REȚELE PETRI COLORATE DINAMIC RECONFIGURABILE PENTRU MODELAREA PROCESELOR DE CALCUL ORIENTATE PE SERVICII

Iurie ȚURCANU, Emilian GUȚULEAC

Universitatea Tehnică a Moldovei

Abstract: *Rețelele Petri (RP) sunt un formalism de modelare şi analiză a sistemelor concurente cu evenimente discrete. Oricum, acestea nu oferă o cale directă care ar reprezenta schimbări dinamice de structură a modelului, o paradigmă foarte importantă la proiectarea sistemelor de calcul orientate pe servicii. În lucrare sunt definite RP colorate (RPC) reconfigurabile marcaj – controlabile (RPCR), care permit de a obține un model mai compact ce descrie funcționarea sistemelor orientate pe servicii, unde în mod dinamic sunt prezente schimbări de structură.*

Cuvinte cheie: *modelare, rețele Petri colorate, reconfigurabilitate, sisteme orientate pe servicii*

1. Introducere

Actualmente, sistemele de calcul cu arhitecturi orientate pe servicii (SCOS) în timp real cunosc o dezvoltare rapidă, atât sub aspectul complexității şi/sau performanţelor, cât şi al ariei de răspândire [1, 4]. Acest tip de sisteme trebuie să aibă o flexibilitate, disponibilitate şi siguranță în funcționare deosebită. Astfel, un sistem SCOS poate fi considerat şi implementat drept fiind o colecție de configurații, unde fiecare din acestea este o rețea de componente ce comunică între ele. Diferite configurații pot fi folosite pentru procesarea a diferitor servicii sau condiții de operare ale aplicațiilor. Ca urmare, facilitățile trecerii în timp real de la o configurație către alta pe parcursul rulării, duc la creşterea s*iguranței în funcționare* şi a flexibilității utilizării sistemului [2, 3, 6], asigurarea cărora în timpul reconfigurării dinamice este dificilă din cauza interacțiunii între serviciile de aplicație care sistemul le oferă utilizatorului.

Unul dintre cele mai răspândite formalisme moderne, folosite pentru modelarea şi analiza sistemelor paralele/distribuite cu evenimente discrete, sunt rețelele Petri (RP) de diferite extensii [4, 5]. Totodată, apare necesitatea de a dezvolta aceste formalisme pentru a descrie mai adecvat, mai flexibil şi mai comod sisteme SCOS cu structuri complexe dinamic restructurabile. În [3, 6], au fost introduse RP *reconfigurabile* (RPR), pentru simularea, verificarea şi analiza sistemelor concurente care în mod dinamic sunt supuse schimbărilor de structură. Însă pentru sisteme reale modelele RPR sunt prea vaste şi dificil de elaborat. Pe de altă parte, RP colorate (RPC) au un limbaj orientat grafic care oferă un cadru eficient pentru proiectarea, specificarea, simularea, validarea, verificarea şi implementarea sistemelor cu evenimente discrete vaste [5]. Mai mult, există multe instrumente software care susțin specificațiile lor, simularea şi analizele formale, cum ar fi CPN Tools [5] care au fost utilizate în mai multe proiecte industriale. RPC combină puterea de modelare a RP obişnuite cu cea a limbajelor de programare.

În lucrare sunt considerate unele aspecte de modelare şi analiză a proceselor de calcul orientate pe servicii reconfigurabile prin RPR colorate.

2. Rețele Petri colorate

Rețelele RPC [5] sunt, în esență, o extensie de RP care oferă posibilitatea de a modela sisteme SCOS într-un mod mult mai compact decât RP obişnuite, deoarece ele folosesc mecanisme de nivel înalt similare cu cel al limbajelor de programare: cu fiecare jeton este asociat o valoare a unui anumit tip de date (set de culori). Într-o RPC o locație poate să conțină jetoane de diferite culori, iar o tranziție poate fi declanşată în diferite moduri în conformitate cu culoarea selectată. Acest fapt este realizat prin ataşarea unui domeniu de culori la fiecare locație şi fiecare tranziție. Astfel, pentru o mulțime de locații şi tranziții identice, numărul de comportamente ce pot fi exprimate de o RPC este cu mult mai mare decât cel prin RP obişnuite.

