УДК 519.711

В.А. БАШКИН, И.А. ЛОМАЗОВА

МОДЕЛИРОВАНИЕ МУЛЬТИАГЕНТНЫХ СИСТЕМ С ПОМОЩЬЮ ОБОБЩЕННЫХ СЕТЕЙ АКТИВНЫХ РЕСУРСОВ[1]

Ключевые слова: *моделирование, верификация, мультиагентные системы, сети Петри.*

ВВЕДЕНИЕ

Одним из наиболее распространенных способов моделирования сложных распределенных систем являются сети Петри. Это достаточно выразительная модель параллелизма, обладающая в то же время значительным набором разрешимых свойств. В частности, для обыкновенных сетей Петри разрешимы проблемы достижимости, останова, живости, ограниченности, покрытия. Еще одной причиной популярности сетей Петри является простота и наглядность описания параллелизма, а также удобное графическое представление модели.

Однако синтаксис сетей Петри достаточно статичен и не всегда пригоден для моделирования сложных мультиагентных систем, поэтому разрабатываются формализмы, в которых тем или иным способом более явно выделяется понятие агента [1–3].

В работе [4] представлен новый способ моделирования распределенных систем — сети активных ресурсов (АР-сети). Данный формализм обладает той же выразительной мощностью, что и обыкновенные сети Петри, однако в нем используется иной принцип моделирования: вместо разделения компонентов модели (вершин графа) на агенты и ресурсы (переходы и позиции) вводится разделение способов взаимодействия (дуг графа) на производство и потребление. Такой подход позволяет получить модель, в которой более явно выделены не типы компонентов системы, а способы их взаимодействия между собой. В ряде случаев это позволяет более компактно и наглядно формализовать семантические свойства мультиагентных систем со сложным поведением агентов.

В данной работе рассматриваются возможности обобщения формализма АР-сетей. Используется понятие метаструктур — формального способа описания новых классов обобщенных АР-сетей. Метаструктура задает набор способов взаимодействия активных ресурсов (видов/цветов дуг) и правило срабатывания отдельного узла (вершины сети). Таким образом, метаструктуры можно рассматривать как способ описания новых инструментов моделирования (обобщений АР-сетей).

В терминах метаструктур определяются новые формализмы — параметризованные АР-сети, двухуровневые АР-схемы и двухуровневые АР-сети. В параметризованных АР-сетях вводятся выражения на дугах (аналог высокоуровневых сетей Петри [5]), что позволяет существенно упростить графическое представление плоских систем с большим количеством однотипных агентов. Вложенные АР-сети сочетают в себе ключевые свойства двух формализмов: простоту синтаксиса АР-сетей и композициональность вложенных сетей Петри [3]. Это позволяет удобно моделировать достаточно сложные мультиагентные системы с модульной и иерархической структурой.

[1] Работа выполнена при поддержке РФФИ (09-01-00277) и ФАНИ (02.740.11.0207).

В качестве модельного примера рассматривается классическая «проблема обедающих философов» (например, [5]). Исследуется выразительная мощность новых формализмов. В частности, доказано, что параметризованные АР-сети и двухуровневые АР-схемы эквивалентны обыкновенным сетям Петри, двухуровневые АР-системы эквивалентны вложенным сетям Петри.

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Пусть Nat — множество неотрицательных целых чисел, X — непустое множество. Мультимножеством M над множеством X называется функция $M: X \to \mathrm{Nat}$. Мощность мультимножества запишем $|M| = \sum_{x \in X} M(x)$. Числа $\{M(x) \mid x \in X\}$ называются коэффициентом мультимножества, коэффициент $M(x)$ определяет число экземпляров элемента x в M.

Операции и отношения теории множеств естественно расширяются на конечные мультимножества. Определим также дополнительные операции — сложение и вычитание мультимножеств. Пусть M_1, M_2 — конечные мультимножества над X. Полагаем $\forall x \in X$ $(M_1 + M_2)(x) =_{\mathrm{def}} M_1(x) + M_2(x)$; $(M_1 - M_2)(x) =_{\mathrm{def}} M_1(x) \dot{-} M_2(x)$ (где $\dot{-}$ — усеченное вычитание).

Сетью Петри называется набор $N = (P, T, F)$, где P — конечное множество позиций; T — конечное множество переходов, $P \cap T = \varnothing$; $F: (P \times T) \cup \cup (T \times P) \to \mathrm{Nat}$— функция инцидентности (конечное множество дуг).

