

УДК 681.5

Графодинамическое моделирование организационно-технических систем на основе триадных агентов

Юдицкий С. А.

Актуальность. В различных предметных областях широко применяются сетевые структуры, представляющие собой множество взаимодействующих автономных функциональных единиц – агентов. Среди факторов, определяющих поведение агента, первостепенными являются: цели, зависящие от назначения агента, его внутренних «ценностей», «убеждений» и приоритетов, а также от складывающейся внешней и внутренней ситуации; действия, направленные на достижение целей; ключевые параметры агента, вектор которых определяет его состояние. Цели, действия и параметры образуют триаду, что отражено в названии «триадный агент». Структуру такого агента образуют взаимодействующие между собой три компонента, описываемые «нагруженными» (помеченными) логическими выражениями графами (сетями Петри) – соответственно граф целей, граф действий, граф параметров. Ведущим в триаде является граф действий, который переключается в дискретные моменты времени при выполнении логических условий, формируемых внешней средой и/или двумя остальными графами триадной структуры. Результатом переключения графа действий в текущий дискретный момент являются: переход к новому действию и/или изменение некоторых параметров агента и/или коррекция его целей. Таким образом, сам триадный агент, по сути, является сетью с динамическим поведением и позволяет описать широкий класс динамических процессов в организационно-технических системах (ОТС). При этом, сложные ОТС отличаются большим числом параметров, разместить которые на соответствующем графе одного агента с точки зрения обозримости и наглядности модели не представляется возможным. С другой стороны, число целей и действий в ОТС существенно меньше числа ее параметров. В связи с этим для обеспечения обозримости и наглядности целесообразно декомпозировать сложную ОТС на несколько параллельно работающих триадных агентов (не более 7-10). **Цель статьи** заключается в создании графодинамической модели триадного агента (далее, просто агента) и инструментария на ее основе путем решения следующих частных задач. 1) Формализации и анализа графов целей, действий и параметров агента. 2) Формализации и анализа связей между графами в рамках агента. 3) Введения операций над графами агента («графохирургия»). 4) Алгебраического (символьного) представления графов агента с переходом от описания в форме графа к символьному описанию и обратно. 5) Формализации и анализа связей между агентами в рамках многоагентной триадной сети. **Научная новизна** работы заключается в создании новой триадной ветви агенто-центрического имитационного моделирования ОТС и поддерживающего ее инструментария, позволяющей упростить процедуру моделирования и повысить его наглядность. Кроме того, к элементам новизны работы стоит отнести: исследование механизма функционирования триадной модели агентов; обоснование нового способа преобразования модели агента с помощью «графохирургической» операции; разработку алгебраического представления графов в виде строки символов, компактно описывающей графы большой размерности (язык СЛОГов – Структурное Логическое Описание Графов), и операции над СЛОГами. **Практическая значимость** работы подтверждается разработанным в Белгородском государственном технологическом университете им. В.Г. Шухова программным продуктом, предназначенным для использования экспертами при имитационном агентном моделировании сетевых структур, и применении этого продукта при решении промышленных задач.

Ключевые слова: графодинамика, триадные агенты, триадные сети, целеполагание, логическое управление, взаимовлияние параметров, механизм взаимодействия графов, графохирургия, алгебра графов, язык слогов, операции над слогами, программная поддержка моделирования.

Введение

В последние десятилетия в развитых странах наблюдается всё возрастающее применение в различных предметных областях сетевых структур и рост интереса к сетевым моделям в управлении. Резко вырос поток публикаций по этой тематике, в том числе на русском языке. Опубликованы работы по общим вопросам теории сетей [1, 2, 3, 4], по сетевым методам принятия решений [5, 6, 7], по теории социальных сетей [8], сетецентрическому управлению в многоагентных системах [9, 10, 11, 12], рассмотрены многочисленные примеры технологических сетей (в газо- и нефтетранспортных системах [13], электроэнергетике, в компьютерных сетях [14] и т.д.).

Сетевая структура определяется как множество автономных функциональных единиц – агентов, которые могут вступать во взаимодействия друг с другом. Эффективность управления поведением сетевой структуры во многом определяется предварительным имитационным моделированием, в ходе которого определяются временные и ресурсные характеристики системы, конфликтные ситуации, «узкие места» и т.д. Целью моделирования является также прогнозирование процессов развития сетевой структуры, с определением её возможных конфигураций и динамики их преобразования. Модель развития («видение будущего») должна работать на дискретной временной шкале и учитывать не только ожидаемые, но и неожиданные (маловероятные) внешние события. Проблема формального описания и моделирования процессов развития сетевых структур проработана еще в недостаточной мере, одной из попыток продвижения на пути к ее решению следует считать и данную работу.

В статье предложена графодинамическая модель триадного агента на базе классической сети Петри [15, 16], дополненной введенными новыми языковыми конструкциями:

- индикатором сравнения численной переменной с другой переменной или с константой;
- индикатором операции преобразования численной переменной;
- индикаторной логической (булевой) функцией;
- индикаторной продукцией ЕСЛИ – ТО;
- индикаторной сетью (сеть Петри с переходами, помеченными индикаторными продуктами);
- моделью триадного агента в виде тройки индикаторных сетей для графов действий, целей, параметров с буферными позициями между переходами разных графов.

Примеры этих конструкций будут даны ниже. Индикаторы являются булевыми переменными, принимающими два значения – истина (1) и ложь (0). Численные переменные выражают лингвистические значения параметров (типа «низкий», «средний», «высокий» и т.д.) в виде числа баллов.

1. Графодинамика триадного агента

Для определения триадной модели агента следует:

- сформировать состав и структуру целей, поставленных перед агентом, установить причинно-следственные связи на множестве целей (целеполагание [6, 7, 17]);
- задать состав и порядок выполнения действий (логическое управление [18, 19, 20]);
- выбрать ключевые параметры, характеризующие работу агента, определить взаимовлияния параметров (когнитивный анализ [21, 22, 23]);
- установить связи между целями, действиями и параметрами.

Решение указанных задач относится к тематике графодинамики [24] – направления в системном анализе, оперирующего переменными в форме графов и отношениями, определенными на графах.

1.1. Моделирование целеполагания

На множестве целей, определяющих деятельность агента, введем отношение подчиненности «цель–подцель», где достижение цели является непосредственным следствием достижения подцелей (подцели детализируют цель). Цели, не подчиненные никакой другой цели, назовем *конечными*, а подцели, которым не подчинены другие цели, – *начальными*.

Конечные цели формулируются абстрактно, в общем виде. Последовательность подчиненных целей, вплоть до начальных, дает необходимые уточнения. Если конечных целей несколько, то они достигаются либо в определенной очередности, либо независимо друг от друга.

Целеполагание является предметом интеллектуальной деятельности специалиста, проводящего моделирование. Психология целеполагания с акцентом на допускаемые типовые ошибки, приводящие к неудачам, описаны в [25].

Формальный аппарат целеполагания базируется на индикаторных сетях, позиции которых сопоставлены целям c_i , $i = 1, \dots, r$, а переходы помечаются индикаторными продукциями. Пример такого графа дан на рис. 1.

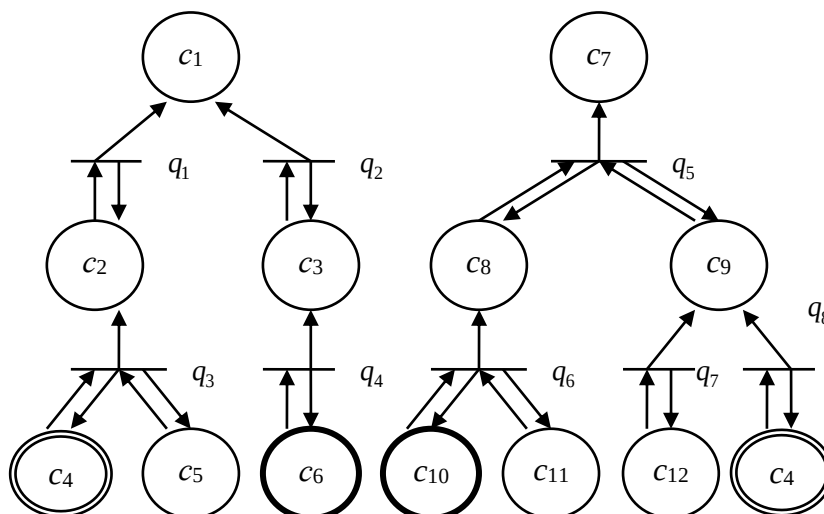


Рис. 1. Пример сети Петри, являющейся основой графа целей

Переходы q_j , $j = 1, \dots, s$, связывают цель c_i с подчинёнными ей подцелями, где каждая подцель соединена с переходом двумя противоположно направленными стрелками, а переход соединен с каждой своей выходной целью – одной стрелкой. Каждая позиция c_i либо пуста ($c_i = 0$ – цель не достигнута), либо содержит один маркер ($c_i = 1$ – цель достигнута).

Если во всех входных позициях перехода q_j есть по маркеру и все его выходные позиции пусты, и, кроме того, выполняется условие соотнесенной переходу продукции (ее левая часть), то переход срабатывает, вносит маркер в каждую выходную позицию c_i , сохраняя маркирование входных позиций, и выполняет действие (оператор, указанный в правой части продукции). После этого дуга $q_j c_i$ «запирается», что исключает попадание второго маркера в выходную позицию c_i .

Дадим необходимые определения, поясняющие использованную в статье терминологию [26].

Индикатором сравнения – называется булева функция $z=(x\#y)$, принимающая единичное значение, если выполняется отношение $\#$, и нулевое, если не выполняется. Где: x, y – численные переменные, « y » может быть константой, $\#$ – знак отношения сравнения из множества $\{=, >, \geq, <, \leq\}$.

Индикаторной логической функцией (ИЛФ) называется выражение, полученное применением конечного числа раз к индикаторам сравнения булевых операций конъюнкции (И), дизъюнкции (ИЛИ) и отрицания (НЕ), изображаемых соответственно $\wedge$, $\vee$, чертой над символом.

