Modelarea Sistemelor Orientate pe Servicii prin RPS Reconfigurabile cu Trăsături Dinamice

Emilian GUȚULEAC, Sergiu ZAPOROJAN, Ion GÎRLEANU, Alexei SCLIFOS, Inga IAVORSCHI
Technical University of Moldova
Emilian.Gutuleac@calc.utm.md,

***Abstract* — În lucrare sunt definite rețele Petri stocastice (RPS) reconfigurabile marcaj – controlabile cu trăsături dinamice (ReRPST), care permit de a descrie formal, în mod compact și flexibil, funcționarea sistemelor orientate pe servicii cu evenimente discrete (SEDS) orientate pe servicii cu un grad ridicat de variabilitate, unde în mod dinamic sunt prezente schimbări de structură și trăsături funcționale cu atribute specifice. Expresiile de actualizare trăsături, asociate cu tranzițiile și regulile de reconfigurare ale modelului ReRPF, permit de a reda în mod natural și concis schimbări dinamice de structură și selectare a trăsăturilor funcționale specificate ale sistemului pe parcursul funcționării lui.**

***Index Terms* — rețele Petri stocastice, reconfigurare, sisteme, servicii, trăsături dinamice.**

I. INTRODUCERE

Actualmente, sistemele cu evenimente discrete (SED) orientate pe servicii (SEDS) în timp real cunosc o dezvoltare rapidă, atât sub aspectul complexității și/sau performanțelor, cât și al ariei de răspândire [2,11]. Acest tip de sisteme (de exemplu, sisteme multiprocessor, sisteme de fabricație reconfigurabile (SFR), aplicații de control reconfigurabil, etc.) trebuie să aibă o flexibilitate, agilitate, disponibilitate și siguranță în funcționare deosebită. Astfel, un sistem SEDS poate fi considerat și implementat drept fiind o colecție de configurații, unde fiecare din acestea este o rețea de subsisteme 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 creșterea s*iguranței în funcționare* și a flexibilității utilizării sistemului [2, 3, 11], asigurarea cărora în timpul reconfigurării dinamice este dificilă din cauza interacțiunii între serviciile de aplicație care sistemul le oferă utilizatorului.

Un SEDS are capacitatea de a se actualiza el însuși, să răspundă cerințelor dinamice sau defecțiunilor imprevizibile. Reconfigurabilitatea unui SEDS se face în timpul funcționării. Prin urmare, sistemul nu trebuie să înceteze să funcționeze, ci el trebuie să lucreze și să se reconfigureze conform unor reguli specificate. Această activitate de reconfigurare necesită timp și cost.

Nevoia de flexibilitate și adaptabilitate conduce inevitabil la integrarea *trăsăturilor* de reconfigurabilitate în aceste modele, dar face sistemul mai complex și dezvoltarea lui este o sarcină dificilă. Prin urmare, este o necesitate crucială de folosi noi abordări pentru proiectarea, verificarea și utilizarea a astfel de sisteme sigure cu proprietăți de reconfigurabilitate.

Utilizarea metodelor formale (rețele Petri (RP), automate, logici formale, etc) la proiectarea SEDS-uri aduce mai multe avantaje: specificație la nivel înalt; simulare; verificarea proprietăților comportamentale; analiza performanțelor, etc. Având în vedere aspectul de reconfigurare, s-au propus numeroase RP-uri extinse [9] pentru a modela și analiza reconfigurabilitatea SEDS în timpul rulării. În [12] în baza unor gramatici ale grafurilor sunt definite reguli de transformare ale RP, utilizate la rafinamentul sistemelor de fabricație. Pe lângă tehnicile bazate pe transformări ale grafurilor, s-au folosit sisteme de rescriere a RP, prezentate în [1, 4, 8] care au fost extinse în [5, 7] în formalismul INRS (Improved Net Rewriting Systems). Cu toate acestea, chiar dacă metodele bazate pe reconfigurabilitatea RP oferă abordări explicite și intuitive de modelare în care structura formalismului mapează structura reconfigurabilă a sistemului, acestor formalisme le lipsesc tehnici de analiză, absența timpului și aspecte stochastice. În această lucrare suntem interesați să definim o abordare a reconfigurării RP generalizate stochastice (RPGS) fără a pierde proprietățile comportamentale ale sistemului în care sunt preponderente următoarele funcționalități ale acestora: abstractizarea, dinamicitatea, preemțiunea, recursivitatea și reconfigurabilitatea.

Pentru a descrie în mod concis și flexibil unele *trăsături* funcționale, care pot fi *schimbate* în mod dynamic, în această lucrare, definim o extensie a RPGS reconfigurabile (RPGSR), definite în [5], cu *trăsături* dinamice, numite ReRPSF, care sunt obținute prin îmbogățirea RPGSR în baza conceptului de selectare și actualizare *trăsături* (eng. *features*), introduse în [10] pentru modelarea sistemelor liniilor de produse software cu un grad ridicat de variabilitate, o abordare cu o provocare importantă în ingineria elaborării produselor program. O *trăsătură* este definită în [10] ca fiind un aspect proeminent de calitate sau caracteristică al unui sistem distinctiv vizibil de către utilizatorul final al unui sistem.

