MODELAREA CONTROLULUI SISTEMELOR DE FABRICAŢIE RECONFIGURABILE PRIN REŢELE PETRI COLORATE MEMBRANALE

E. Guţuleac, dr. hab.prof.univ., S. Zaporojan, dr.conf.univ.
Universitatea Tehnică a Moldovei

INTRODUCERE

Sistemele de fabricaţie au fost în ultimii ani şi sunt în continuare supuse unui proces profund de transformare cu o dinamică ascendentă. Elementul de noutate în concepţia actuală îl reprezintă Sistemele de Fabricaţie Reconfigurabile (SFR) [9, 12, 16, 18]. Acesta presupune capabilitatea schimbării structurii SFR pentru aşi modifica funcţionalitatea şi a prelucra noi familii de produse care cuprinde atât reconfigurarea controlului, căt şi mobilitatea modulelor hardware/software, acţiuni ce trebuie descrise formal [6, 17]. Tot odată, acest tip de sisteme trebuie să aibă o flexibilitate, disponibilitate şi siguranţă în funcţionare deosebită.

Utilizarea principiilor de control distribuit orientat pe servicii (CDOS) al SFR este o soluţie promiţătoare pentru a atinge modularitatea, flexibilitatea, reconfigurarea şi interoperabilitatea componentelor acestuia [9, 12]. Aspectele cruciale ale acestui tip de control sunt descrierea formală şi coordonarea executării serviciilor oferite de către entităţile distribuite. Pentru alocarea şi procesarea a diferitor servicii sau condiţii de operare ale aplicaţiilor pot fi folosite diferite configuraţii de algoritmi şi entităţi hardware/software ce comunică între ele. Ca urmare, facilităţile trecerii în timp real de la o configuraţie către alta, pe parcursul rulării, duc la creşterea s*iguranţei în funcţionare* şi a flexibilităţii utilizării sistemului [13, 18], asigurarea cărora în timpul reconfigurării dinamice este dificilă din cauza interacţiunii între serviciile de aplicaţie care sistemul le oferă SFR.

Modelarea acestor tipuri de sisteme este o sarcină dificilă, iar analiza lor se complică din cauza apariţiei unor defecţiuni sau schimbări de produse.

Modelarea formală şi tehnicile de verificare sunt utilizate pe scară largă la proiectarea şi analiza multor tipuri de sisteme cu procese discret-continue, în baza verificării cărora se pot identifica lacunele de proiectare la stadiul incipient al ciclului lor de viaţă. Ca rezultat, aceste probleme pot fi eliminate mai devreme, iar costurile de depanare, întreţinere şi mentenanţă pot fi reduse semnificativ.

Unul dintre cele mai răspândite formalisme moderne, folosite pentru modelarea şi analiza a sistemelor cu procese discrete, sunt reţelele Petri (RP) de diferite extensii [1, 2, 5, 6, 7, 8]. Acest demers este utilizat pentru a reda şi menţine modelul RP într-o formă simplă, concisă şi precisă, deoarece el oferă o interpretare grafică a sistemelor complexe care este uşor de înţeles, în timp ce baza lor matematică este lipsită de ambiguitate. RP clasice nu au puterea de expresivitate suficientă pentru a face faţă unor aspecte actuale de modelare cum ar fi reconfigurabilitatea şi mobilitatea. În acest context, au fost propuse mai multe lucrari, care încearcă să extindă RP cu capacitatea de a specifica mobilitatea şi, în general, reconfigurabilitatea [6, 8, 14]. Însă aceste extensii nu permit de a descrie în mod explicit organizarea ierarhică, mobilitatea şi dinamica reconfigurării modelului dependentă de starea curentă a acestuia, apariţia unor evenimente şi/sau la schimbarea valorilor unor atribute asociate cu acest model.

În lucrarea dată sunt considerate unele aspecte de modelare ale CDOS cu procese dinamice reconfigurabile şi entităţi mobile ale SFR. Pentru aceasta sunt introduse reţele Petri colorate reconfigurabile (RPCR) membranale care permit de a construi modele capabile să descrie cazuri în care structura modelului şi atributele lui pot să se schimbe în dependenţă de starea sa curentă şi/sau la apariţia unor evenimente. Acestea, fiind foarte flexibile pentru descrierea mobilităţii şi reconfigurabilităţii proceselor CDOS, sunt traduse şi prin posibilitatea de a avea o adaptare dinamică a modelului care urmează să fie reconfigurat în mod ierarhic sau chiar eventual pentru a se autoreconfigura.

1. REŢELE PETRI COLORATE DINAMIC RECONFIGURABILE

Reţelele Petri colorate (RPC) [5] sunt o extensie de RP care oferă posibilitatea de a modela sisteme cu evenimente discrete într-un mod mult mai compact decât RP obişnuite, deoarece ele folosesc mecanisme de nivel înalt, similare cu cele ale limbajelor de programare: cu fiecare jeton este asociat o valoare a unui anumit tip de date (set de culori). Într-o RPC o locaţie poate să conţină jetoane de diferite culori, iar o tranziţie poate fi declanşată în diferite *moduri* în conformitate cu culoarea selectată. Acest fapt este realizat prin ataşarea unui domeniu de culori la fiecare locaţie şi fiecare tranziţie. Astfel, pentru un marcaj iniţial dat numărul de comportamente ce pot fi exprimate de o RPC este cu mult mai mare decât cel prin RP.

