

## ФОРМАЛЬНА МОДЕЛЬ ФУНКЦІОНУВАННЯ КОЛАБОРАТИВНОГО СЕРЕДОВИЩА

Гломозда Д.К., Глибовець М.М.

### Вступ

За останні роки відбулося значне поширення Інтернет-послуг. Але розвиток технологій спільної роботи, яка ґрунтується на спеціальному програмному забезпеченні та середовищі взаємодії для спільного розв'язання конкретних задач, сповільнений через те, що навіть на неформальному рівні досить складно описати дистанційне співробітництво, яке дає змогу людям спільно використовувати і маніпулювати мультимедійною інформацією в реальному часі і на різних рівнях модальності.

Проектування системи дистанційного співробітництва достатньо складне, оскільки охоплює користувачів, мережу та проблеми з головними комп'ютерами (хостами), наприклад, різномірні платформи для "застосувань". Користувачі прагнуть отримати середовище дистанційного співробітництва, яке має забезпечити якість взаємодії, близької до "зустрічі віч-на-віч". В цьому випадку стають необхідними координаційні механізми досягнення консенсусу для спільного й ефективного використання ресурсів.

Першим кроком вирішення проблеми побудови ефективної комп'ютерної системи підтримки середовища колаборативного типу є створення формальної моделі. Незважаючи на стрімкий розвиток мережних технологій, питання побудови адекватної формальної моделі середовища співпраці все ще залишається відкритим і актуальним. В даній роботі пропонується така формальна модель, створена за допомогою мереж Петрі. При цьому ми намагалися якнайбільше абстрагуватися від конкретних предметних областей застосування та конкретних програмних реалізацій засобів підтримки таких середовищ. Вирішення проблем, пов'язаних з координацією, на базі абстрактної моделі допоможе згодом побудувати програмні реалізації, що матимуть аналогічні властивості.

Формальна модель колаборативного середовища повинна ґрунтуватися на детальній структурі формальної специфікації та перевірки властивостей колаборативних систем. При побудові нашої моделі ми взяли за основу структуру, запропоновану Доммелем та Гарсія-Луна-Ачевесом [4]. Колаборативне середовище вони подають як набір  $\Gamma = \langle S, U, R, F \rangle$ , де  $S$  — множина сеансів,  $U$  — множина користувачів,  $R$  — набір спільних ресурсів,  $F$  — множина рівнів (floor), що керують ресурсами. Сеанс надає інфраструктуру для взаємодії та співпраці користувачів. Користувачами, в залежності від контексту, можуть бути не лише люди, що працюють в системі, а й прикладні програми чи засоби самої системи. Ресурсом також може бути що завгодно, від online конференції чи прикладної програми до окремого вікна чи спільної змінної. Що

стосується рівнів, то вони є складовими фундаментального протоколу взаємодії учасників колаборативного середовища — рівневого контролю (floor control), детально описаного в працях [3; 5]. Володіти рівнем — означає мати привілейований доступ до відповідного ресурсу. Одному ресурсу можуть відповідати кілька рівнів для керування різними його фрагментами, але кожен рівень керує виключно одним ресурсом. Контролери кожного з цих елементів сукупно утворюють координаційний механізм, що забезпечує комфортну роботу користувачів колаборативної системи.

В рамках проекту, результати якого було оприлюднено на конференції УкрПРОГ2006 та надруковано у збірнику праць конференції [1], було побудовано моделі цих чотирьох складових колаборативної системи мовами скінченних автоматів та мереж Петрі. Мережні моделі було досліджено на обмеженість і активність та доведено, що всі вони мають ці властивості. В даній роботі розглядаються принципи побудови мережної моделі системи в цілому як сукупності цих чотирьох елементів та досліджуються одержані мережні моделі для систем з одним та двома користувачами.

### Побудова моделі системи з одним користувачем

Структуру зв'язків між складовими системи зображено на рис. 1.



Рис.1 Зв'язок між складовими системи

Як бачимо, всі операції з ресурсами, за винятком створення, користувач може виконувати лише тоді, коли займає відповідний рівень. Крім цього, хоча з малюнку цього явно не видно, всі свої дії користувач виконує в рамках певного сеансу зв'язку з іншими учасниками системи. В усьому іншому контролер сеансу працює окремо від інших, і його реакція на функціонування інших елементів системи проявляється лише у випадку падіння системи.

В процесі об'єднання мереж окремих складових в одне ціле постає низка проблем. По-перше, це забезпечення контролю за командами, які віддає користувач. Через недетерміновану природу мереж Петрі [2, С. 40—44] цілком вірогідна ситуація, коли користувач буде постійно віддавати системі одну й ту саму команду, наприклад, «створити ресурс», або ж віддаватиме команди у неправильній послідовності (спочатку «вилучити ресурс», потім «створити ресурс» і т. д.) Таким чином, прив'язки дозволу до віддачу певної команди до одержаного від системи повідомлення про виконання попередньої операції, як це передбачалося при побудові моделей окремих складових, виявилася недостатньою. Навіть якщо мережа матиме такий вигляд, як на рис. 2, перехід  $t_1$  все одно залишиться дозволеним.

