УДК 681.3

Гломозда Д. К.

МОДЕЛЮВАННЯ РОБОТИ КОЛАБОРАТИВНОГО СЕРЕДОВИЩА ЗА ДОПОМОГОЮ КОЛЬОРОВИХ МЕРЕЖ ПЕТРІ

У статті описано модель координаційного механізму програмної системи підтримки асинхронної дистанційної взаємодії в мережі Інтернет, створену за допомогою кольорових мереж Петрі. Наведено результати дослідження запропонованої моделі на безпечність, активність та коректність.

Ключові слова: кольорові мережі Петрі, CPN Tools, координація, колаборативне середовище, дистанційна взаємодія.

Вступ

Стаття продовжує цикл робіт [1; 2; 3], присвячених побудові моделі програмної системи підтримки асинхронної дистанційної взаємодії в масштабах Інтернет (далі – ПСПАДВІ), що ґрунтується на структурі колаборативного середовища, складовими елементами якої є сеанси, користувачі, спільні ресурси та рівні [5], за допомогою яких реалізується рівневий протокол доступу до ресурсів [6]. У попередніх працях було запропоновано модель ПСПАДВІ мовою мереж Петрі, описано багатоагентну систему, яка моделювала роботу складових такої системи, та побудовано прототип координаційної системи, що пов'язує автоматизовану систему управління навчальним закладом із системою керування навчальним процесом. У цій статті пропоновано модель ПСПАДВІ мовою кольорових мереж Петрі [7] – потужнішою від використаної у попередніх працях мову низькорівневих мереж Петрі, а тому перспективнішою з погляду проведення подальших поглиблених досліджень.

Побудова моделі

У праці [2] описувалася низькорівнева мережна модель колаборативної системи, до якої входять N користувачів, M сеансів, L ресурсів та координаційний механізм – сукупність позицій та переходів мережі Петрі, що зв'язує користувачів, сеанси та ресурси (рис. 1). Координаційним механізмом називався блок, до якого входили контролери рівнів (кожному рівню відповідає один ресурс) та механізм забезпечення взаємовиключення при створенні сеансу, і складався з таких позицій мережної моделі:

а) механізм забезпечення взаємовиключного створення сеансу для кожного з користувачів та для кожного сеансу. Це позиції «Користувач U_i не є керівником сеансу S_j», «Користувач U_i є керівником сеансу S_j», «Користувач U_i запитує дозвіл на створення сеансу S_j» та переходи, «користувач U_i хоче розпочати сеанс S_j», «користувач U_i хоче завершити сеанс S_j», «користувачу U_i дозволено створити сеанс S_j», «користувачу U_i не дозволено створити сеанс S_j»;

б) контролер рівня. Позиції: «Рівень F_k вільний», «Рівень F_k обробляє запит», «Рівень F_k в стані очікування», «Надати ресурс R_k», «Ресурс R_k надано», «Звільнити ресурс R_k», «Ресурс R_k звільнено», «Видалити ресурс R_k», «Ресурс R_k видалено»;

в) механізм забезпечення взаємовиключного доступу кожного з користувачів до кожного з рівнів. Позиції: «Користувач U_i не є

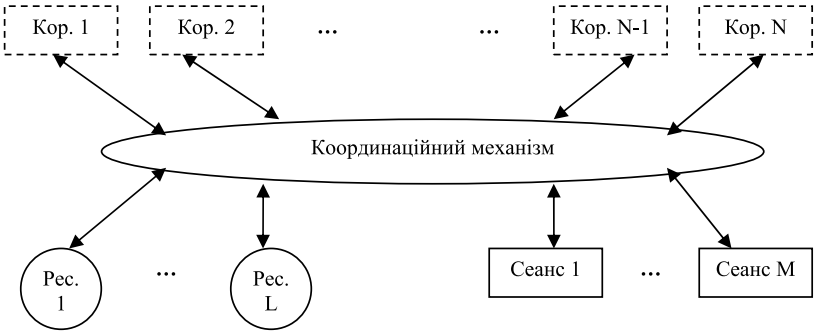

Рис. 1. Структура мережної моделі колаборативного середовища

утримувачем рівня F_k», «Користувач U_i запитує доступ до рівня F_k», «Користувач U_i є утримувачем рівня F_k», «Рівень F_k активно використовується користувачем U_i», «Рівень F_k пасивно використовується користувачем U_i». Переходи: «користувач U_i хоче отримати доступ до рівня F_k», «користувачеві U_i відмовлено у доступі до рівня F_k», «користувачеві U_i дозволено доступ до рівня F_k», «рівень F_k зайнято користувачем U_i», «користувач U_i призупиняє використання рівня F_k», «користувач U_i поновлює використання рівня F_k», «користувач U_i хоче видалити ресурс R_k», «ресурс R_k видалено користувачем U_i», «користувач U_i хоче звільнити рівень F_k», «користувач U_i звільнив рівень F_k».