Разметкой (состоянием) сети N называется функция $M: P \to \mathrm{Nat}$, сопоставляющая каждой позиции сети некоторое натуральное число (или ноль). Разметка может рассматриваться как мультимножество над множеством позиций сети.

Графически сеть Петри изображается как двудольный ориентированный граф. Вершины-позиции изображаются кружками и характеризуют локальные состояния сети, вершины-переходы — прямоугольниками и соответствуют действиям. Дуги соответствуют элементам F. Позиции могут содержать маркеры (фишки), изображаемые черными точками. При разметке M в каждую позицию p помещается $M(p)$ фишек.

Переход $t \in T$ активен при разметке M, если $\forall p \in P$ $M(p) \geq F(p, t)$ (все входные позиции содержат достаточное количество фишек).

Активный при разметке M переход t может сработать, порождая при этом новую разметку M', где $\forall p \in P$ $M'(p) =_{\mathrm{def}} M(p) - F(p, t) + F(t, p)$.

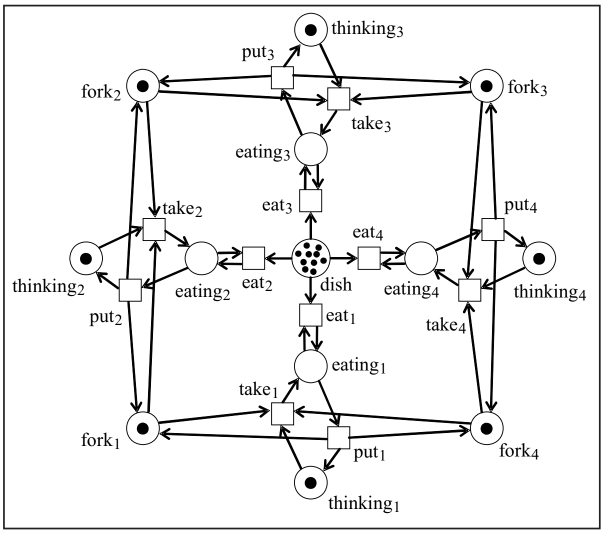

Рис. 1. Сеть Петри (четыре обедающих философа)

Рассмотрим в качестве примера «проблему обедающих философов» (рис. 1), которая получила статус классической и упоминается в разных вариантах во многих учебниках. Это объясняется тем, что в ней при кажущейся простоте имеются все элементы распределенных систем: выбор, конфликт, распараллеливание и синхронизация. Философы (в данном случае их четверо) сидят за столом, посередине общее блюдо — спагетти (dish). Между каждыми двумя философами на салфетке лежит вилка (fork). Есть спагетти можно только одновременно двумя вилками, поэтому соседи потенциально конкурируют. Каждый философ может находиться в одном из двух состояний: он либо думает (thinking), либо ест (eating). Переходы между состояниями — take (взял обе вилки) и put (вернул на место). В состоянии eating он также может выполнить действие eat (взять спагетти).

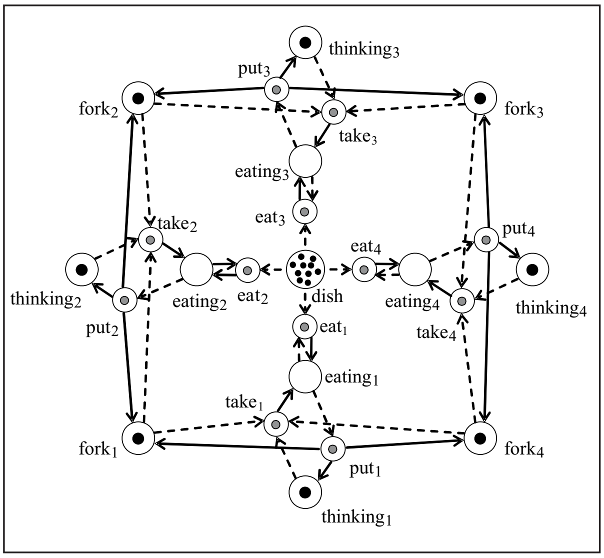

Рис. 2. АР-сеть (четыре обедающих философа)

Сетью активных ресурсов [4] назовем набор $AR=(V,I,O)$, где V — конечное множество вершин; $I:(V\times V)\to \mathrm{Nat}$ — множество потребляющих дуг; $O:(V\times V)\to \mathrm{Nat}$— множество производящих дуг.

