

# PNStudio -Toolkit de Simulare Animată a Rețelelor Petri Hibrade Reconfigurabile

Iurie ȚURCANU, Artiom CIUMAC, Emilian GUȚULEAC

Universitatea Tehnică a Moldovei

Iurie.Turcanu@endava.com, artiom.ciumac@gmail.com, egutuleac@mail.utm.md

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

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

## I. INTRODUCERE

Concepția unui sistem de calcul bazat pe paralelism și coperarea componentelor de la specificația sa inițială până la verificarea faptului că dezvoltarea propusă *satisface* cerințelor de performanță, necesită un mediu de dezvoltare. În acest context, este de dorit ca într-un astfel de mediu să fie folosit *un singur formalism*, care să aibă capacități suficiente pentru descrierea acțiunilor proceselor cooperante, protocoalelor de comunicație, reconfigurarea dinamică în construirea și validarea modelului sistemului analizat [1, 3, 8, 9].

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, 7].

Rețelele Petri (RP) și extensiile lor pot fi considerate ca modele de stare-tranziție, care permit de efectua verificarea și evaluarea performanțelor proceselor componente ale unui sistem cu evenimente discrete [5-10].

În [5, 7, 8] au fost definite și studiate o clasă nouă de rețele Petri hibrade generalizate stocastice descriptiv-restructurabile, care descriu în mod dinamic, pe parcursul funcționării acestora, schimbarea structurii, atributele și stările curente ale modelului.

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 PNStudio -Toolkit.

În continuare, succint sunt prezentate unele elemente ale rețelelor Petri hibrade reconfigurabile (RPHR) și aspecte de elaborare și utilizare ale PNStudio -Toolkit.

## II. REȚELE PETRI HIBRIDE RECONFIGURABILE

O rețea Petri hibridă temporizată (RPHT) este tripletul:  $HT = \langle NG, \psi, V \rangle$  în care  $NG$  este o structură de rețea Petri generalizată temporizată (RPGT) [7], unde mulțimea finită a locațiilor  $P$  și tranzițiilor temporizate  $T$  constituie partițiile:  $P = P_d \cup P_c$ ,  $P_d \cap P_c = \emptyset$  astfel, încât

$P_d = \{p_1, \dots, p_{n_d}\}$ ,  $n_d = |P_d|$  este mulțimea locațiilor discrete, iar  $P_c = \{b_1, \dots, b_{n_c}\}$ ,  $n_c = |P_c|$  este mulțimea locațiilor continue (buffere). Locațiile discrete pot conține un număr întreg de jetoane, iar locațiile continue pot conține o cantitate (număr real) de fluid. Grafic, locațiile discrete sunt reprezentate prin ceruțele, iar cele continue, prin ceruțele concentrice duble;  $T = T_d \cup T_c$ ,  $T_d \cap T_c = \emptyset$ , astfel încât  $T_d = \{t_1, \dots, t_{k_d}\}$ ,  $k_d = |T_d|$  este mulțimea tranzițiilor discrete, iar  $T_c = \{u_1, \dots, u_{k_c}\}$ ,  $k_c = |T_c|$  este mulțimea tranzițiilor continue. Grafic, tranzițiile continue sunt reprezentate prin dreptunghiuri.  $\psi : P \cup T \rightarrow \{d, c\}$  este funcția hibridă care indică tipul fiecărui nod al rețelei, adică tip discret  $d$  sau tip continuu  $c$ .  $Pre, Test$  și  $Inh : P \times T \times Bag(P) \rightarrow Bag(P)$ , respectiv sunt funcția de incidență înainte ale arcelor normale, test și inhibiție.  $Bag(P)$  este multisetul mulțimii  $P$  (discrete sau continue). Funcția de incidență înapoi ale arcelor normale este  $Post : T \times P \times Bag(P) \rightarrow Bag(P)$ . Aceste funcții descriu cardinalitatea marcaj-dependentă a mulțimii arcelor  $A$  care este partiționată în submulțimile de arce discrete (continue) normale, inhibitoare și test.