Un arc ce conectează o locație (tranziție) cu o tranziție (locație) este etichetat cu expresie ce este o funcție de culoare marcaj dependentă. Această funcție determină pentru fiecare culoare a tranziției (sau instanță de declanşare a tranziției) numărul de jetoane de culoarea aleasă ce trebuie consumate sau produse în locația respectivă la declanşarea tranziției relativ la culoarea selectată. Astfel, o rețea RPC este constituită din: a) locații, tranziții şi arce în acelaşi mod cum şi RP obişnuite; b) jetoane individuale diferențiabile unele fată de altele, de exemplu prin culori, de unde au numele de rețele colorate; c) un marcaj inițial ce indică pentru fiecare locație numărul şi tipul de jetoane pe care le conține; d) funcții de colorare ale arcelor ce fac referință la tipuri de jetoane date. Cum şi pentru RP obişnuite, alegerea culorii de declanşare a unei tranziții a RPC este efectuată în mod indeterminist: dacă o tranziție *t* este validată atât relativ la culoarea c_1, cât şi la culoarea c_2 se va selecta în mod întâmplător pentru a fi declanşată tranziția relativ numai la o singură culoare.

3. Rețele Petri colorate dinamic descriptiv -reconfigurabile

În acest compartiment, cu scopul de a trata unele probleme menționate la modelarea şi analiza proceselor de calcul orientate pe servicii reconfigurabile, definim un model de rețele *RPC dinamic descriptiv-reconfigurabile*, în care introducem o mulțime de reguli de rescriere colorate $R = \{r_1,..., r_k\}$ a rețelei *RPC* curente ce poate modifica atât marcajul curent, cât şi structura ei la ocurența unor evenimente specificate. O regulă de rescriere $r_j \in R$ este o generalizare a noțiunii de tranziție discretă $t_j \in T$, folosită în sens clasic. Condiția de validare de către marcajul curent M a unei tranziții $t_j \in T$ şi/sau reguli $r_j \in R$ este similară cu cea a unei *RPC*.

Fie $X\rho Y$ este o relație binară ρ între două mulțimi *X* şi *Y*, în care *domeniul* lui ρ este *Dom*(ρ) = ρY, iar *codomeniul* său este *Cod*(ρ) = $X\rho$.

Definiția 1**.** O multimulțime μ de elemente ale unei mulțimi finite nevide C este orice aplicație a lui C în mulțimea numerilor întregi naturale IN. Exprimăm $\mu = \sum_{x\in C} \mu(x).x$, unde $\mu(x)$ denotă numărul de ocurențe ale lui x în multimulțimea μ. Mulțimea tuturor multimulțimilor lui C este notată $Bag(C)$. Pentru două multimulțimi $\mu,\ \mu' \in Bag(C),\ \mu \le \mu'$ dacă $\forall c \in C, \mu(c) \le \mu'(c)$. Mulțimea de submulțimi ale lui C este notată prin $\wp(C)$. ■

Exemplul 1. Fie $C = \{a,b,d,e\}$, $\mu = \{a,a,b,b,b,b,e,e,e\}$ este o multimulțime pe C astfel încât: $\mu(a) = 2,\ \mu(b) = 4,\ \mu(d) = 0,\ \mu(e) = 3$. Această multimulțime μ este notată ca o sumă formală redată astfel: $M = 2a + 4b + 3e$.

Definiția 2**.** O rețea *RPC descriptiv-reconfigurabilă*, abreviat ΓC, este o structură de obiecte constituită din următorul 15-tuplu:

$$\Gamma C = < P, E, C, C_E, C_P, Pre, Post, Test, Inh, Pri, G_E, G_R, K_p, \phi, M_0 >, \text{ unde :}$$

