

# Sistem Software de Simulare Animată a Reţelelor Petri Diferenţiale Reconfigurabile

Iurie ȚURCANU, Emilian GUȚULEAC, Alexei CORDUNENU

Universitatea Tehnică a Moldovei

[Iurie.Turcanu@endava.com](mailto:Iurie.Turcanu@endava.com), [egutuleac@mail.utm.md](mailto:egutuleac@mail.utm.md), [Corduneanu\\_Alexei@yahoo.com](mailto:Corduneanu_Alexei@yahoo.com)

**Abstract** — În lucrare sunt prezentate aspecte de elaborare și utilizare a unui sistem software instrumental, realizat pentru modelarea, verificarea funcțională, simularea animată și evaluarea performanțelor proceselor discret-continue ale sistemelor de calcul, descrise prin rețele Petri diferențiale reconfigurabile (RPDR).

**Index Terms** — modelare, rețele Petri diferențiale, reconfigurabilitate, sistem instrumental, simulare animată.

## I. INTRODUCERE

Studierea performanțelor constituie una din cele mai importante componente ale activităților de concepție, realizare și întreținere ale proceselor sistemelor de calcul paralele/distribuite reconfigurabile [3, 6, 7].

Sistemele de calcul cu arhitecturi avansate au o structură ierarhică cu mai multe nivele reconfigurabile ce se restructurează, adaptându-se la schimbarea cerințelor și a mediului ambiant, modificându-și funcționalitatea și propriile configurații de hardware și/sau software pe parcursul procesării aplicațiilor prin adăugarea și/sau înlăturarea unor componente sau resurse în sistem fără întreruperea procesării aplicației curente [2, 4, 5].

Amploarea efortului de cercetare, investit recent în domeniul modelării și evaluării performanțelor sistemelor cu evenimente discrete și hibride, inclusiv și a sistemelor de calcul, este de largă notorietate, fiind reflectate de numeroasele articole pe această temă, publicate în cele mai prestigioase reviste de specialitate [1, 3, 8].

În [5] au fost definite și studiate o clasă nouă de rețele Petri diferențiale generalizate stocastice descriptiv-restructurabile, care descriu în mod dinamic, pe parcursul funcționării acestora, schimbarea structurii, atributele și stările curente ale modelului. Aceasta permite de a efectua modelarea discret-continuă, verificarea funcțională, simularea animată și evaluarea performanțelor proceselor reconfigurabile ale aplicațiilor sistemelor de calcul.

Pentru a avea posibilitatea de a efectua în mod automat verificarea funcțională, simularea animată și evaluarea indicatorilor de performanță specificate în baza modelelor acestor tip de rețele Petri a fost elaborat un produs software instrumental, numit RDPRS.

În continuare, succint sunt prezentate unele elemente ale rețelele Petri diferențiale reconfigurabile (RPDR) și aspecte de elaborare și utilizare ale RDPNS.

## II. REȚELE PETRI DIFERENȚIALE RECONFIGURABILE

Spre deosebire de rețelele Petri diferențiale generalizate (RPD) [7], rețelele RPDR au și proprietăți de a se restructura la ocurența unui eveniment [5]. Reamintim succint definiția RPDR:  $RN = \langle \Gamma, R, \phi, G_r, G_r, M_0 \rangle$ , unde este o rețea RPG;  $R = \{r_1, \dots, r_k\}$  – o mulțime finită de reguli de reconfigurare;  $\phi: E \rightarrow \{T, R\}$ ,  $E = T \cup R$  –

funcție, care indică tipul evenimentului validat ce poate fi declanșat; cu fiecare regulă  $r$  de rescriere a rețelei sunt asociate funcții de gardă:  $G_r: R \times Bag(P) \rightarrow \{true, false\}$  ce determină validarea lui  $r$ , iar  $G_r: R \times Bag(P) \rightarrow \{true, false\}$  ce implică rescrierea rețelei curente. Implicit,  $\forall r \in R$  în marcajul curent  $M: g_r(M) \in G_r$  este “true”, iar  $g_r(M) \in G_r$  este “false”.