Un arc ce conectează o locaţie (tranziţie) cu o tranziţie (locaţie) este etichetat cu o expresie ce este o funcţie de culoare marcaj dependentă. Această funcţie determină pentru fiecare culoare a tranziţiei (sau instanţă de declanşare a tranziţiei) numărul de jetoane de culoarea aleasă ce trebuie *consumate* sau *produse* în locaţia respectivă la declanşarea tranziţiei relativ la culoarea selectată. Cum şi pentru RP obişnuite, alegerea culorii de declanşare a unei tranziţii a RPC este efectuată în mod nedeterminist: dacă o tranziţie *t* este validată atât relativ la culoarea c_1, cât şi la culoarea c_2 se va selecta în mod nedeterminist pentru a fi declanşată tranziţia relativ numai la o singură culoare. Cu toate acestea RPC nu pot fi folosite pentru a descrie funcţionarea sistemelor ce trebuie reconfigurate în mod dinamic sau mobilitatea lor.

În continuare, cu scopul de a trata unele probleme menţionate mai înainte la modelarea şi analiza SFR orientate pe servicii reconfigurabile, definim o extensie de RPC, numite reţele RPC *dinamic descriptiv-reconfigurabile*, în care sunt introduse o mulţime de reguli de rescriere colorate $R = \{r_1,..., r_k\}$ a reţelei RPC curente ce poate modifica atât marcajul curent, cât şi structura ei la ocurenţa unor evenimente specificate. O regulă de rescriere $r_j \in R$ este o generalizare a noţiunii de tranziţie discretă $t_j \in T$, folosită în sens clasic. Condiţia de validare de către marcajul curent M a unei tranziţii $t_j \in T$ şi/sau reguli $r_j \in R$ este similară cu cea a unei RPC .

Definiţia 1. O reţea RPC *dinamic reconfigurabilă* (RPCR), notată ΓC, este o structură de obiecte constituită din 14-tuplu:

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

- P este o mulţime nevidă de locaţii; E este mulţimea 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 mulţimea tranziţiilor, declanşarea cărora pot să modifice numai marcajul curent, iar $R = \{r_1,..., r_k\}$, $P \cap T \cap R = \varnothing$ este mulţimea *regulilor de rescriere*, care poate să modifice în mod dinamic marcajul curent şi/sau structura cu toate atributele reţelei curente. Grafic, tranziţiile sunt reprezentate prin bare groase, iar regulile de rescriere sunt reprezentate prin dreptunghiuri imbricate.
- $C = \{c_1, c_2,..., c_n\}$ este mulţimea finită nevidă de culori, definită pentru funcţii de colorare ale mulţimilor P şi E respectiv, astfel încât $C_E : E \to \wp(C)$ şi $C_P : P \to \wp(C)$, unde $\wp(C)$ este mulţimea submulţimilor lui C;
- Arcele, redate de funcţiile colorate *Pre, Test* şi *Inh* (resp. *Post*) sunt aplicaţii de incidenţă *înainte* (resp. *înapoi*), *test* şi *inhibiţie* definite pe $P \times E$ (resp. $E \times P$), astfel încât sunt verificate pentru orice eveniment $e \in E$ funcţiile *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 funcţia de ordonare parţială a lui *E*, care introduce priorităţi dinamice de declanşare a evenimentelor validate de marcajul curent. Implicit, priorităţile ce nu sunt menţionate ale unor evenimente e_j relativ la culoarea c sunt considerate nule, adică *Pri*(e_j)(*c*)=0. IN_+ este mulţimea numerelor întregi naturale, iar $Bag(C)$ este mulţimea multimulţimilor lui C;
- $G_E : E \times Bag(C_P(p)) \to \{true, false\}$ şi $G_R : R \times Bag(C_P(p)) \to \{true, false\}$ sunt respectiv nişte funcţii de gardă (eng. *Guard-function*), care pentru orice $e_j \in E$ şi $r_k \in R$ determină respectiv funcţii 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*' - acesta nu este validat. Implicit $g_j^E(M)(c)$='*true*'. În cazul în care e_j validat este o tranziţie sau o regulă de rescriere şi $g_j^R(M)(c)$='*false*', atunci acesta, fiind declanşat, va *schimba numai* marcajul curent al reţelei ΓC, însă dacă $g_j^R(M)(c)$='*true*' el va modifica atât structura cu unele atribute curente ale ΓC, cât şi marcajul ei curent în conformitate cu specificaţiile acestei reguli;
- $K_p : P \times Bag(C_P(p)) \to (IN_+ \cup +\infty)$ este funcţia de capacitate a locaţiilor 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 locaţia p_i. Implicit, $K_p(p_i)(c)$ este nelimitată;

- $M_0 : P \to Bag(C)$ este marcajul iniţial ce determină o funcţie de marcare definită pe mulţimea locaţiilor P, astfel încât $\forall p \in P$, $M(p) \in Bag(C_P(p))$ ■

În Fig. 1 sunt prezentate primitivele unei ΓC.

Pentru a defini regulile de funcţionare ale unei ΓC notăm ${}^{\bullet}e$ şi $e^{\bullet}$ - mulţimea de locaţii incidente respectiv la intrarea şi la ieşirea evenimentului *e*, iar ${}^{o}e$ şi ${}^{*}t$ este, respectiv, mulţimea locaţiilor de control al lui *e* prin arce inhibitoare şi arce test.

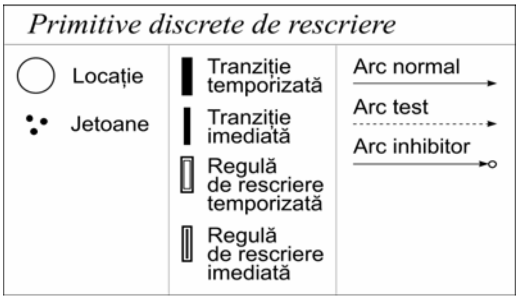


Figura 1. Primitivele unei reţele ΓC.

