PETRI NETS EXPLORER PENTRU SIMULAREA ŞI EVALUAREA ARHITECTURILOR MULTIAGENT

Alexei CORDUNEANU, Emilian GUŢULEAC

Universitatea Tehnică a Moldovei

Abstarct: *În lucrare sunt prezentate unele aspecte de elaborare şi realizare a sistemului Petri Nets Explorer destinat pentru simularea vizuală şi analiza funcţionării arhitecturilor multiagent. Acest sistem oferă o interfaţă grafică utilizator intuitivă pentru simularea diferitor elemente ale arhitecturilor multiagent, modelate prin reţele Petri membranale reconfigurabile. În cadrul acestui sistem este introdus şi realizat calculul matriceal al atributelor modelului simulat, ceia ce duce la o reprezentare mai compactă a modelelor sistemelor multiagent cu procese discret-continue care sunt reprezentate prin seturi de date multidimensionale.*

Cuvinte cheie: *Arhitecturi multiagent, calcul matriceal, modelare, reţele Petri membranle, simulare vizuală.*

1. Introducere

Conceptul de sisteme multiagent [1] este introdus recent în domeniul ştiinţific, dar acesta a câştigat multă simpatie printre cercetători datorită generalităţii modelului, a posibilităţilor multiple de utilizare şi abordare si, de asemenea, datorită rezultatelor remarcabile obţinute în urma folosirii sale. Sistemele cu arhitecturi multiagent sunt sisteme sau procese compuse din mai multe entităţi de calcul raţionale care interacţionează între ele, cunoscute sub numele de agenţi, care posedă două capacităţi importante. În primul rând, aceştia sunt capabili să acţioneze în mod autonom – de a decide de sine stătător ceea ce trebuie să execute pentru a-şi satisface obiectivele, în scopul cărora au fost proiectate. În al doilea rând, ei sunt capabili să interacţioneze cu alţi agenţi – nu doar prin schimbul de date, ci şi prin angajarea analogilor săi în activitatea sa de cooperare, coordonare, negociere, etc. Agenţii şi sistemele multiagent reprezintă o nouă modalitate de analiză, modelare şi implementare a sistemelor complexe cu procese discret-continue. Viziunea bazată pe agenţi oferă astăzi o gamă largă de instrumente, tehnici şi paradigme cu un uriaş potenţial de a îmbunătăţi modul în care sunt concepute şi utilizate noi tehnologii informaţionale. De asemenea, sistemele multiagent cu inteligenţă colectivă sunt considerate astăzi ca fiind tipul de sisteme care se va impune tot mai mult în aplicaţiile practice ale viitorului. Prin sistem bazat pe agenţi se înţelege un sistem de calcul în care elementul cheie îl reprezintă agentul reprezentat de o entitate hardware şi/sau software. În principiu, un astfel de sistem poate fi proiectat în funcţie de agenţi, dar implementat fără ca structurile sale să corespundă într-un fel agenţilor.

În lucrare sunt prezentate unele aspecte de elaborare şi realizare a sistemului Petri Nets Explorer destinat pentru simularea vizuală şi analiza funcţionării arhitecturilor multiagent. Acest sistem oferă o interfaţă grafică utilizator intuitivă pentru simularea diferitor elemente ale arhitecturilor multiagent, modelate prin reţele Petri membranale reconfigurabile [3]. În cadrul acestui sistem este introdus şi realizat calculul matriceal al atributelor modelului simulat, ceia ce duce la o reprezentare mai compactă a modelelor şi îl face util la modelarea sistemelor multiagent cu procese discret-continue care sunt reprezentate prin seturi de date multidimensionale.

2. Formalisme de modelare

La modelarea proceselor sistemelor de calcul se folosesc diferite formalisme şi instrumente cum ar fi: GPSS, LabVIEW, reţele Petri, algoritmii genetici, reţele neuronale şi altele [4]. Toate aceste instrumente de modelare, fiind dotate cu un aparat matematic specific de descriere a modelului, pot să descrie procesele sistemelor de calcul cât mai variate şi mai complexe. Însă, pe parcursul anilor s-a observat că sistemele de modelare la baza cărora stau elemente de inteligenţă artificială au o capacitate sporită de a da soluţii mai simple pentru probleme de complexitate sporită. Ceia ce duce la utilizarea acestor sisteme/formalisme (bazate pe elemente de inteligenţă artificială) pe scară largă in domeniul de modelare a sistemelor de calcul.