Aceste funcții booleene se calculează la fiecare iterație de marcarea a rețelei, iar în cazul îndeplinirii condiției ele modifică rețeaua inițială. Validarea unei reguli  $r$  în  $RN$  este efectuată cum și pentru RPD când  $g_r(M)$  este “true”. La declanșarea lui  $r$ , în cazul în care  $g_r(M)$  are valoarea “false”,  $r$  va schimba numai marcajul curent  $M$  al  $RN$  într-un marcaj nou  $M'$ . Însă, dacă  $g_r(M)$  are valoarea “true”,  $r$  va reconfigura rețeaua  $RN$  în altă rețea  $RN'$  după regula:  $r: RN_L \triangleright RN_W$ .

Modificarea rețelei  $RN$  are loc în două etape. La prima etapă are loc eliminarea elementelor specificate din subrețeaua  $RN_L$  ale rețelei curente, obținând astfel o nouă rețea  $RN\rho = RN \setminus RN_L$ . Etapa a doua constă din adăugarea sub-rețelei noi  $RN_W$  în rețeaua  $RN\rho$  curente cu elemente și atribute noi ale rețelei, obținând astfel  $RN' = RN\rho \cup RN_W$ . Dacă în rețeaua curentă sunt elemente cu același nume ca la elementele ce urmează a fi adăugate, ele se suprascriu, sau cu alte cuvinte se contopesc. Astfel, ca rezultat se va obține o rețea nouă  $RN'$ , care va funcționa după noi reguli. La o definiție corespunzătoare a reguliilor de reconfigurare este posibilă funcționarea rețelei în regim flip-flop, adică în unele condiții rețeaua se modifică într-un anumit fel, iar apoi poate reveni la configurația inițială.

Starea curentă a rețelei  $RN$  este  $s = (M, v^*(M))$ , unde  $M$  este marcajul curent al rețelei  $RN$ , iar  $v^*(M)$  este balanța dinamică în acest marcaj [5]. Configurația rețelei curente  $RN$  este  $(RN, s)$ .

În Fig. 1 sunt prezentate primitivele unei rețele RPDR.

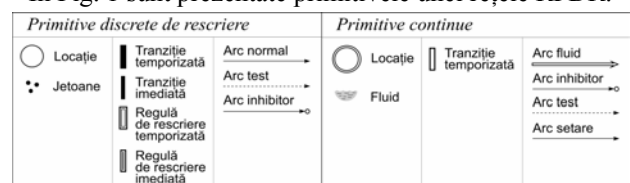


Fig.1. Primitivele rețelei RPDR.

### III. DESCRIEREA SISTEMULUI RDPNS

Sistemul software instrumental RDPNS este o aplicație desktop și dispune de o interfață grafică User-friendly. Acest sistem a fost elaborat pe platforma Microsoft .NET, ceea ce permite realizarea într-un timp relativ scurt a aplicațiilor complexe.

În continuare, sunt descrise formele de bază ale RDPNS, precum și modul de creare a modelelor de rețele RPDR, setarea parametrilor elementelor constitutive ale acestora, lansarea simulării și colectarea datelor statistice.

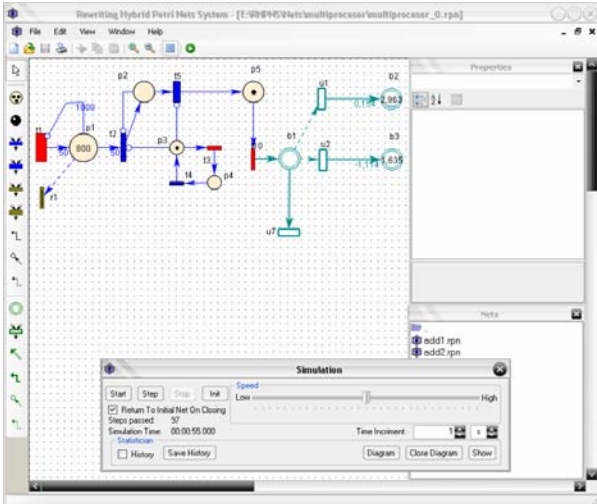


Fig.2. Interfața grafică Utilizator a sistemului RDPNS.