При цьому $i \in 1…N$; $j \in 1…M$; $k \in 1…L$.

Таким чином, зазвичай низькорівнева мережа Петрі, що моделює роботу координаційного механізму колаборативної системи, складатиметься з:

- 3 * N * M позицій та 4 * N * M переходів, що регламентують взаємовиключне створення сеансів;
- 9 * L позицій, що відповідають контролерам рівнів;
- 5 * N * L позицій та 10 * N * L переходів, що відповідають зв'язкам між контролерами рівнів та користувачами.

При переході на мову кольорових мереж Петрі загальна структура мережної моделі залишилася такою самою, а от перелік позицій і переходів, а також їх функціональне наповнення зазнали змін, які стали можливими завдяки механізму надання фішкам певних типів даних і присвоєння конкретних значень, які обробляються позиціями та переходами, і на основі яких визначається відкритість/закритість переходів та правила заповнення позицій, з якими пов'язуються так звані множини кольорів (colour sets), що визначають типи фішок, які можуть перебувати на тій чи іншій позиції. В нашій моделі використовуються такі множини кольорів:

colset BOOL = bool (булеві значення);
colset ID = int (цілі числа);
colset IDxID = product ID * ID (пари цілих чисел);
colset IDRECLIST = list IDxID (список пар цілих чисел).

Механізм типізації дозволив скоротити кількість необхідних позицій. Так, позиції «Користувач U_i не є керівником сеансу S_j» та «Користувач U_i є керівником сеансу S_j» стали однією позицією «Користувач U_i є керівником сеансу S_j» з типом даних BOOL. На старті вони містять фішки зі значенням «хиба», що означає, що користувач не є керівником сеансу, а коли він стає керівником сеансу, в позицію поміщається фішка зі значенням «істина». Аналогічним чином були об'єднані позиції «Користувач U_i не є утримувачем рівня F_k» і «Користувач U_i є утримувачем рівня F_k».

Означені множини кольорів використовуються для опису не лише позицій мережі, а й змінних, що входять до скалу дугових виразів (arc expressions), які визначають функціональність дуг мережі. В нашій моделі використовуються такі змінні:

var idu : ID (ідентифікатор користувача);
var ids : ID (ідентифікатор сеансу);
var idf : ID (ідентифікатор рівня);
var idr : ID (ідентифікатор ресурсу);
var idreclist : IDRECLIST (список користувачів-учасників сеансу).

Наявність таких засобів, як типізація і дугові вирази дає можливість наочно і зручно змоделювати механізм реєстрації користувачів, що працюють в сеансі, але не є його керівниками. В низькорівневій мережі Петрі для цього, на додачу до N * M переходів «користувач U_i приєднується до сеансу S_j», треба було б створити ще N * M позицій виду «Користувач U_i є учасником сеансу S_j». У кольоровій мережі Петрі достатньо встановити для позицій «Сеанс S_j активний» тип даних IDRECLIST, а для вхідних дуг, які з'єднують ці позиції з переходами «користувачу U_i дозволено створити сеанс S_j» і «користувач U_i приєднується до сеансу S_j», задати такі дугові вирази:

– для дуги від переходу «користувачу U_i дозволено створити сеанс S_j»: *ins idreclist (ids,idu);*

– для дуги від переходу «користувач U_i приєднується до сеансу S_j»: *ins idreclist (#1(List.hd idreclist),idu),*

– де *ids* – ідентифікатор сеансу, *idu* – ідентифікатор користувача, idreclist – список приєднаних до сеансу користувачів, який міститься у фішці в позиції «Сеанс S_j активний» і подається на перехід по вихідній дузі. Команди ins та List.hd є інструкціями функціональної мови CPN ML [7; Ch. 3] та означають, відповідно, вставку елемента в хвіст зазначеного списка та отримання голови списка. Інструкція #1 означає, що треба взяти перше поле добутку (список складається з елементів типу IDxID), яке містить ідентифікатор сеансу.