Definiţia **2.** *(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 condiţia de validare $ec(e_j, M)(c))$ care este redată de următoarea expresie logică:

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

$ec^{\Pr e}(e_j, M)(c) = \bigwedge_{\forall p_i \in {}^{\bullet}e_j} (M(p_i) \geq \Pr e(e_j, p_i)(c))$ - condiţia de validare relativ la arcele normale incidente înainte la evenimentul e_j;

$ec^{Inh}(e_j, M)(c) = \bigwedge_{\forall p_i \in {}^{o}e_j} (M(p_i) < Inh(e_j, p_i)(c))$ - condiţia de validare relativ la arcele inhibitoare incidente înainte la evenimentul e_j; $g^E(e_j, M)(c)$ - funcţia de gardă a evenimentului e_j;

$ec^{Test}(e_j, M)(c) = \bigwedge_{\forall p_i \in {}^{*}e_j} (M(p_i) \geq Test(e_j, p_i)(c))$ - condiţia 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 condiţia de validare relativ la capacitatea locaţiilor incidente înapoi la evenimentul e_j. ■

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

Operatorul $\rhd$ de rescriere a structurii reţelei curente reprezintă o operaţie binară, care produce o modificare a structurii în $DE_{\Gamma C}$ şi, deci, a lui ΓC curentă prin rescrierea ei (eng. rewriting). La *schimbarea* expresiei descriptive DE_L specificate a subreţelei $\Gamma C_L \subseteq \Gamma C$ (DE_L şi ΓC_L sunt eliminate, obţinându-se respectiv $DE_{\Gamma C} \setminus DE_L$ şi $\Gamma C \setminus \Gamma C_L$) cu o nouă expresie descriptivă DE_w specificată a subreţelei ΓC_w (DE_w şi ΓC_w sunt adăugate respectiv la $DE_{\Gamma C} \setminus DE_L$ şi $\Gamma C \setminus \Gamma C_L$). Astfel, rezultă o nouă expresie descriptivă $DE'_{\Gamma C'}$ a reţelei noi modificate $\Gamma C' = (\Gamma C \setminus \Gamma C_L) \cup \Gamma C_w$ astfel, încât $P' = (P \setminus P_L) \cup P_w$, $E' = (E \setminus E_L) \cup E_w$ şi $A' = (A \setminus A_L) \cup A_w$. Aici, operatorul \ (respectiv $\cup$) indică operaţia de eliminare a ΓC_L din (adăugarea ΓC_w în) ΓC. La eliminarea unor locaţii şi/sau evenimente arcele ce le conectează se vor elimina în mod implicit. În această nouă reţea modificată $\Gamma C'$, obţinută la declanşarea regulii de rescriere $r_j \in E(M)(c)$ validate de marcajul curent M, respectiv locaţiile şi evenimentele, ce au aceleaşi atribute sunt contopite. Implicit, $r : DE_L \rhd \varnothing$ şi $r : \varnothing \rhd DE_w$ descriu respectiv $\Gamma C' = \Gamma C \setminus \Gamma C_L$ şi $\Gamma C' = \Gamma C \cup \Gamma C_w$. Menţionăm, de asemenea, că în ΓC_L şi ΓC_w pot fi considerate aparte şi ca submulţimi ale P_L şi/sau P_w cu marcaje respective, E_L şi/sau E_w şi A_L şi/sau A_w.

Definiţia 3. *Regula de declanşare* a unui eveniment al reţelei ΓC. Fie $\Gamma C = (RN, M)$ este configuraţia reţelei curente, numită *configuraţie sursă*. Evenimentul $e_j \in E(M)(c)$ validat de marcajul curent M relativ la culoarea $c \in C_E$ este declanşat, dacă *nu există* un alt e_k cu o prioritate superioară lui, adică $\neg\exists\,(Pri(e_j)(c) > Pri(e_k)(c))$, pentru care sunt verificate precondiţiile sale de validare. La declanşarea lui e_j, acesta va modifica, în dependenţă de valoarea lui $g_j^R(M)(c)$, fie numai marcajul curent, fie va modifica atât structura şi unele atribute ale reţelei curente, cât şi marcajul ei. Astfel, pentru acest eveniment avem:

If $((\phi_j = t_j) \vee (\phi_j = r_j) \wedge (g_j^R(M)(c) = False))$ ***then*** (declanşarea tranziţiei t_j sau a regulii de rescriere r_j validate de marcajul curent relativ la culoarea *c* va schimba numai marcajul curent în această configuraţie de reţea, adică:

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

else (declanşarea regulii $r_j \in R(M)(c)$ de rescriere va schimba atât structura reţelei cu atributele ei, cât şi starea curentă în configuraţia de reţea sursă, adică $r_j \in R(M)(c)$ fiind declanşată va induce o modificare în configuraţia *sursă* de la $\Gamma C = (RN, M)$ la *configuraţia destinaţie* $\Gamma C' = (RN', M')$, astfel încât::

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

Starea obţinută după declanşarea regulii *r* este $\gamma' = (RN^{'}, M')$. Configuraţia reţelei ΓC_0 iniţiale este (RN_0, M_0), iar (RN, M) este configuraţia reţelei curente ΓC.

Graful de stări accesibile $GA(\Gamma C_0)$ ale reţelei $\Gamma C_0 =< RN_0, M_0 >$ este un graf orientat etichetat în care vârfurile sunt etichetate cu stările de configuraţii (RN, M), iar arcele ce leagă aceste vârfuri sunt etichetate cu evenimentele de tip tranziţii sau cu reguli de rescriere respective, astfel încât:

a) declanşarea $e_j \in E(M)(c)$ determină un arc, etichetat cu $e_j = t_j$ sau $e_j = r_k$ pentru $g_k^R(M)(c) = "false"$, de la (RN, M) - starea sursă către starea nouă (RN, M') în care structura reţelei rămâne aceeaşi, iar marcajul curent M este modificat într-un nou marcaj:

$$M' = M + Post(e_j, \cdot)(c) - Pre(e_j, \cdot)(c);$$

b) declanşarea regulii $r_j \in R(M)(c)$ din (RN, M) pentru $g_k^R(M)(c) = "true"$ conduce la modificarea configuraţiei surse (RN, M) într-o configuraţie nouă (RN', M') conform specificaţiilor operatorului $\triangleright$ de reconfigurare a modelului.