Графически вершины сети изображаются кружками, потребляющие дуги — пунктирными стрелками, производящие дуги — непрерывными стрелками (рис. 2).

Разметкой сети AR называется функция вида $M:V\to \mathrm{Nat}$.

Ресурс (агент) в вершине $v\in V$ активен при разметке M, если $M(v)>0$ (узел v непустой) и $\forall w\in V\;\; M(w)\ge I(w,v)$ (в потребляемых узлах содержится достаточное количество фишек). Активный при разметке M ресурс v может сработать, порождая при этом новую разметку M', где $\forall w\in V\;\; M'(w)=_{\mathrm{def}}\; M(w)-I(w,v)+O(v,w)$.

Таким образом, в срабатывании ресурса участвуют его входные и выходные дуги, в трансформации ресурса (изменении разметки соответствующей вершины) — потребляющие и производящие его дуги.

Пример тривиального варианта моделирования приведен на рис. 2. В данном случае позиции и переходы сети Петри просто заменили на соответствующие узлы АР-сети. Однако синтаксис АР-сетей предоставляет и некоторые новые возможности. Например, поведение отдельного философа может быть промоделировано еще двумя способами (рис. 3).

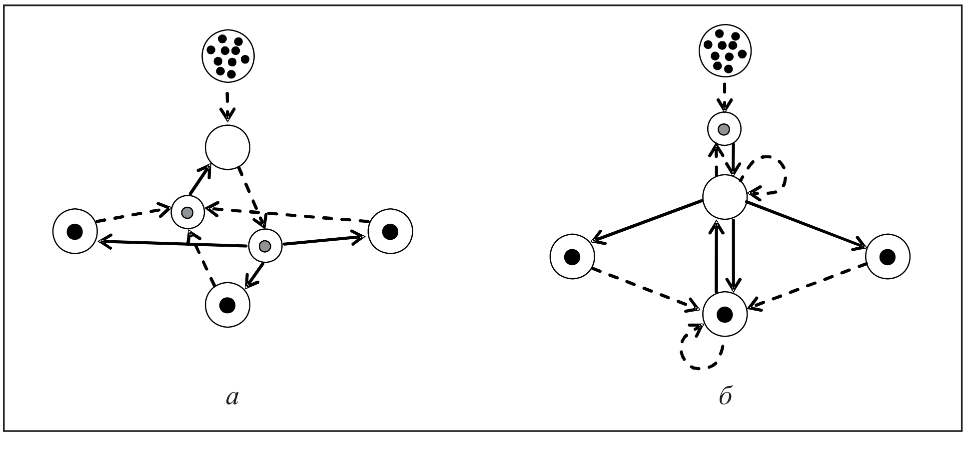

Рис. 3. Варианты моделирования поведения отдельного философа: несамостоятельная смена состояния агента (философа), самостоятельное потребление ресурса (спагетти) (*а*); самостоятельная смена состояния, несамостоятельное потребление (*б*)

Сети активных ресурсов по выразительной мощности совпадают с сетями Петри [4], но при этом в них добавляется возможность «двойного» использования ресурса (фишки): не только в качестве пассивного ресурса, но и в качестве активного агента. Это чисто синтаксическое расширение полезно на практике при моделировании систем с динамической структурой (например, сетей веб-сервисов). В настоящее время изучен ряд способов повышения выразительности и удобства моделирования сетями активных ресурсов. В частности, предложен ряд естественных расширений АР-сетей до универсальных систем, а также упрощений их синтаксиса без потери выразительности [6].

ОБОБЩЕННЫЕ АР-СЕТИ

Одно из основных отличий АР-сетей от сетей Петри — более наглядное разграничение функций дуг (потребление и производство). Фактически дуги в сетях Петри также распадаются на два класса (в силу двудольности графа), однако в АР-сетях типы дуг получили собственные обозначения. Такой чисто синтаксический прием также позволил избавиться от разделения вершин на позиции и переходы, что существенно изменило возможности моделирования.

Рассмотрим дальнейшее естественное обобщение синтаксиса АР-сетей, полученное введением произвольных типов дуг.

Метаструктурой назовем набор $\Lambda = (\Gamma, \Phi)$, где $\Gamma = \{\Re_1, \ldots, \Re_k\}$ — конечное множество типов дуг, Φ — ограничения на структуру графа и правила срабатывания узла.