Индикатором преобразования численной переменной $x(\tau)$ при наступлении следующего момента $\tau+1$ называется булева функция $y(\tau+1)=(x(\tau+1)=\Phi(x(\tau)))$, где: Φ – оператор преобразования; $y=1$ в момент $\tau+1$ при выполнении преобразования Φ , и $y=0$ во все остальные моменты.

Индикаторной продукцией называется причинно-следственное выражение вида $Y \rightarrow O$, где: Y – условие, выражаемое индикаторной логической функцией; O – оператор, являющийся следствием, представленный конъюнкцией индикаторов преобразования.

Индикаторной сетью называется сеть Петри с переходами, помеченными индикаторными продуктами.

При работе сетевой модели целеполагания маркеры продвигаются «снизу–вверх» по древовидным графам целей (граф целей – ациклический, любой переход в нем срабатывает не более одного раза). При этом могут использоваться два вида отношения «цель–подцели», при числе подцелей не менее двух: конъюнктивное и альтернативное.

При конъюнктивном отношении обязательно выполнение всех подцелей, которые являются составными частями цели.

При альтернативном отношении необходимо и достаточно выполнение только одной подцели (при недетерминированном выборе этой подцели).

В примере на рис. 1 конъюнктивными являются отношения: $(c_2 - c_4, c_5)$, $(c_7 - c_8, c_9)$, $(c_8 - c_{10}, c_{11})$, а альтернативными: $(c_1 - c_2, c_3)$, $(c_9 - c_4, c_{12})$.

Среди начальных подцелей (позиций) на рис. 1 выделим *противоречивые подцели* (изображаются жирными кружками) и *совпадающие подцели* (изображаются двойными кружками).

Противоречивые начальные позиции c_i, c_j находятся в отношении альтернативности, обеспечиваемом логической функцией $\alpha = \bar{c}_i \wedge \bar{c}_j$ называемой семафором. Маркирование противоречивых позиций возможно только при $\alpha = 1$, что имеет место при пустых позициях. Если первым маркер попадает в одну из противоречивых позиций, например $c_i = 1$, то $\alpha = 0$, и вход в c_j блокируется. Совпадающие позиции являются экземплярами одной и той же начальной подцели. В примере на рис. 1 противоречивы позиции c_6, c_{10} и совпадают две позиции c_4 .

Результатом моделирования целеполагания является линейный график достижения целей на заданном временном горизонте $[\tau = 0, \tau = N]$, пример которого дан на рис. 2. Интервалы горизонта, на которых цели c_i достигнуты, заштрихованы. График достижения целей строится непосредственно по графу целей.

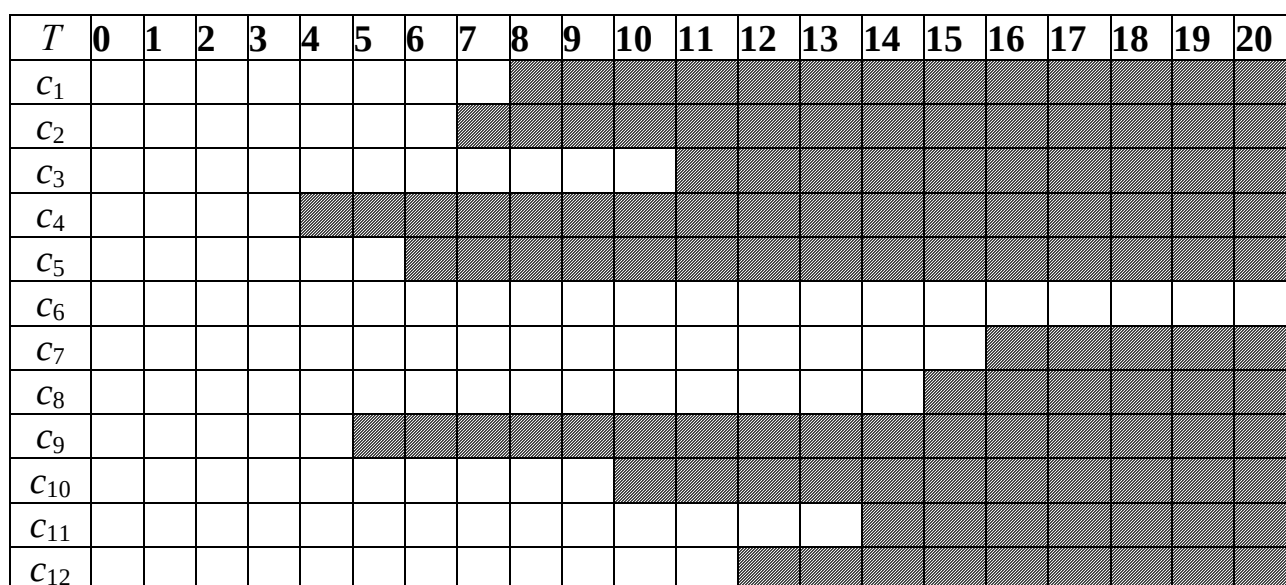


Рис. 2. Линейный график достижения целей

В примере на рис. 2 в момент $\tau = 4$ достигаются (вследствие внешнего воздействия) оба экземпляра подцели c_4 . Левая подцель не влияет на свою цель c_2 , правая вызывает в следующий момент $\tau = 5$ достижение цели c_9 . Дальнейших переключений (без воздействия на начальные позиции) не происходит, в графе целей устанавливается равновесие. В момент $\tau = 6$ маркер заносится в начальную позицию c_5 , срабатывает переход q_3 и в момент $\tau = 7$ маркер вносится в позицию c_2 . Далее в момент $\tau = 8$ срабатывает q_1 и маркер вносится в позицию c_1 , достигнута первая конечная цель. Граф целей продолжает функционировать подобным образом. В момент $\tau = 15$ маркеры находятся в позициях c_8, c_9 и выполняется условие $c_1 \wedge \bar{c}_7 = 1 \wedge 1 = 1$. Это приводит к

срабатыванию перехода q_5 и внесению маркера в позицию c_7 . Таким образом, достигнута вторая конечная цель c_7 , причём строго после первой c_1 .

На графике на рис. 2 есть несогласованность (на внимание читателя): строка c_3 не должна быть заштрихована, т.к. начальная цель c_6 не достигнута, и правая цепочка, ведущая в конечную цель c_1 , в данном имитационном эксперименте не задействована.

1.2. Моделирование логического управления действиями агента

Для моделирования порядка выполнения действий и обусловленного этим порядком логического управления агентами применяются *графы действий*, называемые также *графами операций* [18].

Пример сети Петри, как основы графа действий, дан на рис. 3. При этом, переходы изображены прямоугольниками, а соответствующие этим переходам индикаторные продукции, которые будут далее рассмотрены отдельно — опущены. Для обоснования разветвлений в описании поведения агента даны воздействия V внешней среды: v_1 и v_2 .

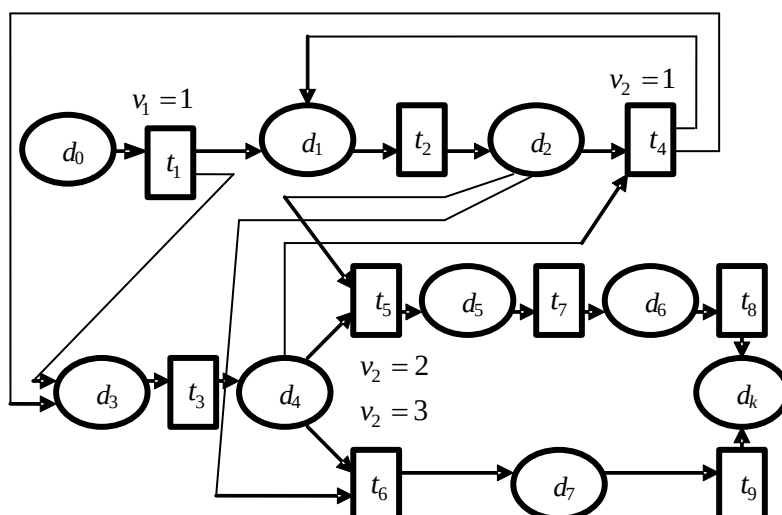


Рис. 3. Сеть Петри как основа графа действий

В любой позиции d_i находится один маркер, если соответствующее действие выполняется, и позиция пуста, если не выполняется. Переход t_j срабатывает мгновенно, если одновременно выполняются следующие условия:

- во всех входных позициях перехода есть по маркеру;
- выполняются соотнесенное переходу внешнее условие v_k и условия, формируемые графами целей и параметров;
- с момента активирования перехода прошло не менее заданного числа единиц модельного времени.

В результате срабатывания перехода из всех его входных позиций удаляются, а во все выходные позиции вносятся маркеры. Временной график логического управления действиями агента является частью итоговой диаграммы моделирования, помещенной на рис. 4.