Notăm faptul, că reţelele tip RPC pot fi obţinute din reţele RPCR ca un caz particular pentru care, mulţimea regulilor de rescriere este vidă.

2. EXPRESII DESCIPTIVE RPCR

Pentru a reda proprietăţi compoziţionale analitice modelelor de reţele RPCR în [1] este introdusă *noţiunea* de *dexel* (***de****scriptive* ***ex****pression* ***el****ement*) şi un set de operaţii compoziţionale, cu atribute respective, care permit de a construi expresii descriptive ce sunt apoi mapate direct în mod grafic de către utilizator.

În scopul de a facilita expunerea lucrării date, prezentăm succint doar unele operaţii compoziţionale. Mai detaliat cititorul poate consulta lucrările [1, 2] pentru compunerea RP generalizate.

Un *dexel* $bRDE$ al unei RPCR este expresia:

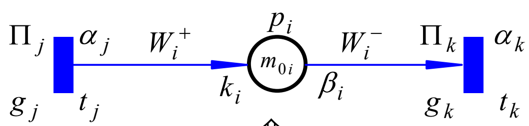
$$bHDE =< {}^{\Pi_j}_{g_j}|^{\alpha_j}_{e_j}\, m_{0i}^{h\ k_i}\, \vec{y}_i^{\beta_i}\, [W_i^+, W_i^-]\, {}^{\Pi_k}_{g_k}|^{\alpha_k}_{e_k} >,$$

unde $\vec{y} \in \{p, \overline{p}, \widetilde{p}\}$ este simbolul-locaţie care reprezintă locaţia discretă p la evenimentul $e \in \{t, r\}$, care este simbolul-eveniment *t* sau *r* ce determină respectiv tipul de arc ({ *p - arc normal*, $\overline{p}$ – *arc inhibitor*, $\widetilde{p}$ – *arc test* }) cu ponderea W_i^- incident înainte la evenimentul e_k, notat $|_{e_k}$:

$$W_i^- \in \{Pre(e_k, \vec{y}_i)(c),\ Inh(e_k, \vec{y}_i)(c),$$
$$Test(e_k, \vec{y}_i)(c)\}$$, iar $W_i^+ \in \{Post(e_j, \vec{y}_i)(c)\}$ este ponderea arcului normal ce iese din evenimentul e_j, notat $|_{e_j}$ şi intră în locaţia $\vec{y}_i$. Atributele locaţiei $\vec{y}_i$ sunt, respectiv: m_{0i} - marcajul iniţial; k_i - capacităţile locaţiei; β_i-eticheta locaţiei ce redă tipul de condiţii. Atributele evenimentelor $\vec{z}_j$ şi $\vec{z}_k$ sunt, respectiv: g_j şi g_k - funcţia de gardă respectivă, unde $g \in \{g^E, g^E\}$; Π_j şi Π_k - funcţia de prioritate; α_j şi α_k - eticheta ce redă tipul de acţiune sau activitate. Unele atribute pot fi omise, fiind implicit

date. De exemplu: $m_{0i} = 0$; $K_{p_i}^{\min} = 0$, iar $K_{p_i}^{\max}$ este considerată nelimitată. În cazul în care $W_i^- = W_i^+ = 1$ paranteza pătrată se va omite.

Maparea unor derivative ale *bHDE* pentru partea discretă a *RPHS* este prezentată în Fig. 2.



$$\Updownarrow$$

$$^{\Pi_j}_{g_j}|^{\alpha_j}_{t_j}\ m_{0i}{}^{k_i}\, p_i^{\beta_i}\ [W_i^+, W_i^-]\ ^{\Pi_k}_{g_k}|^{\alpha_k}_{t_k}$$

Figura 2. Maparea unor derivative *bRDE*.

Cu ajutorul diferitor derivative ale *bHDE* şi operaţii *compoziţionale unare* şi/sau *binare*, folosind un raţionament adecvat ce redă interacţiunea condiţiilor şi evenimentelor sistemului specificat, putem compune expresii descriptive ale modelelor RPCR (sub)sistemelor considerate.

O expresie descriptivă (*DE*) a unei reţele ΓC tip RPCR este:

DE :: = *bDE* | $DE_i * DE_j$ | $\circ DE$, unde $*$ reprezintă operatorul unei *operaţii binare*, iar $\circ$ reprezintă operatorul unei operaţii unare.

Implicit, la aplicarea acestor operaţii, locaţiile şi tranziţiile ce au acelaşi nume se vor contopi în mod respectiv. Într-o *DE*, orice simbol-locaţie sau simbol-eveniment poate fi folosit în orice ordine de mai multe ori. Astfel, se va subînţelege că în reţele RPCR respective, redate de expresia *DE*, aceleaşi locaţii (evenimente), cu acelaşi simbol vor fi *contopite* într-un singur *simbol-locaţie* (*simbol-eveniment*). La eliminarea unor locaţii şi/sau evenimente arcele conectate cu aceste noduri vor fi, de asemenea, eliminate.

Redăm în continuare unele operaţii compoziţionale ce vor fi aplicate în această lucrare:

- *Operaţia Inhibiţie,* redată de *operatorul* "$\overline{p}_i$", este o operaţie unară. Ea descrie faptul că la ocurenţa pre-condiţiei p_i, nu mai poate avea loc ocurenţa evenimentului specificat.

- *Operaţia Test* unară cu *operatorul* "$\widetilde{p}_i$", descrie o buclă a reţelei impure, redată de $DE1 = m_{0i}\widetilde{p}_i\ [W_i]\,|^{\alpha_j}_{t_j}$, ea reprezintă arcul *test*.