Avantajul îmbinării unor astfel de abordări constă în faptul că modelele ReRPST descriu mai nuanțat comportamentul așteptat al unor SEDS specific.

În continuare, definim formalismul ReRPST în care reconfigurarea descriptiv-compozițională este efectuată în mod dinamic prin reguli de rescriere, combinate cu întreruperi și rafinamente la activarea/dezactivarea tranzițiilor și a regulilor de rescriere, care sunt asigurate prin condiții de aplicare *trăsături* și actualizări ale *expresiilor de trăsături*. Mai mult, formalismul ReRPSF captează și comportamentul întregului SEDS de control reconfigurabil într-un model modular compozițional concis, deschizând astfel calea spre o posibilă analiză și verificare eficientă.

II. RPGS RECONFIGURABILE

În continuare, cu scopul de a trata unele probleme menţionate mai înainte la modelarea şi analiza aplicaţiilor SEDS reconfigurabile şi a introduce ReRPST, prezentăm succint definiţia formalismului RPGSR, descris mai detaliat în [5, 6], cu o mulţime de reguli de rescriere $R = \{r_1, ..., r_k\}$, care la apariţia unor evenimente specificate, modifică atât marcajul curent, cât şi structura curentă a lui RPGSR cu atributele respective. 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 al unei tranziţii $t_j \in T$ şi/sau reguli de rescriere $r_j \in R$ este similară cu cea a unei RP generalizate [5, 9].

Definiţia 1. O reţea RPGS *dinamic descriptiv-reconfigurabilă* (RPGSR), notată Γ, este o structură constituită din următorul 13- tuplu: $\Gamma = < P, E, Pre, Post, Test, Inh, Pri, G_E, G_R, K_p, \Lambda, \omega, M_0 >$, unde :

- P este o mulţime nevidă de locaţii; E este o mulţime 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 reţelei curente cu atributele sale respective.

la rândul lor, sunt partiţionate $E = E_\tau \cup E_0$, $E_\tau \cap E_0 = \varnothing$ astfel, încât: E_τ este mulţimea evenimentelor temporizate cu durate de declanşare aleatoare ce au o distribuţie exponenţial-negativă, iar E_0 este mulţimea evenimentelor imediate ce au o durată de declanşare nulă. Grafic, tranziţiile temporizate sunt reprezente prin bare groase, cele imediate - prin bare subţiri, iar regulile de rescriere sunt reprezentate prin dreptunghiuri imbricate respective;

- Arcele, redate de funcţiile *Pre, Test* şi $Inh : P \times E \times IN^{|P|} \to IN$ (resp. $Post : E \times P \times IN^{|P|} \to IN$) sunt aplicaţii de incidenţă *înainte* (resp. *înapoi*), *test* şi *inhibiţie* definite pe $P \times E$ (resp. $E \times P$). Aceste funcţii sunt dependente de marcajul curent. Implicit, ponderile arcelor ce sunt egale cu 1 nu sunt menţionate. *IN* este mulţimea numerilor întregi naturale;
- $Pri : E \times IN^{|P|} \to IN$ 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 evenimete e_j sunt considerate nule. Implicit totdeauna $Pri(E_0) > Pri(E_\tau)$;
- Funcţiile de gardă $G_E : E \times IN^{|P|} \to \{true, false\}$ şi $G_r : R \times IN^{|P|} \to \{true, false\}$ care pentru orice $e_j \in E$ şi $r_k \in R$ determină respectiv funcţii Booleene $g_j^E(M)$ şi $g_k^R(M)$ în marcajul curent *M*. Astfel, dacă evenimentul e_j este validat de marcajul curent M, notat $M[e_j >$ şi $g_j^E(M)$='*true*', atunci e_j rămâne validat şi, eventual, el poate fi declanşat, altfel acesta nu este validat. Implicit $g_j^E(M)$='*true*'. În cazul în care e_j este validat şi $g_j^R(M)$='*false*', atunci acesta, fiind declanşat, va *shimba numai* marcajul curent al reţelei Γ, însă dacă $e_j = r_j$ şi $g_j^R(M)$='*true*', el va modifica atât structura cu unele atribute curente ale Γ, cât şi marcajul ei curent în conformitate cu specificaţiile acestei reguli. Implicit $g_j^E(M)$ este o constantă '*true*' ;
- $K_p : P \times IN^{|P|} \to (IN \cup +\infty)$ este funcţia de capacitate a locaţiilor astfel, încât $\forall p_i \in P$, aceasta este redată de capacitatea maximă de jetoane $0 < K_p(p_i) < +\infty$ care poate să se afle în locaţia p_i. Implicit, $K_p(p_i)$ este nelimitată;
- $\Lambda : E_\tau \times IN^{|P|} \to IR_+$ este funcţia ce descrie rata de declanşare $\lambda(e, M)$ a evenimentului temporizat $e \in E_\tau(M)$ validat de marcajul curent M, adică rata de declanşare este parametrul legii exponenţial-negative;
- $\omega : E_0 \times IN^{|P|} \to IR_+$ este funcţia ce descrie viteza ponderată de declanşare $\omega(e, M)$ a evenimentului imediat validat de marcajul curent M, în baza căreia este determinată probabilitatea de declanşare a acestui eveniment. IR_+ este mulţimea numerilor reale;
- $M_0 : P \to IN$ este marcajul iniţial ce determină o funcţie de marcare a locaţiilor *P*. ■

Regulile de funcţionare ale unei reţele Γ şi cele de construire a lanţului Markov timp continuu (LMTC) inclus sunt descrise mai detaliat în [5]. În următorul compartiment sunt prezentate succinct aceste reguli de funcţionare. Menţionam că aceste reguli *R* permit de a efectua reconfigurarea on-line a modelelor RPGS, cu deosebire de abordările [1, 7, 8] care rescriu manual modele RP .