Unul din instrumentele de modelare moderne este formalismul reţelelor Petri. Acestea reprezintă un instrument de modelare şi verificare a sistemelor paralele şi concurente cu evenimente discrete. Există mai multe extensii ale reţelelor Petri, care sunt utilizate în diferite domenii pentru modelarea diverselor sisteme şi de o complexitate sporită.

Elaborarea mijloacelor de modelare, validare şi evaluare a performanţelor funcţionării sistemelor de calcul paralele/distribuite cu procese concurente este o problemă actuală, anume din cauza creşterii complexităţii sistemelor proiectate. Sistemele de calcul cu arhitecturi avansate, deseori, au o structură

ierarhică cu mai multe nivele reconfigurabile ce se restructurează, adaptându-se la schimbarea cerinţelor şi a mediului ambiant.

Mai mult, recent corporaţia IBM a lansat o nouă concepţie de organizare şi proiectare a sistemelor de calcul, conceptul de „Autonomic computing" [6], care reprezintă o arie nouă de studii în ştiinţa calculatoarelor, având ca scop de a elabora şi realiza sisteme de calcul capabile să funcţioneze independent, să se adapteze la diverse circumstanţe şi să-şi pregătească resursele pentru a manevra cât mai eficient sarcinile de lucru şi aplicaţiile curente. Astfel, un sistem de calcul autonom îşi determină starea mediului operaţional, modelează comportamentul său în acel mediu şi face anumite acţiuni pentru a schimba comportamentul său. Printre proprietăţile de bază, menţionăm proprietatea de reconfigurare şi autoconfigurare:

- abilitatea de a se adapta la condiţiile care se schimbă, restructurându-şi propriile configuraţii de hardware şi/sau software pe parcursul procesării aplicaţiilor,
- funcţionalitate care permite adăugarea şi/sau înlăturarea unor componente sau resurse în sistem, fără întreruperea procesării aplicaţiei curente.

Unul din formalismele care poate descrie funcţionarea unui sistem multiagent este calculul membranal [8]. Fiecare membrană din acest model reprezintă un agent raţional şi acesta interacţionează cu alţi agenţi/membrane. Interacţiunea dintre membrane active prezintă o bază pentru studiul comportamentului multiagent.

Calculul membranal este un domeniu nou, provocator şi în continuă dezvoltare, situat la frontiera dintre ştiinţele sistemelor de calcul, informatice, matematice şi cele biologice [8].

Componentele de bază ale unui sistem membranal - sunt structuri de membrană ce constau din membrane ierarhic integrate în membrana exterioară. Fiecare membrană include o regiune ce conţine un multiset de obiecte discrete şi posibil alte membrane. Mai formal, o structură membranală este un arbore, ale cărui noduri sunt alte membrane, rădăcina este numită skin membrană, iar frunzele arborelui sunt numite membrane elementare. Informal, acestea pot reprezenta structuri membranale folosind diagrame Venn (Figura 1).