- P este o mulțime nevidă de locații; E este mulțimea nevidă de evenimente discrete constituită din $E = T \cup R \neq \varnothing$, $T \cap R = \varnothing$, astfel încât $P \cap E = \varnothing$, unde T este mulțimea tranzițiilor, declanşarea cărora pot să modifice numai marcajul curent, iar $R = \{r_1,..., r_k\}$, $P \cap T \cap R = \varnothing$ este mulțimea *regulilor de rescriere*, care poate să modifice în mod dinamic marcajul curent şi/sau structura cu toate atributele rețelei curente. Grafic, tranzițiile sunt reprezentate prin bare groase, iar regulile de rescriere sunt reprezentate prin dreptunghiuri imbricate.
- $C = \{c_1, c_2,..., c_n\}$ este mulțimea finită nevidă de culori, definită pentru funcții de colorare ale mulțimilor *P* şi *E* respectiv, astfel încât $C_E : E \to \wp(C)$ şi $C_P : P \to \wp(C)$;
- Arcele, redate de funcțiile colorate *Pre, Test* şi *Inh* (respectiv *Post*) sunt aplicații de incidență *înainte* (respectiv *înapoi*), *test* şi *inhibiție* definite pe $P \times E$ (respectiv $E \times P$), astfel încât sunt verificate pentru orice eveniment $e \in E$ funcțiile *Pre*(*p*,*e*), *Post*(*p*,*e*), *Test*(*p*,*e*), *Inh*(*p*,*e*): $C_E(e) \times C_P(p) \to Bag(C_P(p))$; *Pri*: $E \times Bag(C_P(p)) \to IN_+$ este funcția de ordonare parțială a lui *E*, care introduce priorități dinamice de declanşare a evenimentelor validate de marcajul curent. Implicit, prioritățile ce nu sunt menționate ale unor evenimete e_j relativ la culoarea *c* sunt considerate nule, adică *Pri*(e_j)(*c*)=0. IN_+ este mulțimea numerilor întregi naturale;
- Funcțiile $G_E : E \times Bag(C_P(p)) \to \{true,\ false\}$ şi $G_R : R \times Bag(C_P(p)) \to \{true,\ false\}$ sunt respectiv nişte funcții de gardă (eng. *Guard-function*), care pentru orice $e_j \in E$ şi $r_k \in R$ determină respectiv funcții Booleene $g_j^E(M)(c)$ şi $g_k^R(M)(c)$ în marcajul curent M relativ la culoarea $c \in C_E$. Astfel, dacă evenimentul e_j este validat de marcajul curent M, notat $M[(e_j.c) >$, relativ la arce pentru culoarea c şi $g_j^E(M)(c)$ ='*true*', atunci evenimentul e_j rămâne validat şi, eventual, el poate fi declanşat, iar dacă

$g_j^E(M)(c)$='*false*' - acest eveniment nu este validat. Implicit $g_j^E(M)(c)$='*true*'. În cazul în care e_j validat este o tranziție sau o regulă de rescriere şi $g_j^R(M)(c)$='*false*', atunci această tranziție sau regulă de rescriere $e_j = r_j$, fiind declanşată, va *shimba numai* marcajul curent al rețelei ΓC, însă dacă $g_j^R(M)(c)$='*true*' ea va modifica atât sructura cu unele atribute curente ale ΓC, cât şi marcajul ei curent în conformitate cu specificațiile acestei reguli;

- $K_p : P \times Bag(C_P(p)) \to (IN_+ \cup +\infty)$ este funcția de capacitate a locațiilor relativ la culoarea c astfel, încât $\forall p_i \in P$, aceasta este redată de capacitatea maximă de jetoane $0 < K_p(p_i)(c) < +\infty$ care poate să se afle în locația p_i. Implicit, $K_p(p_i)(c)$ este nelimitată;
- $\phi : E \to \{T, R^t, R\}$ este funcția care indică tipul de eveniment validat de către marcajul curent al rețelei, adică el este de tipul *t*, $e \in T$ sau $e \in R^t$, declanşarea căruia modifică *numai marcajul curent* sau de tipul $r \in R$, care modifică atât *structura rețelei cu atributele sale*, cât şi *marcajul ei curent*;
- $M_0 : P \to Bag(C)$ este marcajul inițial ce determină o funcție de marcare definită pe mulțimea locațiilor *P*, astfel încât $\forall p \in P, M(p) \in Bag(C_P(p))$. ■

În Fig. 1 sunt prezentate primitivele unei rețele ΓC.