Notăm faptul, că reţelele tip RPGS [5, 9] pot fi obţinute din reţele RPGSR ca un caz particular pentru care mulţimea regulilor de rescriere *R* este vidă.

III. RPGSR CU TRĂSĂTURI DINAMICE

Cum deja s-a menţionat, ReRPST cu *trăsături* dinamice sunt obţinute prin îmbogăţirea modelelor RPGSR în baza conceptului de selectare *trăsături* şi actualizare *trăsături*, introdus în [10], prin asocierea la mulţimea evenimentelor *E* (tranziţii şi reguli de rescriere) a unor condiţii de aplicare *trăsături* şi expresii de actualizare a acestor *trăsături pe parcursul funcţionării*. O condiţie de aplicare *trăsături* este o formulă logică pe o mulţime de *trăsături F*, care descrie combinaţiile de *trăsături* aplicate mulţimii de evenimente *E*. Ea consitute o condiţie necesară (deşi nu suficientă) pentru ca evenimentul validat să declanşeze. De fapt, în cazul în care condiţia de aplicare *trăsături*, asociată evenimentului $e \in E$, este falsă, înseamnă că acest eveniment este dezactivat. O expresie de actualizare *trăsături*, notată $u(e)$, descrie selectarea şi actualizarea *trăsăturilor* ce evoluează după declanşarea lui e.

În continuare, prezentăm unele definiţii şi notaţii, în conformitate cu [10], care sunt necesare pentru a introduce formalismul ReRPST cu regulile lui de funcţionare.

Definiția 2. (*Condiție de aplicare trăsătură* [10]). O condiție de *aplicare trăsătură* φ este o constrângere logică (Booleană) pe o mulțime de *trăsături* F, definită de gramatica: $\varphi ::= true \mid a \mid \varphi \wedge \varphi \mid \neg\varphi$, unde $a \in F$, în care conectorii logici sunt folosiți în mod obișnuit. Pentru a indica mulțimea tuturor condițiilor de aplicare *trăsături* din F folosim notația Φ_F. ■

Definiția 3. (*condiții de aplicare trăsături* [10]). Fie φ, o condiție de aplicare și F_S o submulțime de *trăsături*, numită selectare *trăsături*. F_S satisfice φ, scris $F_S \models \varphi$, dacă și numai dacă (iff): (1) *întotdeauna* $F_S \models true$; (2) $F_S \models a$, iff $a \in F_S$; (3) $F_S \models \neg\varphi$, iff $F_S \not\models \varphi$; (4) $F_S \models \varphi_1 \wedge \varphi_2$, iff $F_S \models \varphi_1$ și $F_S \models \varphi_2$. ■

Definiția 4. (*Actualizare trăsături*). O actualizare (*update*) *trăsături* u este definită de următoarea gramatică [10]: $u ::= noop \mid a\ on \mid a\ off \mid u;u$, unde $a \in F$, iar F este mulțimea de *trăsături.* Vom scrie U_F pentru a indica mulțimea actualizărilor totale în F. Având o selectare *trăsături* curente $F_S \subseteq F$, o expresie actualizare modifică F_S conform următoarelor reguli $\rho =$ (*condiție de aplicare*)/(*expresie de actualizare*) [9]:

$$\rho 1 : F_S \xrightarrow{noop} F_S;\ \rho 2 : F_S \xrightarrow{a\ on} F_S \cup \{a\};$$

$$\rho 3 : F_S \xrightarrow{a\ off} F_S \setminus \{a\};\ \rho 4 : \frac{F_S \xrightarrow{u_0} F_S' \quad F_S \xrightarrow{u_1} F_S''}{F_S \xrightarrow{u_0;u_1} F_S''}.$$

■

În baza RPGSR și a definițiilor enunțate mai sus, relativ la trăsături, introducem formal rețele ReRPST, care sunt obținute prin asocierea a unor *trăsături* funcționale cu tranzițiile și regulile de reconfigurare ale RPGSR în care sunt definite condiții de aplicare și reguli de actualizare ale acestor *trăsături* pe parcursul funcționării ReRPST. O condiție de aplicare *trăsătură* consitute una din condițiile necesare pentru ca tranziția sau regula de reconfigurare să declanșeze. În cazul în care condiția de aplicare *trăsătură* este falsă, înseamnă că tranziția sau regula de reconfigurare este dezactivată. O expresie de actualizare *trăsătură*, descrie selectarea *trăsăturii* ce trebuie să evolueaze după declanșarea evenimentului selectat.