Funcția de capacitate a locațiilor discrete (continue) este definită în mod similar ca și pentru rețele RPGT, însă pentru locațiile continue ea primește valori reale.  $V : T_c \times Bag(P) \rightarrow IR$  este funcția ce descrie viteza de declanșare a tranzițiilor continue validate. Această funcție determină rata fluidului transmis prin arcele continue ce leagă tranzițiile continue cu locațiile continue respective ale rețelei. Implicit, ea este egală cu 1.

În Fig.1 sunt prezentate opțiunile posibile de conectare a arcelor rețelei RPHT cu tranzițiile discrete și continue.

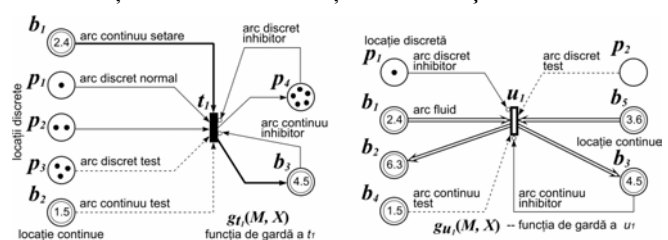


Fig. 1. Opțiunile posibile de conectare a arcelor rețelei RPHT cu tranzițiile discrete și continue.

Regulile de validare și declanșare ale tranzițiilor unei RPHT marcate sunt definite în mod similar cum și pentru RPHG, luând în considerație influența atât a părții discrete, cât și celei continue, respectând opțiunile posibile de conectare a arcelor cu tranzițiile discrete și continue (vezi Fig. 1). O tranziție continuă validată de marcajul curent  $u_j \in T_c(M)$  va declanșa în mod continuu, dacă nu există o altă tranziție cu o prioritate superioară ei,  $Pri(u_j) > Pri(u_k)$ , pentru care sunt verificate precondițiile sale de validare [8,9].

Spre deosebire de rețelele RPHT, rețelele RPHR au și proprietăți de a se restructura la ocurența unui eveniment [5]. Reamintim succint definirea RPHR:

$RN = \langle HG, R, \phi, G_r, G_r, M_0 \rangle$ , HG 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 marcare a rețelei, iar în cazul îndeplinirii condiției ele modifică rețeaua inițială. Validarea unei reguli  $r$  în RN este efectuată cum și pentru RPHT 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:  $r: RN_L \triangleright RN_W$ .

În Fig.2 sunt prezentate opțiunile posibile de conectare a arcelor RPHR cu regulile de rescriere.

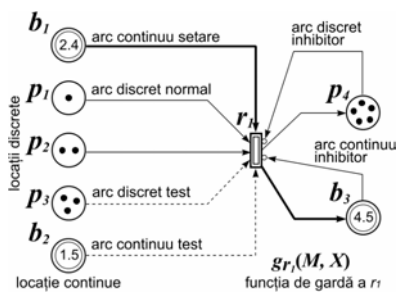


Fig. 2. Opțiunile posibile de conectare a arcelor RPHR cu regulile de rescriere.

Modificarea rețelei RN are loc în două etape. La prima etapă are loc eliminarea elementelor specificate de subrețeaua  $RN_L$  ale rețelei curente, obținând o nouă rețea  $RN\rho = RN \setminus RN_L$ . Etapa a doua constă din adăugarea subrețelei  $RN_W$  în rețeaua  $RN\rho$  curente cu elemente 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 funcțiilor 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 configurarea 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. Configurația rețelei curente RN, este  $(RN, s)$ .

### III. DESCRIEREA SISTEMULUI PNSTUDIO

Sistemul software instrumental PNStudio este un Toolkit cu o aplicație desktop și dispune de o interfață grafică User-friendly. Acest sistem a fost elaborat pe platforma Microsoft .NET Framework 3.5 (Fig.3), ceea ce permite realizarea într-un timp relativ scurt a aplicațiilor complexe.