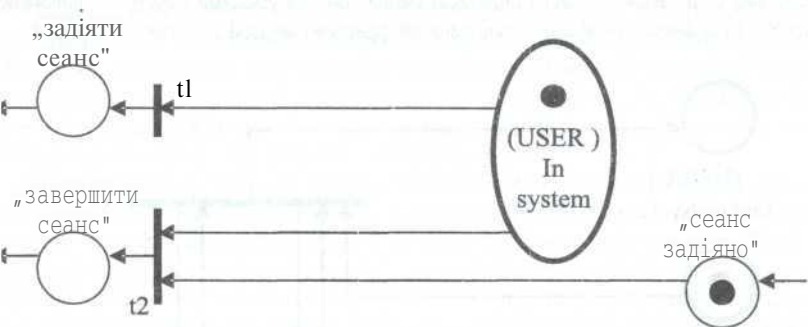


Рис.2. Неefективне блокування

Для розв'язання даної проблеми при побудові моделі системи в цілому ми взагалі відмовилися від команд та повідомлень, замінивши відповідні послідовності переходів безпосередніми зв'язками між станами контролерів, яким віддається команда, при цьому поточний стан підлеглого контролера визначає можливі команди, які йому уже віддати користувач. Тепер ситуація виглядатиме так:

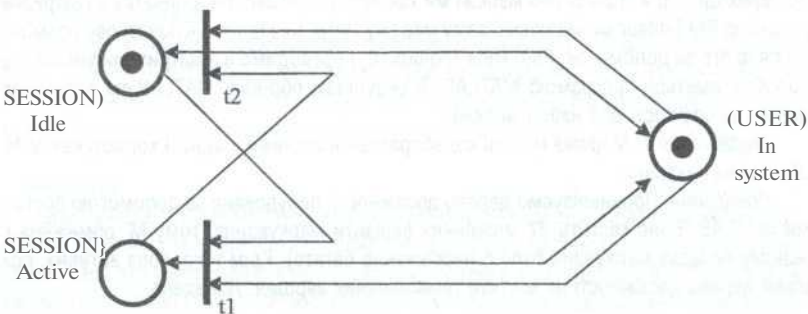


Рис. 3. Еfективне блокування

Очевидно, що в цьому випадку результатом спрацювання переходу  $t_1$  буде перехід контролера сеансу зі стану бездіяльності Idle до стану активності Active. Більш того, після цього перехід  $t_1$  перестає бути дозволеним, тому ситуація з повторенням команди активації сеансу неможлива.

Друга проблема, яку треба вирішити, пов'язана з виходом користувача із системи. Для цього було достатньо спрацювання переходу  $t_{3a}$  [1, рис. 2.6.1]. Коли ми розглядли поведінку користувача як окремої сутності, це не мало великого значення. Але для користувача як складової єдиного цілого подібна поведінка неприпустима, оскільки на момент виходу він може володіти якимось ресурсом, і доки він своєю командою його не звільнить, ніхто інший не зможе до нього доступитися. Тому на вихід користувача з системи слід також накласти додаткові умови. Такими умовами будуть а) звільнений рівень та б) завершений сеанс. Відповідний фрагмент мережі зображено на рис. 4.

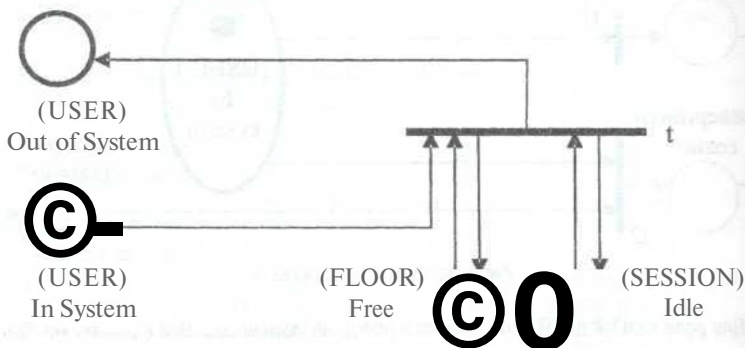


Рис. 4. Умова виходу користувача із системи

В результаті отримаємо мережу (позначимо її  $M_1$ ), яка складається з 21 позиції та 26 переходів. Для аналізу цієї мережі ми такі програмні засоби. Спочатку в графічний редактор PM Editeur ми малюємо саму мережу Петрі та задаємо її початкову розмітку. Після цього за допомогою Petri Nets Toolbox [6] переводимо її в матричний вигляд, що оброблятиметься програмою MATLAB. В результаті обробки MATLAB'OM отримуємо дерево досяжності для нашої мережі.

*Твердження 1.* Мережа моделі колаборативної системи з одним користувачем  $M_1$  обмежена і активна.

*Доведення.* Проаналізуємо дерево досяжності, побудовано за допомогою програми MATLAB. Воно містить 37 можливих варіанти маркування, тому  $M'$  обмежена (в іншому випадку маркувань було б нескінченно багато). Крім того, вона активна, оскільки дерево досяжності не містить термінальних вершин. Доведено.