Definiția 5. O rețea RPGSR *cu trăsături dinamice,* ReRPST, este 4- tuplul $\Gamma F =< \Gamma, F, \phi, u >$, unde:

Γ este o rețea RPGSR conform definiției 1; F este mulțimea de *trăsături*; $\phi : E \to \Phi_F$ este o funcție asociată la fiecare eveniment $e \in E$, fiind drept o condiție de aplicare *trăsături* din Φ_F; $u : E \to U_F$ este o funcție ce asociază la fiecare eveniment $e \in E$ o actualizare din U_F. O stare curentă a unei rețele ΓF este un marcaj extins M^x, redat de cuplul $M^x = (M, F_S)$, unde M este marcajul curent al rețelei Γ, subiacente lui ΓF, iar $F_S \subseteq F$ sunt *trăsăturile* curente selectate. Starea initială a lui ΓF este $M_0^x = (M_0, F_{S0})$. ■

În continuare, vom nota u_e pentru a indica expresia de actualizare $u(e)$, asociată cu un eveniment $e \in E$. De asemenea, vom scrie $F_S \models \phi(e)$ în cazul în care *trăsăturile* selectate F_S îndeplinesc condiția de aplicare asociată cu $e \in E$. Grafic, fiecare eveniment a lui ΓF va fi adnotat cu o condiție de aplicare *trăsături* și cu o expresie de actualizare *trăsături* în următorul mod: (*condiție de aplicare*)/(*expresie de actualizare*).

Definiția 6**.** *(Validarea unui eveniment).* Un eveniment e_j al unei rețele ΓF este validat de marcajul extins current M^x, notat $e_j \in E(M^x)$, dacă este verificată condiția de validare $ec^{\Gamma F}(e_j, M^x)$, care este redată de următoarea expresie logică:

$$ec^{\Gamma F}(e_j, M^x) = ec^{\Gamma}(e_j, M) \wedge (F_S \models \phi(e_j)),$$

unde $ec^{\Gamma}(e_j, M)$ este condiția de validare a evenimentului e_j de către marcajul current al rețelei Γ, suiacentă lui ΓF, iar $F_S \models \phi(e_j)$ este satisfacerea condiției de aplicare a trăsăturii $\phi(e_j)$ la acest eveniment. ■

Fie $E(M^x) = T(M^x) \cup R(M^x)$, este mulțimea de evenimente validate de către marcajul extins M^x al rețelei curente ΓF. De asemenea, fie A=<*Pre*, *Post*, *Test*, *Inh* > este mulțimea arcelor rețelei $\Gamma F =< RN, M^x >$, iar ΓF și RN sunt reprezentate de expresii descriptive DE_Γ și DE_{RN} respective [5, 6]. Regula de rescriere $r \in R$ în mod dinamic a rețelei ΓF curente la declanșarea unui eveniment validat $e_j \in E(M^x)$ 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 $\Gamma F_L \subseteq \Gamma F$ rețelei curente ΓF astfel, încât $P_L \subseteq P$, $E_L \subseteq E$ cu mulțimea arcelor $A_L \subseteq A$. În același mod, *domeniul* $Dom(\triangleright) = \triangleright DE_W$ lui $\triangleright$, determină de o expresie descriptivă DE_w specificată a unei subrețele noi $\Gamma F_w \subseteq \Gamma F'$ a rețelei modificate $\Gamma F'$ 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_Γ și, deci, a ΓF curentă prin *rescrierea* ei (eng. *rewriting*). La *schimbarea* expresiei descriptive DE_L specificate a subrețelei $\Gamma F_L \subseteq \Gamma F$ (DE_L și deci ΓF_L sunt eliminate, obținându-se respectiv $DE_\Gamma \setminus DE_L$ și $\Gamma F \setminus \Gamma F_L$) cu o nouă expresie descriptivă DE_w specificată a subrețelei ΓF_w (DE_w și ΓF_w sunt adăugate respectiv la $DE_\Gamma \setminus DE_L$ și respectiv $\Gamma F \setminus \Gamma F_L$). Astfel, rezultă o nouă expresie descriptivă DE'_Γ a rețelei noi modificate $\Gamma F' = (\Gamma F \setminus \Gamma F_L) \cup \Gamma F_w$ 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 $\setminus$ (resp. $\cup$) indică operația de eliminare a ΓF_L din (resp. adăugarea ΓF_w în) ΓF. La eliminarea unor locații și/sau evenimente *arcele* ce le conectează se vor elimina în mod implicit, iar la adăugarea subrețelei respective ΓF_w nodurile ce au același nume se vor contopi,

iar jetoanele aceloraşi locaţii se vor aduna. Deci, în această nouă reţea modificată $\Gamma F'$, obţinută la declanşarea regulii de rescriere $r_j \in E(M^x)$ validate de marcajul curent M^x, locaţiile şi evenimentele respective, ce au aceleaşi atribute sunt contopite. Implicit, $r: DE_L \triangleright \varnothing$ şi $r: \varnothing \triangleright DE_w$ descriu respectiv $\Gamma F' = \Gamma F \setminus \Gamma F_L$ şi $\Gamma F' = \Gamma F \cup \Gamma F_w$. Menţionăm, că în ΓF_L şi ΓF_w pot fi considerate aparte şi ca submulţimi ale locaţiilor P_L şi/sau P_w cu marcaje respective, ale evenimentelor E_L şi/sau E_w şi ale arcelor A_L şi/sau A_w.

