

Spalvotųjų Petri tinklų taikymas valdymo sistemoms modeliuoti

S. Bartkevičius

Teorinės elektrotechnikos katedra, Kauno technologijos universitetas

Studentų g. 48, LT-3031 Kaunas, Lietuva, tel. +370 37 300253, el. paštas Stanislovas.Bartkevicius@ktu.lt

V. Mačerauskas, K. Šarkauskas

Valdymo technologijų katedra, Kauno technologijos universitetas

Studentų g. 48, LT-3031 Kaunas, Lietuva, tel. +370 37 300 276, el. paštas Kastytis.Sarkauskas@ktu.lt

Įvadas

Šiuolaikinės valdymo sistemos vis labiau kompiuterėja, t.y. vis daugiau svorio įgauna valdymo algoritmas. Netgi modernūs elektronikos prietaisai ar įrenginiai turi mikrokompiuterius, kurie valdo prietaisų darbą, t.y. tokie įrenginiai yra valdymo sistemos. Valdymo kompiuterių, valdiklių, programos, algoritmai darosi vis sudėtingesni ir jų patikimumo, stabilumo, patikrinimas tampa sudėtingu uždaviniu. Čia turima galvoje ne tik programavimo, algoritmo realizavimo klaidų aptikimas, bet ir funkcionavimo logikos sutikimas su valdymo tikslais. Ji nėra fiksuota, nes bendruoju atveju realias sistemas veikia atsitiktiniai trikdžiai ir valdymo sistemos reakcijos pobūdis dažnai priklauso nuo trikdžių ar įvykio sistemoje pobūdžio. Bėne dažniausiai pasitaikantis uždavinys, kurį tenka spręsti lanksčią valdymo logiką turinčioms sistemoms – pasirinkimas iš kelių galimybių, siekiant optimalaus rezultato. Optimalus rezultatas gali būti ne vietinio ar momentinio pobūdžio, o strateginis, kurį kiekybiškai įvertinti galima tik praėjus gana ilgam sistemos darbo laiko intervalui. Todėl kuriant tokias sistemas būtinas modeliavimo etapas, kurio metu ir sukuriamas valdymo algoritmas, tenkinantis keliamus reikalavimus. Valdiklių emuliatoriai netinka šiam tikslui, nes jie imituoja jau parengtos programos veikimą ir algoritmo abstraktesnę analizę čia sudėtinga. Minėtam tikslui geriau tiktų modeliavimo sistema, veikianti su abstrakčiu valdymo algoritmu, nesusietu su konkrečiu programos kodu. Be to, tokia valdymo sistema turi gebėti modeliuoti ir valdymo objektą bei jo ryšius su valdikliu. Kitą vertus, modeliavimo sistemos gautas rezultatas, algoritmo struktūra, turi būti nesunkiai verčiamas į konkretaus valdiklio programą.

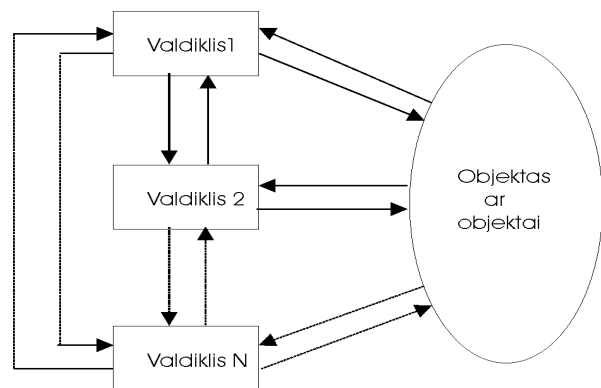
Šio darbo autoriai anksčiau minėtiems tikslams pasiekti pasirinko spalvotuosius Petri tinklus [1]. Šie tinklai puikiai tinka konkurenciniams (pasirinkimo) algoritmams modeliuoti [2,3,4]. Tinklo spalvų aprašams bei šakų ir pereinų reiškiniams užrašyti, remiantis [1], buvo pasirinkta programavimo kalba SML [5]. Pagrindiniai jos privalumai yra notacijų kompaktiškumas

ir griežta sintaksė, neleidžianti atsirasti neapibrėžtoms situacijoms.