Вихід з сеансу реалізується іншим дуговим виразом: *rm (getIDSbyIDU(idreclist,idu), idu) idreclist*. Команда rm означає видалення з вказаного списка вказаного елемента, а функція getIDSbyIDU повертає ідентифікатор сеансу, що входить до складу елемента, друге поле якого містить зазначений ідентифікатор користувача. Ось її код:

```
fun getIDSbyIDU(list : IDRECLIST, idu : ID) : ID =
if list = []
      then 0
else
      if (idu = #2(List.hd list))
            then (#1(List.hd list))
      else getIDSbyIDU(List.tl list, idu).
```

Таким чином, детальна інформація про поточний стан системи кодується безпосередньо у фішці в позиції «Сеанс S_j активний», і ніяких додаткових позицій та переходів не потрібно.

Механізм забезпечення взаємовиключного доступу кожного з користувачів до кожного з рівнів також скоротився. Завдяки тому, що фішка з множини кольорів IDxID може містити пару «ідентифікаційний номер рівня, ідентифікаційний номер користувача», відпала необхідність в окремих позиціях типу «Рівень F_k активно використовується користувачем U_i» та «Рівень F_k пасивно використовується користувачем U_i». На зміну їм прийшли універсальні позиції «Рівень F_k активно використовується» та «Рівень F_k пасивно використовується», що мають множину кольорів IDxID і містять індекс користувача, який у даний момент використовує рівень. Одночасно немає потреби у додаткових переходах, що вели в ці позиції, тож позиції «Рівень F_k в стані очікування», що слугували для синхронізації моделювання перехідного етапу між відкритістю і зайнятістю, і пов'язані з ними переходи також стали непотрібними. Те саме стосується позицій і переходів, що моделювали зв'язок між рівнем та ресурсом.

Загалом, у випадку реалізації мовою кольорових мереж Петрі блок, що відповідає координаційному механізму, складається з таких позицій і переходів:

а) механізм забезпечення взаємовиключного створення сеансу для кожного з користувачів та для кожного сеансу. Це позиції «Користувач U_i є керівником сеансу S_j», «Користувач U_i запитує дозвіл на створення сеансу S_j» та переходи «користувач U_i хоче розпочати сеанс S_j», «користувач U_i хоче завершити сеанс S_j», «користувачу U_i дозволено створити сеанс S_j», «користувачу U_i не дозволено створити сеанс S_j»;

б) контролер рівня. Позиції: «Рівень F_k вільний», «Рівень F_k обробляє запит»;

в) механізм забезпечення взаємовиключного доступу кожного з користувачів до кожного з рівнів. Позиції: «Користувач U_i запитує доступ до рівня F_k», «Користувач U_i є утримувачем рівня F_k», «Рівень F_k активно використовується», «Рівень F_k пасивно використовується». Переходи: «користувач U_i хоче отримати доступ до рівня F_k», «користувачеві U_i відмовлено у доступі до рівня F_k», «користувачеві U_i дозволено доступ до рівня F_k», «користувач U_i призупиняє використання рівня F_k», «користувач U_i поновлює використання рівня F_k», «користувач U_i видаляє ресурс R_k», «користувач U_i звільняє рівень F_k».

При цьому $i \in 1…N$; $j \in 1…M$; $k \in 1…L$.

Отже, в загальному випадку кольорова мережа Петрі, що моделює роботу координаційного механізму колаборативної системи, складатиметься з:

- $2 * N * M$ позицій та $4 * N * M$ переходів, що регламентують взаємовиключне створення сеансів;
- $2 * L$ позицій, що відповідають контролерам рівнів;
- $2 * N * L + 2$ позицій та $7 * N * L$ переходів, що відповідають зв'язкам між контролерами рівнів та користувачами.

Загальний вигляд мережної моделі системи, до складу якої входять двоє користувачів, один сеанс, один рівень і один ресурс (позначимо її u2s1f1), побудованої в середовищі CPN Tools [8], показано на рис. 2–4.