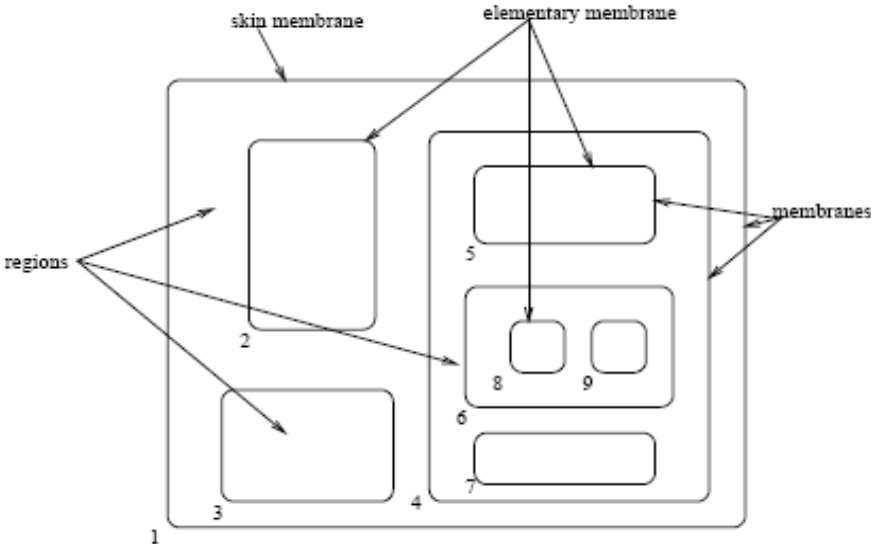

Figura 1. Reprezentarea grafică a unei structuri membranale [8]

Cercetătorii din ştiinţa calculatoarelor şi informatică utilizează, din ce în ce mai mult, concepte şi modele inspirate din sistemele biologice, precum şi simulările software derivate din aceste modele formale. Folosirea acestora duc la mai multe rezultate importante în problemele de luare a deciziei, verificare şi simulare a proceselor de calcul. În această direcţie au fost efectuate unele eforturi pentru a simula acest tip de sisteme prin reţele Petri membranale care permit verificarea multor proprietăţi comportamentale cum ar fi mărginirea, viabilitatea, reversibilitatea, etc. [3].

În caz general, problema efectuării în mod analitic a verificării şi analizei performanţelor modelelor de reţele Petri membranale este extrem de dificilă. Această problemă poate fi soluţionată prin elaborarea şi realizarea unor instrumente software obiect orientate spre simularea vizuală, care va da posibilitatea de a efectua verificarea funcţională şi analiza performanţelor acestor tip de modele.

3. Aplicația Petri Nets Explorer

Petri Nets Explorer este o aplicație desktop pentru simularea vizuală a modelelor de rețele Petri reconfigurabile membranale. Acesta furnizează utilizatorului o interfață grafică intuitivă şi comodă pentru a crea modele de rețele Petri membranale. Sistemul dat este dezvoltat pe platforma Microsoft. NET ce include cele mai moderne tehnologii cum ar fi: WPF, LINQ, Task paralel lybrary, PRISM, UNITY şi altele. Aceasta face ca sistemul dat să fie competitiv în fața aplicațiilor moderne din acest domeniu de aplicare la simularea proceselor de calcul orientate multiagent.

Fereastra de bază a aplicației date este prezentată în figura 2.