Šiame darbe siekiama parodyti pagrindines prielaidas ir apribojimus, būtinus modeliuojant valdymo sistemas Petri tinklais. Visa tai yra patyrimo, įgyto kuriant ir taikant programų paketą CENTAURUS-C, apibendrinimas.

Interfeisas su objektu

Bendruoju atveju valdymo sistemą gali sudaryti keletas valdiklių ir valdymo objektas ar objektai (1 pav.).



1 pav. Valdiklių ir objekto ryšiai

Visas šias grandis sieja abipusiai ryšiai. Diskretiniai ryšiai tarp valdiklių yra nesunkiai modeliuojami jungtimis Petri tinkle. Objektas dažniausiai būna tolydinis, todėl jo negalima modeliuoti diskretiniu Petri tinklu. Programų pakete CENTAURUS-C valdymo objektas yra apibūdinamas struktūrine schema, įprasta valdymo teorijoje ir praktikoje. Struktūrinė schema iš esmės yra diferencialinių lygčių vaizdas. Yra darbų, kuriuose tolydinis objektas yra apibūdinamas diferencialinėmis lygtimis, kurios savo ruožtu yra specialių tolydinių pereinų atributai [9]. Tai verčia objekto lygčių sistemą išskaidyti dalimis, kas, ypač netiesinio objekto atveju, ne visada įmanoma. Todėl projektuojant paketą CENTAURUS-C buvo pasirinktas toks modeliavimo būdas, kai objektas modeliuojamas nepriklausomai nuo Petri tinklo (valdiklių

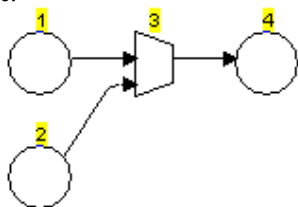
modelio), o tinkle numatomi tik *interfeiso su objektu* elementai. Iš esmės tai atitinka realią sistemą, kur objektas egzistuoja atskirai, o valdikliai susieti su juo per interfeisą.

Darbo su struktūrinėmis schemomis posistemis yra pavaldėtas iš paketo CENTAURUS [7]. Todėl šiame darbe apie jį daugiau nekalbama.

Autoriai siūlo Petri tinkle ryšiui su objektu naudoti specializuotą pereigą (2 pav., elementas 3). Šios pereigos aplinkoje naudojami pakete numatyti kintamieji pažymėti simboliu \$. Bendroju atveju ryšiui su objektu naudojamos pereigos aplinkos atributai yra tokie:

- a) kurios nors vienos įėjimo šakos reiškinyje REAL spalvos kintamajam \$Out priskiriama vertė;
- b) pereigos kontrolės (guard) reiškinyje naudojami du REAL spalvos kintamieji – \$In ir \$Prev.

Kintamieji \$Out, \$In ir \$Prev yra į objekto struktūra perduodamas signalas, iš objekto gaunamas signalas ir iš objekto gauto signalo ankstesniame modeliavimo žingsnyje vertė.



2 pav. Interfeisas tarp objekto struktūros ir Petri tinklo

Taip apibūdinamas procesas, kuriame vyksta abipusis pasikeitimas signalais tarp objekto modelio ir Petri tinklo. Signalas \$Out gali būti perduodamas į bet kurio objekto struktūros bloko bet kurią įėjimą, o signalai \$In ir \$Prev gali būti gaunami iš bet kurio struktūros bloko išėjimo. Struktūros įėjimo bei išėjimo blokų numeriai ir įėjimo numeris yra proceso atributai. Specifiškas ir pereigos veikimas – kontrolės reiškinys lemia pereigos atsідarymą, bet neturi įtakos signalo \$Out perdavimui į objektą. Bendroju atveju signalo \$Out perdavimas ir pereigos atsідarymas nėra sinchroniški. Toliau šitokia pereiga bus vadinama *procesu*. Procesui paleisti (inicijuoti) pakanka, kad visų įėjimo šakų reiškinų rezultatais, multiaibėmis, nusakomi žetonai būtų atitinkamose pozicijose (2 pav., elementuose 1 ir 2), o procesas pasibaigia, kai pereigos kontrolės reiškinys įgyja vertę TRUE. Procesą galima laikyti pereiga, kurios vėlinimas nėra apibrėžtas iš anksto.