Definiţia 7. (*Regula de declanşare* a unui eveniment al reţelei ΓF). Fie $\Gamma F = (RN, M^x)$ este configuraţia reţelei curente, numită *configuraţie sursă*. Evenimentul $e_j \in E(M^x)$, validat de marcajul extins M^x curent, este declanşat dacă *nu există* un alt $e_k \in E(M^x)$ cu o prioritate superioară lui, adică $\neg\exists\,(Pri(e_j) > Pri(e_k))$, pentru care sunt verificate precondiţiile sale de validare. La declanşarea lui e_j, acesta va modifica, în dependenţă de valoarea lui $g_j^R(M^x)$ şi $F_S \models \phi(e_j)$, fie numai marcajul extins curent sau va modifica atât structura şi marcajul extins, cât şi unele atribute ale reţelei curente. Astfel, la declanşarea acestui eveniment avem:

If $(((e_j = t_j) \vee (e_j = r_j)) \wedge (g_j^R(M^x) = False))$ ***then*** (*Declanşarea tranziţiei* t_j sau a regulii de rescriere r_j validate de marcajul extins curent M^x va schimba numai acest marcaj curent şi va actualiza trăsăturile funcţionale respective în această configuraţie de reţea curentă, adică:

$$(RN, M^x) \xrightarrow{e_j} (RN, M'^x) \Leftrightarrow$$
$$(RN = RN,\ M^x[e_j > M'^x,\ F_S \xrightarrow{u(e_j)} F_S'))$$

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

$$(RN, M^x) \xrightarrow{r_j} (RN', M'^x) \Leftrightarrow$$
$$(RN = RN',\ M^x[r_j > M'^x,\ F_S \xrightarrow{u(e_j)} F_S')).$$

Starea obţinută după declanşarea regulii r_j este $\gamma' = (RN', M'^x)$. Configuraţia reţelei ΓF_0 iniţiale este (RN_0, M_0^x), iar (RN, M^x) este configuraţia reţelei curente ΓF.

Graful de stări accesibile $GA(\Gamma F_0)$ ale reţelei $\Gamma F_0 =< RN_0, M_0^x >$ este un graf orientat etichetat în care vârfurile sunt etichetate cu stările de configuraţii (RN, M^x), iar arcele ce leagă aceste vârfuri sunt etichetate cu evenimente e_j de tip tranziţii sau reguli de rescriere respective, astfel încât: a) declanşarea $e_j \in E(M^x)$ determină un arc ce este etichetat cu e_j, unde $e_j = t_j$ sau $e_j = r_k$ pentru care $g_k^R(M^x) = "false"$, de la (RN, M^x) - starea sursă către starea nouă (RN, M'^x) în care structura reţelei ΓF rămâne aceeaşi, iar marcajul extins curent M^x este modificat într-un nou marcaj M'^x şi de asemenea, sunt actualizate trăsăturile curente F_S în F_S': $M'^x = M^x + Post(e_j, \cdot) - Pre(e_j, \cdot)$ şi $F_S \xrightarrow{u(e_j)} F_S'$;

b) declanşarea regulii de rescriere validate $r_j \in R(M^x)$ din configuraţia curentă (RN, M^x) pentru $g_k^R(M^x) = "true"$ conform specificaţiilor operatorului $\triangleright$ de reconfigurare al modelului de reţea ΓF va conduce la modificarea configuraţiei curente (RN, M^x) într-o nouă configuraţie (RN', M'^x) ■

În Fig.1este prezentat un exemplu de RPGSR în care r_3 aparţine reţelei $\Gamma 1$, iar r_4 aparţine lui $\Gamma 2$.

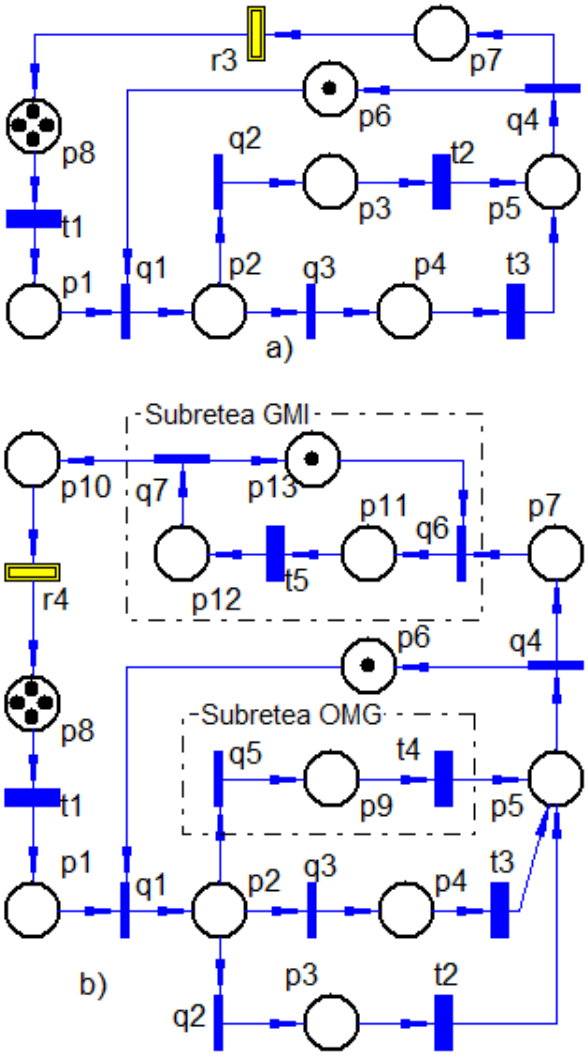

Fig. 1. Exemplu de reconfigurare RPGSR:
a) modelul $\Gamma 1$; b) modelul $\Gamma 2$.