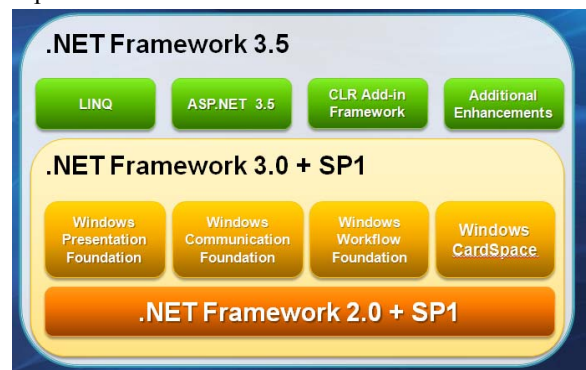


Fig.3. Platforma Microsoft .NET Framework 3.5. Tehnologii utilizate pentru realizarea PNStudio:

- Limbaj – C#;
- IDE – Visual Studio 2008
- Multithreading - performanță;
- LINQ – manipulare date;
- Windows Forms – l'IHM;
- XML Serialization – stocare date;
- ZedGraph – vizualizare date;
- GDI+ - vizualizare și animarea funcționări

Structura sistemului. Modulele sistemului sunt prezentate în diagrama din Fig. 3.

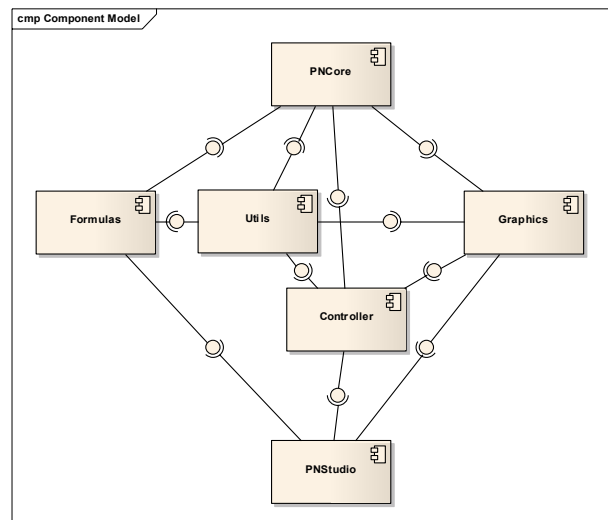


Fig.3. Structura sistemului PNStudio.

Această structură dă posibilitate de a schimba sau extinde funcționalitățile sistemului, fiind constituit din mai multe module. Modulele au fost proiectate astfel, încât să fie cât mai puțin dependente unul de altul, ceea ce permite, de exemplu, de a reprezenta o rețea Petri, atât sub forma clasică 2D, cât și sub forma 3D sau de a elabora un instrument fără interfață grafică pentru o analiză rapidă a modelelor salvate. Modulele aplicației sunt descrise în paragraful ce urmează.

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

Descrierea modulelor urmează conform ordinii de dependență a unuia față de altul.

1. *Utils* – acest modul conține funcționalități utilitare, care sînt utilizate în toate celelalte, exemplu logarea evenimentelor și a erorilor, algoritmi matematici, interacțiunea la nivel jos cu sistemul de operare, etc.
2. *Formulas* – aici sînt realizate funcționalitățile de compilare și calcul dinamic a formulelor matematice. Se utilizează pentru a reprezenta majoritatea proprietăților elementelor unei rețele Petri. De exemplu ponderea unui arc poate fi o formulă dependentă de numărul de jetoate dintr-o locație oarecare.
3. *PNCore* – aici sînt realizate toate funcționalitățile legate de rețele Petri (algoritmi de simulare, serializare în XML, reprezentare logică s.a.m.d.). Acest modul este complet independent de reprezentarea grafică.
4. *Graphics* – modulul conține totul ce e legat de reprezentarea grafică a unei rețele Petri. Deasemenea aici se conțin algoritmi de animare și manipulare a figurilor grafice. Grafica e realizată complet, utilizînd doar sistemul GDI+ din sistemul de operare Windows, ceea ce face această bibliotecă compatibilă practic cu orice hardware ce satisface cerințele minime ale .NET Framework.
5. *Controller* – la proiectarea programului a fost utilizat patternul MVC, și anume în acest modul e realizată grafica controlorului – acest obiect gestionează interacțiunea dintre utilizator și modelul de date, controlează animația și salvarea sau citirea datelor din memoria permanentă.
6. *PNStudio* – acest modul conține interfața grafică utilizator (GUI) a sistemului și modulele ce realizează algoritmi de transmitere a evenimentelor produse de utilizator către controlor.