Galimi ir specializuoti procesai – autonominis ir be atsako bei pusiau abipusis procesas. Pirmasis skirtas signalams iš objekto perduoti į Petri tinklą ir visada laikomas aktyviu. Procesas be atsako skirtas perduoti signalams į objekto struktūrą nelaukiant atsako, t.y. proceso pabaigos. Šis procesas taip pat visada laikomas aktyviu. Iš viso galimos keturios procesų modifikacijos, kurios surašytos 1 lentelėje.

Lentelė

Nr.	Proceso apibūdinimas	Naud. \$Out	Naud. \$In	Turi įėjimo šakų	Turi išėjimo šakų
1	Autonominis	-	+	-	+
2	Pusiau abipusis	-	+	+	+
3	Abipusis	+	+	+	+
4	Be atsako	+	-	+	-

Bandymas paleisti nepasibaigusį abipusį ar pusiau abipusį procesą laikomas klaida.

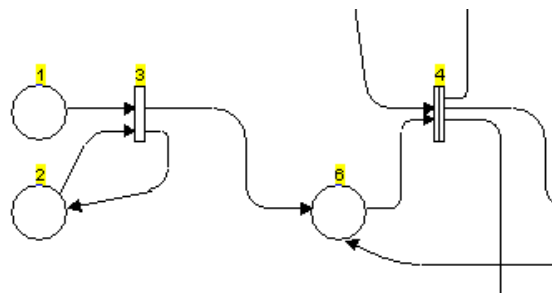
Laikas Petri tinkle

Laikas spalvotuose Petri tinkluose įvertinamas trejopai – jis gali būti žetono [1], pozicijos [8] arba pereigos [6] atributas. Pirmieji du atvejai patogūs, kai reikia modeliuoti, pavyzdžiui, transportavimo sistemas.

Valdymo sistemoje visi procesai vyksta laikui bėgant. Jei valdymo objekto laiko pastoviosios yra didelės (pvz., šiluminiai įrenginiai), tai valdiklio programos sugaištamasis laikas gali būti ir nevertinamas. Tačiau kitais atvejais, kai, pavyzdžiui, objektas yra elektroninis įrenginys, programos gaištamasis laikas yra tos pačios eilės kaip ir procesų objekte trukmė. Bendroju atveju modelyje tenka įvertinti vėlinimus valdymo objekte ir valdiklyje. Tiek valdymo objektą, tiek valdymo algoritmą galima laikyti visuma *procesų*, kurių pradžia ir pabaiga susijusi su *įvykiais* valdymo sistemoje. Jei įvykiu laikysime vienos ar keleto pereigų atsідarymą sistemos Petri modelyje, tai racionali laiką (vėlinimus) susieti su pereigomis.

Modeliuojant valdymo sistemas laikas (laiko momentas) gali būti priežastis, sukelianti sistemoje naujus procesus. Todėl negalima taikyti [6] aprašyto pereigų vėlinimų normalizavimo pakeičiant realius vėlinimus normuotais sveikaisiais skaičiais, išreiškiamais dydžiais, normavimo baze laikant mažiausiąjį vėlinimą.

Modeliavimo pakete CENTAURUS-C laikas yra globalus dydis apibūdinamas kintamuoju \$Time, kurio vertę vartotojas gali tik skaityti. Laiko inkrementą sukelia pereigų aktyvūs vėlinimai, procesai arba, jeigu tokie yra, kintamąjį \$Time turintys pereigų ar šakų reiškiniai. Laiko skaitiklio inkrementas Δt , kuris turi būti daug mažesnis už mažiausią pereigos vėlinimą ar proceso trukmę, ir apibūdina laiko vaizdavimo modelyje tikslumą.



3 pav. Vėlinančioji ir nevēlinančioji pereigos

Vėlinančioji ir nevēlinančioji pereigos vaizduojamos skirtingais simboliais – vėlinimas žymimas papildomu brūkšniu pereigos simbolio viduje (3 pav., elementas 4).

Jeigu pereiga yra vėlinančioji, tai įprastinės pereigos atsідarymo sąlygos tik pradeda vėlinimo skaičiavimą. Pereiga atsідaro, kai baigiasi vėlinimas. Dažnai pereigos vėlinimas nėra pastovus dydis, todėl pakete CENTAURUS-C pasiūlytas pereigų vėlinimų apibūdinimas REAL spalvos reiškinium, kuriuo ir apskaičiuojama vėlinimo trukmė jo pradžios metu. Nevēlinančioms pereigoms nebūdingas ir šis reiškinys.