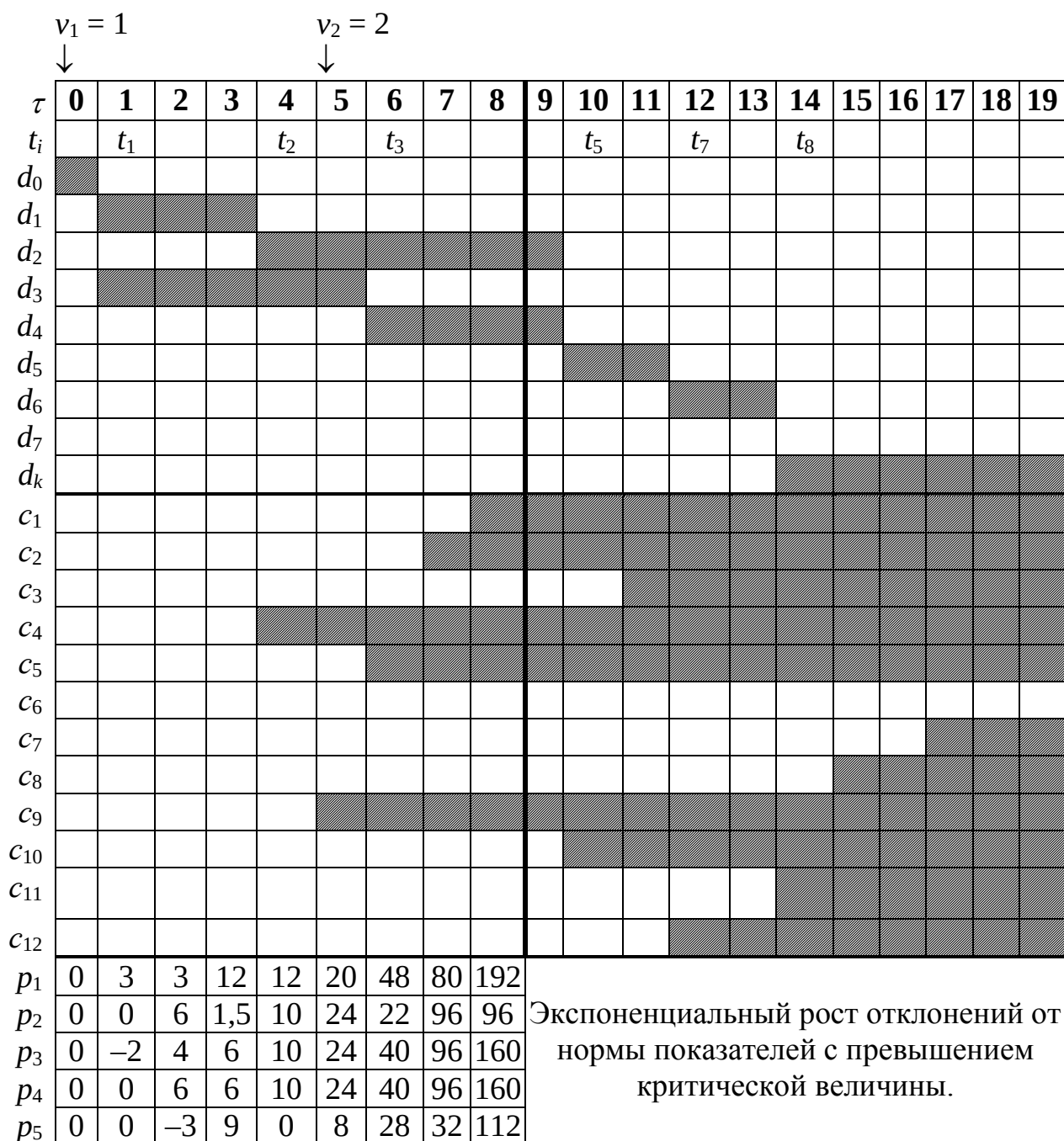


Рис. 4. Итоговая диаграмма моделирования триадной структуры агента

В верхней строке диаграммы указаны моменты времени, образующие горизонт моделирования. В следующей строке проставлены соотнесенные определенному моменту переходы графа действий, в которые происходит смена действий, достижение целей, изменение величины параметров. В верхнем ярусе диаграммы даны линейные графики булевых переменных, сопоставленных действиям (d_0, d_k – начальное и конечное «пустые» действия, обозначающие подготовку к моделированию и его завершение). В среднем ярусе представлены линейные графики достижения целей, в нижнем дана таблица бальных значений параметров. Вертикальная жирная линия разделяет допустимый и недопустимый интервалы горизонта моделирования.

Итог моделирования поведения агента: при заданной триадной структуре полностью выполняются действия d_1 , d_3 и лишь частично d_2 , d_4 ; достигаются цели c_1 , c_2 , c_4 , c_5 , c_9 ; в момент $\tau = 8$ значения параметров p_1 , p_3 , p_4 , p_5 выходят за критическую отметку $p_i^{max}=100$, а отклонение p_2 вплотную приблизилось к ней. Поэтому в момент $\tau = 8$ имитационный эксперимент прекращаем.

1.3. Моделирование взаимовлияния параметров агента

Лингвистическая (словесная) оценка величины параметров, выражаемая числом условных баллов, и прогнозы изменения параметров при работе создаваемой системы, выполняются экспертами на предпроектной стадии, и являются достаточно грубыми. Результатом такого предварительного моделирования могут стать тенденции, которые по мнению экспертов проявятся при работе системы. Но для этого надо ввести в модель некоторые гипотетические уточнения.

1. Примем, что параметры агента могут изменяться в моменты, непосредственно следующие за наступлением нового такта (интервала) $\tau = 1, 2, \dots, N$ на заданной временной шкале, как следствие срабатывания перехода t на графе действий в начальный момент такта τ , а также выполнения условий достижения определенных целей при определенном состоянии внешней среды.

2. Изменение параметров может вызвать изменение других параметров и т.д. с установлением равновесия в том же такте. Процесс изменения описывается графом параметров (рис. 5), а равновесие обеспечивается индикаторными продукциями переходов r_i этого графа.

3. Параметры изменяются на бальной предметной шкале, содержащей, например, 100 пунктов, где каждый 10-й пункт имеет лингвистическую интерпретацию. Эксперты формулируют свои оценки в лингвистических терминах.

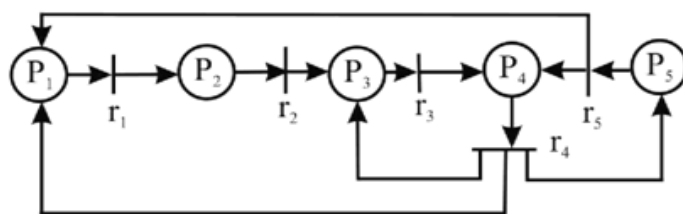


Рис. 5. Сеть Петри как основа графа параметров

Приведем пример бальной предметной шкалы: 0 – без оценки, 10 – очень плохо, 20 – плохо, 30 – между плохо и удовлетворительно, 40 – удовлетворительно, 50 – между удовлетворительно и хорошо, 60 – хорошо, 70 – между хорошо и отлично, 80 – отлично, 90 – между отлично и превосходно, 100 – превосходно.

1.4. Механизм взаимодействия графов действий, целей и параметров при работе агента

Рассмотрим вначале механизм взаимодействия сетей Петри, составляющих основу графов агента, при выполнении описанного выше переходного процесса, инициированного наступлением нового такта в момент τ (рис. 6).

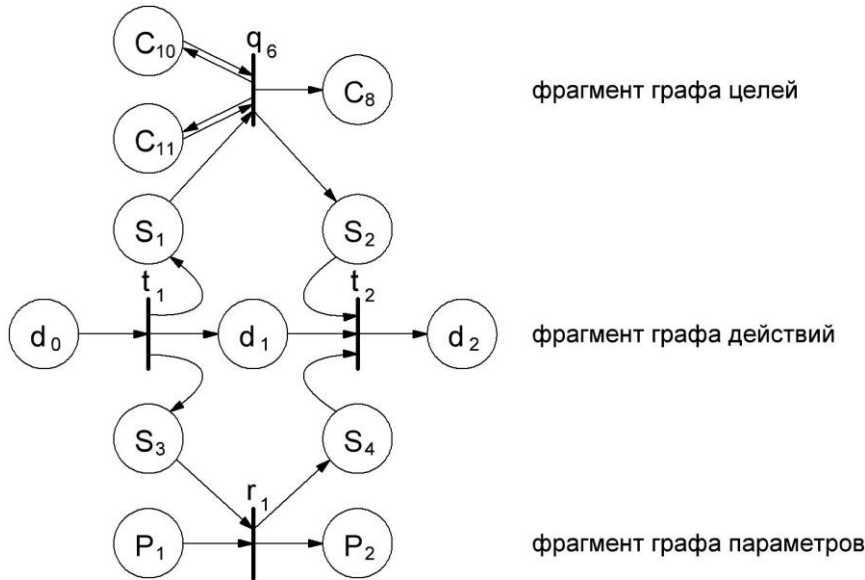


Рис. 6. Механизм взаимодействия графов агента

Каждую из трех сетей будем представлять фрагментом с переходом (переходами), помеченным индикаторной продукцией. Связи между сетями (фрагментами) обеспечиваются введением дополнительных «буферных» позиций s_i и структурой продукций. Буферная позиция принимает только два значения: 1 и 0, т.е. является булевой.

Переходы, срабатывающие в переходном процессе, – t_1 , t_2 (граф действий), q_6 (граф целей), r_1 (граф параметров), соотнесены следующим индикаторным продукциям:

$$\begin{aligned} \Pi(t_1): & (d_0(\tau)=1) \wedge (d_1(\tau)=0) \wedge (s_1(\tau)=0) \wedge (s_3(\tau)=0) \wedge (\tau=1) \rightarrow \\ & \rightarrow (d_0(\tau+1)=0) \wedge (d_1(\tau+1)=1) \wedge (s_1(\tau+1)=1) \wedge (s_3(\tau+1)=1); \end{aligned} \quad (1)$$

$$\begin{aligned} \Pi(t_2): & (d_1(\tau)=1) \wedge (d_2(\tau)=0) \wedge (s_2(\tau)=1) \wedge (s_4(\tau)=1) \wedge (\tau=2) \rightarrow \\ & \rightarrow (d_1(\tau+1)=0) \wedge (d_2(\tau+1)=1) \wedge (s_2(\tau+1)=0) \wedge (s_4(\tau+1)=0); \end{aligned} \quad (2)$$

$$\begin{aligned} \Pi(q_6): & (c_8(\tau)=0) \wedge (c_{10}(\tau)=1) \wedge (c_{11}(\tau)=1) \wedge (s_1(\tau)=1) \wedge (s_2(\tau)=0) \wedge (\tau=1) \rightarrow \\ & \rightarrow (c_{10}(\tau+1)=1) \wedge (c_{11}(\tau+1)=1) \wedge (s_1(\tau+1)=0) \wedge (c_8(\tau+1)=1) \wedge (s_2(\tau+1)=1); \end{aligned} \quad (3)$$

$$\begin{aligned} \Pi(r_1): & (p_1(\tau) > 20) \wedge (p_1(\tau) < 60) \wedge (p_2(\tau) < 80) \wedge (s_3(\tau)=1) \wedge (s_4(\tau)=0) \wedge (\tau=1) \rightarrow \\ & \rightarrow (p_1(\tau+1)=p_1(\tau)-10) \wedge (p_2(\tau+1)=p_2(\tau)+10) \wedge (s_3(\tau+1)=0) \wedge (s_4(\tau+1)=1). \end{aligned} \quad (4)$$

Для моделирования взаимодействий между графами агента задаются начальные условия в виде значений переменных – булевых для действий d ,

целей c , буферных позиций s , и бальные оценки для параметров p , в начальный момент $\tau=0$. Далее, с помощью индикаторных продукций типа (1)–(4) вычисляются значения переменных для последующих тактов $\tau=1, 2, \dots, N$.