- *Operaţia Sincronizare,* redată de *operatorul* "$\bullet$" sau "$\wedge$", este o operaţie binară *comutativă*, *asociativă* şi *reflexivă* ce descrie *sincronizarea* pre-condiţiilor legate cu $p_i \in {}^{\bullet}e_j$, ale unui eveniment e_j, apariţia căruia va avea loc numai atunci, când concomitent aceste pre-condiţii sunt verificate, fiind descrise de *DE*2:

$$DE2 = (m_{01}{}^{K_{p_1}}\, y_1\ [W_1] \bullet \ldots \bullet m_{0n}{}^{K_{p_n}}\, y_n\ [W_n])\,|^{\alpha_j}_{e_j}\ .$$

- *Operaţia Secvenţialitate*, redată de *operatorul* "|", este o operaţie binară ce determină logica "cauză-consecinţă" a relaţiei dintre două stări locale p_i (pre-condiţie) şi p_k (post-condiţie), determinată de acţiunea e_j. Această o*peraţie*, exprimată de expresia *DE*3, este *asociativă, reflexivă* şi *tranzitivă*, însă *necomutativă*:

$$DE3 = m_{0i}p_i\ [W_i]\,|^{\alpha_j}_{e_j} m_{0k}p_k\ [W_k]\,.$$

- *Operaţia AND-Split*, redată de *operatorul* "$\Diamond$", descrie faptul că la apariţia unui eveniment specificat e_j se vor produce concomitent mai multe post-condiţii. Aceasta operaţie binară, fiind *comutativă*, *asociativă* şi *reflexivă,* este redată de următoarea expresia *DE*4:

$$DE4 = |^{\alpha_j}_{e_j}(\ m_{01}p_1\ [W_1]\ \Diamond \ldots \Diamond m_{0n}p_n\ [W_n]\ )\ .$$

- *Operaţia Paralelism competitiv*, redată de *operatorul* "$\veebar$" sau "$\mp$", descrie *relaţiile logice de paralelism competitiv* ale condiţiilor şi evenimentelor între două sau mai multe procese concurente. Ea este aplicată pentru a efectua compunerea unor submodele de subreţele RPCR, ce descriu funcţionarea subsistemelor respective, într-un model rezultant al sistemului considerat.

Fie două subreţele N_A şi N_B sunt redate de expresiile respective $DE_A = A$ şi $DE_B = B$, atunci la compunerea lor prin aplicarea operatorului "$\veebar$", relativ la aceste două expresii descriptive, obţinem o reţea rezultantă N_R redată de $DE_R = C = A \veebar B$ în care locaţiile şi tranziţiile ce au acelaşi nume, respectiv, vor fi contopite. Nodurile contopite vor păstra atributele şi incidenţa arcelor din fiecare subreţea. Această operaţie este *comutativă*, *asociativă* şi *reflexivă*.

În [1] este arătat cum se poate efectua maparea DE a unei RP generalizate în reprezentarea sa grafică şi invers.

3. MODELAREA MOBILITĂŢII SERVICIILOR CDOS

În continuare, pentru a ilustra aplicarea acestui tip de reţele, vom considera un exemplu de modelare a proceselor de control distribuit orientat pe servicii reconfigurabile mobile – CDOS . Acest sistem este structurat ierarhic în mod explicit într-un set de subsisteme, fiecare din care poate reprezenta un agent (o entitate de serviciu, cod mobil) sau un site unde pot să se afle mai mulţi agenţi. Fiecare subsistem este identificat printr-un titlu (identificator). Structura ierarhică curentă a unui CDOS este explicit determinată. De exemplu, Agentul_1, (notat A1) şi Agentul_3, (notat A3) sunt agenţi imobili care se află pe două site-uri diferite (S4 şi S5). Agentul_2, (notat A2) este un agent mobil, care se află iniţial pe site-ul S4, Site_4. Pe acest site, Agentul_2 comunică cu Agentul_1, care se află pe acelaşi site, printr-un canal de comunicare C_1. Agentul_2 primeşte o informaţie (un anumit tip de date indicate $<x>$: *Informaţii*) din C_1, apoi se deplasează spre site-ul S5, unde se află Agentul_3. Pe site-ul Site_5, Agentul_2, reconfigurându-şi structura sa, transmite, prin canalul de comunicare C_2, către Agentul_3 informaţiile primite de la Agentul_1.

Pentru a descrie organizarea ierarhică a sistemului dat şi funcţionarea lui prin compunerea submodelelor RPCR, ΓC_i ale subsistemelor_*i*, vom folosi, în mod similar cu [1, 10, 11], paradigma de reţele RPCR membranale (RPCRM), care sunt o generalizare a ΓC. Modelul de reţea ΓC_i membranală este structurat într-o membrană $[_i \quad]_i$ redat de expresia descriptivă Z_i, adică $\Gamma C_i = [_i\ Z_i\ ]_i$. În Z_i vom folosi două *indexuri* pentru a codifica simbolurile-locaţii, $p_{i,k}$ şi cele ale tranziţiilor, $t_{i,j}$, unde primul index *i* arată numărul de ordine al subsistemului_*i*, adică al membranei $[_i \quad]_i$, iar al doilea index arată numărul de ordine al simbolului respectiv în modelul ΓC_i. De asemenea, pentru a exprima în mod explicit mobilitatea membranelor [1, 11] vom modifica şi regula de rescriere a RPCR în modul următor: $r = \{Loc_i : DE_L\} \triangleright \{Loc_j : DE_W\}$, unde Loc_i şi Loc_j sunt etichete ce indică localitatea submodelelor, adică faptul că subreţeaua DE_L modelului curent ce se află în membrana $[_i \quad]_i$ este eliminată, iar în membrana $[_j \quad]_j$ este plasată o nouă subreţea DE_W.

