfikację przepełnień w systemie (np. buforów, rejestrów) czy identyfikację martwego kodu (brak zakleszczeń). W głównym obszarze zainteresowań przy konstruowaniu proponowanej metody modelowania SPK znalazły się czasowe kolorowane sieci Petriego (*Timed Coloured Petri Nets - TCPN*) [15] oraz hierarchiczne i stochastyczne rozszerzenia tych sieci.



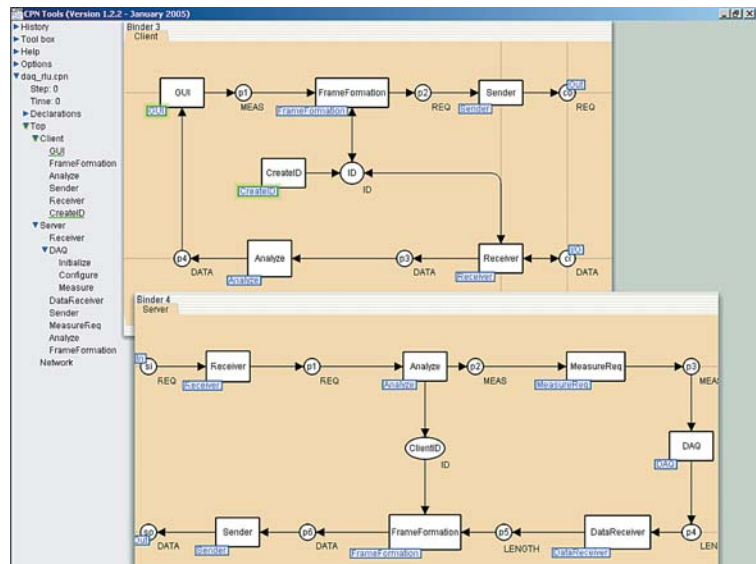
Rys. 1. Schemat modelowanego rozproszonego systemu pomiarowego
 Fig. 1. Diagram of modelled distributed measuring system

W środowisku CPN Tools [16] stworzono model rozproszonego systemu (rys. 1) o architekturze klient-serwer. Tworzenie modelu systemu w środowisku CPN/Tools polega na zdefiniowaniu hierarchicznej sieci Petriego określającej dynamiczne właściwości elementów systemu. Model wykorzystuje globalny zegar oraz zmienne i kolory czasowe. Blokowa perspektywa opisu systemu prowadzi do zdefiniowania modelu wysokiego poziomu (*supernet*) Top (rys. 2). Model składa się z trzech elementów wysokiego poziomu określających część klienta i serwera systemu oraz medium transmisyjne. Sieć zawiera cztery miejsca określające interfejsy sieciowe serwera i klienta systemu pomiarowego. Elementy systemu w modelu stworzonym w CPN/Tools zdefiniowano jako trzy substytucje tranzycji skojarzone z sieciami niższego poziomu (*subnet*): Client, Network oraz Server (rys. 2).

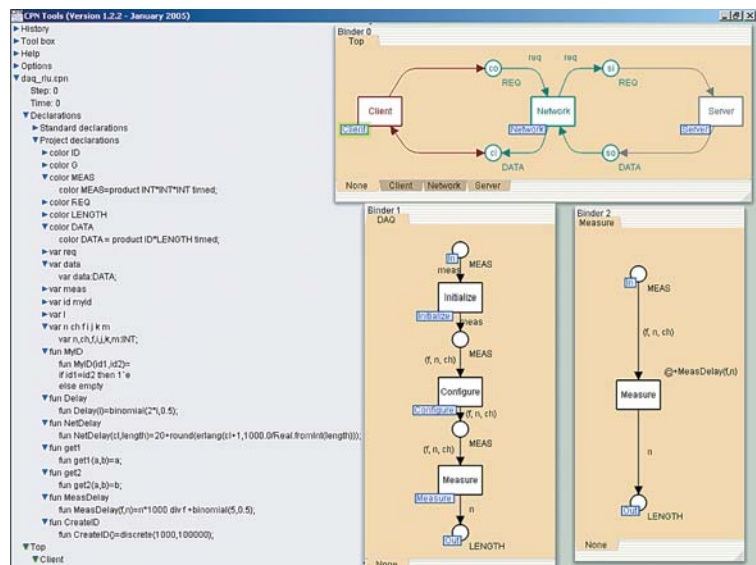
Zastosowano bardzo uproszczony model sieci (*Network*).



Rys. 2. Model RSPK w CPN/Tools (supernet Top, podsieci: GUI, Network)
 Fig. 2. DMS model in CPN/Tools (supernet Top, subnets: GUI, Network)



Rys. 3. Podsieci: Server i Client
 Fig. 3. Subnets: Server and Client



Rys. 4. Podsieci: DAQ i Measure
 Fig. 4. Subnets: DAQ and Measure

Czas transmisji komunikatów określono rozkładem prawdopodobieństwa w języku ML funkcją *NetDelay*: $\text{fun NetDelay}(cl, \text{length}) = 20 + \text{round}(\text{erlang}(cl+1, 1000.0/\text{Real.fromInt}(\text{length})))$, gdzie: *erlang* jest funkcją rozkładu *n*-Erlang. Część serwerowa i kliencka modelowanego SPK jest określona przez podsieci *Client* i *Server* (rys. 3).