$$r_3 : DE_{l1} \triangleright DE_{w1},\ DE_{l1} = r_3,\ g^{r_3}(M) = (M(p_7) = 4),$$
$$DE_{w1} = p_2 \,|_{q_5}\, p_9 \,|_{t_4}\, p_5 \,\tilde{\vee}\, (p_7 \bullet 1 p_{13}) \,|_{q_6}$$
$$p_{11} \,|_{t_5}\, p_{12} \,|_{q_7}\, (p_{13} \Diamond p_{10} \,|_{r_4}\, 4 p_8).$$

$$r_4 : DE_{l2} \triangleright DE_{w2},\ DE_{w2} = p_7 \mid_{r_3} 4p_8,$$
$$DE_{l2} = \{q_5, q_6, q_7, t_4, t_5, , r_4, p_9, p_{10}, p_{11}, p_{12}, p_{13}\},$$
$$g_4^R(M) = (M(p_7) = 3) \wedge (M(p_8) = 1).$$

Pentru a păstra proprietățile comprtamentale BLR (adică cele de: mărginire, B; viabilitate, L; reversibilitate, R) la reconfigurarea unui model ReRPST vom adapta abordarea IRNS, prezentată în [6, 7], care este un sistem de rescriere rețea RP, utilizat pentru a reconfigura "rapid" modele RP. Prin expresia "rapid", se înțelege că nu este nevoie de a verifica proprietățile comportamentale BLR ale modelelor reconfigurate, deoarece se demonstrează că ele nu vor pierde aceste proprietăți după aplicarea unei reguli de rescriere utilizând abordarea IRNS. Ideea principală de a adapta această abordare este de a avea o bibliotecă de clase de blocuri subrețea "*bine – formate*" ReRPST și apoi a substitui o subrețea bine - formată a oricărei ReRPST1 curente cu proprietăți BLR cu o structură de subrețea bine-formată de același tip de interfață. Această transformare duce întotdeauna la o altă ReRPST2 cu proprietăți BLR.

Cu toate acestea, aplicarea directă a abordări [6, 7] la un model RPGSR1 (sau ReRPST1) nu garantează că RPGSR2 (sau ReRPST2) reconfigurat își va păstra aceleași proprietăți comportamentale. Astfel, în cazul unui conflict structural este necesar ca toate evenimentele e incidente înapoi la locația p, $\forall e \in p^\bullet$, să fie immediate sau temporizate. Aici pentru un nod x (locație sau eveniment) în Γ sau ΓF, notația ${}^\bullet x$ (respectiv $x^\bullet$) este mulțimea de noduri, diferită de x, incidentă înainte (respectiv înapoi) la nodul x. Pentru a depăși acest tip de limitări, extindem abordarea [6, 7] relativ la rețele Γ sau ΓF. În continuare folosim /modificăm câteva specificații de clase bloc rețea utilizate în [6, 7] pentru permite reconfigurarea unei rețele Γ sau ΓF cu proprietăți BLR la o altă rețea Γ' sau $\Gamma F'$ cu aceleași proprietăți.

Blocurile propuse corespund naturii Γ și ΓF. Urmăm aceleași notații ce sunt prezentate în [6]. Astfel, considerăm două tipuri de locații $P^o = \{p \mid M_0(p) = 0\}$ - numite subseturi locații de operații și $P^r = \{p \mid M_0(p) \geq 1\}$ - numite subseturi locații de resurse.

O rețea Γ sau ΓF este numită subrețea cu o *Singură Locație* (SL), dacă $P = P^o = \{p\}$, $E = A = \varnothing$, A este mulțimea arcelor incidente la P. În acest caz, interfața este de tipul locație care este $p_{in} = p_{out} = p$.

O rețea Γ sau ΓF este numită subrețea cu un *Singur Eveniment* (SE), dacă $E = \{e\}$, $P = A = \varnothing$. În acest caz, interfața este $e_{in} = e_{out} = e$.

O rețea Γ sau ΓF este numită *Mașină de Stări Deschsă* (MSD), dacă sunt îndeplinite următoarele condiții: 1) $P = P^o, |P| > 1, |E| \geq 1$ și nu există nici un circuit direct în graful lui Γ sau ΓF ; 2) $\exists p_i,\ p_j \in P^o$, $i \neq j$ astfel încât ${}^\bullet p_i = p_j^\bullet = \varnothing$; 3. $|{}^\bullet e| = |e^\bullet| = 1$, $\forall e \in E$; 4. $p^\bullet \subseteq E_0$ sau $p^\bullet \subseteq E_\tau$, $\forall p \in P^o$.