Fie $SC_0 = [_0 [_4\ S_4]_4\ [_5\ S_5]_5\ ]_0$ este *DE* a modelului curent de reţea membranală RPCRM1 cu $\Gamma C_i = [_i\ Z_i\ ]_i$ ce redă organizarea ierarhică şi funcţionarea CDOS descris mai sus în mod informal. În acest model vom considera că *DE* ale submodelelor acestui CDOS sunt determinate în modul următor:

$$S_4 = [_4 [_1\ Z_1]_1 \dot{\nabla} Z_4 \dot{\nabla} [_2\ Z_2]_2\ ]_4;\ S_5 = [_5\ Z_5 \dot{\nabla}\ [_3\ Z_3]_3\ ]_5;$$

$$Z_1 = p_{1,1} |_{t_{1,1}} p_{1,2} |_{t_{1,2}} p_{1,3} |_{r_{1,1}} p_{1,4} |_{t_{1,3}} p_{1,5};$$

$$Z_2 = Z_{2,1} \dot{\nabla} p_{2,3} |_{t_{2,3}} p_{2,5} |_{t_{2,5}} p_{2,2} \dot{\nabla} Z_{2,2}, \text{ unde}$$

$$Z_{2,1} = p_{2,1} |_{t_{2,1}} p_{2,2}[<x>,<x>] |_{t_{2,2}} p_{2,3},$$

$$Z_{2,2} = p_{2,3}[<x>] |_{t_{2,4}} p_{2,4} |_{r_{2,1}} p_{2,1} \dot{\nabla} p_{2,2} |_{t_{2,6}} p_{2,4};$$

$$Z_3 = Z_{3,1} \dot{\nabla} p_{3,3} |_{r_{3,1}} p_{3,4} |_{t_{3,3}} p_{3,5}, \text{ unde}$$

$$Z_{3,1} = p_{3,1} |_{t_{3,1}} p_{3,2}[<x>,<x>] |_{t_{3,2}} p_{3,3};$$

$$Z_4 = |_{t_{1,1}} p_{4,1}[<x>,<x>] |_{t_{2,1}},\ Z_5 = p_{5,1}[<x>] |_{t_{3,1}};$$

Funcţiile de gardă: $g^R_{1,1} = g^R_{3,1} = "false"$, $g^R_{2,1} = "true"$.

Regula de reconfigurare a agentului A2 este: $r_{2,1} = \{Loc_4 : DE1\} \triangleright \{Loc_5 : (Y_5 \dot{\nabla} [_2\ Z_2]_2)\}$, unde

$$DE1 = p_{4,1}[<x>] |_{t_{2,1}} \dot{\nabla} [_2\ Z_2]_2 .$$

Regulile de reconfigurare $r_{1,1}$ şi $r_{1,1}$ ale lui A1 şi A3 nu sunt specificate.

Maparea lui SC_0 în reprezentare grafică a modelului RPCRM1 este prezentată în figura 3.

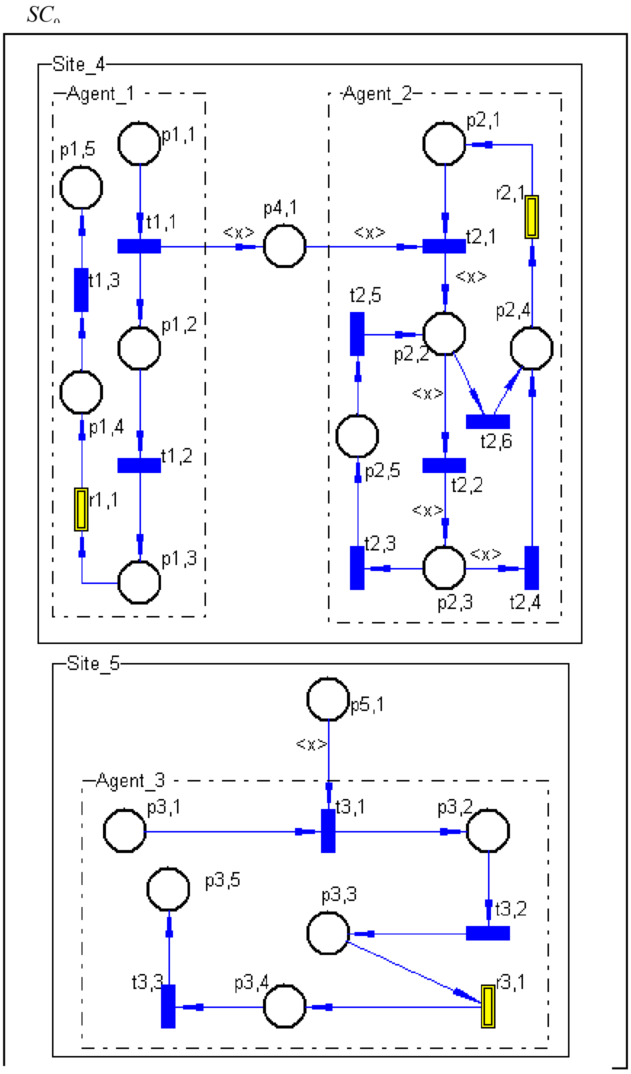


Figura 3. Modelul RPCRM1 al SC_0.

Marcajul iniţial al locaţiilor este $M_0(p_{1,1}) = <x>$, $M_0(p_{2,1}) = M_0(p_{3,1}) = <\bullet>$, unde variabila $<x>$ sunt date de tipul *Informaţii*, iar $<\bullet>$

reprezintă constanta jeton-negru (un tip care conţine doar o valoare care este <•>). Arcele neetichetate implicit au eticheta <•>.