Formalus tinklo veikimo, esant vėlinimams, aprašas

Spalvotieji Petri tinklai *CPN* apibūdinami tokiu rinkiniu [1]:

$$CPN = (\Sigma, P, T, A, N, C, G, E, I);$$

čia Σ – baigtinė, ne tuščia spalvų (duomenų tipų) aibė; P – baigtinė pozicijų aibė; T – baigtinė pereinų aibė; A – baigtinė, ne tuščia jungčių aibė; N – mazgų funkcija, siejanti kiekvieną jungtį su pora (pozicija, pereiną) arba (pereiną, pozicija); C – pozicijų spalvų funkcija, siejanti kiekvieną poziciją iš aibės P su spalva iš aibės Σ ; G – pereinų kontrolės reiškiniių (funkcijų) aibė; E – jungčių reiškiniių (funkcijų) aibė; I – inicijavimo reiškiniių (funkcijų) aibė.

Toliau aibių elementai žymimi mažosiomis aibės, kuriai jie priklauso, simbolių atitinkančiomis raidėmis, pvz., a yra jungčių aibės A elementas.

Simboliu M pažymėjus tinklo žymėjimą – multiaibę, apimančią pozicijose esančius žetonus, o einamąjį tinklo žingsnį – Y , Petri tinklo veikimą galima nusakyti taip :

a) pradinis tinklo žymėjimas:

$$\forall p \in P: M_0(p) = I(p); \quad (1)$$

b) $n+1$ -asis tinklo žymėjimas:

$$\forall p \in P: M_{n+1}(p) = \left(M_n(p) - \sum_{(t,b) \in Y} E(p,t) \right) + \sum_{(t,b) \in Y} E(t,p) . \quad (2)$$

Simbolis $$ prie į pereiną įeinančios šakos reiškinio $E(p,t)$ ir iš pereinų išeinančios šakos reiškinio $E(t,p)$ žymi pereinų atsідarymo faktą.

Tinklo aprašą papildykime dar dviem aibėmis – vėlinimų skaičiavimo reiškiniių (funkcijų) aibe DE ir pereinų vėlinimu aibe D . Einamasis Petri tinklo žingsnio aprašas tuomet apibūdinamas papildomu nariu, kuris atitinka pereiną, atsідarančias, kai pasibaigia anksčiau inicijuoti jų vėlinimai:

$$\forall p \in P: M_{n+1}(p) = \left(M_n(p) - \sum_{(t,b) \in Y} E(p,t) \right) + \sum_{(t,b) \in Y} E(t,p) + \sum_{(t,b) \in Y} E(t,p) <D(t) <= 0>. \quad (3)$$

Pereinų vėlinimų inicijavimas apibūdinamas taip:

$$\forall t \in T: D(t) = DE(t) . \quad (4)$$

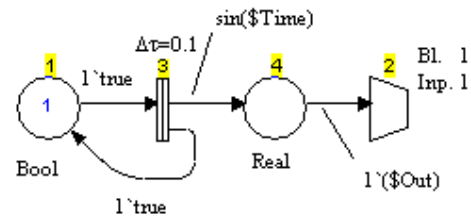
Konkurencinis pereinų aktyvinimas

Spalvotuosiuose Petri tinkluose galimas ir konkurencinis pereinų aktyvinimas. Konkurenciniu vadinamas toks atvejis, kai pereinai atsідaryti tinka keli vienoje pozicijoje esantys žetonai ar kelios skirtingose pozicijose esančių žetonų kombinacijos. Įprastinėje Petri tinklų sintaksėje nėra jokių apribojimų, draudžiančių konkurenciją. Valdymo sistemose konkurencinis pereinų atsідarymas gali sukelti neapibrėžtas situacijas, nes, atsідarius pereinai, iš įėjimo pozicijų pašalinami visi

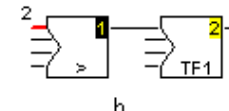
konkuruojantys žetonai [1] arba kombinacija pasirenkama atsitiktinai [9]. Tokiu būdu galima ne tik prarasti informaciją, bet konkurencinis pereinų aktyvinimas reiškia ir *neišspręstą konfliktą*. Spalvotuosiuose Petri tinkluose vartotojo deklaruoti kintamieji gali įgyti konkrečias vertes tik įėjimo šakų reiškiniiuose – *konstruktoriuose*. Konstruktoriumi vadinamas rinkinys, sudarytas tik iš kintamųjų. Kintamieji naudojami jungčių ir pereinų reiškiniiuose. Konkurencinio pereinų aktyvinimo atveju kintamųjų vertės yra neapibrėžtos, tiksliau – daugiareikšmės, todėl pakete CENTAURUS-C konkurencinis pereinų atsідarymas uždraustas ir laikomas tinklo funkcionavimo klaida.