**Interfața Grafică Utilisator.** Pentru a deschide 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 principală în care se prezintă rețeaua, cum este arătat în Fig. 4. Din această figură se observă că fereastra GUI conține 3 meniuri de bază: meniul principal, meniul comenzilor de bază și meniul instrumentelor de creare/editare a rețelei.

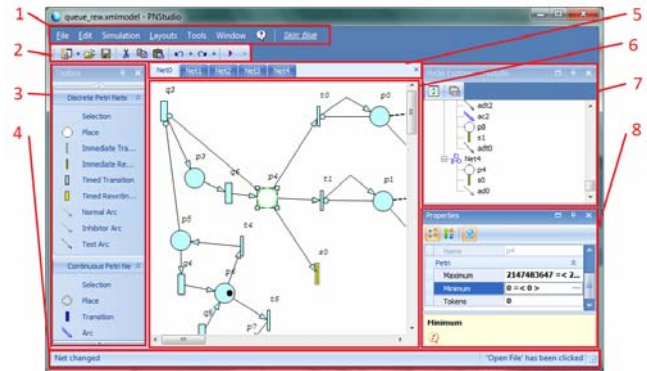


Fig.4. Interfața grafică Utilizator a sistemului PNStudio.

Funcționalitatea elementelor GUI:

1. *Bara principală de meniuri* – de aici pot fi accesate toate funcționalitățile de încărcare sau salvare în fișiere. Descrierea meniurilor urmează în continuare.
2. *Bara de instrumente* – aici sunt plasate cele mai des utilizate funcționalități din bara principală de meniuri.
3. *Panelul de elemente* – aici sunt plasate toate elementele de editare a unei rețele Petri. Pentru a selecta un element trebuie de apăsat pe denumirea lui (și nu pe iconiță).
4. *Bara de stare* – prezintă diferite mesaje de stare a sistemului.
5. *Paginile rețelelor* – de aici pot fi accesate toate rețelele deschise curent din modelul dat.
6. *Suprafața de editare* – în zona aceasta se reprezintă rețeaua Petri activă la moment. Dacă apăsam cu click drept pe această suprafață – apare meniul de acces rapid, ce este descris în continuare în acest capitol.
7. *Arborele logic al modelului* – aici este reprezentată structura modelului sub forma unui arbore. Pentru a activa o rețea (daca ea nu este activă sau fereastra ei a fost închisă) trebuie de apăsat de doua ori (double-click) pe numele rețelei. Pentru a selecta un obiect al rețelei trebuie de apăsat de doua ori pe numele obiectului – în acest caz va fi activată rețeaua ce conține acest obiect, obiectul va fi marcat ca selectat și vor fi deschise proprietățile lui.
8. *Panelul de proprietăți* – aici sînt arătate toate proprietățile obiectului selectat la moment. Dacă sunt selectate mai multe obiecte – se va afișa proprietățile lor comune.

Modulul grafic (*Graphics*), de asemenea, a fost elaborat pentru un echilibru optim dintre performanță și consum de resurse și reprezintă cel mai mare consumator modul din punct de vedere al utilizării resurselor calculatorului. Din funcționalități grafice putem menționa deplasarea și redimensionarea arbitrară a unui singur obiect sau unui grup de obiecte, ajustarea numărului de jetoane vizibile în dependență de dimensiunea locației, ajustarea automată a punctului de conectare a arcelor la obiecte (arcul niciodată nu se intersectează cu obiectul, ci se conectează automat la



marginea obiectului, indiferent de poziția lor relativă). Pentru configurații complexe de rețele, la arce pot fi adăugate sau șterse arbitrare puncte de control – astfel arcele pot „înconjura” obiectele din calea sa.

Procesul de editare a rețelei a fost simplificat la maximum posibil.

**Crearea și editarea rețelei.** 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 de un arc precum și două tranziții, dacă va fi efectuată încercarea de a efectua o așa interconectare editorul rețelei va afișa mesajul de eroare respectiv.