După efectuarea reconfigurării obţinem:

$SC_0' = \big[_0 \big[_4\, S_4' \big]_4 \; \big[_5\, S_5' \big]_5 \; \big]_0$, unde

$S_4' = \big[_4 \big[_1 Z_1 \big]_1 \overset{\leftrightarrow}{\triangledown} Y_4 \overset{\leftrightarrow}{\triangledown} \big[_2 Y_2 \big]_2 \; \big]_4 ;\; S_5' = \big[_5\; Y_5 \overset{\leftrightarrow}{\triangledown} \big[_3\, Y_3 \big]_3 \; \big]_5 ;$

$Y_3 = Z_3 ;\; Y_2 = Z_{2,1} \overset{\leftrightarrow}{\triangledown} Y_{2,1} \overset{\leftrightarrow}{\triangledown} Y_{2,2}$, unde

$$Y_{2,1} = p_{2,3} \,|_{t_{2,3}}\; p_{2,5} \,|_{t_{2,5}}\; p_{2,2} ,$$

$$Y_{2,2} = p_{2,3}[<x>] \,|_{r_{2,2}}\; p_{2,6} \,|_{t_{2,6}}\; p_{2,4} \,|_{t_{2,4}} ;$$

$$Y_4 = |_{t_{1,1}}\; p_{4,1}[<x>] ; Y_5 = |_{r_{2,2}}\; p_{5,1}[<x>,<x>] \,|_{t_{3,1}} .$$

Cu funcţiile de gardă: $g_{1,1}^R = g_{2,2}^R = g_{3,1}^R = "false"$, $g_{2,1}^R = "true"$. În acest caz regulile de reconfigurare nu sunt specificate.

În figura 4 este prezentat modelul RPCRM2 al SC_0' după mişcarea lui A2 din site-ul S4 în site-ul S5, adică după reconfigurarea lui SC_0.

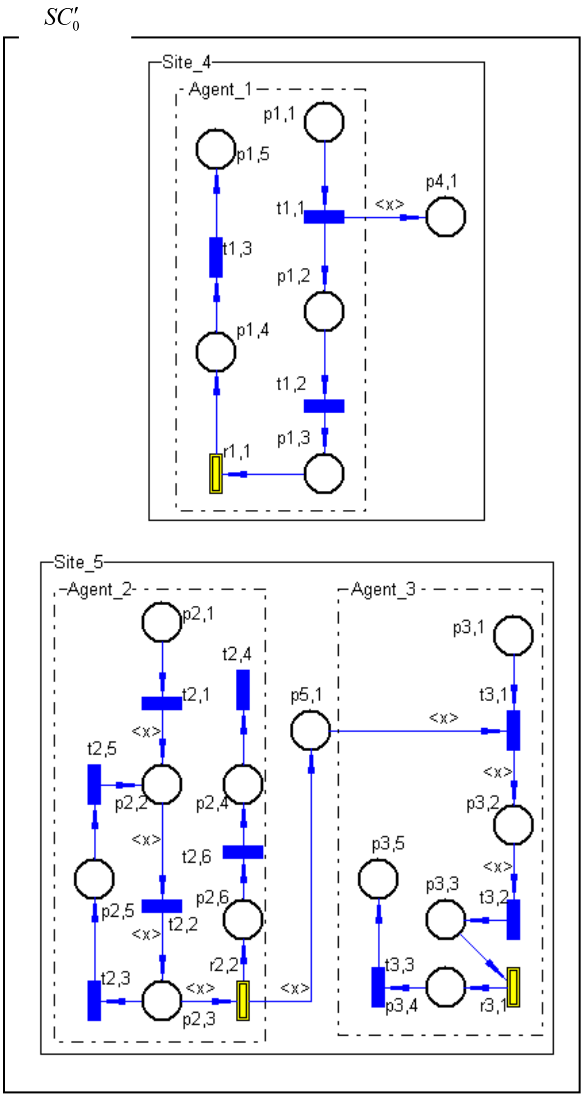


Figura 4. Modelul RPCRM1 al SC_0' .

Pentru a verifica proprietăţile comportamentale (mărginire, siguranţă, viabilitate, reversibilitate, conflicte etc) ale acestor tipuri de modele putem folosi produsul program instrumental de simulare a RP reconfigurabile care este descris în [15].

Deseori, în astfel de sisteme duratele activităţilor şi momentele de timp ale apariţiei evenimentelor, care condiţionează reconfigurările şi deplasările agenţilor mobili [7, 14, 17], sunt determinate de un proces stocastic. În acest caz este necesar de a introduce RPCR markoviene pentru a descrie şi a avea posibilitatea de a evalua unii indicatori de performanţă.

Definiţia 4. O reţea RPCR markoviană, notată $S\Gamma C$, este sistemul redat de către tripletul $S\Gamma C = <\Gamma C, \Lambda, \omega>$, unde:

- ΓC este o reţea RPCR în care evenimentele sunt temporizate, astfel încât $E = E_\tau \cup E_0$, $E_0(M) \cap E_\tau(M) = \varnothing$. Aici E_τ este mulţimea evenimentelor temporizate cu o durată aleatorie de declanşare ce are o distribuţie exponenţial–negativă (grafic sunt reprezentate prin dreptunghiuri groase), iar E_0 este mulţimea evenimentelor imediate cu o durată de declanşare nulă (grafic sunt reprezentate prin bare subţiri), astfel încât $\Pr i(E_0) > \Pr i(E_\tau)$. Acestea descriu selectoare probabilistice;
- $\Lambda : E_\tau \times Bag(C_P(p)) \to IR^+$ este funcţia ce determină rata $0 \le \lambda(e,M)(c) < +\infty$ de declanşare a evenimentului validat în marcajul curent *M*. IR^+ este mulţimea mărimelor reale nenegative.
- $\omega : E_0 \times Bag(C_P(p)) \to IR^+$ este funcţia matriceală de pondere $0 \le \omega(t,M) < +\infty$ ce determină probabilitatea de declanşare a evenimentelor imediate validate de marcajul curent *M*. ■

Acest tip de modele vor permite de a elabora o tehnologie de modelare a unor aspecte dinamice de funcţionare, simulare vizuală si evaluare a indicatorilor de performanţă ai SFR concrete.