Interfeiso elementų ir vėlinimo modeliuose pavyzdžiai

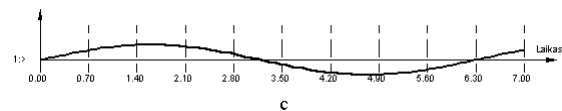
Pirmasis pavyzdys skirtas parodyti, kaip *procesas be atsako* gali būti panaudotas informacijai į objektą perduoti (4 pav., a). Šitoks procesas gali būti keitklio iš kodo į analoginį signalą atitikmuo. 1-oji pozicija ir 3-ioji pereiną, kurios vėlinimas $\Delta\tau = 0,1$, sudaro generatorių. Periodiškai atsідarant 3-iajai pereinai jos išėjimo į 4-ąją poziciją šakos reiškinio rezultatas yra sinuso funkcijos verčių, kurios atsimenamos 4-ojoje pozicijoje, seka. Šios vertės šakos jungiančios 4-ąją poziciją su 2-uoju procesu reiškiniije (konstruktoriuje) priskiriamos kintamajam \$Out, t.y. perduodamos į analoginės struktūros (4 pav., b) 1-ojo bloko 1-ąjį įėjimą. Tokiu būdu šio bloko išėjime gaunamas laiptuotas artimas sinusoidai signalas (4 pav., c), kuris, siekiant gauti sklandžią sinusoidę, gali būti filtruojamas pirmosios eilės inercine grandimi (4 pav., c, 2-asis blokas).



a



b

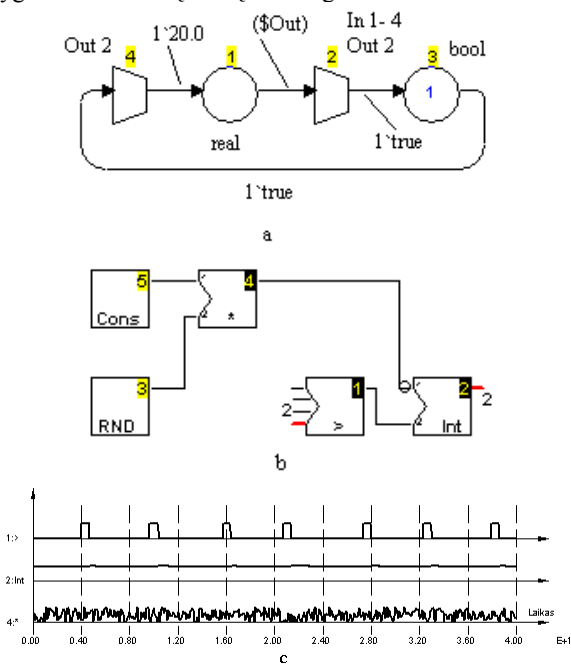


c

4 pav. Procesas be atsako – sinusinio signalo šaltinis: a – Petri tinklas, b–analoginė struktūra, c–signalas 1-ojo struktūros bloko išėjime

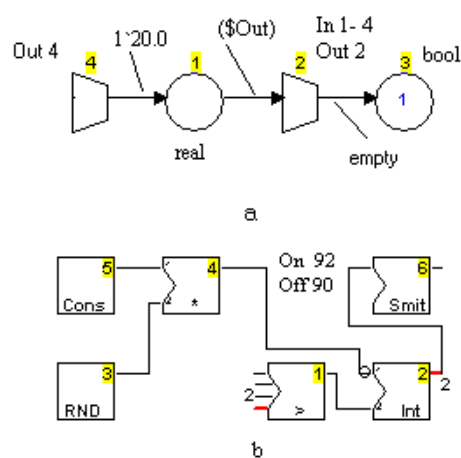
Antrasis pavyzdys modeliuoja vandens lygio palaikymo vandentiekio bokšte valdymo sistemą (5 pav.,a). Objekto struktūra parodyta 5 pav., b. Vandentiekio bokšto talpa modeliuojama integratoriumi (3 blokas), jo išėjimo signalas proporcingas vandens lygio aukščiui. Atsitiktines vandens sąnaudas modeliuoja 3,4 ir 5 blokai. Vandens papildymo siurblių atitinka 1-asis blokas.