Пусть в начальный момент $\tau=0$: $d_0=1$, $d_1=0$, $d_2=0$, $c_{10}=1$, $c_{11}=1$, $c_8=0$, $p_1 \in [20, 60]$, $p_2=40$, $s_1=s_2=s_3=s_4=0$, ни один переход не активирован.

В момент начала такта $\tau=1$ срабатывает переход t_1 , изымающий маркер из d_0 и вносящий маркеры в d_1 , s_1 , s_3 . Это приводит к активированию двух переходов – q_6 и r_1 , которые могут сработать в любой последовательности либо одновременно (описываемая структура является расширением классической сети Петри, в которой запрещается одновременное срабатывание двух и более переходов). При срабатывании перехода q_6 маркеры остаются в позициях c_{10} и c_{11} , обнуляется буферная позиция s_1 и маркеры вносятся в позиции c_8 , s_2 . При срабатывании перехода r_1 в его входной позиции p_1 , значением которой эксперт указал только принадлежность к интервалу на бальной шкале, согласно оператору продукции (4) верхняя и нижняя границы этого интервала уменьшаются на 10, т.е. он принимает вид $p_1=[10, 50]$, а выходные позиции принимают значения $p_2=40+10=50$, $s_4=1$. При установившемся маркировании переменных активируется и срабатывает в следующем такте $\tau=2$ переход t_2 , удаляются маркеры из всех его входных позиций, и вносится маркер в выходную позицию d_2 . Переходный процесс в структуре на рис. 6 в такте $\tau=1$ завершен, и структура пришла в равновесное состояние.

1.5. Операции над графами – графохирургия

Существует содержательная аналогия между медицинскими хирургическими операциями типа пересадки тканей и органов, «вживления» (имплантации) искусственных устройств – кардиостимуляторов, сердечных клапанов, протезов суставов и т.д., и операциями преобразования графов. При классической медицинской операции последовательно выполняются следующие этапы:

- 1) разрез тканей и доступ к пораженному органу;
- 2) «отключение» органа от системы кровообращения (и, возможно, других систем организма);
- 3) удаление пораженного органа или его части (например, новообразования);
- 4) при имплантации – «вставка» замещающего органа; сшивание сосудов и тканей.

При выполнении операции преобразования графа будем применять следующий базовый алгоритм.

1. Выделяем (замкнутой пунктирной линией) фрагмент графа, который подлежит замене. Граничные переходы выделенного фрагмента представляем в виде связок – совокупности синхронизированных компонентов перехода, которые по определению срабатывают одновременно, и только одновременно [26, 27] (понятие «связка» вписывается в используемое в работе расширение классических сетей Петри). Каждый компонент связки принадлежит либо выделенному

фрагменту, либо остальной части графа. Таким образом, выделенный фрагмент оказывается отделенным от исходного графа.

2. Удаляем фрагмент, отделенный от остальной части графа.
3. Вставляем новый фрагмент, замещающий удаленный, с формированием связей, синхронизирующих источники нового фрагмента и стоки остальной части графа (источниками и стоками согласно работе [27] названы переходы, которые соответственно не содержат входных или выходных дуг). Замещающий фрагмент может быть «пустым», не содержащим ни одной позиции.
4. Каждую введенную связку заменяем одним переходом («сшивание» компонентов связки).

Ввиду отмеченного выше подобия между медицинскими операциями и операциями преобразования графов последние будем называть *графохирургическими операциями*.

Проиллюстрируем базовый алгоритм проведения графохирургических операций на примерах.

В графе на рис. 7 а выделены два фрагмента: верхний и нижний, содержащие соответственно позиции a_3, a_5 и a_4, a_6 . Граничными переходами для этих фрагментов служат b_2, b_7 , представимые связками $b_2 = \{b_2^1, b_2^2, b_2^3\}$, $b_7 = \{b_7^1, b_7^2, b_7^3\}$. Удаляем верхний и нижние фрагменты, выделенные на рис. 7 а верхний черным, а нижний красным пунктиром, и замещаем их фрагментами, показанными на рис. 7 б, – верхним на базе позиций a_8, a_9 и нижним пустым фрагментом. Далее «сшиваем» связки b_2, b_7 и получаем граф, изображенный на рис. 7 в.

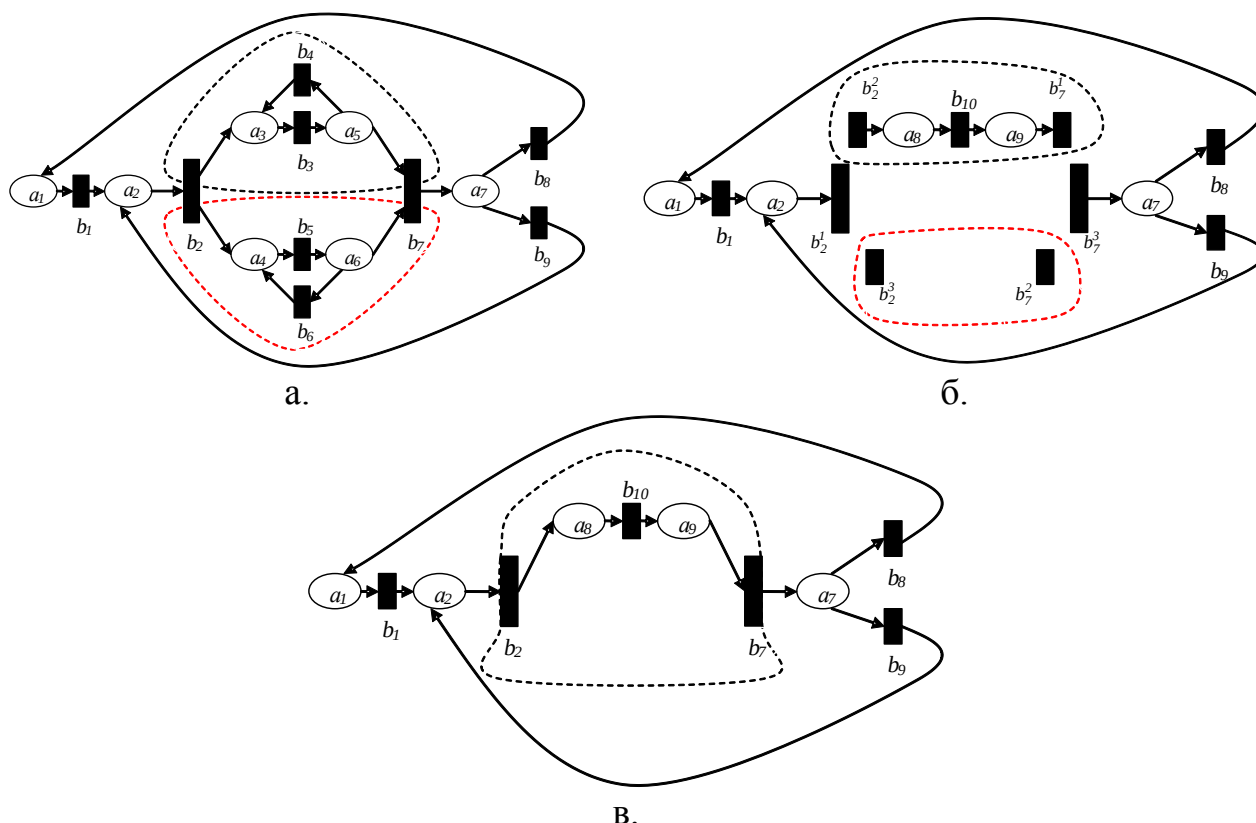


Рис. 7. Пример графохирургической операции

Базовый алгоритм может быть применен для различных вариантов графохирургических операций.

На рис. 8 показана операция совмещения двух позиций графа с заменой их одной позицией. Входными (выходными) переходами совмещенной позиции становятся входные (выходные) переходы совмещаемых позиций. Пусть, например, требуется совместить позиции a_2 и a_7 . Выделяем (рис. 8 а) и удаляем позицию a_7 (рис. 8 б) и переходы b_7 , b_8 , b_9 соединяем дугами с позицией a_2 (рис. 8 в). Из b_7 в a_2 выводится выходная дуга, из a_2 в b_8 и из b_8 в a_1 – также выходные дуги, переход b_9 связан с позицией a_2 входной и выходной дугами.