4. CONCLUZII

În lucrare este introdusă o nouă clasă de reţele Petri colorate membranale marcaj–controlabile temporizate stocastic, care permit de a obţine un model compact şi flexibil pentru descrierea funcţionării controlului distribuit al SFP, unde în mod dinamic sunt prezente organizarea ierarhică, mobilitatea şi reconfigurabilitatea în cazul în care structura şi atributele sistemului se pot schimba în timpul funcţionării.

Aplicabilitatea acestui demers este ilustrată printr-un exemplu ce descrie procesele de control distribuit al SFR cu servicii-agenţi mobili în care sunt prelucrate diferite tipuri de cereri ale diferitor tipuri de servicii reconfigurabile.

Cercetările de imediată perspectivă se vor axa pe dezvoltarea unui produs program instrumental pentru a efectua simularea vizuală şi analiza modelelor RPCR temporizate stocastic, în care, comod şi flexibil, pot fi redate structuri ierarhice ce descriu, adecvat şi natural, arhitectura şi funcţionalitatea aplicaţiilor SFR.

Totodată, actualmente nu sunt încă studiate complexitatea şi decidabilitatea RPCR. Aceste aspecte urmează a fi dezvoltate în lucrările de viitor, dată fiind importanţa lor.

Lucrarea dată a fost efectuată în cadrul proiectului instituţional de cercetări ştiinţifice aplicative 15.817.02.28A „Modele, metode şi interfeţe pentru conducerea şi optimizarea sistemelor de fabricaţie inteligente”.

Bibliografie

1. **Guţuleac, E.** *Descriptive compositional HSPN modeling of computer systems. Annals of the University of Craiova, România, Series: Automation, Computers, Electronics and Mechatronics, Vol. 3(30), No.2, pp. 82-87, 2006.*

2. **Guţuleac, E.** *Descriptive Timed Membrane Petri Nets for Modeling of Parallel Computing. International Journal of Computers, Communications & Control, No. 3, Vol. I, Agora University Editing House, Oradea, România, pp. 33-39, 2006.*

3. http://www.informatik.uni-hamburg.de/TGI/ PetriNets/tools/quick.html

4. http://wiki.daimi.au.dk/cpntools/cpntools.wiki

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), pp. 213-254, 2007.*

6. **Kahloul, L., Chaoui, A.** *Coloured Reconfigurable Nets for Code Mobility Modeling, International Journal of Computers, Communications & Control, Suppl. issue, 3(S), pp. 358-363, 2008.*

7. **Khler, M., Moldt, D., Rlke, H.** *Modelling mobility and mobile agents using nets within nets. In W. van der Aalst and E. Best, eds. Proceeding, Applications and Theory of Petri Nets, vol. 2679 of LNCS, pp.121-139, 2003.*

8. **Llorens, M., Oliver, J.** *Structural and dynamic changes in concurrent systems: Reconfigurable Petri nets. In IEEE Transactions on Computers, Vol.53, No.9, pp. 1147-1158, 2004.*

9. **Marco Mendes, J., Leitão, P., Colombo, A. W., Restivo, F..** *High-level Petri nets for the process description and control in service-oriented manufacturing systems. International Journal of Production Research, 50:6, pp.1650-1665, 2012.*

10. ***P systems web page***. http://ppage.psystems.eu/

11. **Păun, Gh.** *Membrane Computing. An Introduction. Natural computing Series. ed. G. Rozenberg, Th. Back, A.E. Eiben .N. Kok, H.P. Spaink, Leiden Center for Natural Computing, Springer–Verlag, Berlin, 2002, p. 420.*

12. **Popescu, C., Martinez Lastra, J.L.** *An incrementtal Petri Net-derived approach to modeling of flow and resources in service-oriented manufacturing systems. IEEE 8th International Conference on Industrial Informatics (INDIN 10), pp. 253-259, 13-16 July 2010.*

13. **Richta, T., Janouek, V., Ko, R.** *Petri Nets-Based Development of Dynamically Reconfigurable Embedded Systems. In proceeding of Petri Nets and Software Engineering.* pp. 203-218, 2013.

14. **Rosa-Velardo, F., Alonso, O.M., Escrig, D. F.** *Mobile Synchronizing Petri Nets: a choreographic approach for coordination in Ubiquitous Systems. Electronic Notes in Theoretical Computer Science 150, pp.103–126, 2006.*

15. **Ţurcanu, Iu., Guţuleac, E., Cordunenu, A.** *Sistem Software de Simulare Animată a Reţelelor Petri Diferenţiale Reconfigurabile. In Proceedings of the 6-th International Conference, ICMCS 2009, vol. 1, 1-3 October, pp. 307-311, 2009.*

16. **Vlad, V., Ciufudean, C., Graur, A. , Filote, C**. *An Example of Modeling Manufacturing Systems Using Petri Nets and the IEC 61499 Standard. Proceedings of the 13th WSEAS International Conference on SYSTEMS, pp. 357-363, 2009.*

17. **Xu, D., Yi, D.** *Modeling Mobile Agent Systems with High Level Petri Nets, IEEE International Conference on Systems, Man, and Cybernetics, 5, pp. 3177-3182, 2000.*

18. **Zhang, L., Rodrigues, B.** *A Petri Net-based Approach to Reconfigurable Manufacturing Systems Modeling. Systemics, Cybernetics and Informatics, Vol. 7, no. 1, pp. 18-24, 2009.*

Recomandat spre publicare: 15.09.2015.