**Interfața Grafică Utilizator (GUI).** Pentru a activa o rețea putem să o creăm sau să deschidem o rețea existentă din colecția de rețele de lucru afișată în fereastra *Nets*. În rezultatul deschiderii unei rețele existente se va deschide o fereastră în care se prezintă rețeaua, cum este arătat în Fig. 2. Această fereastră GUI conține 3 meniuri de bază: meniul principal, meniul comenzilor de bază și meniul instrumentelor de creare/editare a rețelei.

Meniul principal este construit după un șablon clasic și conține:

- *File* – îndeplinirea operațiilor cu fișiere:
  - *New* – crearea unui proiect nou;
  - *Open* – deschiderea unui proiect existent;
  - *Save* – salvarea schimbărilor în proiectul curent (nu este activ dacă nu au fost introduse schimbări în rețea);
  - *Save as* – salvarea proiectului curent sub alt nume;
  - *Export* – exportarea proiectului curent în format grafic (\*.jpg, \*.gif, \*.png).
  - *Print* – imprimarea rețelei curente;
  - *Print Preview* – vizualizarea preventivă;
  - *Exit* – ieșirea din program.
- *Edit* – efectuarea operațiilor de copiere/înscriere asupra elementelor rețelei:
  - *Cut* – șterge elementele selectate înscriindu-le în bufer;
  - *Copy* – copierea în bufer a subrețelei selectate;
  - *Paste* – înscrierea conținutului buferului;
  - *View* – alegerea modului de vizualizare a ferestrelor secundare ale sistemului (Windows) și meniului de instrumente (*Toolbars*):
  - *Toolbars*: Discrete, Continuous, Design;

- *Windows*: Properties, Samples;
- *Show All (Hide All)*:
  - *Places'Labels* – vizualizarea (ascunderea) tuturor denumirilor locațiilor;
  - *Transitions'Labels* – vizualizarea (ascunderea) tuturor denumirilor ale tranzițiilor;
  - *Arcs'Labels* – vizualizarea (ascunderea) tuturor denumirilor ale arcelor;
- *Window* – aranjarea automată a ferestrelor rețelelor deschise;
  - *Cascade* – toate rețelele se vizualizează în cascadă;
  - *Horizontal* – ordonarea rețelelor de la stânga la dreapta, de sus în jos de la rețeaua activă;
  - *Vertical* – ordonarea rețelelor de sus în jos, de la stânga la dreapta.

De asemenea, meniul conține denumirile tuturor rețelelor deschise.

- *Help* – conține forma *About* cu informații despre elaboratorul proiectului.

În figura 3 sunt prezentate butoanele corespunzătoare comenzilor de bază ale sistemului: *New*, *Open*, *Save*, *Print*, *Cut*, *Copy*, *Paste* și operațiile principale:

- *Zoom In*, *Zoom Out* – scalarea imaginii rețelei;
- *Show Grid* – vizualizarea plasei de referință;
- *Run* – declanșarea simulării.



Fig.3. Meniul comenzilor de bază.

Este de menționat că la deplasarea cursorului asupra butonului apare descrierea acestuia.

De la stânga la dreapta (de sus) în meniul dat sunt prezentate instrumentele disponibile (Fig.4) pentru crearea modelelor de rețele:

- *Pointer* – dezactivarea tuturor instrumentelor;
- *Discrete Place* – crearea locației discrete;
- *Token* – adăugarea jetonului într-o locație discretă;
- *Immediate Transition* – crearea tranziției imediate;
- *Time Transition* – crearea tranziției temporizate;
- *Rewriting Immediate Transition* – crearea regulii de reconfigurare imediate;
- *Rewriting Time Transition* – crearea regulii de reconfigurare temporizate;
- *Normal Arc* – crearea arcului discret;
- *Inhibitor Arc* – crearea arcului inhibitor discret;
- *Test Arc* – crearea arcului test discret;
- *Continuous Place* – crearea locației continue;
- *Continuous Transition* – crearea tranziției continue;
- *Continuous Arc* – crearea arcului continuu;
- *Flow Arc* – crearea arcului fluid;
- *Inhibitor Arc* – crearea arcului inhibitor continuu;
- *Test Arc* – crearea arcului test continuu;

Fig.4. Meniul de instrumente.