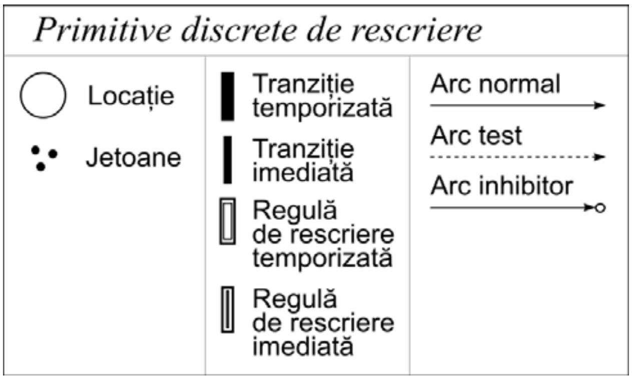

Figura 1. Primitivele unei rețele RPC dinamic descriptiv-reconfigurabilă ΓC.

În cele ce urmează, pentru a defini regulile de funcționare a unei ΓC, introducem următoarele notații:

- ${}^{\bullet}e = \{p \in P / Pre(e,p) \neq 0\}$ şi $e^{\bullet} = \{p \in P / Post(p,e) \neq 0\}$ este, respectiv, mulțimea de locații incidente la intrarea şi la ieşirea evenimentului *e*;
- ${}^{o}e = \{p \in P / Inh(e,p) \neq 0\}$ şi ${}^{*}t = \{p \in P / Test(e,p) \neq 0\}$ este, respectiv, mulțimea locațiilor de control al evenimentului *e* prin arce inhibitoare şi arce test.

Definiția 3**.** *(Regula de validare a unui eveniment).* Un eveniment e_j este validat de marcajul curent M relativ la culoarea $c \in C_E$, notat $e_j \in E(M)(c)$, dacă este verificată condiția de validare $ec(e_j, M)(c))$ care este redată de următoarea expresie logică:

$ec(e_j,M)(c) = ec^{\mathrm{Pre}}(e_j,M)(c) \wedge ec^{Inh}(e_j,M)(c) \wedge ec^{Test}(e_j,M)(c) \wedge ec^{K_p}(e_j,M)(c) \wedge g^E(e_j,M)(c)$, cu:

$ec^{\mathrm{Pre}}(e_j,M)(c) = \bigwedge_{\forall p_i \in {}^{\bullet}e_j} (M(p_i) \geq \mathrm{Pre}(e_j,p_i)(c))$ - condiția de validare relativ la arcele normale incidente înainte la evenimentul e_j; $ec^{Inh}(e_j,M)(c) = \bigwedge_{\forall p_i \in {}^{o}e_j} (M(p_i) < Inh(e_j,p_i)(c))$ - condiția de validare relativ la arcele inhibitoare incidente înainte la evenimentul e_j; $g^E(e_j,M)(c)$ - funcția de gardă a evenimentului e_j; $ec^{Test}(e_j,M)(c) = \bigwedge_{\forall p_i \in {}^{*}e_j} (M(p_i) \geq Test(e_j,p_i)(c))$ - condiția de validare relativ la arcele test; $ec^{K_p}(e_j,M)(c) = \bigwedge_{\forall p_i \in e_j^{\bullet}} ((K^p(p_i) - M(p_i)) \geq Post(e_j,p_i)(c))$ este condiția de validare relativ la capacitatea locațiilor incidente înapoi la evenimentul e_j. ■

Fie $E(M)(c) = T(M)(c) \cup R(M)(c)$, $T(M)(c) \cap R(M)(c) = \varnothing$ este mulțimea de evenimente validate de marcajul curent M al rețelei ΓC. De asemenea, fie A=<Pre, $Post$, $Test$, Inh > este mulțimea arcelor rețelei $\Gamma C =< RN, M >$, iar ΓC și RN sunt reprezentate de expresii descriptive $DE_{\Gamma C}$ și DE_{RN} respective [4]. Regula de rescriere $r \in R$ în mod dinamic a rețelei ΓC la declanșarea unui eveniment validat $e_j \in E(M)(c)$ constă în maparea $r: DE_L \triangleright DE_W$, în care *codomeniul operatorului de rescriere* $\triangleright$ este *Cod*($\triangleright$) $= DE_L \triangleright$, care este o expresie descriptivă DE_L specificată a subrețelei $\Gamma C_L \subseteq \Gamma C$ rețelei curente ΓC astfel, încât $P_L \subseteq P$, $E_L \subseteq E$ cu mulțimea arcelor $A_L \subseteq A$. În același mod, *domeniul* lui $\triangleright$ este *Dom*($\triangleright$) $= \triangleright DE_W$, determină o expresie descriptivă DE_w specificată a unei subrețele noi $\Gamma C_w \subseteq \Gamma C'$ a rețelei modificate $\Gamma C'$ cu P_w, E_w și mulțimea arcelor A_L respective.