Valdiklio modelį sudaro vieno kontūro Petri tinklas. Prasdėdamas 2-asis procesas perduoda \$Out signalą į objekto struktūros 1-ojo bloko 4-ąjį įėjimą, t.y. paleidžia siurblių. Šis procesas pasibaigia, kai vandens lygis pasiekia 100% lygį, t.y. kai jo kontrolės reiškinio $\$In > 100,0$ rezultatas tampa TRUE. Tada į 3-iąją pozicija dedamas žetonas, kurio vertė yra TRUE, ir leidžia veikti 4-ajam procesui. Šis procesas yra pusiau abipusis, nes neperduoda signalo į objekto struktūrą. Šis procesas (pereiga) atsidaro ir perduoda siurblio našumą apibūdinantį žetoną į 1-ąją poziciją tada, kai vandens lygis nukrinta žemiau 90% lygio, t.y. kai proceso kontrolės reiškinys $\$In < 90,0$ įgyja vertę TRUE. Paskui jau galimas 2-asis procesas ir t.t. 5 pav, c, parodyti siurblio darbo, vandens lygio ir vandens sąnaudų kitimo grafikai.



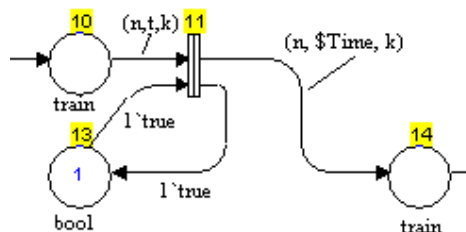
5 pav. Vandentiekio bokšto lygio reguliatorius: a – Petri tinklas, b–analoginė struktūra, c–signalai 1-ojo, 2-ojo ir 4-ojo struktūros blokų išėjimuose

Trečiasis pavyzdys modeliuoja tą pačią sistemą kaip ir 2-ajame pavyzdyje, tik čia įdiegtas valdymas su valdiklio *pertraukimais*. 4-asis *autonominis* procesas perduoda signalą į 1-ąją poziciją tik tada, kai tenkinamas jo kontrolės reiškinys $\$Prev > \In . Kol nėra šio signalo, valdiklio modelyje niekas nevyksta. Signalai $\$Prev$ ir $\$In$ yra vienu simuliacijos žingsniu laiko atžvilgiu nutolusios viena nuo kitos dvi Shmidto trigerio (6 pav., b, 6-asis blokas) išėjimo signalo vertės. Kontrolės reiškinys tenkinamas, kai trigeris pereina iš aukšto lygio į žemą. Tai atsitinka, kai vandens lygis nukrinta žemiau 90%. Toliau sistema veikia kaip ir 2-ajame pavyzdyje atsidarius 4-ajam procesui. 3-ioji pozicija šiame pavyzdyje reikalinga tik tam, kad 2-asis procesas būtų abipusis, todėl 2-ąjį procesą ir 3-iąją poziciją jungiančios šakos reiškinys yra *empty*, t.y. tuščia aibė. Vandens lygio ir vandens sąnaudų bei siurblio darbo grafikai tokie pat kaip ir antrajame pavyzdyje, todėl nerodomi. Analogiškai gali būti modeliuojama vandens pakėlimo sistema su keliais siurbliais [10].



6 pav. Vandens lygio reguliatorius su pertraukimais

Petri tinklus galima naudoti ir traukinių maršrutams geležinkelio sudaryti [11]. Todėl ketvirtasis pavyzdys yra geležinkelio ruožo modelis (7 pav.). Šiuo ruožu važiuojančių traukinių greitis yra skirtingas ir priklauso nuo traukinio tipo (keleivinis ar prekinis). Traukinys apibūdinamas *train* spalva (Color train = product int*real*kind), o traukinio tipas – spalva *kind* (color kind = with kel | prek).