Дослідження моделі

Окрім графічного редактора для створення мереж, середовище CPN Tools включає засоби автоматизованого аналізу побудованої мережної моделі в просторі станів [9] шляхом, зокрема, побудови дерева досяжності [4, с. 79–106]. Дерево досяжності для мережі u2s1f1 зображено на рис. 5. Завершення процесу побудови дерева досяжності означає, що множина можливих маркувань нашої мережі скінченна, а отже, наша мережа обмежена. Додаткові властивості можна дізнатись у звіті аналізатора, який складається з кількох розділів. Для мережі u2s1f1 він матиме такий вигляд:

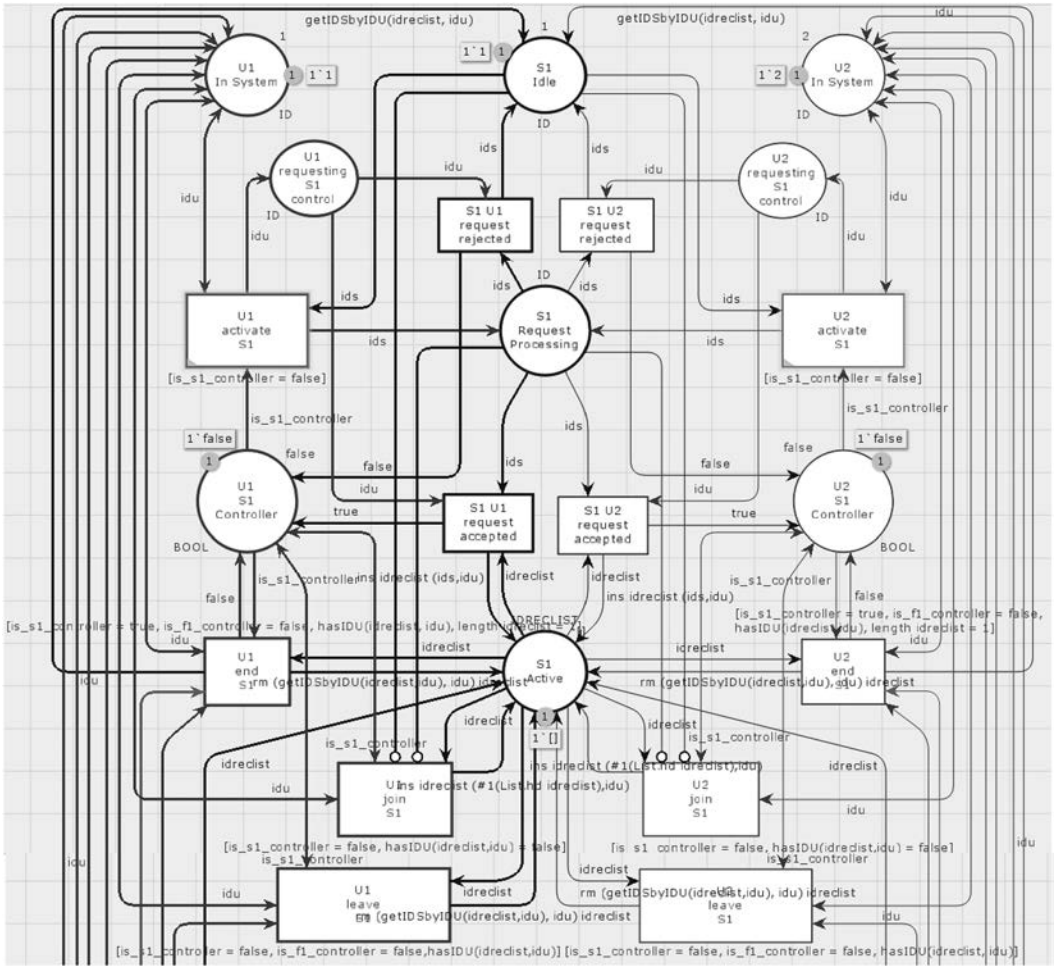

Рис. 2. Мережна модель колаборативної системи (u2s1f1).
Блок взаємодії користувачів із контролером сеансу