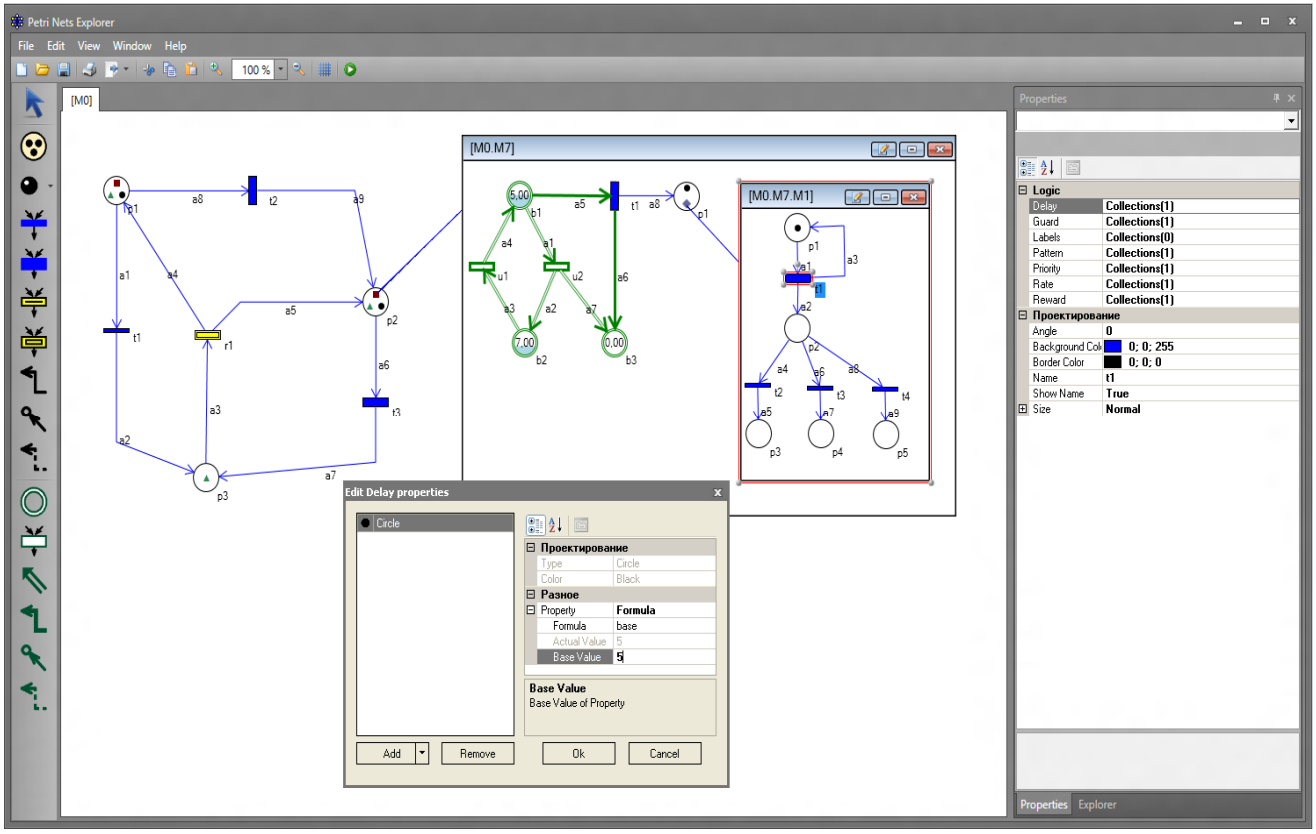

Figura 1. Fereastra principală a sistemului Petri Nets Explorer

Fapt că acest sistem instrumental prevede posibilități de modelare şi simulare a sistemelor de calcul membranale reconfigurabile îl face ca fiind unul din pionerii din acest domeniu. De asemenea, posibilitatea de a opera cu seturi de date multidimensionale îl face atractiv pentru aplicații cu sisteme vectoriale, care utilizează modelul de calcul vectorial/matriceal.

În partea dreapta a ferestrei principale este situat meniul de instrumente, numit „Tools” (Figura 3). Pentru crearea şi editarea modelelor de rețele Petri membranale acest meniu dat conține următoarele instrumente:

- *Pointer* – resetează instrumentul activ;
- *Discrete Place* – crearea locației discrete;
- *Token* –crează un jeton (membrană) în rețeaua (membrana) curentă;
- *Immediate Transition* – crearea unei tranzi'ii imediate;
- *Time Transition* – crearea a unei tranziții temporizate;
- *Rewriting Immediate Transition* – crearea unei reguli imediate de reconfigurare a structurii modelului;
- *Rewriting Immediate Transition* – crearea unei reguli temporizate de reconfigurare a structurii modelului;
- *Normal Arc* – creare unui arc normal (discret);
- *Inhibitor Arc* – creare unui arc inhibitor;
- *Test Arc* – crearea unui arc test;
- *Continous Place* – crearea unei locații continue;
- *Continous Transition* – crearea unei tranziții continue;
- *Continous Arc* – crearea unui arc continuu;
- *Flow Arc* – creare unui arc de setare;
- *Inhibitor Arc* – crearea unui arc continuu inhibitor;
- *Test Arc* – crearea unui arc continuu test.