În Fig. 5 este prezentat algoritmul de creare și editare a rețelei în GUI.

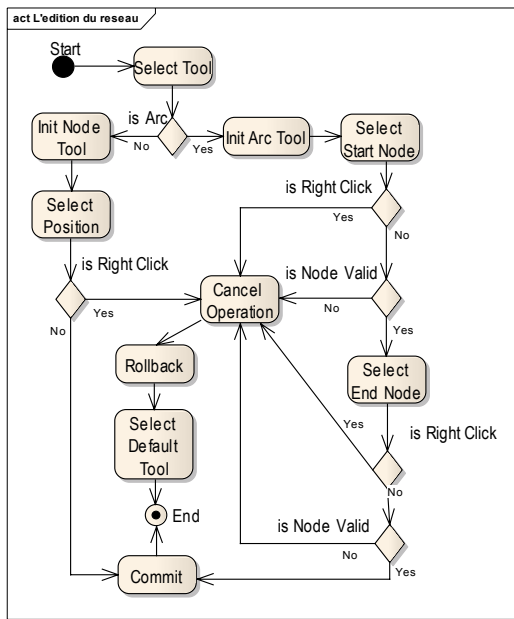


Fig. 5. Algoritmul de editare a rețelei în GUI.

La adăugarea unui element nou lui i se atribuie un nume după șablonul predefinit. Într-o rețea toate elementele sunt unice, deci unele din ele 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.

Pentru a adăuga un nod (locație sau tranziție) trebuie de selectat tipul dorit de pe panelul de obiecte al GUI (apăsând pe *Denumirea* nodului) (1) apoi, după poziționarea nodului pe suprafața rețelei, de-l fixa apăsând încă o dată suprafața (2), așa cum este arătat în Fig. 6a.

Pentru a adăuga un arc – trebuie de selectat tipul de arc dorit de pe panelul de obiecte (apăsând *Denumirea* obiectului) (1) apoi de apăsare pe nodul sursă (2) și pe nodul destinație (3). Dacă a fost selectat un nod incompatibil cu tipul curent al arcului – instrumentul curent se resetează la selecție („Selection” de pe panelul de obiecte), așa cum este arătat în Fig. 6b.

Orice obiect din rețea are o denumire care constă din prefix și indice. Prefixul se setează automat în dependență de tipul obiectului (de exemplu locațiile continue au prefixul „p”). Indicele, de asemenea, se atribuie automat, dar poate fi schimbat. Numerotarea obiectelor este unică pentru fiecare tip (două obiecte de același tip nu pot avea același indice), se începe de la 0 (zero) și programul utilizează întotdeauna cel mai mic indice disponibil. De exemplu, dacă am creat obiectele cu indicii 1, 2, 3 și apoi am șters obiectul cu indicele 2 – următorul obiect creat va utiliza acest indice eliberat – 2.

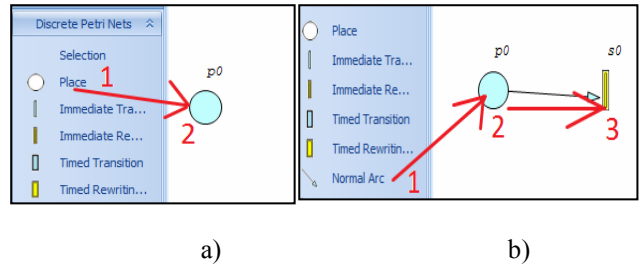


Fig. 6. Adăugarea unui: (a) nod; (b) arc.

Panelul de proprietăți permite accesul la toate proprietățile logice ale obiectelor selectate. Unele proprietăți nu pot fi schimbate – ele se colorează în sur.

Proprietățile reprezentate de formule pot fi editate în felul următor – în panelul de proprietăți (Fig. 7) trebuie de selectat proprietatea dorită și apăsare pe butonul cu trei puncte, cum este indicat în imaginea următoare. Până la semnul „=” este indicată valoarea formulei, după aceea, în paranteze unghiulare, – însăși textul formulei. Editorul de formule va fi descris în continuare.

