УДК 681.3

Н.Н. ГЛИБОВЕЦ, Д.К. ГЛОМОЗДА

СЛОЖНОСТЬ ЗАДАЧИ ВЕРИФИКАЦИИ КООРДИНАЦИОННОГО МЕХАНИЗМА СИСТЕМЫ ПРОГРАММНОЙ ПОДДЕРЖКИ СОВМЕСТНОЙ СЕТЕВОЙ РАБОТЫ

Ключевые слова: *координация, сети Петри, вычислительная сложность, co-NP-полнота, программные агенты.*

Рассмотрим задачу верификации координационного механизма (КМ) для коллаборативной системы, построенной в рамках проекта по созданию модели системы программной поддержки коллаборативной среды на языке сетей Петри, начатого в работе [1]. Предложенная в [2] сетевая модель коллаборативной системы базируется на структуре коллаборативной среды, составляющие которой — сеансы, пользователи, общие ресурсы и уровни. С помощью таких уровней реализован протокол доступа к ресурсам. Схема сетевой модели системы приведена на рис. 1.

Коллаборативная система состоит из N пользователей, M сеансов, L ресурсов и координационного механизма — совокупности позиций и переходов сети Петри, которые связывают пользователей U, сеансы S и ресурсы R. КМ в рассматриваемой модели представляет собой блок, состоящий из контроллеров уровней (каждому уровню соответствует один ресурс) и механизма обеспечения взаимоисключения при создании сеанса. Он включает такие позиции сетевой модели.

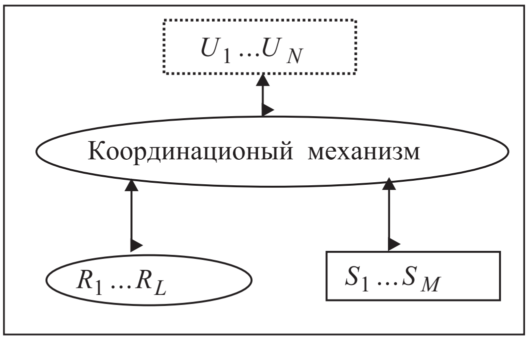

Рис. 1. Схема сетевой модели для коллаборативной системы

1. Блок обеспечения взаимоисключения при создании сеанса для каждого пользователя и каждого сеанса. В его состав входят позиции: «Пользователь U_i не является руководителем сеанса S_j», «Пользователь U_i является руководителем сеанса S_j», «Пользователь U_i запрашивает разрешение на создание сеанса S_j». Переходы сети Петри: «Пользователь U_i хочет начать сеанс S_j», «Пользователь U_i хочет завершить сеанс S_j», «Пользователю U_i разрешено создать сеанс S_j», «Пользователю U_i не разрешено создать сеанс S_j» (при этом $i \in 1 \ldots N$; $j \in 1 \ldots M$).

2. Контроллер уровня. Позиции: «Уровень F_k свободен», «Уровень F_k обрабатывает запрос», «Уровень F_k в состоянии ожидания», «Предоставить ресурс R_k», «Ресурс R_k предоставлен», «Освободить ресурс R_k», «Ресурс R_k освобожден», «Удалить ресурс R_k», «Ресурс R_k удален» (при этом $k \in 1 \ldots L$).

3. Блок обеспечения взаимоисключения доступа каждого пользователя к каждому уровню. Его позиции: «Пользователь U_i не является держателем уровня F_k», «Пользователь U_i запрашивает доступ к уровню F_k», «Пользователь U_i является держателем уровня F_k», «Уровень F_k активно используется пользователем U_i», «Уровень F_k пассивно используется пользователем U_i». Переходы: «Пользователь U_i хочет получить доступ к уровню F_k», «Пользователю U_i отказано в доступе к уровню F_k», «Пользователю U_i разрешен доступ к уровню F_k», «Уровень F_k занят пользователем U_i», «Пользователь U_i приостанавливает использование уровня F_k», «Пользователь U_i возобновляет использование уровня F_k», «Пользователь U_i хочет удалить ресурс R_k», «Ресурс R_k удален пользователем U_i», «Пользователь U_i хочет освободить уровень F_k», «пользователь U_i освободил уровень F_k».