Figura 2. Meniul de instrumente

Crearea şi editarea unei rețele Petri poate fi efectuată prin plasarea pe suprafața de lucru a reprezentărilor grafice pentru elemente respective şi interconectarea lor într-un mod prestabilit.

La adăugarea acestui element lui i se va genera un identificator unic după un anumit şablon. Fiecare model conține un set de proprietăți care pot fi setate de utilizator în fereastra flotantă din dreapta a ferestrei de bază.

Odată ce modelul a fost construit acesta poate fi simulat şi astfel, putem să studiem comportamentul sistemului. Simularea sistemului este controlată de fereastra de simulare (Figura 3) în care pot fi aleşi parametrii de simulare, gradul de detaliere al simulării precum şi caracteristicile necesare pentru studiu.

În timpul simulării procesele decizionale sunt controlate de elemente reconfigurabile.

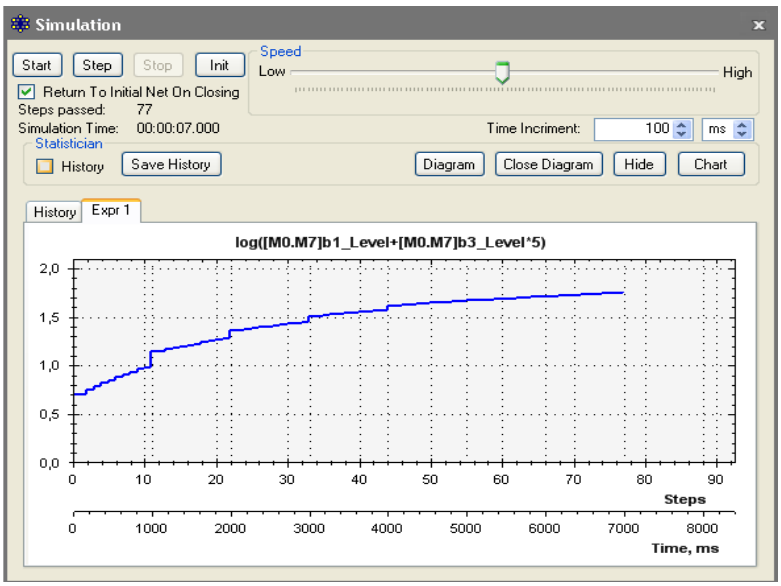

Figura 4. Fereastra de simulare

4. Calculul matriceal al atributelor modelului

Pentru a asigura universalitatea sistemului în domeniul modelării agenților raționali este nevoie de a introduce calculul vectorial/matriceal.

Pentru a defini atributele matriceale ale unei rețele Petri membranale cel puțin este necesar de matrice tridimensionale din cauza faptului ca avem două tipuri de entități noduri (locații şi tranziții) şi diferite tipuri de arce. În acest scop, pentru simularea vizuală a modelelor multiagent, în nucleul sistemului sunt folosite atribute matriceale şi acestea trebuie să implementeze următoarea interfață:

```
publicinterfaceIMatrix<T>
    {
IList<int>Size{ get; set; }
IList<T>Items{ get; set; }
TValue{ get; }

voidSetValue(IEnumerable<int?> indexes, IMatrix<T> value);
IMatrix<T>GetValue(IEnumerable<int?> indexes);

IList<IMatrix<T>>Rows{ get; }
IList<IMatrix<T>>Columns{ get; }
IList<IMatrix<T>>Pages{ get; }

//Operations
IMatrix<T>Inverse();
IMatrix<T>Transpose();
IMatrix<T>Add(IMatrix<T> other);
IMatrix<T>Substract(IMatrix<T> other);
IMatrix<T>Multiply(IMatrix<T> other);
    }
```

Astfel, structura de date folosită la implementarea interfeței grafice utilizator face posibilă păstrarea şi prelucrarea datelor multidimensionale. În figura 5 este prezentat un exemplu de matrice tridimensională.

Una din problemele întâlnite la reprezentarea atributelor cantitative ale modelelor este înmulțirea matricelor de diferite dimensiuni. Soluția acceptată spre implementare este MULTIPROD framework (Multiple matrix multiplications, with array expansion enabled) descrisă de Paolo de Leva în [8].