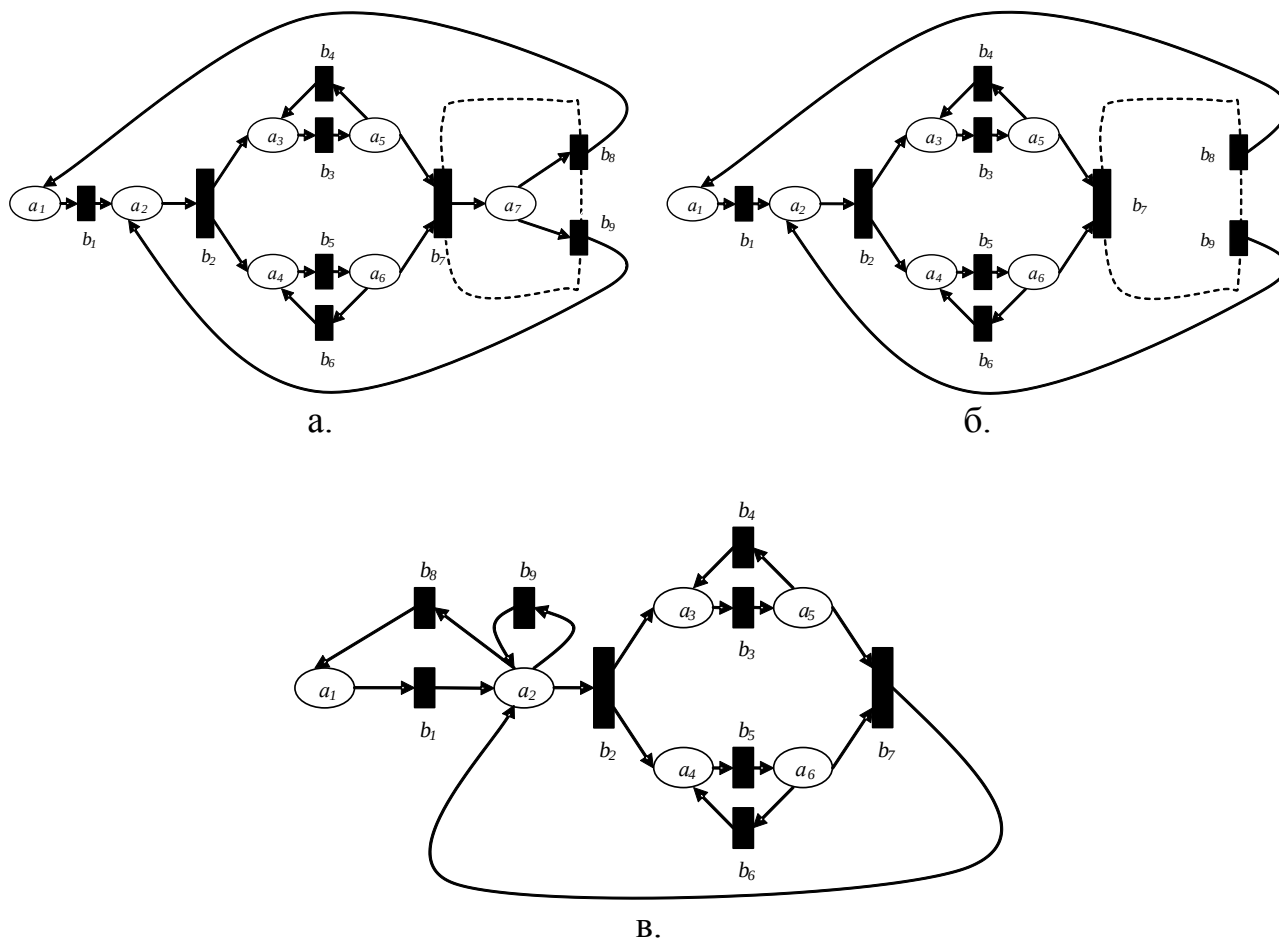
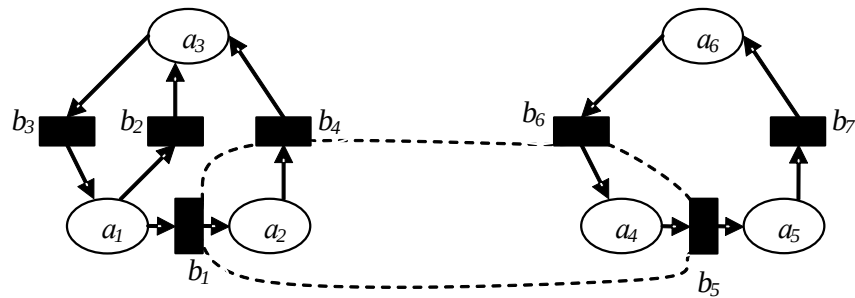


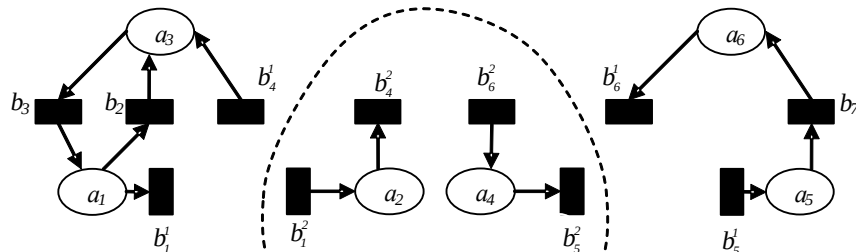
Рис. 8. Пример операции совмещения позиций

На рис. 9 даны примеры преобразования пары графов: соединения их в один граф и введения в графы дополнительных дуг. На рис. 9 а выделен фрагмент, состоящий из позиций a_2 , a_4 , принадлежащих разным графам. Далее по переходам b_1 , b_4 , b_5 , b_6 вводятся связки и отделяется центральный фрагмент (рис. 9 б). После удаления этого фрагмента, вставки замещающего фрагмента с внутренними переходами b_8 , b_9 и «сшивания» соответствующих связок, приходим к единому графу, показанному на рис. 9 в. Другой вариант преобразования графов на рис. 9 а заключается в том, что в оставшихся после удаления центрального фрагмента частях формируются связки, содержащие помимо компонентов «разрезанных» переходов и «неразрезанные» переходы:

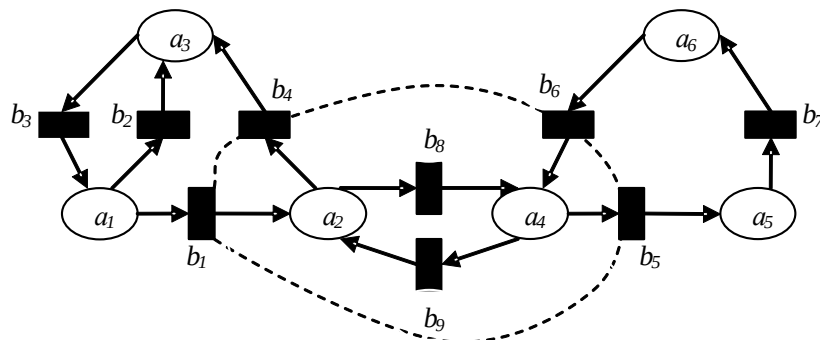
$\{b_1^1, b_3, b_4^1\}$, $\{b_5^1, b_6^1, b_7\}$. После «сшивания» связок получим графы, приведенные на рис. 9 г.



а.

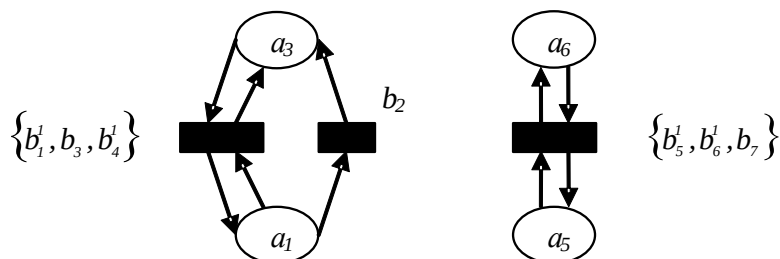


б.



$$b_1 = \{b_1^1, b_1^2\}; \quad b_4 = \{b_4^1, b_4^2\}; \quad b_5 = \{b_5^1, b_5^2\}; \quad b_6 = \{b_6^1, b_6^2\}$$

в.



г.

Рис. 9. Графохирургическое преобразование пары графов:

в – введение соединительного фрагмента; г – введение дополнительных дуг

Управление процессом функционирования сети предусматривает возможность изменения ее структуры – как в части действующих графов, так и

взаимодействий между ними. Известен ряд типовых операций, отображающих преобразование графов: введение/удаление элемента сети (позиции или перехода), слияние позиций, копирование подграфа и т.д. [13, 28, 29]. Список типовых операций является открытым и при необходимости может пополняться.

Таким образом, в этой части работы предложен общий (графохирургический) метод преобразования графов, на основе которого могут порождаться типовые операции (интерпретируемые как частные случаи общего решения). Метод позволяет вносить выборочные изменения в определенные фрагменты системы, сохраняя преемственность в отношении остальной ее части.

Графохирургия включает:

- «разрезание» некоторых переходов сети с формированием так называемых связок – подмножеств синхронизированных переходов;
- отделение и удаление фрагмента сети;
- вставку на его место другого фрагмента; «сшивание» связок – замена их неструктурированными переходами.

1.6. Взаимодействие агентов в многоагентной системе

Многоагентная триадная сеть представляет собой множество взаимосвязанных агентов (субъектов или объектов), в которой [10]:

- каждый агент сохраняет свою индивидуальность, а именно имеет собственные (индивидуальные) цели, выполняет направленные на достижение этих целей индивидуальные действия, характеризуется индивидуальными параметрами;
- связанность агентов заключается в том, что их деятельность может координироваться во времени, и в определенные моменты они могут передавать друг другу сообщения и ресурсы;
- результатом индивидуальной деятельности агентов является достижение определенных коллективных целей и определенная динамика коллективных параметров.

Задача синтеза многоагентной сети в общем виде формулируется следующим образом. Известен набор агентов, представленных, например, триадными структурами, отображающими взаимовлияние графов индивидуальных целей, действий, параметров, и заданы коллективные цели и параметры. Требуется определить, можно ли путем организации связей между агентами создать сеть, в которой наряду с индивидуальными целями достигались бы и желаемые коллективные цели при допустимых значениях индивидуальных и коллективных параметров. Если желаемых целей достичь нельзя, то какие коллективные цели реально достижимы? Аналитическое решение задачи синтеза сети с получением соответствующих необходимых и достаточных условий представляется весьма трудным и может стать предметом специальной работы. В данной статье автор ограничивается построением модели, позволяющей подбирать компоненты многоагентной сети и проверять эффективность такого подбора с помощью имитационного моделирования.

Рассмотрим некоторые аспекты внешнего поведения сети, определяемые взаимодействиями между агентами. Два агента считаются связанными, если их индивидуальная деятельность (как уже говорилось) координируется во времени, и в определенные моменты они могут передавать друг другу информационные или/и материальные ресурсы.

Переходы b_1, b_2 , принадлежащие разным агентам и помеченные соответственно продуктами $Y_1 \rightarrow O_1, Y_2 \rightarrow O_2$, где Y – логическое условие, O – оператор, будем называть сильно связанными, если они являются компонентами связки $\{b_1, b_2\}$ [27], т.е. срабатывают только одновременно. Сильно связанные переходы помечаются продуктами $Y_1 \wedge Y_2 \rightarrow O_1, Y_1 \wedge Y_2 \rightarrow O_2$, обеспечивающими их одновременное срабатывание. Оператор агента, передающего порцию ресурса R , уменьшает значение соответствующей переменной на величину этой порции, а оператор принимающего агента, наоборот, увеличивает значение переменной на ту же величину. Бинарное отношение сильной связанности является временным в том смысле, что участвующие в нем агенты могут поменяться ролями, агент может вступить в связку с другим партнером и т.д. Мотивация к таким переменам – разумный баланс между личными и коллективными интересами и целями (знакомо, правда?).