## **Побудова моделі системи з двома користувачами**

При розширенні системи до двох користувачів треба зробити наступне. По-перше, в цій схемі користувачі конкуруватимуть між собою не лише з приводу ресурсу, але й з приводу сеансу. Зрозуміло, що активувати сеанс, а отже, і мати право його завершити, може лише один з них. Відповідно, треба ввести блок координації, який слідкуватиме за тим, щоб господарем сеансу був лише один з користувачів. Для цього слугуватимуть такі позиції як „не контролює сеанс” та „контролює сеанс” (для кожного користувача вони свої). Аналогічні позиції введемо для організації взаємовиключного доступу до рівня. Вважатимемо, що інший користувач, якщо він перебуває у системі, автоматично приєднується до вже активованого сеансу.

По-друге, вихід користувачів із системи треба регламентувати таким чином, щоб господар рівня або сеансу не міг покинути систему, попередньо не звільнивши рівень або не припинивши сеанс. При цьому господар сеансу не може його завершити, якщо інший користувач працює в цей час з ресурсом. Для забезпечення цього перехід „вихід із системи” буде дозволенним лише тоді, коли відповідний користувач одночасно „не контролюватиме сеанс” та „не контролюватиме ресурс”.

По-третє, для забезпечення взаємовиключної обробки запитів на активацію сеансу та зайняття рівня, а також для того, щоб контролер сеансу або рівня розрізняв, від кого саме з користувачів надійшов запит, для кожного з них вводяться позиції „претендує на сеанс” та „претендує на ресурс”, а для контролерів рівня і сеансу — окремі переходи для запитів кожного користувача (наприклад, „запит користувача 1 відхилено”, „запит користувача 2 відхилено” тощо). Також контролер рівня розрізнятиме, який саме користувач його зайняв, завдяки позиціям „Зайнятий Користувачем-1”, „Зайнятий Користувачем-2”, „Призупинений Користувачем-1” та „Призупинений Користувачем-2”.

Контролер ресурсу змін не зазнає, оскільки всі питання регламентування доступу вирішуватиме контролер рівня.

Отримана мережа (позначимо її M2) складається з 38 позицій та 42 переходів.

*Твердження 2.* Мережа моделі колаборативної системи з двома користувачами M2 обмежена і активна.

*Доведення.* Проаналізуємо дерево досяжності, побудовано за допомогою програми MATLAB. Воно містить 132 можливих варіанти маркування, тому наша мережа обмежена (в іншому випадку маркувань було б нескінченно багато). Крім того, вона активна, оскільки дерево досяжності не містить термінальних вершин. Доведено.

## **Висновки**

Запропонована нами модель колаборативного середовища є базисом як для досліджень властивостей вже існуючих систем організації дистанційної співпраці користувачів, так і для проєтування нових систем. Завдяки тому, що для її побудови були використані мережі Петрі, її можна розширювати за рахунок додавання нових користувачів та ресурсів, передбачення можливості одночасного існування двох або більше сеансів зв'язку. Крім того, для більш глибокого дослідження системи її можна перевести на мову часових або стохастичних мереж Петрі.

Як зазначалося вище, при побудові цієї моделі ми намагалися абстрагуватися від предметної області застосування колаборативної системи, конкретних політик, що регламентують взаємодію користувачів, механізмів реагування на помилки та їх виправлення. Знов-таки, до моделі можна додавати відповідні уточнення, розкриваючи, фактично, кожну позицію мереж M1 та M2. Головна мета, якої ми досягли створенням цієї моделі, є побудова надійного (без зациклювань та тупиків) механізму координації дій як зовнішніх користувачів, так і складових самої системи.

## ЛІТЕРАТУРА

1. Глибовець М.М., Гломозда Д.К. Формальна модель координаційно-орієнтованої мережі для колаборативної системи навчання // Проблеми програмування. — 2006. — № 2—3. Спец. вип. — С. 402—412.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. — М.: Мир, 1984. — 264 с.
3. Dommel H-P., Garcia-Luna-Aceves J.J. Floor control for multimedia conferencing and collaboration // Multimedia Systems. — Vol. 5. — Springer-Verlag, 1997. — P. 23—38.
4. Dommel H-P., Garcia-Luna-Aceves J.J. Networking Foundations for Collaborative Computing at Internet Scope // International ICSC Congress on Intelligent Systems and Applications, Symposium on Interactive and Collaborative Computing (ICC2000). — Woiiongong, Australia. — 2000.
5. Myers B.A., Chuang Yu Shan A., Tjandra M. et al. Floor Control in a Highly Collaborative Co-Located Task. — [Submitted for publication]. — May 22, 2006. — <[www.cs.cmu.edu/~pebbles/papers/pebblesfloorcontrol.pdf](http://www.cs.cmu.edu/~pebbles/papers/pebblesfloorcontrol.pdf)>.
6. Svadova M., Hanzalek Z. Matlab Toolbox for Petri Nets // INOVACE 2000. — Praha. — 2000. — P. 15.