Обобщенной АР-сетью метаструктуры Λ (сокращенно Λ-сетью) назовем набор $G = (V, R)$, где V — конечное множество вершин, $R: (V \times V \times \Gamma) \to \mathrm{Nat}$ — множество кратных дуг, помеченных типами из Γ.

В данном определении не указан конкретный способ задания правила срабатывания узла. Для известных типов дуг (производящих, потребляющих, ингибиторных, обнуляющих и т.д.) правила могут описываться в стандартной терминологии сетей Петри, однако возможны и другие способы.

ПАРАМЕТРИЗОВАННЫЕ («ВЫСОКОУРОВНЕВЫЕ») АР-СЕТИ

Рассмотрим обобщение, при котором тип дуги определяется приписанным ей параметрическим выражением. Похожий прием используется при определении так называемых «высокоуровневых сетей Петри» (например, [5]).

Пусть Const — конечное множество констант, Var — конечное множество переменных, Func — конечное множество всюду определенных функций из Const в Const. Рассмотрим простой язык L, определяемый следующим образом:

1) $\mathrm{Const} \subseteq L$, $\mathrm{Var} \subseteq L$;

2) если $v \in \mathrm{Var}$ и $f \in \mathrm{Func}$, то $f(v) \in L$;

3) если $E_1 \in L$ и $E_2 \in L$, то $E_1 + E_2 \in L$.

Таким образом, выражения построены с помощью операции сложения из констант, переменных и вызовов функций от переменных.

Означивание $b: \mathrm{Var} \to \mathrm{Const}$ сопоставляет всем переменным конкретные значения констант. Заметим, что при этом могут быть однозначно вычислены значения всех функций из Func. Таким образом, выражение E при означивании b (обозначается $E[b]$) определяет мультимножество констант.

Константам в данной сети будут соответствовать различные виды фишек. Переменные и функции будут играть роль параметров в выражениях на дугах.

Пусть $\mathrm{Labels} \subseteq L$ — некоторое конечное подмножество выражений языка L, используемых для обозначения дуг (для дуги v ее метку обозначим $\mathrm{Label}(v)$), cons и prod — обычные типы дуг АР-сети — дуги потребления и производства ресурсов.

Параметризованной называется метаструктура $\Lambda^{\mathrm{par}} = (\{\mathrm{cons}, \mathrm{prod}\} \times \times \mathrm{Labels}, \Phi^{\mathrm{par}})$, где Φ^{par} определяет следующее правило поведения:

1) фишками являются константы, разметка определяется как M: $(V \times \mathrm{Const}) \to \mathrm{Nat}$;

2) ресурс v активен при разметке M и означивании b, если $M(v) > 0$ (узел v непустой) и $\forall w \in V$, $c \in \mathrm{Const}$ $M(w, c) \geq (\mathrm{Label}(I(w, v))[b])(c)$ (в потребляемых узлах достаточно нужных фишек-констант). Активный ресурс v может сработать, порождая разметку M', где $\forall w \in V$, $c \in \mathrm{Const}$ имеем $M'(w, c) =_{\mathrm{def}} M(w, c) - -(\mathrm{Label}(I(w, v))[b])(c) + (\mathrm{Label}(O(v, w))[b])(c)$;

3) если при данной разметке возможно несколько означиваний с активными ресурсами, то выбор конкретного означивания недетерминирован.

Параметризованные сети предоставляют новые возможности компактного моделирования систем с большим количеством однотипных агентов. Пример приведен на рис. 4, где $\mathrm{Const} = \{\bullet, \mathrm{ph}_1, \mathrm{ph}_2, \mathrm{ph}_3, \mathrm{ph}_4, \mathrm{f}_1, \mathrm{f}_2, \mathrm{f}_3, \mathrm{f}_4\}$, $\mathrm{Var} = \{x\}$, $\mathrm{Func} = \{l, r\}$, $l(\mathrm{ph}_1) = r(\mathrm{ph}_2) = \mathrm{f}_1$, $l(\mathrm{ph}_2) = r(\mathrm{ph}_3) = \mathrm{f}_2$, $l(\mathrm{ph}_3) = r(\mathrm{ph}_4) = \mathrm{f}_3$, $l(\mathrm{ph}_4) = r(\mathrm{ph}_1) = \mathrm{f}_4$. Здесь параметризация существенно уменьшила размер графической модели.