В общем случае сеть Петри, моделирующая работу координационного механизма рассматриваемой коллаборативной системы, имеет:

— $3 \cdot N \cdot M$ позиций и $4 \cdot N \cdot M$ переходов, регламентирующих взаимоисключающее создание сеансов;

— $9 \cdot L$ позиций, соответствующих контроллерам уровней;

— $5 \cdot N \cdot L$ позиций и $10 \cdot N \cdot L$ переходов, соответствующих связям между контроллерами уровней и пользователями.

Рассмотрим задачу верификации КМ и найдем ее вычислительную сложность. На неформальном уровне задача верификации КМ заключается в определении пригодности данного КМ к использованию в коллаборативной системе или, другими словами, в проверке соответствия КМ определенным спецификациям, регламентирующим принципы и результаты работы системы. Прежде чем дать формальную постановку этой задачи, введем следующие определения.

Недопустимыми будем называть такие состояния системы, в которых два или более пользователей одновременно являются руководителями[1] одного и того же сеанса и/или держателями одного и того же уровня. Все другие состояния будем называть допустимыми. Тогда на формальном уровне задачу верификации КМ можно определить таким образом.

Дано: N пользователей, L уровней (каждому уровню соответствует один отдельный ресурс), M сеансов, координационный механизм.

Ответ: если для каждого достижимого варианта маркировки сети Петри, соответствующей КМ, выполняется условие допустимости маркировки (т.е. маркировка соответствует допустимому состоянию), то КМ пригоден. В противном случае — не пригоден.

В таком виде задача верификации КМ подобна задаче верификации агентов, рассмотренной в работе [3], где описывается зависимость сложности задачи от характеристик среды, в которой действует агент, и от сложности спецификации задачи Ψ, определенной как предикат над множеством пробегов **R** агента в системе

$$\Psi : \mathbf{R} \to \{\text{true}, \text{false}\}.$$

Пробег агента в системе представляет собой последовательность состояний среды и действий агента, в результате чего происходит смена состояний среды.

КМ можно сопоставить с агентом, выполняющим задачу поддержки, т.е. удерживающим среду в одном из допустимых состояний.

В случае коллаборативной среды возможных пробегов бесконечно много. Но для проверки надежности данного координационного механизма достаточно определить возможность каждого пользователя поочередно в качестве руководителя каждого из возможных сеансов получить доступ к каждому имеющемуся ресурсу. При этом целесообразно ввести дополнительное условие, согласно которому только руководитель сеанса может получить доступ к ресурсу, поскольку присоединение к уже активированному сеансу не является взаимоисключающим. Таким образом, все множество возможных пробегов сводится к одному конечному пробегу (условие остановки — все пользователи в качестве руководителей всех возможных сеансов воспользовались всеми имеющимися ресурсами). Такой пробег $\mathbf{R}'$ удобно представить в виде последовательности определенных этапов. Этапу пробега соответствует такое состояние системы, когда все составляющие среды используются максимально полно. Это означает, что если, например, $N = M = L$, то каждый пользователь руководит конкретно одним своим сеансом и удерживает конкретно один уровень. В действительности все другие варианты, когда количество пользователей, сеансов и уровней различно ($N = n$, $M = m$, $L = l$; $(n \neq m) \neq (l \neq n) \neq (m \neq l)$), можно не рассматривать отдельно, поскольку они сводятся к случаю, когда $N = M = L = \min(n, m, l) = n$ (при $n \geq 2$). Поясним это.

1. За контроль над избыточными сеансами и/или уровнями пользователи не конкурируют, поэтому нет необходимости отдельно координировать их использование.

2. Если пользователей больше, чем сеансов, то на каждом этапе работа КМ сводится к согласованию действий m пользователей (по одному на сеанс). Те пользователи, кому не хватило сеанса для КМ, находятся вне системы, поскольку в соответствии с дополнительным условием ресурсы доступны исключительно руководителям сеансов, поэтому нет необходимости координировать действия этих пользователей.