Многоагентная система – это своеобразное «общество», в котором постоянно возникает и разрушается огромное число связей (верхняя граница при N агентах равна C_N^2 – числу сочетаний из N по 2).

Проследить за динамикой связей в системе с большим числом агентов можно только по усредненным показателям. Тем не менее, конкретизируя интересы и цели реальных агентов и зная назначение и стратегию реальной многоагентной системы, при ограниченном числе агентов (не более 10) можно спрогнозировать наиболее вероятные линии развития системы и провести их имитационное моделирование. Однако, этот вопрос выходит за пределы данной статьи.

2. Алгебраическое представление графов

Модель сети характеризуется статической и динамической составляющей, где статическая составляющая представляется в графовой форме, а динамическая – в алгебраической (в виде индикаторных выражений), что делает модель неоднородной. Кроме того, сложность (а тем самым и наглядность) графового описания экспоненциально возрастает при увеличении его размерности. Следовательно, для моделирования статической составляющей имеет смысл применить алгебраическое описание, линейно зависящее от размерности агентов и сети. В статье для этого введен формальный язык СЛОГов (Структурное Линейное Описание Графов), отображающих ориентированный граф в виде строки, составленной из символов вершин (применительно к двудольному графу – из чередующихся символов позиций и переходов) и нумерованных вертикальных стрелок.

2.1. Язык СЛОГов

СЛОГ взаимно однозначно отображает ориентированный граф – вершины, дуги, функцию инциденций. Примем, что вершины обозначаются символами из алфавита $E=\{e_i, i=1, \dots, n\}$, где вершина e_i – это позиция или переход. Если из вершины e_i исходят дуги, ведущие в вершины $e_{j1}, e_{j2}, \dots, e_{jk}$, то первая дуга отображается последовательностью $e_i e_{j1}$ символов в строке СЛОГа, остальные – упорядоченными парами пронумерованных вертикальных стрелок. Стрелки подразделяются на выходные, направленные вверх, и на входные, направленные вниз. Выходная стрелка проставляется справа от символа e_i , входные стрелки – слева от символов $e_{j2}, \dots, e_{jk}$. Пара стрелок, соответствующая одной дуге, нумеруется одинаковыми верхними индексами. Применительно к дуге $e_i e_{j1}$ стрелки, находящиеся между e_i и e_{j1} , во внимание не принимаются. Если таких стрелок несколько, то вначале проставляются подряд все выходные стрелки, а затем подряд все входные стрелки. Справа от терминальной вершины, из которой не исходит ни одной дуги, ставится восклицательный знак – «!».

Алгебраическое представление СЛОГов проиллюстрируем на примере графа, изображенного на рис. 8 а:

$$s_1 = \downarrow^7 a_1 \ b_1 \ \downarrow^8 a_2 \ b_2 \uparrow^1 \downarrow^2 a_3 \ b_3 \ a_5 \ \uparrow^3 b_4 \ \uparrow^2 \ ! \downarrow^1 \downarrow^4 a_4 \ b_5 \ a_6 \uparrow^5 b_6 \ \uparrow^4 \downarrow^3 \downarrow^5 b_7 \\ a_7 \uparrow^6 b_8 \uparrow^7 \downarrow^6 b_9 \uparrow^8 \quad (5)$$

Следование буквы b_1 непосредственно за a_1 отображает дугу $a_1 b_1$ на графе на рис. 8 а, a_2 за b_1 – дугу $b_1 a_2$ и т.д. Цепочка чередующихся позиций и переходов в СЛОГе завершается буквой b_6 , справа от которой стоит знак «!». Другая цепочка – $b_7 a_7 b_8$. Пара «выходная стрелка – входная стрелка» с номером 1 отображает дугу $b_2 a_4$, пара стрелок с номером 2 – дугу $b_4 a_3$ и т.д.

В общем случае исходному графу соответствует множество эквивалентных СЛОГов, отличающихся порядком букв в строке, числом и местом расположения вертикальных стрелок. Эквивалентность заключается в том, что полученные на их основе графы изоморфны. Среди СЛОГов может быть выбран оптимальный, например, с наименьшим числом стрелок. Однако здесь такая задача не рассматривается. Номера стрелок в СЛОГе (5) соответствуют следующим дугам графа на рис. 8 а: 1 – $b_2 a_4$, 2 – $b_4 a_3$, 3 – $a_5 b_7$, 4 – $b_6 a_4$, 5 – $a_6 b_7$, 6 – $a_7 b_9$, 7 – $b_8 a_1$, 8 – $b_9 a_2$. Непомеченные цифрами дуги графа отражают последовательность букв в строке СЛОГа в порядке слева направо.

Ввод СЛОГа в компьютер удобно выполнять на основе приема, позволяющего сократить число перенастроек клавиатуры: на первом шаге формируется строка из букв a_i, b_j , на втором шаге расставляются стрелки, на третьем вводятся номера стрелок.

2.2. Операции над СЛОГаами

Рассмотрим элементарные операции над СЛОГаами, подобные тем, которые выполняются при графохирургических преобразованиях на графах (раздел 1.5 статьи). Алгоритм преобразования СЛОГа состоит из следующих шагов.

1. *Формирование связок синхронизированных переходов* $b_i = \{b_i^1, \dots, b_i^k\}$, где компоненты связки срабатывают одновременно, и только одновременно. В качестве примера рассмотрим СЛОГ (5), в котором заместим связками переходы b_2 и b_7 :

$$s_2 = \downarrow^7 a_1 b_1 \downarrow^8 a_2 \{b_2^1, b_2^2, b_2^3\} \uparrow^1 \downarrow^2 a_3 b_3 a_5 \uparrow^3 b_4 \uparrow^2! \downarrow^1 \downarrow^4 a_4 b_5 a_6 \uparrow^5 b_6 \uparrow^4! \downarrow^3 \downarrow^5 \{b_7^1, b_7^2, b_7^3\} a_7 \uparrow^6 b_8 \uparrow^7! \downarrow^6 b_9 \uparrow^8 \quad (6)$$

2. *Преобразование связок.* В связке выделяются компоненты, которые будут играть роль источников и стоков для подСЛОГов, образованных в результате разбиения СЛОГа на части (источники и стоки – это переходы, ведущие в подСЛОГ и из него). Стоки и источники в связке будем разделять косой чертой, слева от которой помещаются стоки, а справа – источники. Если непосредственно перед открывающей фигурной скобкой связки (после закрывающей скобки) проставлены входные (выходные) стрелки, то эти стрелки переносятся внутрь скобок, соответственно перед стоками или после источников. В СЛОГе (6) в первой по порядку связке переход b_2^1 объявляем стоком, а b_2^2, b_2^3 – источниками. Стрелку $\uparrow^1$ помещаем внутрь скобок после источника b_2^3 , за источником b_2^2 следует буква a_3 , помещенная в строке СЛОГа справа от закрывающей скобки связки. Во второй связке перед открывающей скобкой проставлены входные стрелки $\downarrow^3, \downarrow^5$, которые помещаем внутри скобок перед стоками b_7^2, b_7^3 . Источник b_7^1 не имеет предшественника. В итоге получаем СЛОГ:

$$s_3 = \downarrow^7 a_1 b_1 \downarrow^8 a_2 \{b_2^1 / b_2^2 b_2^3 \uparrow^1\} \downarrow^2 a_3 b_3 a_5 \uparrow^3 b_4 \uparrow^2! \downarrow^1 \downarrow^4 a_4 b_5 a_6 \uparrow^5 b_6 \uparrow^4! \{\downarrow^3 b_7^2 \downarrow^5 b_7^3 / b_7^1\} a_7 \uparrow^6 b_8 \uparrow^7! \downarrow^6 b_9 \uparrow^8 \quad (7)$$

3. *Разбиение СЛОГа на подСЛОГи.* ПодСЛОГи выделяются из СЛОГа на основе следующего алгоритма. В СЛОГе фиксируются начальный и конечный символы, которые могут быть как буквой, так и стрелкой. В первом подСЛОГе к начальному символу СЛОГа справа приписывается следующий символ. Если он является стоком данного подСЛОГа или конечным символом СЛОГа, то формирование подСЛОГа завершается. В противном случае приписывается следующий символ, и процедура повторяется. Для последующих подСЛОГов применяется тот же алгоритм с использованием в качестве начального символа соответствующего источника.

Например, для СЛОГа (7) получаем четыре подСЛОГа:

$$\begin{aligned} s_4^1 &= \downarrow^7 a_1 b_1 \downarrow^8 a_2 b_2^1 \\ s_4^2 &= b_2^2 \downarrow^2 a_3 b_3 a_5 \uparrow^3 b_4 \uparrow^2! \downarrow^3 b_7^2 \\ s_4^3 &= b_2^3 \uparrow^1 \downarrow^1 \downarrow^4 a_4 b_5 a_6 \uparrow^5 b_6 \uparrow^4! \downarrow^5 b_7^3 \\ s_4^4 &= b_7^1 a_7 \uparrow^6 b_8 \uparrow^7! \downarrow^6 b_9 \uparrow^8 \end{aligned} \quad (8)$$

Заметим, что подСЛОГи s_4^2, s_4^3 являются замкнутыми, т.к. все стрелки ведут в тот же подСЛОГ. Для s_4^1, s_4^4 это не так, последний подСЛОГ взаимодействует с первым по стрелкам 7 и 8. Пара расположенных рядом стрелок с номером 1 в третьем подСЛОГе, полученная формально по алгоритму, является излишней и может быть удалена.