Operatorul $\triangleright$ de rescriere a structurii rețelei curente reprezintă o operație binară, care produce o modificare a structurii în $DE_{\Gamma C}$ și, deci, în ΓC curentă prin rescrierea lor (eng. rewriting). La *schimbarea* expresiei descriptive DE_L specificate a subrețelei $\Gamma C_L \subseteq \Gamma C$ (DE_L și ΓC_L sunt eliminate, obținându-se respectiv $DE_{\Gamma C} \setminus DE_L$ și $\Gamma C \setminus \Gamma C_L$) cu o nouă expresie descriptivă DE_w specificată a subrețelei ΓC_w (DE_w și ΓC_w sunt adăugate respectiv la $DE_{\Gamma C} \setminus DE_L$ și $\Gamma C \setminus \Gamma C_L$). Astfel, rezultă o nouă expresie descriptivă $DE'_{\Gamma C'}$ a rețelei noi modificate $\Gamma C' = (\Gamma C \setminus \Gamma C_L) \cup \Gamma C_w$ astfel, încât $P' = (P \setminus P_L) \cup P_w$, $E' = (E \setminus E_L) \cup E_w$ și $A' = (A \setminus A_L) \cup A_w$. Aici, operatorul \ (respectiv $\cup$) indică operația de eliminare a ΓC_L din (adăugarea ΓC_w în) ΓC. La eliminarea unor locații și/sau evenimente arcele ce le conectează se vor elimina în mod implicit. În această nouă rețea modificată $\Gamma C'$, obținută la declanșarea regulii de rescriere $r_j \in E(M)(c)$ validate de marcajul curent M, respectiv locațiile și evenimentele, ce au aceleași atribute sunt contopite. Implicit, $r: DE_L \triangleright \varnothing$ și $r: \varnothing \triangleright DE_w$ descriu respectiv $\Gamma C' = \Gamma C \setminus \Gamma C_L$ și $\Gamma C' = \Gamma C \cup \Gamma C_w$. Menționăm, de asemenea, că în ΓC_L și ΓC_w pot fi considerate aparte și ca submulțimi ale P_L și/sau P_w cu marcaje respective, E_L și/sau E_w și A_L și/sau A_w.

Definiția 4. *Regula de declanșare* a unui eveniment al rețelei ΓC. Fie $\Gamma C = (RN, M)$ este configurația rețelei curente, numită *configurație sursă*. Evenimentul $e_j \in E(M)(c)$ validat de marcajul curent M relativ la culoarea $c \in C_E$ este declanșat, dacă *nu există* un alt eveniment e_k cu o prioritate superioară lui, adică $\neg\exists(Pri(e_j)(c) > Pri(e_k)(c))$, pentru care sunt verificate precondițiile sale de validare $e_k \in E(M)(c)$. La declanșarea lui e_j, acesta va modifica, în dependență de valoarea lui $g_j^R(M)(c)$, fie numai marcajul curent, fie va modifica atât structura și unele atribute ale rețelei curente, cât și marcajul ei. Astfel, pentru acest eveniment: ***if*** $((\phi_j = t_j) \vee (\phi_j = r_j) \wedge (g_j^R(M)(c) = False))$ ***then*** (declanșarea tranziției t_j sau a regulei de rescriere r_j validate de marcajul curent relativ la culoarea *c* va schimba numai marcajul curent în această configurație de rețea, adică: $(RN, M) \xrightarrow{e_j(c)} (RN, M') \Leftrightarrow (RN = RN,\ M[e_j(c) > M'))$ ***else*** (declanșarea regulii de rescriere $r_j \in R(M)(c)$ va schimba atât structura rețelei cu atributele ei, cât și starea curentă în configurația de rețea sursă, adică $r_j \in R(M)(c)$ fiind declanșată va induce o modificare în configurația *sursă* de la $\Gamma C = (RN, M)$ la *configurația destinație* $\Gamma C' = (RN', M')$, astfel încât::

$$(RN, M) \xrightarrow{r_j(c)} (RN', M') \Leftrightarrow (RN = RN',\ M[r_j(c) > M')).$$