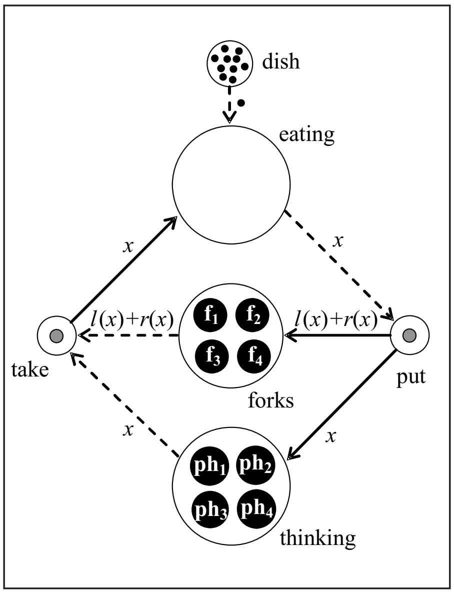

Рис. 4. Параметризованная АР-сеть

Теорема 1. Класс параметризованных АР-сетей по выразительной мощности совпадает с классом обыкновенных сетей Петри.

Доказательство. Обыкновенные сети Петри эквивалентны обыкновенным АР-сетям [4], которые являются частным случаем параметризованных (при $\mathrm{Const} = \{\bullet\}$). Следовательно, класс сетей Петри вкладывается в класс параметризованных АР-сетей.

Для доказательства обратного вложения достаточно показать, что любая параметризованная АР-сеть может быть преобразована в эквивалентную обыкновенную АР-сеть (сеть Петри). Рассмотрим следующее преобразование: заменим в исходной сети каждую вершину на конечное множество вершин, каждая из которых соответствует конкретной константе. Затем дополнительно размножим каждую из полученных вершин по числу допустимых означиваний переменных (это число также конечно в силу конечности множеств Const, Var и Func). Затем очевидным образом соединим вершины дугами, имитируя преобразования разметки при соответствующих означиваниях. □

ДВУХУРОВНЕВЫЕ АР-СХЕМЫ

Для моделирования сложных иерархических систем используются различные виды вложенных сетей Петри, в которых фишки сами могут быть сетями [3]. При этом для синхронизации уровней (слоев) используются или переходы синхронизации (срабатывающие только одновременно на двух соседних уровнях), или охранные функции на переходах верхнего слоя, получающие в качестве параметров «значения» используемых вложенных фишек. Заметим, что в обоих случаях одним из инициаторов события синхронизации является верхний слой модели — именно там срабатывает переход.

Рассмотрим новый способ моделирования иерархических систем, при котором синхронизация нижнего и верхнего (внутреннего и внешнего) слоев системы осуществляется снизу. Объектная сеть взаимодействует с системной посредством потребления и производства системных ресурсов.

В простейшем случае внешний уровень не обладает собственным поведением (событиями), а служит только схемой взаимодействия вложенных объектов (хранилищем общих ресурсов и «картой взаимосвязей»).

Пусть Syn — конечное множество портов, где $\mathrm{cons} \notin \mathrm{Syn}$, $\mathrm{prod} \notin \mathrm{Syn}$.

Схемной называется метаструктура $\Lambda^{\mathrm{sch}} = (\mathrm{Syn}, \Phi^{\mathrm{sch}})$, где Φ^{sch} определяет следующее ограничение структуры сети: граф является двудольным, все множество V распадается на два непересекающихся класса — V_{res} и V_{ag} «ресурсных» и «агентных» вершин. В агентных вершинах будут находиться объекты (вложенные сети), в ресурсных — системные ресурсы (обычные фишки, не обладающие собственным поведением). Дуги могут вести только от агентных узлов к ресурсным (фактически это ребра, показывающие, какие типы ресурсов могут потребоваться агентам данного типа).