La deplasarea cursorului asupra instrumentului respectiv apare denumirea acestuia.

Din Fig.1 se observă că suprafața de lucru în mare măsură este ocupată de redactorul rețelei. Funcțiile sale de

bază cuprind crearea și editarea (sub)rețelelor și vizualizarea animată a reconfigurării (sub)rețelei în decursul simulării.

Pentru comoditatea utilizatorului este realizat un meniu de context care se activează la apăsarea *click drept* pe forma de redactare a rețelei. Meniul dat conține la fel unele din comenzi de bază ale aplicației, dar și proprietățile specifice ale unor obiecte.

Primele trei operații efectuează aceleași operații ca și în meniul principal. Comanda *Delete* șterge obiectul / obiectele selectate. Comanda *Size* permite setarea dimensiunilor potrivite pentru obiectele selectate, aceasta fiind proprie doar locațiilor și tranzițiilor. Comanda *Rotate* este proprie doar tranzițiilor și permite rotirea acestora în jurul centrului său cu un unghi de 45°.

**Formele secundare ale aplicației.** La formele secundare ale aplicației se referă în primul rând forma de editare a proprietăților obiectelor din rețea (*Properties*) și forma de vizualizare a rețelelor de lucru. Aceste ferestre se vizualizează la *activarea* aplicației. La fel, există și un set de forme care nu se lansează cu lansarea aplicației. În continuare vom analiza ambele tipuri de ferestre secundare.

**Editorul proprietăților obiectelor.** În Fig. 5 este prezentată forma de editare a proprietăților și atributele obiectelor constituente ale rețelei. Implicit, ea se poziționează în colțul dreapta - sus al formei de bază, dar poate fi repositionată, sau închisă. Această formă este compusă din următoarele 4 componente:

- lista obiectelor existente în rețea;
- butoanele de gestiune ale modului de vizualizare a proprietăților obiectului selectat (cu grupare pe categorii sau fără);
- ghidul de vizualizare a proprietăților elementului selectat;
- câmpul de descriere a proprietății selectate.

La selectarea obiectului din editorul rețelei, el automat se selectează în fereastra *Properties*.

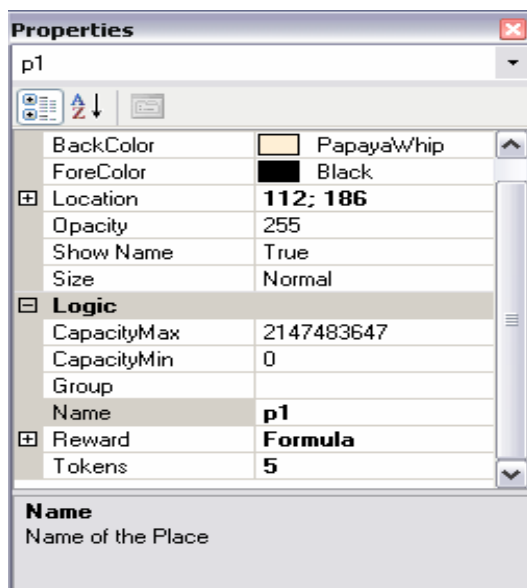


Fig 5. Forma *Properties*.

Toate proprietățile se împart în două clase: *Logic* și *Design*. La selecția unui grup de obiecte vor fi vizualizate doar proprietățile comune ale acestora, în așa fel schimbarea atributelor acestei proprietăți va duce la

schimbarea proprietăților tuturor elementelor selectați.

**Directorul de lucru.** O altă fereastră secundară este forma *Nets*. În ea sunt prezentate toate rețelele din directorul de lucru a proiectelor (Fig. 6). Implicit, această formă este vizualizată în colțul dreapta - jos, dar poate fi repositionată sau închisă.