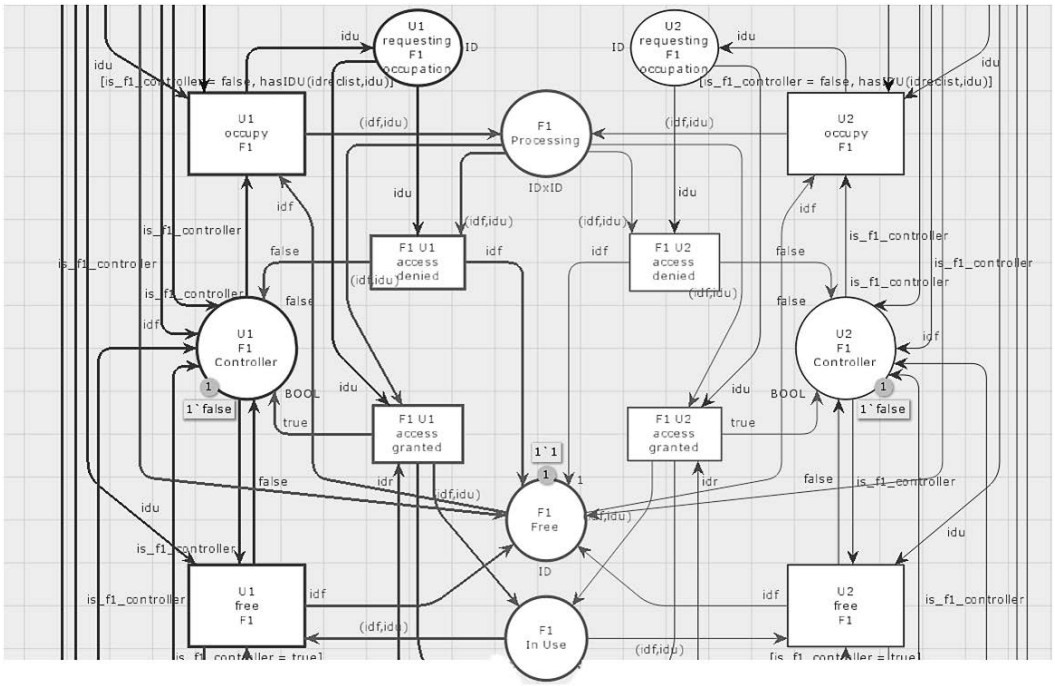

Рис. 3. Мережна модель колаборативної системи (u2s1f1).
Блок взаємодії користувачів із контролером рівня

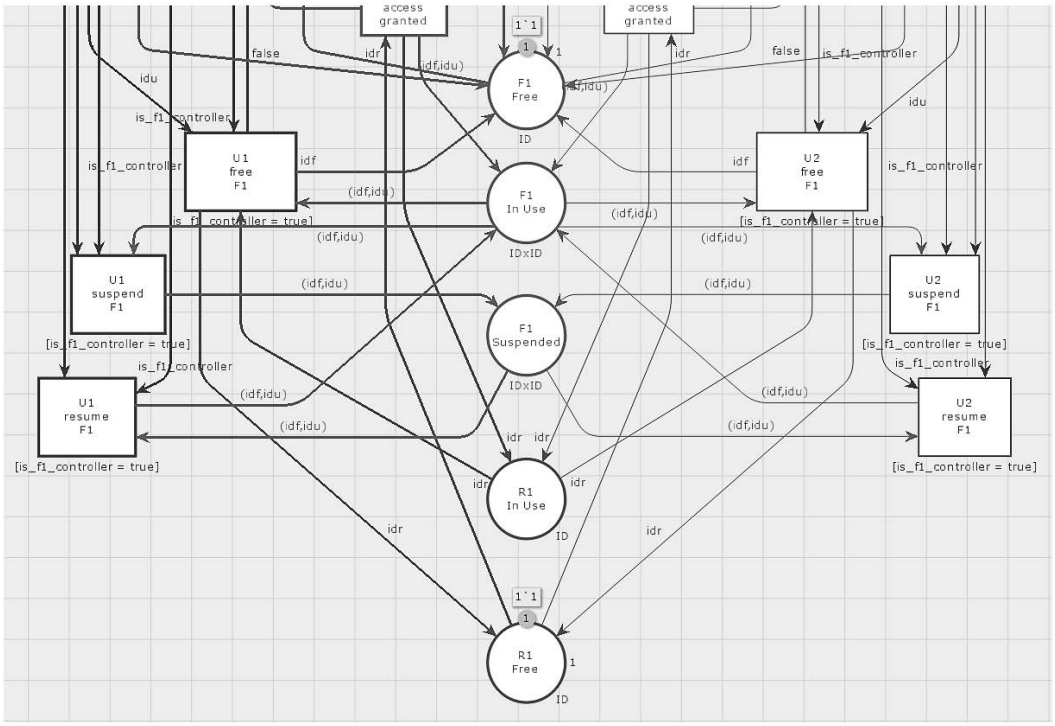

Рис. 4. Мережна модель колаборативної системи (u2s1f1). Блок взаємодії користувачів із контролером рівня і контролером ресурсу (можливість створення/видалення ресурсу користувачем відсутня)