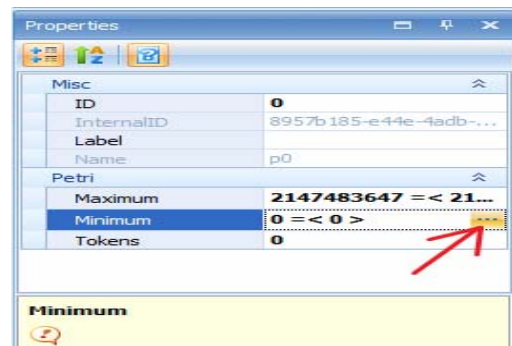


Fig. 7. Editarea formulei.

Editorul de formule permite elaborarea simplă și eficientă a formulelor matematice, practic de orice complexitate, pentru redarea atributelor cantitative marcat-dependente ale rețelelor RPHR. Elementele editorului sunt prezentate în Fig.8.

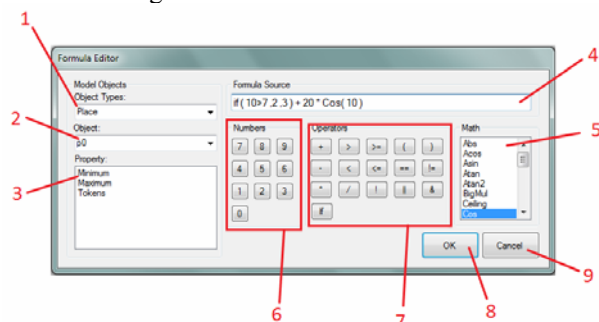


Fig. 8. Editorul de formule.

Elementele editorului de formule sunt:

- 1) Lista tipurilor obiectelor prezente în model. Se arată tipurile din toate reţelele.
- 2) Lista obiectelor de tipul selectat la moment. Se arată obiectele din toate reţelele.
- 3) Lista de proprietăţi a obiectului selectat la moment. Click dublu pe o proprietate - o inserează în textul formulei (4).
- 4) Aici se reprezintă textul formulei. El poate fi redactat utilizând controlul din această formă, dar pentru o redactare mai rapidă se recomandă scrierea manuală.
- 5) Lista de funcţii matematice ce pot fi utilizate în formulă. Separatorul argumentelor funcţiei (dacă sunt mai multe de unu) este „, ” (virgulă).
- 6) Panelul de numere. Aici separatorul părţii zecimale în numerele fracţionare este „.” (punct).
- 7) Panelul de operatori. Aici sunt indicaţi operatorii implementaţi la moment. La apăsarea pe vre-un buton – în textul formulei se inserează operatorul selectat.
- 8) La apăsarea acestui buton are loc compilarea formulei. Dacă compilarea a reuşit cu succes – formula este creată şi transmisă în structura grafică a reţelei sau la proprietatea obiectului. Dacă nu – este afişat un mesaj de interpretare a acestei erori. În cazul în care utilizatorul apasă butonul „Yes” – din nou se deschide editorul de formule pentru a o corecta, dacă se apasă butonul „No” – acţiunea este anulată.
- 9) Acest buton anulează editarea formulei. În acest caz se pierd toate datele schimbate în timpul editării.

În cazul în care formula nu se calculează şi rezultatul calculului este zero, aceasta se întâmplă când formula este scrisă corect, dar valoarea ei nu poate fi calculată la moment, de exemplu, pentru formula „ $p7.Tokens + 10$ ”, dacă locaţia  $p7$  nu este prezentă în reţeaua curentă, rezultatul va fi zero şi el va fi marcat cu „!” în panelul de proprietăţi, de exemplu „ $0 (!) \leq p7.Tokens + 10 >$ ”.

În formule se poate de utilizat şi operatorul „if” – acest operator se utilizează ca o funcţie de 3 parametri. Primul parametru este condiţia, al doilea – rezultatul, dacă condiţia se evaluează ca *true* (sau un număr diferit de zero) şi al treilea parametru este rezultatul, dacă condiţia se evaluează ca *false* (sau zero). De asemenea, în formulă este posibil de scris „ $(1 < 2) * 5$ ”, ceea ce se evaluează spre valoarea 5. Aceasta se datorează faptului că valorile booleene se convertesc automat în numere şi invers conform următoarelor reguli: *true* spre număr se converteşte ca 1 (unu), *false* – spre 0 (zero); orice număr diferit de zero se converteşte ca *true*, iar zero se converteşte ca *false*.