Starea atinsă după declanșarea regulii r este $\gamma' = (RN', M')$. Configurația rețelei ΓC_0 inițiale este (RN_0, M_0), iar (RN, M) este configurația rețelei curente ΓC.

Graful de stări accesibile $GA(\Gamma C_0)$ ale rețelei $\Gamma C_0 =< RN_0, M_0 >$ este un graf orientat etichetat în care vârfurile sunt etichetate cu stările de configurații (RN, M), iar arcele ce leagă aceste vârfuri sunt etichetate cu evenimentele $e_j \in R(M)(c)$ de tip tranziții sau cu reguli de rescriere respective, astfel încât: a) declanșarea evenimentului $e_j \in E(M)(c)$ determină un arc, etichetat cu $e_j = t_j$ sau $e_j = r_k$ pentru $g_k^R(M)(c) = "false"$, de la starea sursă (RN, M) către starea nouă (RN, M') în care structura rețelei rămâne aceeași, iar marcajul curent M este modificat într-un marcaj nou $M' = M + Post(e_j, \cdot)(c) - Pre(e_j, \cdot)(c)$;

b) declanşarea regulii $r_j \in R(M)(c)$ din (RN, M) pentru $g_k^R(M)(c) = "true"$ conduce la modificarea configurației surse (RN, M) într-o configurație nouă (RN', M') conform specificațiilor operatorului $\triangleright$.

Notăm faptul, că rețelele tip RPC pot fi obținute din rețele ΓC ca un caz particular pentru care, mulțimea regulilor de rescriere este vidă, adică $R = \varnothing$.

Pentru a ilustra funcționarea acestui tip de rețele, în continuare, vom considera un exemplu de modelare a proceselor de calcul orientate pe servicii reconfigurabile.

Exemplul 1. Rețelele colorate descriptiv-reconfigurabile $\Gamma C1$ şi $\Gamma C2$ din figura 1 descrie funcționarea unui proces al sistem de calcul orientat pe servicii, în care *n* diferiți clienți cer şi obțin *k* tipuri de servicii. Fiecare cerere a clienților (respectiv fiecare serviciu furnizat) este identificat prin culoarea sa $< x > \in C_{cl} = \{< c_i >, i = \overline{1, n}\}$ (respectiv $< y > \in C_{sv} = \{< s_i >, i = \overline{1, k}\}$).