3. Условие $n \geq 2$ следует соблюдать, чтобы в результате сокращения не получить вырожденный случай с одним пользователем, одним сеансом и одним уровнем, поскольку тогда уже ни о какой конкуренции и координации не будет идти речь. Если имеется лишь один сеанс и/или уровень, то количество пользователей приравнивается к двум (наименьшее число, при котором имеется конкуренция за ресурс).

Для построения спецификации Ψ введем такие вспомогательные предикаты:

— РС (i, j) — значение «истина» $\not\subset$ в сети Петри в позиции «Пользователь U_i является руководителем сеанса S_j» стоит фишка, иначе «ложь»;

— ДУ (i, k) — значения «истина» $\not\subset$ в сети Петри в позиции «Пользователь U_i является держателем уровня F_k» стоит фишка, иначе «ложь»;

— КСВИ $(j) \equiv \forall i_1 \; \forall i_2 \, (\neg\, (\text{РС}\,(i_1, j) \wedge \text{РС}\,(i_2, j) \wedge (i_1 \neq i_2)))$, где $i_1, i_2, j \in \in 1 \ldots n$. «Истина» означает, что контроль сеанса взаимоисключающий. Выражение без кванторов обозначим КСВИ$'(j)$;

— УУВИ $(k) \equiv \forall i_1 \; \forall i_2 \, (\neg(\text{ДУ}\,(i_1, k) \wedge \text{ДУ}\,(i_2, k) \wedge (i_1 \neq i_2)))$, где i_1, i_2, $k \in 1 \ldots n$. «Истина» означает, что удерживание уровня взаимоисключающее. Выражение без кванторов обозначим УУВИ$'(k)$;

— Доп_состоние (КМ) $\equiv \forall j \, \forall k$ (КСВИ $(j) \wedge$ УУВИ $(k)) \equiv \forall i_1 \; \forall i_2 \; \forall j \, \forall k$ (КСВИ$'(j) \wedge$ УУВИ$'(k)$).

Тогда предикат $\Psi(\mathbf{R}')$ будет иметь вид

$$\Psi(\mathbf{R}') = \begin{cases} \text{истина, если предикат Доп\_состоние (КМ)} \\ \qquad \text{выполнется для всех маркировок пробега;} \\ \text{ложь} - \text{inache.} \end{cases} \quad (1)$$

Для дальнейшей работы целесообразно перейти от предиката к квантифицированной булевой формуле (КБФ):

— РС (i, j) заменяется булевой переменной $x_{i,j}$;

— ДУ (i, k) заменяется булевой переменной $y_{i,k}$;

— КСВИ (j) заменяется формулой

$$\forall x_{1,1} \; \forall x_{1,2} \ldots \forall x_{1,n} \; \forall x_{2,1} \ldots \forall x_{n,n} [\rceil(x_{1,1} \wedge x_{2,1}) \wedge \rceil(x_{1,1} \wedge x_{3,1}) \wedge \ldots]; \quad (2)$$

— УУВИ (k) заменяется формулой

$$\forall y_{1,1} \; \forall y_{1,2} \ldots \forall y_{1,n} \; \forall y_{2,1} \ldots \forall y_{n,n} [\rceil(y_{1,1} \wedge y_{2,1}) \wedge \rceil(y_{1,1} \wedge y_{3,1}) \wedge \ldots]; \quad (3)$$

— Доп_состояние(КМ) заменяется формулой

$$\forall x_{1,1} \; \forall x_{1,2} \ldots \forall x_{1,n} \; \forall x_{2,1} \ldots \forall x_{n,n} \forall y_{1,1} \forall y_{1,2} \ldots \forall y_{1,n} \forall y_{2,1} \ldots \forall y_{n,n} [\rceil(x_{1,1} \wedge x_{2,1}) \wedge \rceil(x_{1,1} \wedge x_{3,1}) \wedge \ldots] \wedge [\rceil(y_{1,1} \wedge y_{2,1}) \wedge \rceil(y_{1,1} \wedge y_{3,1}) \wedge \ldots]. \quad (4)$$