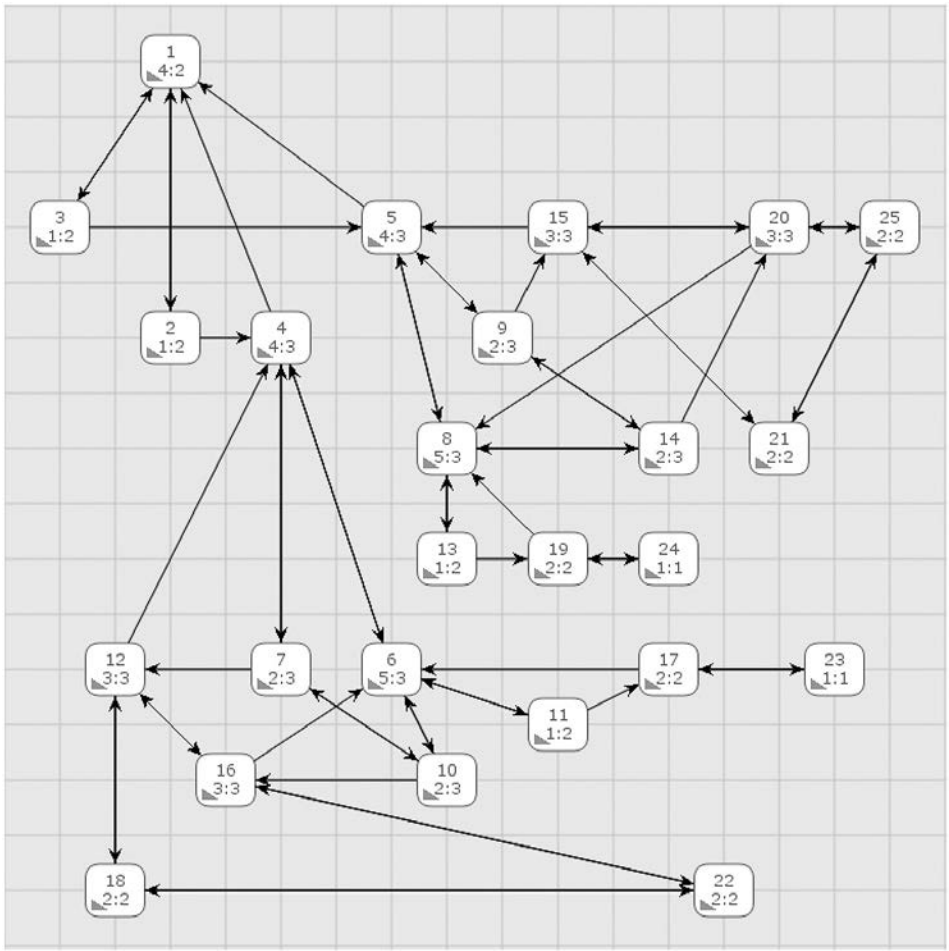

Рис. 5. Дерево досяжності для мережі u2s1f1

Statistics

State Space
Nodes: 25
Arcs: 60
Secs: 0
Status: Full

Scc Graph
Nodes: 1
Arcs: 0
Secs: 0

Ці дані означають, що дерево досяжності складається з 25 вершин і 60 дуг, які входять до однієї загальної сильно зв'язної компоненти.

Boundedness Properties

Best Integer Bounds

		Upper	Lower
Net_Model'F1_Free	*1*	*1*	*0*
Net_Model'F1_In_Use	*1*	*1*	*0*
Net_Model'F1_Processing	*1*	*1*	*0*
Net_Model'F1_Suspended	*1*	*1*	*0*
Net_Model'R1_Free	*1*	*1*	*0*
Net_Model'R1_In_Use	*1*	*1*	*0*
Net_Model'S1_Active	*1*	*1*	*1*
Net_Model'S1_Idle	*1*	*1*	*0*
Net_Model'S1_Request_Proc…	*1*	*1*	*0*
Net_Model'U1_F1_Controller	*1*	*1*	*0*
Net_Model'U1_In_System	*1*	*1*	*1*
Net_Model'U1_S1_Controller	*1*	*1*	*0*
Net_Model'U1_requesting_F1_occ…	*1*	*1*	*0*
Net_Model'U1_requesting_S1_control	*1*	*1*	*0*
Net_Model'U2_F1_Controller	*1*	*1*	*0*
Net_Model'U2_In_System	*1*	*1*	*1*
Net_Model'U2_S1_Controller	*1*	*1*	*0*
Net_Model'U2_requesting_F1_occ…	*1*	*1*	*0*
Net_Model'U2_requesting_S1_control	*1*	*1*	*0*