7 pav. Geležinkelio ruožo modelis

10-oji ir 14-oji pozicijos atitinka stotis. 13 –oji pozicija su jos ryšiais neleidžia įvažiuoti į jau užimtą ruožą. Traukinio ruožo sugaištamas laikas yra pozicijos vėlinimas, kuris priklauso nuo traukinio tipo taip: keleivinio traukinio vėlinimas sudaro 63, o prekinio – 128 laiko vienetus. Todėl 11-osios pereigos vėlinimo reiškinys yra toks: if k=kel then 63,0 else 128,0. Šakos tarp 10-osios pozicijos ir 11-osios pereigos reiškinys yra konstruktorius ir kintamasis *k* įgyja traukinio tipo vertę tik tuomet, kai 10-ojoje pozicijoje yra žetonas, atitinkantis traukinį.

Išvada

Spalvotaisiais Petri tinklais, naudojant pasiūlytą interfeisą su objektu ir laiko įvertinimo būdą, galima modeliuoti sistemas su įvairiarūšiais dinaminiais objektais ir vėlinimais bei valdymo uždavinius, esant asinchroninei procesų eigai.

Literatūra

1. **Jensen Kurt.** Coloured Petri Nets, basic concepts // Springer, 1997.
2. **Allam M., Alla H.** Modelling Production Systems by Hybrid Automata and Hybrid Petri Nets // Proc. Of the IFAC-IFIP-

- IMAC's Conf. On Control industry systems (Belfort, France), May 1997. - P. 463-469.
3. **Di Febraro, Sacone S.** Hybrid Modelling of Transport Systems by means of Petri Nets // Proc. Of 1998 IEEE Int. Conf. On Systems, Man and Cybernetics (San Diego, CA, USA). - P. 131-135.
 4. **Finkel A., Purushothaman Iyer S., Sture G.** Well abstracted transition systems // Proc. of 11 th Int. Conf. Concurrence Theory (CONCUR'2000)(PSU State College, PA, USA), Aug. 2000, Lecture Notes in Computer Science. Springer, 2000.
 5. **Harper Robert.** Introduction to standard ML // School of Computer Science Carnegie Mellon University, Pitsburg.-1993.
 6. **Aura Tuomas, Lilius Johan,** Time processes of time Petri nets // Proc. of 18th Int. Conf. on Application and Theory of Petri Nets (ATPN), Toulouse, June 1997, P. - 136-155.
 7. **Bartkevičius S., Šarkauskas K.** Simulation of complicated Non-linear Systems by Heterogeneous Structures // Elektronika ir elektrotechnika – Kaunas: Technologija, 2002. - Nr. 7(42). - P.56-62.
 8. **Zhibin Jiang, Ming J. Zuo, Richard Y.K. Fung, Paul Yiliu Tu.** Temporized Coloured Petri Nets with Changeable Structure for Performance Modelling of dynamic production Systems // Int. J. of Production Research – Taylor & Francis Ltd. 2000. - Vol 38, No. 8. - P. 1917-1945.
 9. **Wieting Ralf.** Hybrid High-level Nets // Proc. of 1996 winter Simulation Conf., Coronado, California, USA, Dec. 1996.
 10. **Jančaitis E., Mačerauskas V., Šarkauskas K.** Petri tinklai ir jų modeliavimas paketu CENTAURUS. // Konferencijos „Automatika ir valdymo technologijos – 2001“ pranešimų medžiaga. – Kaunas: Technologija. – 2001. – P. 186-190.
 11. **Mačerauskas V., Rudzikas R., Švėgžda O.** Traukinio ir manevro maršrutų sudarymo vaizdavimo monitoriujė programa. // Konferencijos „Transporto technologijos“ pranešimų medžiaga. – Kaunas: Technologija. – 1994. – P. 54-60.

Pateikta spaudai 2003 02 05

S. Bartkevičius, V. Mačerauskas, K. Šarkauskas. Spalvotųjų Petri tinklų taikymas valdymo sistemoms modeliuoti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2003. – Nr. 4(46). – P. 7-11.