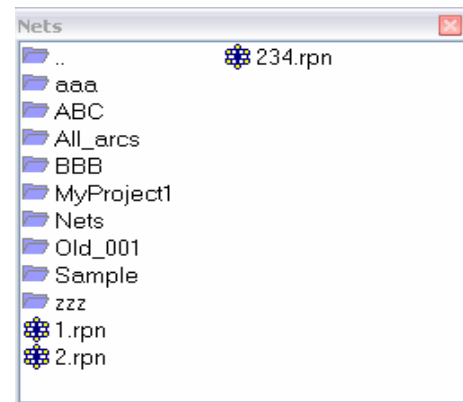


Fig.6. Lista proiectelor din directorul de lucru.

În forma dată se vizualizează doar directoriile și fișierele de lucru cu extensia *\*.rpn*. Directoriile prezintă proiecte separate. Directoriul cu denumirea „..” servește pentru navigare și îndeplinește funcția de deplasare cu un nivel mai sus. Odată cu alegerea unei directorii de proiect în forma dată se încarcă conținutul acestei directorii. Dacă ea conține o rețea de bază pentru proiectul dat (fișier cu denumirea directoriei urmat de „\_0”) se încarcă editorul rețelei.

**Editorul de formule.** Pentru a edita rețele cu automodificare a fost introdusă posibilitatea de a reda unele proprietăți cantitative ale atributelor rețelei ce sunt marcate dependente. Pentru a seta aceste proprietăți se utilizează un editor de formule ce este prezentat în Fig. 7.

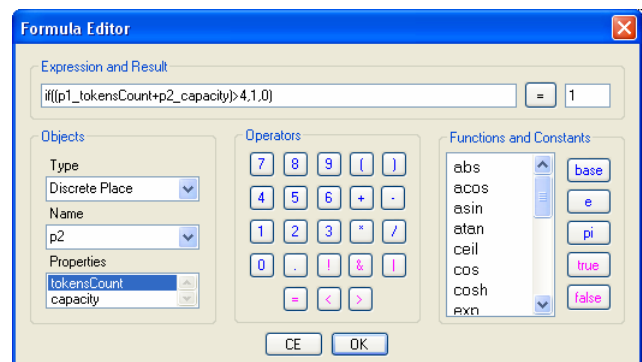


Fig.7. Editor de formule.

Editorul de formule se lansează doar pentru proprietățile cantitative ale atributului, care se setează accesând butonul respectiv în fereastra *Properties* sau pentru formulele necesare la crearea diagramelor de simulare. În partea superioară a acestei forme este situată grupa de componente *Expressions and Result*, care conține textbox-ul de redactare a formulei și textbox-ul rezultatului. După ce formula a fost introdusă putem determina corectitudinea ei și totodată calcularea valorii acesteia tastând butonul „=”. În cazul erorii sintactice în formulă – se afișează mesajul eroarei respective.

Partea inferioară a formei este divizată în trei părți:

➤ *Objects* – permite alegerea proprietăţii necesare. Pentru aceasta se alege tipul obiectului din lista *Type*. Pentru fiecare tip selectat se reînnoieşte conţinutul listei *Name*, care conţine denumirile obiectelor de acest tip. După ce a fost selectat un obiect în lista *Properties*, apar proprietăţile acestuia. Efectuând dublu click asupra proprietăţii, ea se adaugă în formulă. Sintaxa proprietăţii fiind:  $\langle \text{nume obiect} \rangle_{\langle \text{nume proprietate} \rangle}$ . Proprietăţile obiectelor pot fi, de asemenea, introduse de utilizator păstrând sintaxa acestora.

➤ *Operators* – calculator permite introducerea cifrelor, parantezelor, punctului zecimal etc, utilizând mouse-ul. De asemenea, cifrele pot fi introduse şi de pe tastatură.

➤ *Functions and Constants* – funcţiile şi constante menţinute de editorul de formule respectiv. La poziţionarea cursorului asupra oricărei funcţii apare descrierea acesteia, inclusiv sintaxa ei. Utilizatorul poate să introducă funcţiile şi de la tastatură. Este de remarcat faptul că editorul de formule conţine setul întreg de operaţii şi funcţii matematice, inclusiv determinarea *maxim*-ului şi *minim*-ului din lista de dimensiuni variabile. La fel editorul de formule oferă posibilitatea de a utiliza şi operatorul condiţional *if*.