Best Upper Multi-set Bounds

Net_Model'F1_Free	*1*	*1`1*
Net_Model'F1_In_Use	*1*	*1`(1,1)++1`(1,2)*
Net_Model'F1_Processing	*1*	*1`(1,1)++1`(1,2)*
Net_Model'F1_Suspended	*1*	*1`(1,1)++1`(1,2)*
Net_Model'R1_Free	*1*	*1`1*
Net_Model'R1_In_Use	*1*	*1`1*
Net_Model'S1_Active	*1*	*1`[]++1`[(1,1)]++ 1`[(1,1),(1,2)]++ 1`[(1,2)]++1`[(1,2),(1,1)]*
Net_Model'S1_Idle	*1*	*1`1*
Net_Model'S1_Request_Proc…	*1*	*1`1*
Net_Model'U1_F1_Controller	*1*	*1`false++1`true*

Net_Model'U1_In_System	*1*	*1`1*
Net_Model'U1_S1_Controller	*1*	*1`false++1`true*
Net_Model'U1_requesting_F1_occ…	*1*	*1`1*
Net_Model'U1_requesting_S1_control	*1*	*1`1*
Net_Model'U2_F1_Controller	*1*	*1`false++1`true*
Net_Model'U2_In_System	*1*	*1`2*
Net_Model'U2_S1_Controller	*1*	*1`false++1`true*
Net_Model'U2_requesting_F1_occ…	*1*	*1`2*
Net_Model'U2_requesting_S1_control	*1*	*1`2*

Best Lower Multi-set Bounds

Net_Model'F1_Free	*1*	*empty*
Net_Model'F1_In_Use	*1*	*empty*
Net_Model'F1_Processing	*1*	*empty*
Net_Model'F1_Suspended	*1*	*empty*
Net_Model'R1_Free	*1*	*empty*
Net_Model'R1_In_Use	*1*	*empty*
Net_Model'S1_Active	*1*	*empty*
Net_Model'S1_Idle	*1*	*empty*
Net_Model'S1_Request_Processing	*1*	*empty*
Net_Model'U1_F1_Controller	*1*	*empty*
Net_Model'U1_In_System	*1*	*1`1*
Net_Model'U1_S1_Controller	*1*	*empty*
Net_Model'U1_requesting_F1_occ…	*1*	*empty*
Net_Model'U1_requesting_S1_control	*1*	*empty*
Net_Model'U2_F1_Controller	*1*	*empty*
Net_Model'U2_In_System	*1*	*1`2*
Net_Model'U2_S1_Controller	*1*	*empty*
Net_Model'U2_requesting_F1_occ…	*1*	*empty*
Net_Model'U2_requesting_S1_control	*1*	*empty*

Цей блок містить інформацію про кількість і різновиди (алфавіт) фішок, які можуть опинятися в тій чи іншій позиції, а також про верхні і нижні межі їх кількості.

Home Properties

Home Markings

All

Цей розділ містить перелік домашніх маркувань, тобто маркувань, досяжних з будь-якого іншого маркування. Бачимо, що домашніми є всі без винятку маркування нашої мережі, що свідчить про її зворотність.

Liveness Properties

Dead Markings

None

Dead Transition Instances

None

Live Transition Instances

All

Цей блок містить дані щодо активності нашої мережі. Тупикові маркування і мертві переходи відсутні, всі переходи активні.

Fairness Properties

Impartial Transition Instances
None

Fair Transition Instances
None

Just Transition Instances

Net_Model'S1_U1_request_accepted	*1*
Net_Model'S1_U1_request_rejected	*1*
Net_Model'S1_U2_request_accepted	*1*
Net_Model'S1_U2_request_rejected	*1*
Net_Model'U1_activate_S1	*1*
Net_Model'U1_end_S1	*1*
Net_Model'U2_activate_S1	*1*
Net_Model'U2_end_S1	*1*