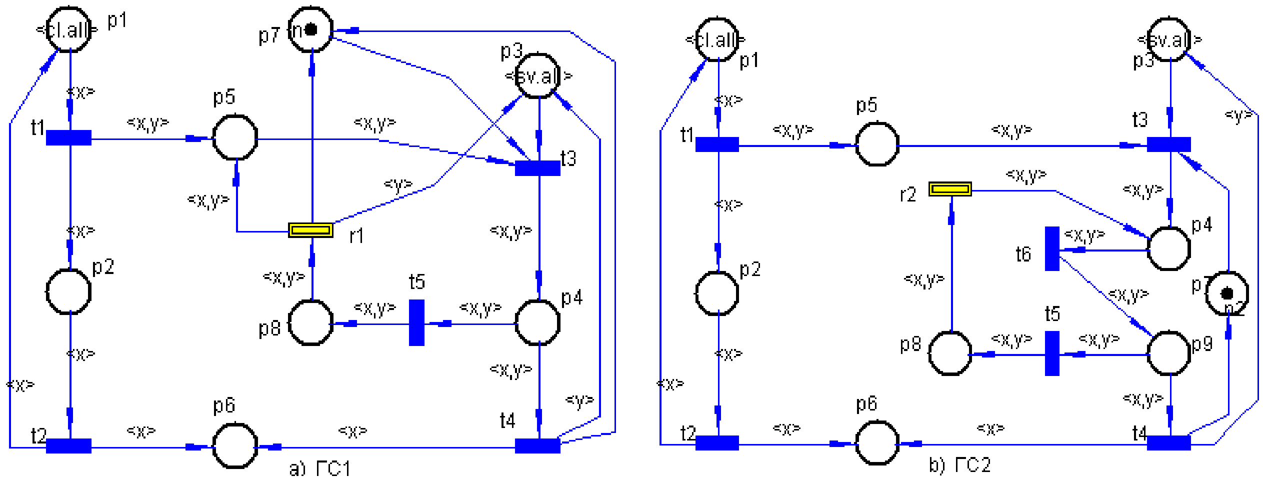

Figura 2. Rețele colorate descriptiv-reconfigurabile: a) $\Gamma C1$ şi b) $\Gamma C2$.

Specificațiile locațiilor, tranzițiilor şi a regulilor de rescriere ale $\Gamma C1$ şi $\Gamma C2$ sunt următoarele:

p_1 - clientul este în stare activă de formare a cererii; p_2 - clientul este în starea de aşteptare a răspunsului; p_3 - serviciile sunt gata pentru a fi procesate; p_4 - serviciile cerute sunt procesate de către servere; p_5 - memorie tampon de stocare a mesajelor de tipul $< x >$ care cer a un serviciu de tipul $< y >$; p_6 - memorie tampon de stocare a răspunsului pentru mesajul de tipul $< x >$ care a obținut un serviciu de tipul $< y >$; p_7 - servere ce procesează serviciile cerute sunt libere (marcajul inițial $M_0(p_7) = n$); p_8 - servicii procesate nu sunt conforme cu specificațiile cerute; p_9 - servicii procesate la faza de verificare a conformității cu specificațiile cerute.

t_1 - clientul de tipul $< x >$ trimite un mesaj serverului (locația p_5) pentru a obține un serviciu de tipul $< y >$ şi apoi clientul va trece în starea de aşteptare; t_2 - recepția răspunsului pentru clientul $< x >$ şi trecerea lui în starea activă; t_3 - recepția mesajului clientului $< x >$ ce cere serviciul $< y >$ pentru a fi procesat de un server liber; t_4 - trimiterea răspunsului la cererea clientului $< x >$ la serviciul $< y >$ obținut, eliberarea serverului şi a acestui serviciu; t_5 - verificarea conformității procesării serviciului cerut; t_6 - faza de preprocesare a serviciilor cerute.

r_1 - reconfigurarea rețelei $\Gamma C1$ în $\Gamma C2$ dacă numărul de servicii neconforme în p_8 depăşeşte celui specificat; r_2 - reconfigurarea rețelei $\Gamma C2$ în $\Gamma C1$ dacă numărul de servicii procesate este mai mic decât cel specificat;

Locațiile $P_{cl} = \{p_1, p_2, p_6\}$ şi respectiv tranzițiile $T_{cl} = \{t_1, t_2\}$ au acelaşi domeniu de culori C_{cl} , adică aceste locații nu pot să conțină decât numai jetoane de culoarea $< c_i >, i = \overline{1, n}$, iar tranzițiile t_1 şi t_2 pot fi declanşate relativ numai la acest tip de culori. Jetoanele în locația p_7 au o culoare neutră $< \bullet >$.

Marcajul inițial $M_0(p_1) = (\sum_{i=1}^{n} < c_i >) =< cl.all >$ indică că toți cei diferiți clienți sunt în starea activă de formare a unei cereri către server pentru a obține un serviciu de tipul $< y > \in C_{sv} = \{< s_i >, i = \overline{1,k}\}$. Declanşarea tranziției t_1 indică că clientul de tipul $< x >$ trimite un mesaj serverului (locația p_5) pentru a obține un serviciu de tipul $< y >$ şi apoi el va trece în starea de aşteptare. Locațiile $P_{sv} = \{p_4, p_5, p_8, p_9\}$, tranzițiile $T_{sv} = \{t_3, t_4, t_5, t_6\}$ şi regulile de rescriere $R_{sv} = \{r_1, r_2\}$ respectiv au ca domeniu de culori cuplul de culori $< x, y > \in C_{cl} \times C_{sv}$. La recepția unui răspuns $< x >$ (locația p_6) el se va reîntoarce în starea activă prin declanşarea tranziției t_2 relativ la $< x >$. Locația p_3,cu $M_0(p_3) = (\sum_{i=1}^{k} < s_i >) =< sv.all >$, indică la faptul că serverul este gata să prelucreze mesaje ce cer serviciul de tipul $< y >$.