Butonul „OK” situat în partea inferioară a acestei forme serveşte pentru a confirma crearea formulei. La tastarea acestui buton are loc verificarea sintactică a formulei şi calcularea valorii expresiei, salvarea expresiei şi închiderea forme de editare a formulelor. Butonul „CE” sau închiderea forme duce la anularea formulei şi închiderea forme.

**Forma de dirijare a simulării.** O altă fereastră secundară a aplicaţiei este forma „Simulation”, prezentată în Fig. 8. Această formă se lansează la alegerea butonului „Run” al meniului principal.

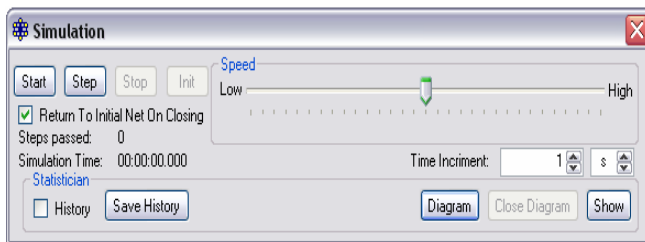


Fig. 8. Forma de dirijare a simulării.

În partea superioară a forme se află următorul grup de butoane:

- *Start* – startează procesul de simulare. Dacă o simulare anterioară a fost oprită butonul Start o va prelunge.
- *Step* – efectuează un singur pas de simulare.
- *Stop* – opreşte procesul de simulare.
- *Init* – reinițializează rețeaua dacă ea a efectuat cel puțin un pas de simulare.

Mai jos se află checkbox-ul «Return To Initial Net On Closing», validarea lui duce la reinițializarea rețelei în cazul închiderii forme „Simulation”. În partea de sus dreapta se află regulatorul vitezei procesului de simulare – *Speed*. Imediat, sub acest regulator, se află regiunea de setare a quantei de timp / pas. *Time Increment* – arată numărul de unități de timp stabilite (*ms*-milisecunde, *s*-secunde, *min*-minute sau *h*-ore) ce decurg la un pas de simulare.

În partea de jos a forme este situat grupul de butoane *Statistic*. Funcția grupului constă în colectarea datelor statistice. Primul din acestea este indicatorul „History”, care permite de a păstra istoria simulării, dacă este nevoie. Butonul „Save History” permite salvarea istoriei simulării în format textual, chemând dialogul de salvare a fișierului, iar butonul „Diagram” servește pentru crearea diagramelor de simulare prin intermediul formulelor create în forma „Formula Editor”. Butonul „Close Diagram” servește pentru închiderea diagramelor curente, iar butonul „Show” pentru vizualizarea / ascunderea diagramelor.

Un exemplu de diagramă a rezultatelor de simulare este prezentat în figura 9.

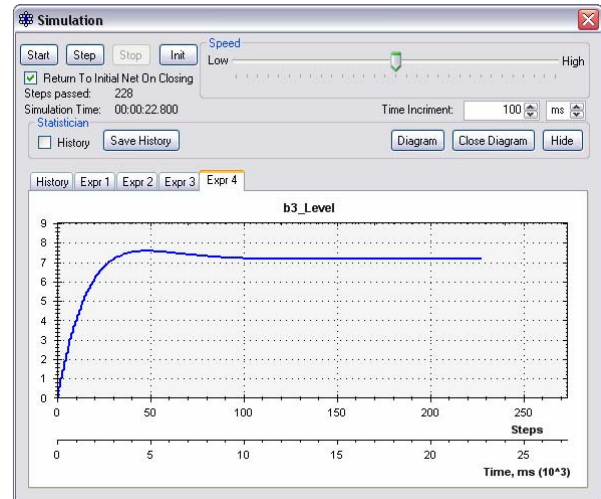


Fig. 9. Diagrama simulării unei rețele.

Din Fig. 10 se observă că istoria simulării este prezentată sub forma unui tabel în care sunt stocate așa date ca: pasul simulării; timpul simulării; marcajul curent; tranzițiile validate și cele declanșate atât cele imediate, cât și cele temporizate.