În acest caz, interfața este de tip locație compusă din cele două locații: $p_{in} = p_i$ și $p_{out} = p_k$. Prezența a două sau mai multe evenimente concurente care aparțin diferitor tipuri (adică eveniment imediat sau temporizat) într-o Γ sau ΓF o face ca acestea să nu fie viabile [3]. Astfel, se adaugă a patra condiție.

O rețea Γ sau ΓF marcată este numită *Graf Marcat Deschis* (GMD), dacă sunt îndeplinite următoarele condiții: 1) $P = P^o, |P| \geq 1, |E| > 1$ și nu există nici un circuit direct; 2) $\exists e_i, e_j \in E, i \neq j$ astfel încât ${}^\bullet e_i = e_j^\bullet = \varnothing$; 3) $|{}^\bullet p| = |p^\bullet| = 1, \forall p \in P^o$. În acest caz, interfața este compusă din două evenimente $e_{in} = e_i$ și $e_{out} = e_j$.

O rețea RPGS marcată Γ este numită *Graf Marcat Închis* (GMÎ), dacă sunt îndeplinite următoarele condiții: 1) $P^r = \{p'\}, |P^o| \geq 1$; 2) $\exists e_i, e_j \in E, i \neq j$ astfel încât ${}^\bullet e_i = \{p'\}$ și $e_j^\bullet = \{p'\}$; 3) Există un bloc de clasă GMD, $\Gamma' = \Gamma F' = < P^o, E, A'; M_0' >$ în Γ' sau $\Gamma F'$ astfel încât avem: $A' = A - \{A(p', e_i)\} - \{A(e_j, p')\}$ și $M_0'(p) = M_0(p),\ \forall p \in P^o$. În acest caz, interfața este compusă din două evenimente $e_{in} = e_i$ și $e_{out} = e_j$.

În Fig. 2 sunt prezentate unele exemple de blocuri subrețele Γ bine-formate.

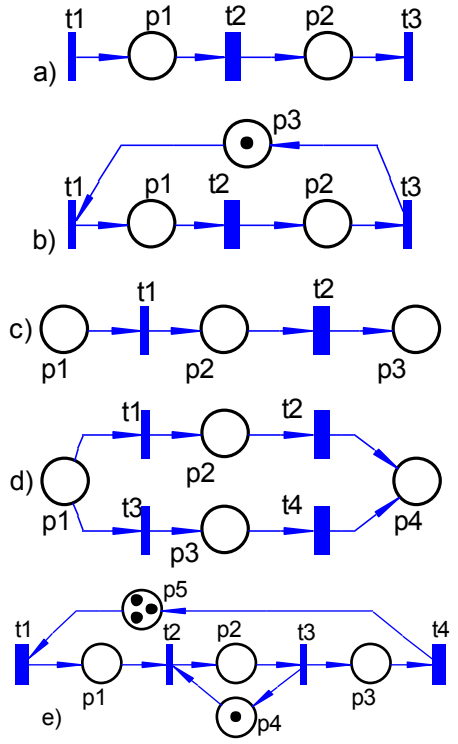

Fig. 2. Exemple de blocuri subrețele Γ bine-formate: a) GMD; b) GMÎ ; c), d) MSD; e) GMÎS.

Menționăm că o rețea Γ sau ΓF marcată, numită *Graf Marcat Închis Sequential* (GMÎS), poate fi generată de la un GMÎ prin rescrierea, eventual de mai multe ori, a unuia din blocurile de subrețele bine-formate din clasa GMD sau SE prin alt bloc de subrețea bine-format de clasă GMÎ, astfel încât nodurile de intrare/ieșire ale celor două blocuri de subrețele aparțin aceluiași tip (adică toate din ele sunt fie evenimente imediate sau evenimente temporizate) [5].

Motivul acestei restricții a fost deja menționat mai sus.

Astfel, folosind blocuri de subrețele bine-formate, definite de mai sus, proiectantul poate specifică și analiza reconfigu-rabilitatea în SEDS folosind modele de tipul Γ sau ΓF marcate.

Din cauza constrângerilor de spațiu prezentăm în Fig. 3 doar un exemplu simplu pentru a ilustra funcționarea modelelor ReRPST în care modelul $\Gamma F1$ este reconfigurat în modelul $\Gamma F2$, prin declanșarea regulii de rescriere r_1 și invers $\Gamma F2$ în $\Gamma F1$ prin declanșarea lui r_2.