Объектной называется метаструктура $\Lambda^{\mathrm{o}} = (\{\mathrm{cons}, \mathrm{prod} \cup \mathrm{Syn}, \Phi^{\mathrm{o}}\})$, где Φ^{o} определяет следующие расширенные правила:

1) дуги типов cons и prod участвуют в срабатывании узлов обычным образом по правилам АР-сетей;

2) порт (элемент Syn) не является в полном смысле слова дугой: он инцидентен ровно одному узлу сети, т.е. либо в него входит, либо из него выходит;

3) узел v объектной сети, находящейся в агентном узле W системной схемы, активен, если для любого его входного порта кратности k в системном слое найдется набор соответствующих (начинающихся в W) портов того же типа. При этом суммарная разметка соответствующих связанных с W ресурсных узлов (с учетом кратности связывающих портов) не меньше k. Активный узел v может сработать, при этом в системной схеме потребляются ресурсы в количестве, соответствующем кратности входящих в v портов, и производятся в количестве, соответствующем кратности выходящих из v портов.

Размеченной двухуровневой АР-схемой называется набор обобщенных АР-сетей $(S, o_1, \ldots, o_n)$, такой что:

1) $S = ((V, R), M)$ — размеченная системная схема, где (V, R) — Λ^{sch}-схема, $M : (V_{\mathrm{res}} \to \mathrm{Nat}, V_{\mathrm{ag}} \to (\{o_1, \ldots, o_n\} \times \mathrm{Nat}))$ — начальная разметка;

2) $o_i = ((v_i, r_i), m_i)$ — размеченная объектная сеть, где (v_i, r_i) — Λ_i^{o}-сеть, $m_i : v_i \to \mathrm{Nat}$ — начальная разметка объекта.

В отличие от сетей Петри синтаксис двухуровневых схем обладает элементами модульности. Рассмотрим пример моделирования (рис. 5). Здесь системная схема задает общую «карту» обеденного стола. Каждый философ может дотянуться: левой рукой — до левой салфетки, правой рукой — до правой салфетки, вилками — до блюда. Соответствующие типы портов обозначены картинками «левая рука», «правая рука» и «вилка». Начальная разметка схемы состоит из философов на всех стульях, вилок на всех салфетках и некоторого количества спагетти на блюде.

Объектная сеть задает поведение одного философа. Вложенность позволяет нам еще больше усложнить модель, разбив действия «взять вилки» и «положить вилки» на наборы из двух независимых действий («взять/положить левую/правую вилку»). Теперь философ — автомат с четырьмя состояниями. Узлы, соответствующие переходам автомата (для наглядности помечены серым), синхронизированы с отношениями в системной схеме. Например, при переходе из RFork в Thinking правая рука философа возвращает вилку (на правую салфетку), при срабатывании цикла Eating-Eating вилки в руках философа уничтожают единичный ресурс типа «спагетти».

Вложенность позволяет весьма просто и наглядно моделировать иерархи-

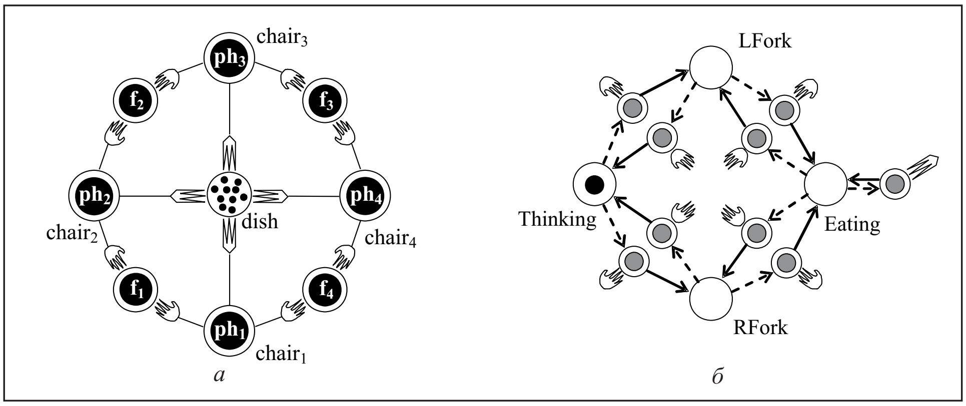

Рис. 5. Двухуровневая АР-схема: системная схема (*а*); объектная сеть ph_i (*б*)