Nr	Time	Marking	Enabled Timed T	Enabled Immediate T	Fired Usual T	Fired
1	00:00:00.000	5p1, p5	r1, u1		r1, u1	
2	00:00:01.000	4p1, p2, p5, 2.75b1	t1, r1, u1, u2		t1, r1, u1, u2	
3	00:00:02.000	3p1, p2, p3, 2.75b1, b2	t2, r1, u1		t2, u1	r1
4	00:00:03.000	2p1, 2p2, p4, 5.5b1, b2	t3, r1, u1, u3		t3, r1, u1, u3	
5	00:00:04.000	2p1, 3p2, p5, 8.25b1	t1, r1, u1, u2		t1, r1, u1, u2	
6	00:00:05.000	p1, 3p2, p3, 8.2b1, b2	t2, r1, u1		t2, r1, u1	
7	00:00:06.000	4p2, p4, 10.95b1, b2	t3, u3		t3, u3	
8	00:00:07.000	p1, 4p2, p5, 10.95b1	t1, r1, u1, u2		t1, r1, u1, u2	
9	00:00:08.000	4p2, p3, 10.9b1, b2	t2		t2	
10	00:00:09.000	4p2, p4, 10.9b1, b2	t3, u3		t3, u3	

Fig.10. Istoria simulării.

De menționat faptul că în timpul simulării este imposibil de redactat modelul rețelei. Modificările acestuia pot fi programate prin proprietățile atributelor elementelor rețelei.

Pentru a salva o eventuală etapă a evoluției rețelei este nevoie de a dezactiva „Return To Initial Net On Closing”.

**Crearea, editarea și simularea rețelelor.** Crearea unei rețele constă în plasarea unui set de elemente pe planșeta forme respective, interconectarea lor și setarea proprietăților lor. Trebuie de ținut cont că două locații nu pot fi interconectate printr-un arc, precum și două tranziții, dacă va fi efectuată încercarea de a efectua o așa interconectare editorul rețelei va afișa mesajul de eroare.

La adăugarea unui element nou, lui i se atribuie un nume după șablonul predefinit. Într-o rețea toate elementele sunt unice deci nu pot avea același nume.

Fiecare din elementele rețelei mai conține și un set de proprietăți care pot fi setate de utilizator.

Pentru o regulă de rescriere subrețeaua nouă și cea veche sunt niște referințe la alte rețele (fișiere). La introducerea denumirii subrețelei se verifică dacă în directoriul de lucru nu există o rețea cu același nume. Dacă ea există atunci ea se deschide pentru redactare, în caz contrar se va crea o rețea nouă cu numele specificat și ea este deschisă pentru creare și redactare.

Fie că avem de construit rețeaua *RPDR1* reconfigurabilă prezentată în Fig. 11.

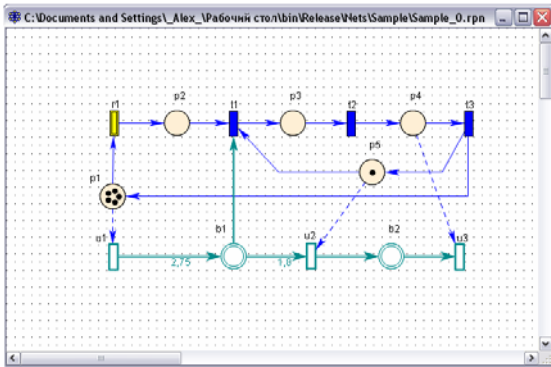


Fig.11. Rețea reconfigurabilă *RPDR1*.

Pentru a construi rețeaua *RPDR1* pe fereastra principală a GUI creăm nodurile respective și le interconectăm prin arce cu atributele lor specificate. Apoi adăugăm jetoanele în locațiile marcajului inițial. Pentru ca rețeaua să fie reconfigurabilă, ea trebuie să conțină regula de rescriere „r1” cu funcția de gardă respectivă. În afară de funcția de gardă avem nevoie de setat cel puțin una din subrețele vechi sau nouă (în cazul setării subrețelei vechi la rescriere rețeaua va degrada, iar în cazul când este setată subrețeaua nouă – va evolua).