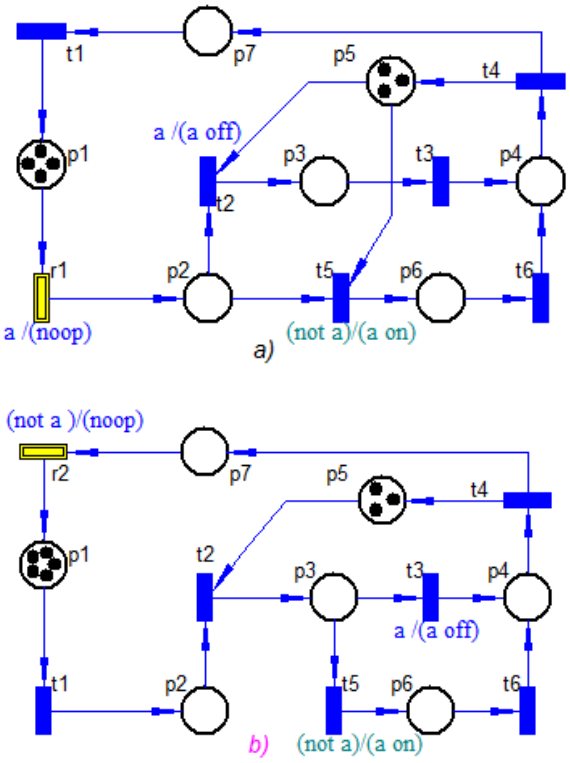

Fig. 3. Exemplu de reconfigurare:
a) modelul $\Gamma F1$; b) modelul $\Gamma F2$.

De asemenea, notăm faptul, că rețelele tip RPGSR pot fi obținute din rețele ReRPST ca un caz particular pentru care mulțimea de *trăsături F* este vidă.

IV. CONCLUZII

În lucrare sunt definite rețele Petri generalizate stocastice reconfigurabile marcaj – controlabile cu trăsături dinamice (ReRPST), care permit de a obține un model compact pentru descrie funcționarea sistemelor orientate pe servicii cu un grad ridicat de variabilitate, unde în mod dinamic sunt prezente schimbări de structură și trăsături funcționale. Expresiile de actualizare trăsături, asociate cu tranzițiile și regulile de reconfigurare a modelului ReRPF, permit de a reda în mod natural și concis schimbări dinamice de structură și selectare a trăsăturilor funcționale specificate ale unui sistem pe parcursul funcționării lui. Din cunoștințele noastre, abordărea ReRPF este una dintre primile care permit de a capta atât variabilitatea trăsăturilor funcționale, cât și aspectele dinamice de scimbare a structurii ale unui SEDS intr-un singur formalism.

Pe viitor, vom considera modele ReRPF în care vom lua în considerație aspectul stochastic fuzzy intuitionist de funcționare al proceselor de calcul cu aplicații orientat pe servicii reconfigurabile și vom prezenta mai detaliat aplicabilitatea acestui demers.

Lucrarea dată a fost efectuată în cadrul Proiectului Național de Cercetări Științifice Aplicative 15.817.02.28A din Republica Moldova.

REFERINȚE

[1] Badouel, M. L. E.; Oliver, J. *Modeling concurrent systems: Reconfigurable nets*. In: Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 2003), CSREA Press, 2003, p. 1568–1574.

[2] Z. Ding, J. Wang, C. Jiang, "An Approach for Synthesis Petri Nets for Modeling and Verifying Composite Web Service," J. Inf. Sci. Eng. 24(5), pp.1309–1328, 2008.

[3] Y. Dong, Y. Xia, T. Sun, Q. Zhu, "Modeling and performance evaluation of service choreography based on stochastic Petri net," JCP 5(4), pp. 516–523, 2010.

[4] S. U. Guan, S. S. Lim, "Modeling adaptable multimedia and self-modifying protocol execution," Future Gener. Comput. Syst. 20(1), pp.123–143, 2004.

[5] E. Guțuleac, "Descriptive self-reconfigurable generalized stochastic Petri nets for performance modeling of computer systems," Buletinul Institutului Politehnic din Iași, Tomul LI (LV), Fasc. 1-4, Secția Automatică și Calculatoare, Ed.: Universitatea Tehnică "Gh. Asachi", Iași, România, pp. 121-136, 2005.

[6] E. Guțuleac, Iu. Țurcanu, Em. Guțuleac, "Compunerea Descriptivă a Modelelor de Rețele Petri pentru Verificarea Sistemelor Orientate pe Servicii," Proceedings of the 3-rd International Conference ICTEI-2010, Vol. 2, Chișinău, R. Moldova, May 20-23, pp. 114-119, 2010.

[7] J. Li, X. Dai, Z. Meng, "Improved net rewriting system-based approach to model reconfiguration of reconfigurable manufacturing systems," Int. J. Adv. Manuf. Technol., pp. 1168–1189, 2008.

[8] M. Llorens, J. Oliver, "Structural and dynamic changes in concurrent systems: reconfigurable Petri nets," IEEE Transactions on Computers, 53(9), pp.1147–1158, 2004.

[9] T. Murata, "Petri nets: Properties, analysis and applications," Proceedings of the IEEE 77(4), pp. 541–580, 1989.

[10] R. Muschevici, D. Clarke, J. Proenca, "Feature Petri Nets," Workshop on Formal Methods and Analysis in Software Product Line Engineering (FMSPLE), 13–17 Sept. 2010, Jeju Island, South Korea. Volume 2 of SPLC '10, Lancaster University, pp. 99–106, 2010.

[11] A. Philip, R. K. Sharma, "A stochastic reward net approach for reliability analysis of a flexible manufacturing module," Int. J. Syst. Assur. Eng. Manage, pp. 293–302, 2013.

[12] U. Prange, H. Ehrig, K. Hoffmann, J. Padberg, "Transformations in reconfigurable place/transition systems," In: Concurrency, Graphs and Models, Springer, pp. 96–113, 2008.