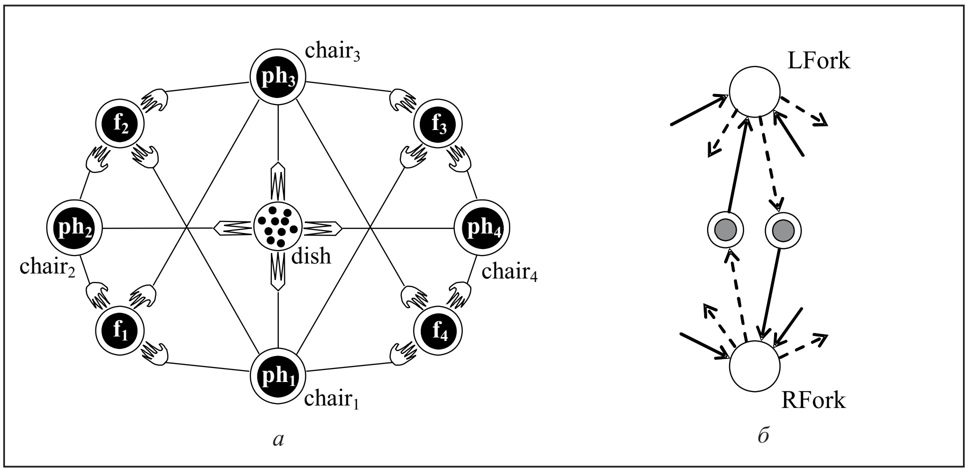

Рис. 6. Структура двухуровневой модели: прямоугольный стол («длиннорукие философы») (*а*); рассеянный философ (фрагмент) (*б*)

ческую структуру. Мы можем менять верхний и нижний слои независимо один от другого, получая новые интересные модели (рис. 6). В одноуровневых моделях это было бы невозможно. Однако выразительная мощность сохраняется, что гарантирует разрешимость всех основных семантических свойств и, следовательно, анализируемость нового формализма.

Теорема 2. Класс двухуровневых схем по выразительной мощности совпадает с классом обыкновенных сетей Петри.

Доказательство. Обыкновенные сети Петри эквивалентны обыкновенным АР-сетям, которые являются частным случаем двухуровневых схем (одна объектная сеть без портов). Следовательно, класс сетей Петри вкладывается в класс двухуровневых схем.

Для доказательства обратного вложения достаточно показать, что любую двухуровневую схему можно преобразовать в эквивалентную АР-сеть (сеть Петри). Рассмотрим следующее преобразование: заменим в исходной системной схеме каждую агентную вершину на подсеть, соответствующую по структуре и разметке всем ранее находившимся в этой вершине объектам (т.е. просто перенесем все объекты на системный уровень). Все порты преобразуем в производящие и потребляющие дуги, связывающие переходы новой подсети с соответствующими ресурсными вершинами схемы. Очевидно, что полученная AP-сеть эквивалентна исходной схеме. □

ДВУХУРОВНЕВЫЕ AP-СИСТЕМЫ

Рассмотрим более сложный случай, когда система обладает собственным поведением. Объекты могут не только трансформироваться (менять свое состояние), но и возникать, исчезать и перемещаться.

Для описания способов управления объектами в системной сети используем параметризованные выражения на потребляющих и производящих дугах. При этом в качестве констант могут выступать AP-сети (объектные фишки).

Системной называется метаструктура $\Lambda^{\text{sys}} = ((\{\text{cons}, \text{prod}\} \times \text{Labels}) \cup \cup \text{Syn}, \Phi^{\text{sys}})$, где Φ^{sys} определяет следующие расширенные правила:

1) дуги типов cons и prod помечены выражениями и участвуют в срабатываниях по правилам параметризованных AP-сетей;

2) дуги-порты участвуют в срабатываниях по правилам двухуровневых AP-схем;

3) в выражениях на потребляющих дугах не может быть констант; в выражениях на различных потребляющих дугах одного узла не может быть одинаковых переменных.

Ограничение 3) может показаться неочевидным, однако оно необходимо, поскольку убирает возможность проверки на ноль (которая привела бы к универсальной вычислительной модели, т.е. машине Тьюринга) [3].

Размеченной двухуровневой AP-сетью называется набор обобщенных AP-сетей $(S, o_1, \ldots, o_n)$ такой, что:

1) $S = ((V, R), M)$ — размеченная системная сеть, где (V, R) — Λ^{sys}-сеть, $M : V \to (\{o_1, \ldots, o_n\} \times \text{Nat})$ — начальная разметка системы;

2) $o_i = ((v_i, r_i), m_i)$ — размеченная объектная сеть, где (v_i, r_i) — Λ_i-сеть, $m_i : v_i \to \text{Nat}$ — начальная разметка объекта.