**Forma de dirijare a simulării.** O altă fereastră secundară a aplicaţiei este forma „Simulation”, prezentată în Fig. 9. Simularea se lansează la activarea butonului „Start” al meniului.

Elementele şi funcţionalităţile butoanelor ferestrei de simulare sunt următoarele:

- 1) Butonul „Start” porneşte simularea. Dacă simularea este deja pornită – acest buton este dezactivat.

- 2) Butonul „Stop” opreşte simularea. El este dezactivat dacă simularea nu este pornită.
- 3) Butonul „Step” efectuează un pas de simulare.
- 4) Butonul „Init” resetează reţeaua în starea sa iniţială.

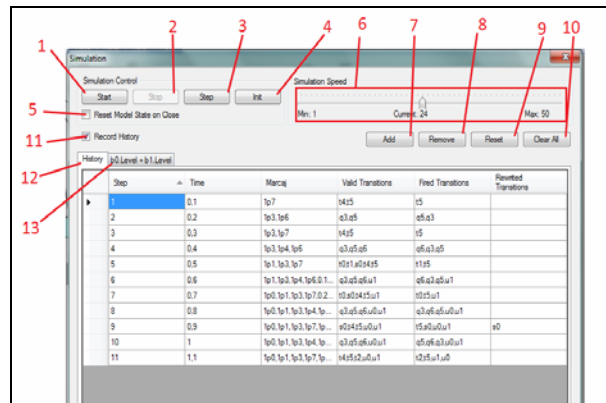


Fig. 9. Forma de dirijare a simulării.

- 5) Dacă acest control este marcat – la închiderea ferestrei reţeaua va fi resetată.
- 6) Aici se indică viteza de simulare.
- 7) Butonul „Add” adaugă o nouă expresie aritmetică pentru obţinerea grafice. În acest caz se deschide editorul de formule, descris mai înainte. Odată cu această acţiune fereastra trece în starea din Fig. 8.
- 8) Şterge graficul curent. Dacă la moment este activă pagina „History” – apăsarea acestui buton este ignorată. Dacă controlul „11” nu este marcat şi a fost şters ultimul grafic – fereastra trece în starea iniţială.
- 9) Şterge conţinutul tuturor graficelor. Însăşi graficele rămân – se şterg doar datele de pe ele.
- 10) Şterge complet toate graficele. Dacă controlul „11” nu este marcat – fereastra trece în starea iniţială.
- 11) Dacă este marcat acest control, fereastra trece în starea din Fig. 9 şi se activează colectarea datelor statistice privind starea reţelei la fiecare pas de simulare. Aceste date se reprezintă în pagina „History” (12). Dacă acest control este dezactivat şi nu este adăugat nici un grafic – fereastra trece în starea iniţială.
- 12) Aici se reprezintă datele statistice colectate privind starea reţelei la fiecare pas de simulare.
- 13) Următoarele pagini reprezintă graficele adăugate cu ajutorul butonului „Add” (7).

PNStudio dispune de un subsistem de configurare, care poate fi ajustat aproape sub orice aspect al funcţionării sistemului, începând cu culoarea şi tipul liniilor pentru orice tip de elemente ale reţelei. Pentru sincronizarea mai simplă a setărilor pe diferite calculatoare toate datele se înscriu în format XML, în fişierele „settings.xml” pentru setările generale a programului, şi „colors.xml” pentru schema de culori a reţelei RPHR.

Din Fig. 9 se observă că istoria simulării este prezentată sub forma unui tabel în care sunt stocate așa date ca: pasul simulării; timpul simulării; marcajul curent; tranziţiile

validate și cele declanșate atât cele imediate, cât și cele temporizate.

În Fig. 10 sunt prezentate operațiile de bază pentru efectuare simulării animate a unui model de rețea RPHR.