4. *Удаление и вставка подСЛОГов.* В наборе подСЛОГов (8) удаляем s_4^2 и замещаем его подСЛОГом $s_4^5 = b_{10} a_8 b_{11} a_9 b_{12}$ с введением связок $b_{13} = \{b_2^1 / b_{10} b_2^3\}$, $b_{14} = \{b_{12} b_7^3 / b_7^1\}$, где b_2^1, b_{12}, b_7^3 – стоки подСЛОГов s_4^1, s_4^5, s_4^3 , а b_{10}, b_2^3, b_7^1 –

источники подСЛОГов s_4^5 , s_4^3 , s_4^4 соответственно. В результате получаем преобразованный СЛОГ, дополненный указанными выше связками:

$$s_5 = \downarrow^7 a_1 \ b_1 \ \downarrow^8 a_2 \ \{b_2^1/b_{10} \ b_2^3 \uparrow^9\} \ a_8 \ b_{11} \ a_9 \ \uparrow^{10} \downarrow^9 \downarrow^4 \ a_4 \ b_5 \ a_6 \ \uparrow^5 b_6 \ \uparrow^4 \downarrow^5 \ \{\downarrow^{10} b_{12} \ b_7^3/b_7^1\} \ a_7 \ \uparrow^6 b_8 \ \uparrow^7 \downarrow^6 b_9 \ \uparrow^8 \quad (9)$$

5. «Сшивание» связок и замещение их одним переходом. В описываемом примере СЛОГ (9) преобразуется в СЛОГ (10), для которого очевидным образом может быть построен граф с неструктурированными переходами.

$$s_6 = \downarrow^7 a_1 \ b_1 \ \downarrow^8 a_2 \ b_{13} \ \uparrow^9 a_8 \ b_{11} \ a_9 \ \uparrow^{10} \downarrow^9 \downarrow^4 \ a_4 \ b_5 \ a_6 \ \uparrow^5 b_6 \ \uparrow^4 \downarrow^5 \downarrow^{10} b_{14} \ a_7 \ \uparrow^6 b_8 \ \uparrow^7 \downarrow^6 b_9 \ \uparrow^8 \quad (10)$$

Моделирование развития триадных сетей происходит в тесном взаимодействии человеческого и компьютерного факторов. Рутинные задачи выполняются компьютерами, а творческие, в первую очередь принятие решений, решаются человеком. Чем в большей степени автоматизирован процесс моделирования, т.е. чем существеннее «компьютерная доля», тем больше возможностей имеет человек для решения творческих прорывных задач. В этом контексте описанный в статье графохирургический подход к преобразованию триадных сетей, базирующийся на элементарных операциях введения связок переходов, «разрезания» связок, удаления и вставки фрагментов сети, «сшивания» связок и т.д., конструктивно поддержанный алгеброй СЛОГов, может способствовать повышению уровня автоматизации моделирования триадных сетей.

3. Программная поддержка моделирования триадных сетей

Для повышения уровня автоматизации моделирования триадных сетей было разработано специальное программное обеспечение для их интерактивного имитационного моделирования. Данное программное обеспечение разработано А.В. Чуевым под руководством проф. В.З. Магергута и при участии научного консультанта разработки проф. С.А. Юдицкого. Описание функциональности этого программного обеспечения представлено в работах [30, 31].

Заключение

Предполагается, что автоматизированная система имитационного моделирования нужна для анализа поведения моделируемой сетевой структуры (в т.ч. определения временных и ресурсных характеристик, конфликтных ситуаций, «узких мест» и т.д.) и прогнозирования процессов развития моделируемой сетевой структуры, с определением её возможных конфигураций и динамики их преобразования. Это нужно для того, чтобы ЛПМ (Лицо, Проводящее Моделирование) могло понять динамику выбранной сетевой структуры, оказать на нее управляющие воздействия и изменить направление ее развития.

Имитационное моделирование процесса развития сетевой структуры реализуется в форме диалога введенной модели с ЛПМ согласно следующей схеме:

1) на каждом временном интервале ЛПМ контролирует функционирование модели и фиксирует полученные данные: о достижении целей, о тенденциях изменения параметров, о месте и времени нарушений, например, выходе параметров за допустимые пределы, о преобразованиях графов (на основе базовой операции «замещение фрагмента графа»), установлении в сети непредусмотренных циклов и т.д.;

2) ЛПМ анализирует эти данные и принимает решение о моменте следующего изменения модели и необходимых преобразованиях графов (на основе базовой хирургической операции замещения фрагмента графа).

Вышеупомянутые предположения дают возможность сформировать требования к возможностям автоматизированной системы имитационного моделирования триадных (и бинарных) сетей. Общее требование состоит в проведении при помощи автоматизированной системы *управляемых и воспроизводимых имитационных экспериментов*. На практике требование сводится к выбору того или иного вида имитационного моделирования (в нашем случае выбрано агентное моделирование). Перечислим концептуальные требования.

1. Адекватное представление выбранной моделируемой сетевой структуры и внешней по отношению к ней среды.
2. Реактивная архитектура агента, при которой агент управляет своим поведением, определяя текущую ситуацию, при заданных целях и способах их достижения.
3. Управление жизненным циклом агентов предполагает, в частности, создание, инициализацию, взаимодействие и уничтожение агентов.
4. Оповещение ЛПМ о нарушениях в функционировании моделируемой сетевой структуры.
5. Обеспечение анализа сетевой структуры как на уровне всей сети в целом, так и на уровне отдельно взятого агента. Предполагается расчет параметров как сети, так и агентов. Более того, представляет интерес анализ динамики значений параметров с использованием методов статистического анализа и методов анализа сетей.
6. Обеспечение интерактивного взаимодействия с ЛПМ. ЛПМ должен иметь возможность не только исследовать результаты имитационного моделирования, но и иметь возможность во время выполнения имитационного моделирования изменять параметры как агента (управлять целями, действиями и параметрами), так и сети в целом (например, управлять внешними событиями).
7. Визуализация основных показателей и формирование отчетов по результатам анализа.

Литература

1. Адизес И. Как преодолеть кризисы менеджмента? – М.: Юнити, 2005. – 189 с.
2. Блюмин С. Л. Оргиперграфы: матричное представление // Управление большими системами, 2010, № 30.1. С. 22-39.

3. Воронин А. А., Губко М. В., Мишин С. П., Новиков Д. А. Математические модели организаций. – М.: Ленанд, 2008. – 359 с.
4. Кульба В. В., Кононов Д. А., Чернов И. В., Рощин П. Е., Шулигина О. А. Сценарное исследование сложных систем: анализ методов группового управления // Управление большими системами. 2010. № 30.1. С. 154-186.
5. Бурков В. Н., Буркова И. В. Метод сетевого программирования в задачах управления проектами // Управление большими системами. 2010. № 30.1. С. 40-61.
6. Ларичев О. И. Теория и методы принятия решений, а также хроника событий в Волшебных Странах. – М.: Логос, 2000. 296 с.
7. Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и связь, 1993. 278 с.
8. Губанов Д. А., Новиков Д. А., Чхартишвили А. Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Физматлит, 2010. – 244 с.
9. Амбарцумян А. А. Сетецентрическое управление на сетях Петри в структурированной дискретно-событийной системе // Управление большими системами. 2010. № 30.1. С. 506-535.
10. Городецкий В. И., Грушинский М. С., Хабалов А. В. Многоагентные системы (обзор) // Новости искусственного интеллекта. 1998. № 2. С. 64-116.
11. Затуливетер Ю. С., Фищенко Е. А. Графодинамические системы с сетецентрическим управлением в математически однородном поле компьютерной информации // Управление большими системами. 2010. № 30.1. С. 567-604.
12. Юдицкий С. А., Владиславлев П. Н., Точ Д. С. Триадный подход к моделированию систем сетецентрического управления // Управление большими системами. 2010. № 28. С. 24-39.
13. Анишев П. А. Редуцируемость сетей Петри // Программирование. 1982. №4. С. 36-43.
14. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. 352 с.
15. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 265 с.
16. Котов В. Е. Сети Петри. – М.: Наука, 1984. – 160 с.
17. Юдицкий С. А., Владиславлев П. Н. Основы предпроектного анализа организационных систем. – М.: Финансы и статистика, 2005. – 144 с.
18. Юдицкий С. А., Магергут В. З. Логическое управление дискретными процессами. – М.: Машиностроение, 1987. – 176 с.
19. Марк Г., МакГоуэн К. Методология структурного анализа и проектирования. – М.: МетаТехнология, 1993. – 231 с.
20. Закревский А. Д. Параллельные алгоритмы логического управления. – М.: Едиториал УРСС, 2003. – 200 с.

21. Кузнецов О. П., Кулинич А. А., Марковский А. В. Анализ влияний при управлении слабоструктурированными ситуациями на основе когнитивных карт // Человеческий фактор в управлении. – М.: КомКнига, 2006. С. 313-344.
22. Максимов В. И. Структурно-целевой анализ развития социально-экономических ситуаций // Проблемы управления. 2005. № 3. С. 30-38.
23. Робертс Ф. С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. – М.: Наука, 1986. – 496 с.
24. Айзерман М. А., Гусев Л. А., Петров С. В., Смирнова И. Н. Динамический подход к анализу структур, описываемых графами (основы графодинамики) // Автоматика и телемеханика. 1977. № 7. С. 135-151.; № 9. С. 123-136.
25. Дернер Д. Логика неудачи. – М.: Смысл, 1997. – 236 с.
26. Юдицкий С. А. Моделирование динамики многоагентных триадных сетей. – М.: Синтег, 2012. – 112 с.
27. Таль А. А., Юдицкий С. А. Иерархия и параллелизм в сетях Петри I, II // Автоматика и телемеханика. 1982. № 7. С. 113-122; № 9. С. 83-88.
28. Семенов А. С. Фрактальные развивающиеся архитектуры // Управление большими системами. 2010. № 30.1. С. 91-103.
29. Юдицкий С. А. Графодинамическое имитационное моделирование развития сетевых структур // Управление большими системами. 2011. № 33. С. 21-34.
30. Юдицкий С. А., Магергут В. З., Чуев А. В. Бинарные сетевые дорожные карты процессов управления проектами // Приборы и системы. Управление, контроль, диагностика. 2013. № 4. С. 1-9.
31. Юдицкий С. А., Магергут В. З., Чуев А. В. Программно-алгоритмическое обеспечение моделирования процессов на бинарных индикаторных сетях // Приборы и системы. Управление, контроль, диагностика. 2014. № 9. С. 10-17.