Усложним модель обедающих философов, добавив им возможность перемещения из столовой в парк (на прогулку) и обратно (на рис. 7 изображен фрагмент сети, соответствующий i-му философу).

С точки зрения философа ситуация та же, что и раньше (ведь он перемещается по парку в том же состоянии Thinking), однако с точки зрения системы при этом освобождается стул. Таким образом, в модели изменилась только системная сеть. Заметим также, что в парке может находиться несколько философов одновременно, но на одном стуле — только один (это свойство обеспечивается использованием узла-семафора is_empty_i).

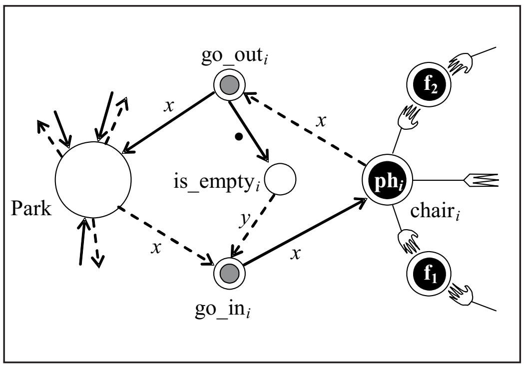

Рис. 7. Фрагмент сети (прогулка в парке)

Теорема 3. Класс двухуровневых AP-сетей с ограниченными объектными сетями [7] совпадает с классом обыкновенных сетей Петри; класс двухуровневых AP-сетей — с классом вложенных сетей Петри [3].

Доказательство. Для доказательства первого утверждения достаточно заметить, что любую ограниченную сеть Петри можно представить в виде автоматной сети Петри, т.е. сети, содержащей в каждый момент времени ровно одну фишку [7]. Таким образом, каждую агентную вершину системной сети можно заменить набором автоматных сетей, по одной для каждого вида объекта. Количество объектов в данной агентной вершине зададим количеством фишек в соответствующих позициях этих автоматных сетей. Порты преобразуем в дуги тем же способом, что и в доказательстве теоремы 2.

При доказательстве второго утверждения присоединим к агентной вершине с помощью пары разнонаправленных дуг узел-переход для каждого способа взаимодействия вложенного объекта с каким-либо портом. Сопоставим этот переход парному узлу-переходу в объекте с помощью пары меток вертикальной синхронизации (см. [3]). Свяжем переход с ресурсными вершинами необходимыми производящими или потребляющими дугами. Затем преобразуем системную и объектные АР-сети в эквивалентные сети Петри с помощью алгоритма, описанного в [4], сохраняя метки синхронизации. □

Вложенные сети Петри — существенно более мощный класс систем, обладающий, тем не менее, обширным набором разрешимых свойств [3]. Таким образом, двухуровневые АР-сети, так же как и одноуровневые сети и двухуровневые схемы, могут использоваться не только для моделирования систем, но и для их верификации.

ЗАКЛЮЧЕНИЕ

Исследованы новые способы представления мультиагентных систем с помощью обобщенных сетей активных ресурсов. Разработан ряд формализмов, предоставляющих удобные возможности моделирования:

1) обыкновенные АР-сети — взаимодействие агентов и ресурсов;

2) параметризованные АР-сети — плоские системы с однотипными агентами;

3) двухуровневые АР-схемы — модульные и иерархические системы;

4) двухуровневые АР-системы — модульные и иерархические системы со сложными взаимодействиями между системным и объектным уровнями модели.

СПИСОК ЛИТЕРАТУРЫ

1. Jensen K., Kristensen L. Coloured Petri nets: modelling and validation of concurrent systems. — N.Y.: Springer, 2009. — 384 p.
2. Kohler M., Rolke H. Properties of super-dual nets // Fundamenta Inform. — 2006. — **72.** — P. 245–254.
3. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. — М.: Научный мир, 2004. — 208 с.
4. Башкин В.А. Сети активных ресурсов // Моделирование и анализ информационных систем. — 2007. — № 14/4. — С. 13–19.
5. Reisig W. Petri net models of distributed algorithms // Lecture Notes in Comput. Sci. — 1995. — **1000.** — 302 p.
6. Bashkin V.A. Nets of active resources for distributed systems modeling // Joint Bulletin of NCC&IIS, C.S. Novosibirsk. — 2008. — **28.** — P. 4–54.
7. Котов В.Е. Сети Петри. — М.: Наука, 1984. — 160 с.

Поступила 14.07.2010