Model klienta (podsieć *Client*) zawiera substytucje tranzycji skojarzone z sieciami niższego poziomu: *GUI*, *FrameFormation*, *Analyze*, *Sender*, *Receiver* i *CreateID* (rys. 3). Na przykład, model graficznego interfejsu użytkownika określa podsieć *GUI* (rys. 2). Model ten składa się z miejsc odpowiadających graficznym kontrolkom interfejsu klienta. Miejsca te w stanie początkowym są pobudzone przez znaczniki o wartościach odpowiadających wartościom kontrolki GUI klienta. Tranzycja *GetArg* określa czas pobrania argumentów z graficznych elementów, wykorzystując zdefiniowaną wcześniej funkcję *Delay* o postaci: $\text{fun Delay}(i) = \text{binomial}(2^i, 0.5)$, gdzie *binomial* jest funkcją rozkładu binomialnego.

Wartości pobranych parametrów (parametrów pomiaru) są następnie przekazywane poprzez port *out* do gniazda *PI* podsieci *Client*. Po wykonaniu pomiaru wartość zmierzona jest dostarczana na port *in* poprzez zdefiniowaną zmienną *data* o zdefiniowanym kolorze *DATA*. Miejsce *display* modeluje wykres na graficznym panelu interfejsu klienta. Czas wyświetlenia wyniku pomiaru określono jako: $\text{Delay}(1 + \text{get}(data) \text{div} 50)$.

Model części serwerowej systemu zawiera kilka podsieci definiujących poszczególne elementy tej części systemu. Na przykład, podsieć *DAQ* (rys. 4) modelująca akwizycję danych w systemie składa się z trzech substytucji tranzycji skojarzonych z sieciami niższego poziomu: *Initialize*, *Configure*, *Measure*.

Zbudowany model systemu pozwala na analizę właściwości dynamicznych SPK przy różnych wartościach parametrów czasowych elementów systemu. Możliwa jest symulacja awarii i analiza zachowania systemu w takich przypadkach. Model pozwala określić graniczne parametry czasowe przepływu informacji. Symulacja umożliwi wyznaczenie granicznych wartości szybkości akwizycji danych przy zadanych, ustalonych parametrach czasowych elementów systemu. Dalsze badania wymagają jednak dokładniejszego zdefiniowania modeli poszczególnych elementów SPK.

6. Podsumowanie

Coraz większa złożoność SPK oraz narzucone wymagania czasowe obligują konstruktorów do systematycznego podejścia do projektowania SPK, które wymaga zastosowania ścisłej metodyki. Określenie takiej metodyki wiąże się z koniecznością stosowania

czasowo-przeptywowych modeli przepływu informacji w systemie. Do stworzenia takiego modelu jest przydatne zastosowanie usystematyzowanego formalnego języka opisu systemu.

Rozszerzenia sieci Petriego pozwalają na modelowanie złożonych systemów z czasem w sposób zwięzły, bez zatracania szczegółowości opisu. Przedstawione dotychczasowe prace i badania wykazują przydatność stosowania sieci Petriego do modelowania SPK.

„Praca naukowa finansowana ze środków na naukę w latach 2004-2006 jako projekt badawczy”

7. Literatura

1. E. Michta, *Modelowanie komunikacyjne sieciowego systemu pomiarowo-sterującego*, WPZ, Zielona Góra 2000.
2. A. Markowski, *Wyznaczanie opóźnień transmisji danych w sieciowych systemach pomiarowo-sterujących*, PAR 7-8/2004, s. 95 – 99.
3. J. Jakubiec, *Błędy powodowane opóźnieniami w systemie pomiarowo-sterującym*, PAR 7-8/2004, s. 71 – 74.
4. H. Al-Raimi, J. Jakubiec, R. Żurkowski, *Błędy wyniku pomiaru w systemie czasu rzeczywistego spowodowane opóźnieniami transmisji*, Mat. Szkół konferencji MWK 2001, Rynia k. Warszawy, 21 – 24 maja 2001, Tom 2, s. 275-280.
5. A.M.K. Cheng, *Real-Time Systems, Scheduling, Analysis, and Verification*, John Wiley & Sons, 2002.
6. J. Carlson, *Languages and Methods for Specifying Real-Time Systems, MRTC (Malardalen Real-Time Research Center)*, Technical Report, August 2002.
7. K. Sacha, *Projektowanie oprogramowania systemów sterujących*, OWP W, Warszawa 1996.
8. T. Szmuc, *Modele i metody inżynierii oprogramowania systemów czasu rzeczywistego*, UWND AGH, Kraków 2001.
9. H. Gomaa, *Designing Concurrent, Distributed, and RT Applications with UML*, Addison-Wesley 2000.
10. <http://vl.fmnet.info/>, Virtual Library formal methods pages, November 2004
11. L. Chen, *Timed Processes: Models, Axioms and Decidability*, Ph. D. Dissertation, University of Edinburgh 1992.
12. T. Murata, *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, Vol. 77, No. 4, April 1989, pp. 541 – 580.
13. E. Vicario, *Static Analysis and Dynamic Steering of Time-Dependent Systems*, IEEE Transactions on Software Engineering, Vol. 27, No. 8, August 2001, pp. 728 – 748.
14. A. Cerone, A. Maggiolo-Shettini, *Time-Based Expressivity of Time Petri Nets for System Specification*, Theoretical Computer Science, Vol. 216, 1999, pp. 1 – 53.
15. K. Jensen, *Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use*, Vol. 1-3, Springer, 1996.
16. <http://www.daimi.au.dk/CPNTools/>, CPN Tools home page, May 2005. ■