References

1. Adizes I. *Kak preodolet' krizisy menedzhmenta?* [How to Solve the Mismanagement Crisis]. Moscow, Iuniti Publ., 2005. 189 p. (in Russian).
2. Blumin S. L. Dihypergraphs: Matrix Representations. *Large-scale Systems Control*, 2010, no. 30.1, pp. 22-39 (in Russian).
3. Voronin A. A., Gubko M. V., Mishin S. P., Novikov D. A. *Matematicheskie Modeli Organizatsii* [Mathematical Models of Organizations]. Moscow, Lenand Publ., 2008, 359 p. (in Russian).
4. Kulba V. V., Kononov D. A., Chernov I. V., Roshchin P. E., Shuligina O. A. Scenario-Based Research of Complex Systems: Analysis of Group Management Methods. *Large-scale Systems Control*, 2010, no. 30.1, pp. 154-186 (in Russian).
5. Burkov V. N., Burkova I. V. Network Programming in Project Management. *Large-scale Systems Control*, 2010, no. 30.1, pp. 40-61 (in Russian).

6. Larichev O. I. *Teoriia i metody priniatiia reshenii, a takzhe khronika sobytii v Volshebnykh Stranakh* [Theory and Methods of Decision-Making, and Also Chronicle of Events in Magic Countries]. Moscow, Logos Publ., 296 p. (in Russian).
7. Saati T. *Priniatie reshenii. Metod analiza ierarkhii* [Decision-Making. Method of Analysis of Hierarchies]. Moscow, Radio i Sviaz Publ., 1993. 278 p. (in Russian).
8. Gubanov D. A., Novikov D. A., Chkhartishvili A. G. *Sotsial'nye seti: modeli informatsionnogo vliianiia, upravleniia i protivoborstva* [Social Networks: Models of Information Influence, Control and Conflict]. Moscow, Izdatelsvo fiziko-matematicheskoi literatury, 2010. 228 p. (in Russian).
9. Ambartsumyan A. A. Network-Centric Control on Petri Nets for Structured Discrete Event System. *Large-scale Systems Control*, 2010, no. 30.1, pp. 506-535 (in Russian).
10. Gorodetskii V. I., Grushinskii M. S., Khabalov A. V. Mnogoagentnye sistemy (obzor) [Multi-Agent System (Review)]. *Novosti iskusstvennogo intellekta*, 1998, no. 2, pp. 64-116 (in Russian).
11. Zatuliveter Yu. S., Fischenko E. A. Graph-Dynamics Systems with Network-Centric Control in Mathematically Uniform Field of Computer Information. *Large-scale Systems Control*, 2010, no. 30.1, pp. 567-604 (in Russian).
12. Yuditskiy S. A., Vladislavlev P. N., Toch D. S. A Triad Approach to Network-Centric Control Systems Modelling. *Large-scale Systems Control*, 2010, no. 28, pp. 24-39 (in Russian).
13. Anishev P. A. Redutsiruemost' setei Petri [Reducyruet Petri Nets]. *Programming and Computer Software*, 1982, no. 4, pp. 36-43 (in Russian).
14. Tarasov V. B. Ot mnogoagentnykh sistem k intellektual'nym organizatsiiam: filosofiia, psikhologiiia, informatika [From Multiagent Systems to Intellectual Organizations: Philosophy, Psychology, Computer Science]. Moscow, Editorial URSS Publ., 2002. 352 p. (in Russian).
15. Peterson James L. *Petri net theory and the modeling of systems*. The University of Texas at Austin, 1981.
16. Kotov V. E. *Seti Petri* [Petri Nets]. Moscow, Nauka Publ, 1984, 160 p. (in Russian).
17. Yuditskiy S. A., Vladislavlev P. N. *Osnovy predproektnogo analiza organizatsionnykh system* [The Basics of Pre-Analysis of Organizational Systems]. Moscow, Finansy i Statistika Publ., 2005, 144 p. (in Russian).
18. Yuditskiy S. A., Magergut V. Z. *The logical control of discrete processes. Model, Analysis, synthesis*. Moscow, Mashinostroyeniye Publ., 1987. 176 p. (in Russian).
19. Mark G., MakGouen K. *Metodologiia strukturnogo analiza i proektirovaniia* [The Methodology of Structural Analysis and Designing]. Moscow, Metatekhnologiiia Publ., 1993. 231 p. (in Russian).
20. Zakrevskii A. D. *Parallel'nye algoritmy logicheskogo upravleniia* [Parallel Algorithms of Logical Control]. Editorial URSS Publ., 2003. 200 p. (in Russian).
21. Kuznetsov O. P., Kulinich A. A., Markovskii A. V. Analiz vliianii pri upravlenii slabostrukturirovannymi situatsiiami na osnove kognitivnykh kart

[Analysis of Effects in the Management of Semi-structured Situations Based on Cognitive Maps]. *Chelovecheskii faktor v upravlenii* [The Human Factor in Management], Moscow, KomKniga Publ., 2006, pp. 313-344 (in Russian).

22. Maximov V. I. The Structurally-Objective Analysis of Socio-Economic Situations Development. *Control Sciences*, 2005, no. 3, pp. 30-38 (in Russian).

23. Roberts F. S. Diskretnye matematicheskie modeli s prilozheniyami k sotsial'nym, biologicheskim i ekologicheskim zadacham [Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems]. Moscow, Nauka Publ., 1986. 496 p. (in Russian).

24. Aizerman M. A., Gusev L. A., Petrov S. V., Smirnova I. N. A Dynamic Approach to Analysis of Structures Represented as Graphs (Fundamentals of Graph Dynamics) *Automation and Remote Control*, 1977, no. 7, pp. 135-151; no. 9, pp. 123-136 (in Russian).

25. Derner D. *Logika neudachi* [The logic of failure]. Moscow, Smysl Publ., 1997. 236 p. (in Russian).

26. Yuditskiy S. A. *Modelirovanie dinamiki mnogoagentnykh triadnykh setej* [Modeling the dynamics of multi-agent triad networks]. Moscow, SINTEG Publ., 2012, 112 p. (in Russian).

27. Tal' A. A., Yuditskiy S. A. Hierarch and Parallelism Petri Nets. *Automation and Remote Control*, 1982, no. 7, pp. 113-122; no. 9, pp. 83-88 (in Russian).

28. Semenov A. S. Fractal Evolutionary Architectures. *Large-scale Systems Control*, 2010, no. 30.1, pp. 91-103 (in Russian).

29. Yuditskiy S. A. Graphodynamic Simulation Modeling of Network Structures Evolution. *Large-scale Systems Control*, 2011, no. 33, pp. 21-34 (in Russian).

30. Yuditskiy S. A., Magergut V. Z., Chuev A. V. Binary network roadmaps project management processes. *Instruments and Systems: Monitoring, Control, and Diagnostics*, 2013, vol. 4, pp. 1-9 (in Russian).

31. Yuditskiy S. A., Magergut V. Z., Chuev A. V. Software for Modeling Processes in Binary Indicator Networks. *Instruments and Systems: Monitoring, Control, and Diagnostics*, 2014, vol. 9, pp. 10-17 (in Russian).

Статья поступила 5 сентября 2016 г.

Информация об авторе

Юдицкий Семен Абрамович – доктор технических наук, профессор, главный научный сотрудник. Институт проблем управления имени В.А. Трапезникова Российской академии наук. Область научных интересов: системный анализ, логическое моделирование, организационно-технологические системы. E-mail: yuseab32@yandex.ru

Адрес: 117321, Москва, Профсоюзная ул. 132, кор. 8, Московский дом-пансионат ветеранов науки, ком. 305.

Graf-Dynamic Modeling of Organizational-Technical Systems Based on the Triad Agents

S. A. Yuditskiy

Relevance. The network structures are the interacting agents that widely used in various subject areas. Agent behavior is determined by several key factors: the goal of the agent; its priorities; the external and internal situation; the actions to achieve the objectives; the main parameters of the agent, which determine its state. Goals, actions and parameters form a triad – «the triad agent». The structure of this agent has three subsystems, which are formalized based on the theory of Petri nets – the purposes graph, the actions graph and the parameters graph. The main graph is the action graph, which switches in discrete moments of time when certain logical conditions are satisfied. If the action graph takes switching then there is activation of a new action. This action may be the transition to a new action, or changing of the agent settings or changing of the agent goals. Thus, the triad agent is network with dynamic behavior, which allows describing a wide class of dynamic processes in organizational-technical systems. Thus the complex organizational-technical systems are characterized by a large number of parameters that are difficult to place in a graph of a single agent. However, the number of goals and actions in the organizational-technical system is substantially less than the number of its parameters. Thus, complex organizational-technical system, it is advisable to decompose into several parallel functioning triad agents (not more than 7-10). This decomposition ensures the visibility of the model. **The purpose of this paper** is creating the graf-dynamic model of the triad agents and methods of its research. This is achieved by the solution of the following subtasks. 1) Formalization and analysis of the goals graph, of the actions graph and the agent settings. 2) Formalization and analysis of the relations between the graphs in the agent. 3) Introduction of operations on the agent graphs (the "graph-surgery"). 4) Algebraic (symbolic) graph representation of the agent with the transition from a description in the form of a graph to a symbolic description and back. 5) Formalization and analysis of the relations between agents within the multi-agent triad network. **Scientific novelty of the paper** is the creation of a new way of building models of organizational-technical system that will simplify its simulation and improve its visibility. In addition, the elements of novelty of the paper include: the research of the mechanism functioning of the triad agents; the rationale of the new method of converting the agent as the "graph-surgery"; the development of an algebraic graph representation as a line of characters (the language of structured Logical Descriptions of Graphs), and operations on them. **Practical relevance of the paper** is the modeling principles which is used in the software that developed in the Belgorod state technological University n. a. V. G. Shukhov. This software is intended for use by experts for simulation-agent modeling of network structures, and for using at industrial applications.

Key words: Graf-dynamic model, triad agent, triad agents net, goal setting, logical control, interference parameters, the mechanism of interaction graphs, graph-surgery, algebra of graphs, language structured Logical Descriptions of Graphs, software support simulation.

Information about Author

Semen Abramovich Yuditskiy – Dr. habil. of Engineering Sciences, Professor. Chief Researcher. V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences. Field of research: system analysis, logic modeling, organizational and technological systems. E-mail: yuseab32@yandex.ru

Address: Russia, 117321, Moscow, Profsoyuznaya street, 132, Bld. 8, Ft. 305, Moscow Pension House of Science Veterans.