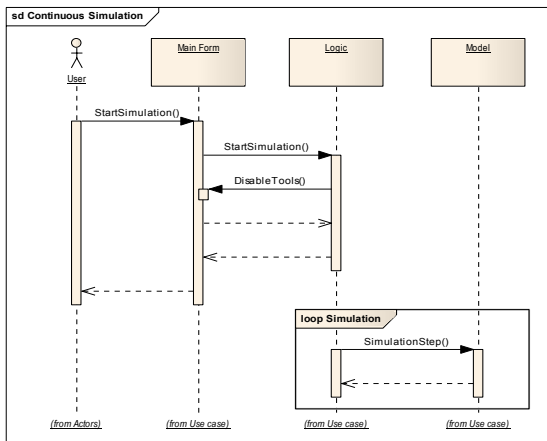


Fig. 10. Operațiile de bază de simulare animate a unui model de rețea RPHR.

Pentru a salva un grafic în fișier – trebuie de apăsat cu el cu *click drept* și de ales punctul corespunzător din meniu.

Un exemplu de diagramă a simulării părții discrete a unei RPHR este prezentat în Fig. 11.

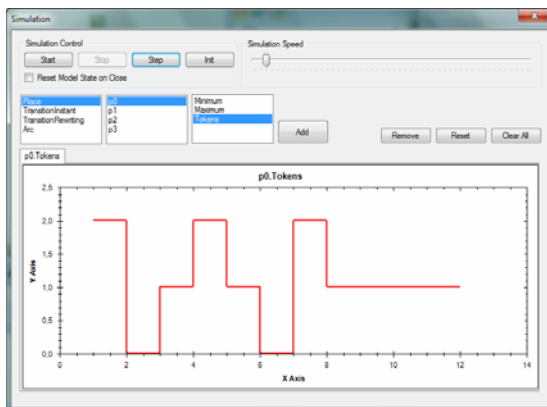


Fig. 11. Diagrama simulării a părții discrete a unei RPHR.

De menționat faptul că în timpul simulării este imposibil de redactat modelul rețelei. Modificările acestia 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”.

Modulul logic (PNCORE) a fost elaborat și testat pentru a avea performanța maximă pe care o permite .NET și LINQ. Astfel 46 mii de pași de simulare a unei rețele reconfigurabile, relativ simple, durează 21.5 secunde, cu o durată medie de 0.46 ms pentru fiecare pas. Modulul acesta și cel grafic au fost optimizate cu ajutorul programului CLRProfiler [11]. Erorile de proiectare și realizare au fost găsite și corectate cu ajutorul programului Microsoft FxCop.

#### IV. CONCLUZIE

Rețelele Petri hibride reconfigurabile sunt un instrument adecvat la modelarea funcționării proceselor discret-

continue ale sistemelor de calcul, informaționale și a proceselor de producție.

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

Actualmente, sunt efectuate elaborări pentru a integra în sistemul PNStudio -Toolkit a unor facilități care vor permite verificarea și simularea animată a modelelor de rețele Petri membranale hibride temporizate [6].

#### BIBLIOGRAFIE

- [1] C. Ciufudean, A. B. Larionescu, “ Estimation of the Performances of The Discrete Events Systems,” Advances in Electrical and Computer Engineering, no. 2, pp. 30-34, 2003.
- [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 Timed Membrane Petri Nets for Modelling of Parallel Computing. International Journal of Computers, Communications & Control,” no. 3, Vol. I, pp. 33-39, Agora University Editing House, Oradea, România, 2006.
- [7] 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.
- [8] E. Guțuleac, “Descriptive compositional HSPN based discrete - continuous modeling of distributed systems,” Analele Științifice ale Universității de Stat din Moldova, Seria Științe fizico-matematice, Ed.: CEP USM, Chişinău, pp. 182-187, 2005.
- [9] E. Guțuleac, Evaluarea performanțelor sistemelor de calcul prin rețele Petri stochastice. Ed.: „Tehnica-Info”, Chişinău, 2004, - 276 p.
- [10] 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.
- [11] [http://ru.wikipedia.org/wiki/.NET\\_Framework](http://ru.wikipedia.org/wiki/.NET_Framework).