Pentru comoditate, ca subrețea veche vom crea o copie a rețelei originale. Iar pentru cea nouă – rețeaua în care trebuie să evolueze rețeaua de bază și anume rețeaua *RPDR2* din Fig. 12, astfel încât  $r1: RPDR1 \triangleright RPDR2$ . În rețeaua dată setăm toate proprietățile după expresia descriptivă dată, iar regula de rescriere o setăm în așa fel ca rețeaua să revină la starea inițială, adică  $r1: RPDR2 \triangleright RPDR1$ . Astfel vom crea un model al procesului reconfigurable ciclic.

Tranzițiile validate, la simularea rețelei, vor fi colorate în roșu, iar cele declanșate în verde. Procesul de declanșare a unei tranziții validate este însoțit de deplasarea jetoanelor pe arce discrete sau scurgerea fluidului prin arce continue.

Proprietatea „*Rewriting Guard*” a regulilor de rescriere poate fi setată prin formulă, astfel încât rescrierea rețelei va deveni dependentă de funcția de gardă de rescriere dată.

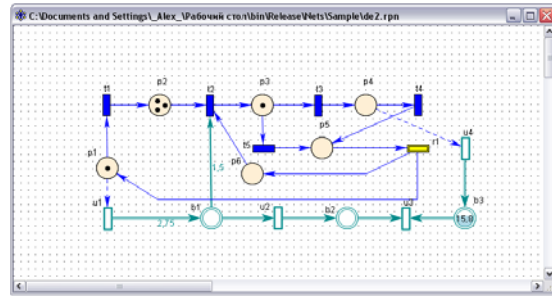


Fig.12. Rețeaua reconfigurabilă *RPDR2*.

#### IV. CONCLUZIE

Rețelele Petri diferențiale reconfigurable sunt un instrument adecvat la modelarea funcționării proceselor discret-continue ale sistemelor de calcul, informaționale și a proceselor de producție ale acestora.

În lucrare sunt prezentate unele aspecte de elaborare și utilizare a unui sistem software instrumental, *RDPDRS*, realizat pentru modelarea, verificarea funcțională, simularea animată și evaluarea performanțelor proceselor discret-continue de calcul, descrise prin rețele Petri diferențiale reconfigurable.

#### BIBLIOGRAFIE

- [1] V. Sudacevski, V. Ababii, V. Negura, “A Hardware Implementation Of Safe Petri Net Models,” *Advances in Electrical and Computer Engineering*, no. 1, Ed.: “Stefan cel Mare” University of Suceava, pp. 54-58, 2006.
- [2] K. Compton, S Hauck, “Reconfigurable Computing: a Survey of Systems and Software,” *ACM Computing Surveys (CSUR)*, vol. 34, no. 2, pp. 171-210, 2002.
- [3] M. Calzarossa, R. Marie, “Tools for Performance Evaluation,” *Perf. Evaluation*, no. 33, pp.1-3, 1998.
- [4] S. Cook, R. Harrison, P Wernik, “A simulation model of self-organising evolvability in software systems,” *Proc. of the 1-st IEEE International Workshop on Software Evolvability, Hungary*, pp. 17-22, 2005.
- [5] E. Guțuleac, “Dynamic rewriting generalized differential Petri nets for discrete-continuous modeling of computer systems,” *Meridian ingineresc*, no. 3, Ed.: UTM, Chişinău, pp.27-32, 2006.
- [6] 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, Automatica și Calculatoare*, Iași, România, pp. 121- 136, 2005.
- [7] E. Guțuleac, *Evaluarea performanțelor sistemelor de calcul prin rețele Petri stochastice*. Ed.: „Tehnica-Info”, Chişinău, 2004, - 276 p.
- [8] M. Llorens, J. Oliver, “Structural and Dynamic Changes in Concurrent Systems: Reconfigurable Nets,” *IEEE Transactions on Computers*, vol. 53, no. 9, pp. 1147-1158, September 2004.