Transition Instances with No Fairness

Net_Model'F1_U1_access_denied	*1*
Net_Model'F1_U1_access_granted	*1*
Net_Model'F1_U2_access_denied	*1*
Net_Model'F1_U2_access_granted	*1*
Net_Model'U1_free_F1	*1*
Net_Model'U1_join_S1	*1*
Net_Model'U1_leave_S1	*1*
Net_Model'U1_occupy_F1	*1*
Net_Model'U1_resume_F1	*1*
Net_Model'U1_suspend_F1	*1*
Net_Model'U2_free_F1	*1*
Net_Model'U2_join_S1	*1*
Net_Model'U2_leave_S1	*1*
Net_Model'U2_occupy_F1	*1*
Net_Model'U2_resume_F1	*1*
Net_Model'U2_suspend_F1	*1*

Останній розділ звіту містить інформацію про чесність нашої мережі. Як бачимо, неупереджених переходів, тобто таких, що спрацьовують нескінченно часто в будь-якій з можливих нескінченних послідовностей спрацювань (infinite firing sequence, IFS), немає. Чесних переходів, тобто таких, що спрацьовують нескінченно часто в кожній НПС, де ці переходи нескінченно часто вмикаються, також немає. Зате є справедливі переходи, тобто такі, що спрацьовують нескінченно часто в кожній НПС, де ці переходи, починаючи з деякого моменту, періодично вмикаються, і ущемлені переходи, тобто такі, для яких існують НПС, в яких ці переходи вмикаються, але так ніколи і не спрацьовують. Утім, це не є проблемою; навпаки, це свідчить про надійність нашої системи, в якій передбачено можливість відкату до початкового стану майже з будь-якого етапу роботи.

Висновки

Запропонована модель є наступним кроком у розвитку єдиної теорії мережної співпраці. Використання високорівневих мереж Петрі дає змогу наблизити модель до реальних практичних розробок і застосувань, зокрема до засобів автоматичної генерації програмного коду, які не

можуть бути застосованими до низькорівневих моделей. Продовженням роботи є побудова легко розширюваної та узагальнюваної ієрархічної модульної мережної моделі з використанням параметризованих модулів, а також імплементація часових механізмів.

Список літератури

1. Глибовець М. М. Формальна модель координаційно-орієнтованої мережі для колаборативної системи навчання / М. М. Глибовець, Д. К. Гломозда // Проблеми програмування. – 2006. – № 2–3. Спец. вип. – С. 402–412.
2. Глибовец Н. Н. Сложность задачи верификации координационного механизма системы программной поддержки совместной сетевой работы / Н. Н. Глибовец, Д. К. Гломозда // Кибернетика и системный анализ. – 2008. – № 4. – С. 15–19.
3. Гломозда Д. К. Координація в асинхронних обчислювальних мережах : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 01.05.03 «Математичне та програмне забезпечення обчислювальних машин і систем» / Д. К. Гломозда. – К., 2011. – 19 с.
4. Котов В. Е. Сети Петри / В. Е. Котов – М. : Наука, 1984. – 160 с.
5. Dommel H.-P. Networking Foundations for Collaborative Computing at Internet Scope / H.-P. Dommel, J. J. Garcia-Luna-Aceves // International ICSC Congress on Intelligent Systems and Applications, Symposium on Interactive and Collaborative Computing (ICC'2000). – Wollongong (Australia), 2000. – 7 p.
6. Dommel H.-P. Floor control for multimedia conferencing and collaboration / H.-P. Dommel, J. J. Garcia-Luna-Aceves // Multimedia Systems. – Vol. 5. – Springer-Verlag, 1997. – P. 23–38.
7. Jensen K. Coloured Petri Nets. Modelling and Validation of Concurrent Systems / K. Jensen, L. M. Kristensen – Berlin : Springer, 2009. – 384 p.
8. Jensen K. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems / K. Jensen, L. M. Kristensen, L. Wells // International Journal on Software Tools for Technology Transfer. – 2007. – Vol. 9. – No. 3–4. – P. 213–254.
9. Jensen K. CPN Tools State Space Manual / K. Jensen, S. Christensen, L. M. Kristensen. – Aarhus : University of Aarhus, 2002. – 49 p.

D. Glomozda

MODELLING OF FUNCTIONING OF COLLABORATIVE ENVIRONMENT WITH COLORED PETRI NETS

A model of coordination mechanism of a program system to support asynchronous distant collaboration in Internet scope, created with a help of coloured Petri nets is described. The results of studying the model in terms of safety, liveness and correctness are presented.

Keywords: coloured Petri nets, CPN Tools, coordination, collaborative environment, remote cooperation.

Матеріал надійшов 15.05.2014