Например, при $n = 2$ соответствующие формулы имеют вид:

$$\forall x_{1,1} \ \forall x_{1,2} \ \forall x_{2,1} \ \forall x_{2,2} [\rceil(x_{1,1} \wedge x_{2,1}) \wedge \rceil(x_{1,2} \wedge x_{2,2})];$$

$$\forall y_{1,1} \ \forall y_{1,2} \ \forall y_{2,1} \ \forall y_{2,2} [\rceil(y_{1,1} \wedge y_{2,1}) \wedge \rceil(y_{1,2} \wedge y_{2,2})];$$

$$\forall x_{1,1} \ \forall x_{1,2} \ \forall x_{2,1} \ \forall x_{2,2} \ \forall y_{1,1} \ \forall y_{1,2} \ \forall y_{2,1} \ \forall y_{2,2} [\rceil(x_{1,1} \wedge x_{2,1})$$

$$\wedge \rceil(x_{1,2} \wedge x_{2,2})] \wedge [\rceil(y_{1,1} \wedge y_{2,1}) \wedge \rceil(y_{1,2} \wedge y_{2,2})].$$

Согласно [3, с. 120, 121] задача верификации агентов для $\sum_u^p$ -сложных спецификаций задачи является по своей вычислительной сложности $\prod_{u+1}^p$-полной. Следует отметить полиномиальные иерархии классов $\sum_u^p$ и $\prod_u^p$. Они охватывают бесчисленное множество классов, каждому из которых соответствует определенное значение $u \in N$ (для $u < 0$ $\sum_u^p = \varnothing$). Предполагается, что спецификацию Ψ можно представить такой машиной Тьюринга T_Ψ, которая принимает на вход только пробеги, удовлетворяющие эту спецификацию. Любую $\sum_u^p$ -спецификацию можно задать альтернирующей машиной Тьюринга, решающей задачу за полиномиальное время с использованием не более *u* переходов между состояниями существования и состояниями всеобщности (соответствующим кванторам существования и кванторам всеобщности). Таким образом, $\sum_0^p \equiv P$, $\sum_1^p \equiv NP$.

Каждый класс сложности $O(t(n))$ представляет собой множество языков, которые можно решить за время $O(t(n))$ на соответствующей машине Тьюринга. В частности, класс *NP* состоит из языков, которые решаются за полиномиальное время на недетерминированных машинах Тьюринга. В общем случае $\prod_u^p \equiv$ co-$\sum_u^p$. Однако непосредственно результат из работы [3] применить к нашей задаче нельзя, так как спецификация данной задачи является co-*NP*-полной (или $\prod_1^p$), поскольку содержит исключительно кванторы всеобщности.

Согласно [4, глава 10] co-*NP* есть класс сложности, которому принадлежат задачи с кратким отверганием, в отличие от класса *NP*, объединяющего задачи с кратким подтверждением. Поскольку задача тавтологичности булевого выражения (boolean validity problem) является co-*NP*-полной, наша спецификация задачи (4) также co-*NP*-полная. Поэтому для определения сложности задачи верификации КМ требуется доказать лемму.

Лемма 1. Задача верификации агентов для co-*NP*-полных спецификаций задачи будет по своей вычислительной сложности co-*NP*-полной.

Доказательство. co-*NP*-сложность непосредственно вытекает из сложности спецификации задачи. Чтобы доказать co-*NP*-полноту, приведем задачу определения истинности КБФ, содержащей только кванторы всеобщности, к задаче верификации агентов. В результате получим

$$\forall \bar{x}_1 \ \forall \bar{x}_2 \ \ldots \ \forall \bar{x}_n \ \chi(\bar{x}_1, \ldots, \bar{x}_n), \tag{5}$$

где каждый $\bar{x}_i$ — конечный набор булевых переменных; $\chi(\bar{x}_1, \ldots, \bar{x}_n)$ — формула логики высказываний на множестве булевых переменных $\bar{x}_1, \ldots, \bar{x}_n$.