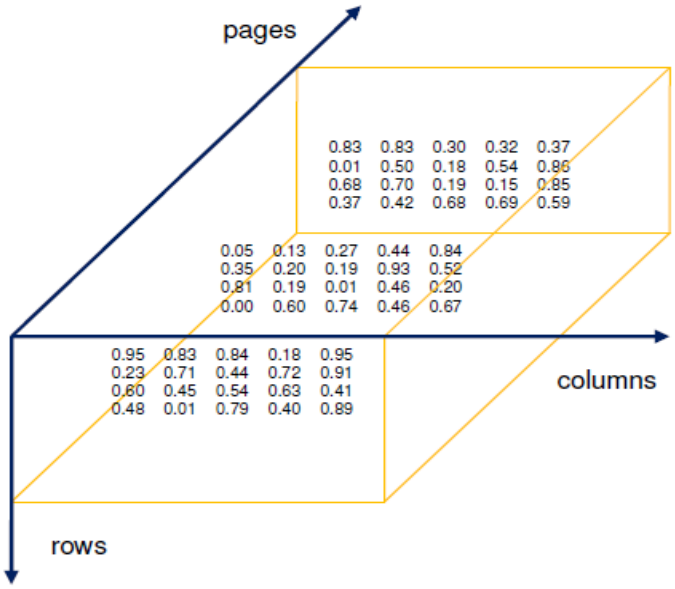

Figura 5. Matrice multidimensională [8].

5. Concluzie

În lucrare sunt prezentate unele aspecte de elaborare şi realizare a unui sistem software instrumental, Petri Nets Explorer, realizat pentru modelarea, verificarea funcțională, simularea animată şi evaluarea performanțelor sistemelor multiagent, descrise prin rețele Petri membranale reconfigurabile. Acest sistem oferă o interfață grafică utilizator intuitivă pentru simularea diferitor elemente ale arhitecturilor multiagent modelate prin rețele Petri membranale reconfigurabile. În cadrul acestui sistem este introdus şi realizat calculul matriceal al atributelor modelului simulat, ceia ce duce la o reprezentare mai compactă a modelelor îl face util la modelarea sistemelor multiagent cu procese discret-continue care sunt reprezentate prin seturi de date multidimensionale.

Bibliografie

1. Wooldridge, M. *Introduction to multiagent system*. T&T Productions Ltd, London, 2002.
2. Compton, K., Hauck, S. *Reconfigurable Computing: a Survey of Systems and Software.* ACM Computing Surveys (CSUR), vol. 34, no. 2, 2002, p. 171-210.
3. Guțuleac, E., Mocanu, M. L., Țurcanu, Iu. *Membrane Differential Petri Nets for Performance Modelling of Hybrid P-Systems*. Control Engineering and Applied Informatics, vol. 9, no. 3-4, Ed.: SRAIT, Bucureşti, România, 2007, p. 12-21.
4. Calzarossa, M., Marie, R. *Tools for Performance Evaluation.* Perf. Evaluation, no. 33, 1998, p.1-3.
5. Cook, S., Harrison, R., Wernik, P. *A simulation model of self-organising evolvability in software systems.* Proc. of the 1-st IEEE International Workshop on Software Evolvability, Hungary, 2005, p. 17-22.
6. Kephart, J. O. *Research challenges of autonomic computing*. In ICSE '05: Proceedings of the 27th International conference on Software engineering. New York, NY, USA: ACM, 2005, p. 15–22.
7. Pujari, Sh., Mukhopadhyay, S. *Petri Net: A Tool for Modeling and Analyze Multi-agent Oriented Systems*. I.J. Intelligent Systems and Applications, 10, 2012, 103-112.
8. Păun, Gh., Rozenberg, G. *A Guide to Membrane Computing*. Theoretical Computer Science, nr. 287, 2002, p. 73-100.
9. http://www.mathworks.de/matlabcentral/fileexchange/8773-multiple-matrix-multiplications-with-array-expansion-enabled.