Šiuolaikinės valdymo sistemos daugiausia yra hibridinės – tolydiniai ar mišrūs objektai valdomi diskretiniais valdikliais. Valdymo algoritmui modeliuoti puikiai tinka spalvotieji Petri tinklai. Tolydiniai objektai aprašomi diferencialinėmis lygtimis, kurias patogiau apibūdinti struktūrinėmis schemomis. Ypatingą reikšmę įgauna interfeisas tarp šių modelio dalių ir laiko įvertinimas, nes laikas (laiko momentas) gali būti vienas iš valdančiųjų faktorių. Darbe aptariama modeliavimo sistema, realizuota programų pakete CENTAURUS-C. Ryšiui tarp tolydinio objekto modelio struktūros ir Petri tinklo naudojami trys sistemoje numatyti kintamieji – \$Out, \$In ir \$Prev. Šiais kintamaisiais Petri tinkle operuoja specializuotos pereinamosios, vadinamos procesais, kurios gali perduoti kintamojo \$Out vertę į objekto struktūrą kaip šuolinį signalą ir priimti bet kurio nurodyto struktūros bloko išėjimo signalą kaip kintamojo \$In vertę. Kintamojo \$Prev vertė yra ankstesnio modeliavimo žingsnio metu atsiminta \$In vertė. Kintamasis \$Out naudojamas įėjimo į procesą šakų reiškinuose ir inicijuoja proceso pradžią, o \$In ir \$Prev – proceso kontrolės reiškinyje naudojami proceso pabaigai, t.y. specializuotos pereinamosios atsaidymui, nustatyti. Laikas pasiūlytoje sistemoje yra procesų ir pereinamųjų atributas - vėlinimas. Pereinamųjų vėlinimams apibūdinami specialiais reiškiniais, todėl gali būti keičiami dinamiškai. Il. 7, bibl. 11 (lietuvių kalba; santraukos lietuvių, anglų ir rusų k.).

S. Bartkevičius, V. Mačerauskas, K. Šarkauskas. Using Colored Petri Nets for Simulation of Control Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2003. – No. 4(46). – P. 7-11.

Modern control systems are hybrid – continuous or mixed objects are controlled by discrete controllers. Colored Petri nets are the excellent mean of simulation of control algorithms. Continuous objects can be described by differential equations. These equations can be represented as a structural scheme. An interface between those to parts of a model is very important as well as evaluation of the time, because it can be one of the factors of control. The simulation system realized in the program package CENTAURUS-C is presented. Interface between both parts of a model is realized via specialized transition, in Petri net, named – process. Three default real variables – \$Out, \$In and \$Prev are used in the interface. A process can carry the value of \$Out into a structure of an object as a step signal and receive signal from any output of blocks in structure as a value of the variable \$In. Value of variable \$Prev is a value of \$In remembered on previous step of simulation. Variable \$Out can be used in an expression of input chord of a process and init the process. Variables \$In and \$Prev can be used in guard expression of a process to set the end of a process – firing of the specialized transition. Time in the presented system is an attribute of processes and transitions – delay. The delays of transitions are represented by special expressions and can be changed dynamically. Ill. 7, bibl. 11 (in Lithuanian; summaries in Lithuanian, English, Russian).

С. Барквявичюс, В. Мачераускас, К. Шаркаускас. Применение цветных сетей Петри для моделирования систем управления // Электроника и электротехника. – Каунас: Технология, 2003. – № 4(46). – С. 7-11.

Современные системы управления являются гибридными – непрерывными или смешанными объектами управляют дискретные контроллеры. Для моделирования алгоритмов управления отлично подходят цветные сети Петри. Непрерывные объекты описываются дифференциальными уравнениями, которые могут быть представлены структурными схемами. Особое значение имеет интерфейс между этими двумя частями и способ представления времени, так как последнее является одним из факторов управления. Представлена система моделирования, реализованная в программном пакете CENTAURUS-C. Интерфейс между структурными схемами и сетью Петри обеспечивается через специализированные передачи сети названные процессами. Для интерфейса используются три предопределенные переменные – \$Out, \$In и \$Prev. Процесс может передать значение переменной \$Out в структуру объекта в виде скачкообразного сигнала и принимать сигнал от любого блока структурной схемы и качестве значения переменной \$In. Переменная \$Prev представляет значение переменной \$In запомненной во время предыдущего шага моделирования. Переменная \$Out используется в выражениях входных хорд передачи процесса и начинает процесс, а \$In и \$Prev используются в “guard” выражениях для определения конца процесса – открытия соответствующей передачи. Время в представленной системе является атрибутом передач – задержкой. Задержки передач определяются специальными выражениями и могут меняться динамически. Ил. 7, библи. 11 (на литовском языке; рефераты на литовском, английском и русском яз.).

DOI: 10.5755/j02.eie.11243