Pentru regulile de rescriere ale $\Gamma C1$ şi $\Gamma C2$ respectiv avem:

$$r_1 : \Gamma C_1 \rhd \Gamma C_2 \text{ cu } g_1^R(M)(x,y) = M(p_8) > 4 \text{ şi } r_2 : \Gamma C_2 \rhd \Gamma C_1 \text{ cu } g_2^R(M)(x,y) = M(p_9) < 3.$$

Este necesar de precizat pentru fiecare variabilă ce apare pe un arc incident cu o tranziție dată la ce componentă a cuplului instanțiat ea este asociată. Pentru rețelele $\Gamma C1$ şi $\Gamma C2$, x este asociat la prima componentă, iar y la a doua. Astfel, la declanşarea tranziției t_4 relativ la cuplul $< c_1, s_3 >$ înseamnă că se va asocia la x valoarea c_1 şi la y valoarea s_3.

Analiza proprietăților comportamentale, efectuată prin metoda P-invarianților colorați, arată că rețelele $\Gamma C1$ şi $\Gamma C2$ sunt mărginite, viabile şi reversibile şi astfel există un regim staționar de funcționare al acestui sistem orientat pe servicii.

4. Concluzie

Rețelele Petri (RP) sunt un formalism de modelare şi analiză a sistemelor concurente cu evenimente discrete şi a proceselor de calcul orientate pe servicii. Însă, acestea nu oferă o cale directă care ar reprezenta schimbări dinamice de structură, o paradigmă foarte importantă la proiectarea sistemelor de calcul orientate pe servicii. În lucrare sunt definite RP colorate (RPC) reconfigurabile marcaj – controlabile (RPCR), care permit de a obține un model mai compact pentru descrie funcționarea sistemelor orientate pe servicii, unde în mod dinamic sunt prezente schimbări de structură. Aplicabilitatea acestui demers este ilustrată printr-un exemplu al proceselor de calcul orientat pe serviciu în care sunt prelucrate diferite tipuri de cereri a diferitor tipuri de servicii.

În lucrările pe viitor, vom defini rețele RPCR în care vom lua în considerație aspectul stochatic de funcționare a proceselor de calcul orientat pe servicii reconfigurabile

Bibliografie

1. Bell, M. *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*. John Willey and Sons, New York, 2008.
2. Compton, K., Hauck, S. *Reconfigurable Computing: a Survey of Systems and Software*. ACM Computing Surveys (CSUR), vol. 34, no. 2, 1998, p. 171-210.
3. Guțuleac, E., Mocanu, M. L., Țurcanu, Iu. *Dynamic Rewriting of Differential Petri Nets for Modeling of Hybrid Systems*. Proceedings of the 2nd International Conference on Intelligent Computer Communication and Processing (ICCP 2006), Cluj-Napoca, 1-2 Sept., România, 2006, p. 105-112.
4. Guțuleac, E., Țurcanu, Iu., Guțuleac, Em. *Compunerea Descriptivă a Modelelor de Rețele Petri pentru Verificarea Sistemelor Orientate pe Servicii*. Proceedings of the 3-rd International Conference ICTEI-2010, Vol. 2, Chişinău, R. Moldova, May 20-23, 2010, p. 114-119.
5. Jensen, K., Kristensen, L. M., Wells, L. *Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems*. Int. Journal on Software Tools for Technology Transfer, 9(3), 2007, p. 213-254.
6. Llorens, M., and Oliver, J. *Structural and Dynamic Changes in Concurrent Systems: Reconfigurable Petri Nets*. IEEE Transactions on Computers, vol. 53(9), 2004, p. 1147-1158.