Задача верификации агентов на основе формулы (5) строится таким образом. Пусть $\bar{x}_1 = x_1^1, x_1^2, \ldots, x_1^m$ — самый внешний набор квантифицированных переменных. Каждой из этих переменных x_1^i будут соответствовать два возможных состояния среды: $e_{x_1^i}$ и $e_{\neg x_1^i}$, соответствующих значениям «истина» и «ложь» переменной x_1^i. Исходное состояние среды обозначим e_0. Среда позволяет агенту выполнять лишь действие a_0, и на *i*-е выполнение этого действия среда отвечает переходом в состояние $e_{x_1^i}$ либо $e_{\neg x_1^i}$. После *m*-го выполнения действия a_0 пробег завершается. Таким образом, каждый пробег приписывает каждой всеобщно квантифицированной переменной из множества $\bar{x}_1 = x_1^1, x_1^2, \ldots, x_1^m$ свое значение, при этом множество всех возможных пробегов соответствует множеству всех возможных комбинаций значений этих переменных. Для данного пробега r $\chi(\bar{x}_1, \ldots, \bar{x}_n)[r / \bar{x}_1]$ — булева формула, полученная из формулы $\chi(\bar{x}_1, \ldots, \bar{x}_n)$ путем замены каждой переменной x_1^i ее значением («истина» либо «ложь»), которое эта переменная приобрела в рамках пробега *r*. Тогда спецификация задачи Ψ будет иметь вид

$$\Psi(r) = \begin{cases} \text{истина, если со} - NP - \text{сложна формула} \\ \qquad \forall \bar{x}_2 \ldots \forall \bar{x}_n \; \chi(\bar{x}_1, \ldots, \bar{x}_n)[r / \bar{x}_1] \\ \qquad \text{приобретает знаcheние «истина»;} \\ \text{ложь} - \text{в противном слу}c\text{hae.} \end{cases} \tag{6}$$

Входная формула (5) приобретает значение «истина» тогда, когда все пробеги агента удовлетворяют спецификации (6). Поскольку приведение полиномиально, задача верификации по своей вычислительной сложности является со-*NP*-полной, что и требовалось доказать. □

Далее определим сложность задачи верификации КМ.

Теорема 1. Задача верификации КМ по своей вычислительной сложности является со-*NP*-полной.

Доказательство. Спецификация задачи верификации КМ (1) по своей вычислительной сложности со-*NP*-полная. Согласно лемме 1 задача верификации агентов, подобная задаче верификации КМ, является по своей вычислительной сложности со-*NP*-полной в случае со-*NP*-полной спецификации задачи. Следовательно, задача верификации КМ по своей вычислительной сложности является со-*NP*-полной, что и требовалось доказать. □

Таким образом, полученный теоретический результат позволяет отнести задачу верификации координационного механизма коллаборативной системы к уже известному и изученному классу со-*NP*-полных задач. Это важный результат для исследования общей модели системы программной поддержки совместной сетевой работы с произвольным количеством пользователей, сеансов и ресурсов, в частности коллаборативного сотрудничества в сети ИНТЕРНЕТ.

СПИСОК ЛИТЕРАТУРЫ

1. Глибовець М.М., Гломозда Д.К. Формальна модель координаційно-орієнтованої мережі для колаборативної системи навчання // Проблеми програмування. — 2006. — № 2–3. — С. 402–412.
2. Глибовець М.М. Моделі і методи створення і супроводу високопродуктивного розподіленого навчального середовища: Автореф. дис ... д-ра фіз.-мат. наук: 01.05.03 / НАН УкраВни; Ін-т кібернетики ім. В.М. Глушкова. — К., 2006. — 36 с.
3. Wooldridge M., Dunne P.E. The computational complexity of agent verification / Intelligent Agents VIII: Proceedings of the Eigth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001). — Berlin: Springer-Verlag, 2002. — P. 115–127.
4. Papadimitriou C.H. Computational Complexity. — New York: Addison-Wesley, 1994. — 523 p.

Поступила 09.11.2007