

УДК 004.832.32

С.А. Зинкин, д-р техн. наук, профессор, Пензенский государственный университет (Россия, 440026, Пенза, ул. Красная, 40) (e-mail: zsa49@yandex.ru)

Мустафа Садек Джафар, аспирант, Пензенский государственный университет (Россия, 440026, Пенза, ул. Красная, 40) (e-mail: mustafajaafar221087@gmail.com)

Н.С. Карамышева, канд. техн. наук, доцент, Пензенский государственный университет (Россия, 440026, Пенза, ул. Красная, 40) (e-mail: karamyshevans@yandex.ru)

КОНЦЕПТУАЛЬНЫЕ ПРЕДСТАВЛЕНИЯ И МОДИФИКАЦИИ СЕТЕЙ ПЕТРИ ДЛЯ ПРИЛОЖЕНИЙ В ОБЛАСТИ СИНТЕЗА ФУНКЦИОНАЛЬНОЙ АРХИТЕКТУРЫ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ПЕРЕМЕННОЙ СТРУКТУРОЙ

Актуальность и цель. Объектом исследования является функциональная архитектура распределенных вычислительных систем с переменной (реконфигурируемой) структурой, характерной для гибридных систем облачно-сетевого (эрид) типа. Несмотря на то, что сети Петри издавна исследуются как в теоретическом, так и в практическом планах, методы их интерпретации продолжают интенсивно развиваться. В настоящее время в недостаточной степени исследована проблема вложения сетей Петри в архитектуру распределенных сетевых приложений, используемых для реализации глобальных вычислений в современных смешанных облачных, эрид и кластерных системах. Показано, что в современных исследованиях сети Петри используются в основном при моделировании дискретных систем и процессов, а не в качестве основы для формализованных спецификаций при разработке распределенных приложений. В этой связи актуальной является интерпретация сетей Петри в приложениях к функциональной архитектуре распределенных вычислительных систем с переменной структурой, основанной на сетевом программном обеспечении промежуточного класса (класса *middleware*). Целью работы являлась интеграция графических представлений концептуальных графов, семантических сетей, сценариев и сетей Петри, что позволило создать эффективный инструментарий с графической поддержкой для проектирования функциональной архитектуры распределенных вычислительных систем с переменной структурой и, в частности, облачной архитектуры вида *NCaaSOD* – *Network Computing as a Service on Demand* (сетевые вычисления как облачный сервис по требованию пользователя).

Материалы и методы. Использованы концептуальные модели распределенных процессов, являющиеся графической интерпретацией исчисления предикатов первого порядка. Предложены концептуальные графы для распределенных сетей Петри смешанного типа, которые позволяют описывать с целью последующей реализации вычислительные процессы в глобальных вычислительных сетях.

Результаты. На основе интеграции графических представлений концептуальных графов, семантических сетей, сценариев и сетей Петри предложены концептуальные представления распределенных реконфигурируемых сетей Петри, позволяющие реализовать их непосредственное вложение в архитектуру вычислительной сети.

Выводы. Предложены новые концептуально-поведенческие модели на основе концептуальных графов распределенных сетей Петри для определения системной и функциональной архитектур распределенных вычислительных систем с переменной структурой, предоставляемой пользователю по его требованию в качестве гибридного облачно-сетевого сервиса; данные модели отличаются возможностью оперативной реконфигурации и непосредственной исполнимостью. Предложена и формализована методика вложения концептуальных сетей Петри в архитектуру облачно-сетевых компьютерных систем типа *NCaaSOD* – сетевые (облачные) вычисления как сервис, организуемый по запросу пользователя. Предложены правила получения отношений связности между позициями и переходами сети Петри, размещаемыми на узлах физической компьютерной сети.

Ключевые слова: распределенные сетевые вычисления; функциональная архитектура; концептуальные сети Петри смешанного типа; реконфигурация сетевых моделей; вложение сетей Петри в архитектуру компьютерной сети.

DOI: 10.21869/2223-1560-2018-22-6-143-167

Ссылка для цитирования: Зинкин С.А., Мустафа Садек Джафар, Карамышева Н.С. Концептуальные представления и модификации сетей Петри для приложений в области синтеза функциональной архитектуры распределенных вычислительных систем с переменной структурой // Известия Юго-Западного государственного университета. 2018. Т. 22, № 6(81). С. 143-167.

Введение

При реализации сетевых приложений, как основы реализации функциональной архитектуры распределенных вычислительных систем с переменной (реконфигурируемой) структурой, возможно использовать различные модификации сетей взаимодействующих автоматов, сетей Петри и других исполнимых сетевых моделей.

Сеть Петри является одной из широко известных и распространенных моделей распределенных и параллельных систем. Сеть Петри представляет собой ориентированный двудольный граф с вершинами, называемыми переходами и позициями. Позиции в сети представляют условия и могут содержать метки. Дуги в таком графе показывают, какие пре- и постусловия соответствуют переходам, представляющим события, происходящие в сети. Сети Петри были первоначально определены К. А. Петри [1]. Приведем распространенное определение сетей Петри с кратными дугами, например, из работы [2], которое в дальнейшем потребуется при определении новых модификаций сетей Петри и их сетевых интерпретаций в компьютерной сетевой архитектуре:

“Сеть Петри формально определяется как пятерка

$$PN = (P, T, F, H, M_0),$$

где $P = \{p_1, p_2, \dots, p_m\}$ – конечное множество позиций;

$T = \{t_1, t_2, \dots, t_n\}$ – конечное множество переходов, причем $P \cup T \neq \emptyset$, и $P \cap T = \emptyset$;

$F: P \times T \rightarrow N$ – “входная” функция инцидентности, определяющая число дуг, ведущих от позиций к переходам; где N –

множество неотрицательных целых чисел;

$H: T \times P \rightarrow N$ – “выходная” функция, определяющая число дуг, ведущих от переходов к позициям;

$M_0: P \rightarrow N$ – функция начальной разметки”.

Работа сети Петри определяется числом и распределением меток в позициях. Переход готов к срабатыванию при условии $(\forall p \in P)[M(p) \geq F(p, t)]$. После срабатывания перехода текущая разметка сети Петри изменяется следующим образом: $(\forall p \in P)[M'(p) = M(p) - F(p, t) + H(t, p)]$.

Аналогичные определения сетей Петри даны и в других работах, например в [3, 4]. Изменение разметки позиций легко воспроизводится в имитационных моделях сетей Петри. В приложениях к компьютерным сетям большой интерес представляют ординарные сети Петри (без кратных дуг). Для ординарных сетей Петри $F: P \times T \rightarrow \{0, 1\}$, $H: T \times P \rightarrow \{0, 1\}$. Для безопасных, или бинарных, сетей Петри, широко используемых в системах логического управления, далее будут использованы следующие предикаты $F: P \times T \rightarrow \{\text{true}, \text{false}\}$, $H: T \times P \rightarrow \{\text{true}, \text{false}\}$, $M_0: P \rightarrow \{\text{true}, \text{false}\}$.

Несмотря на то, что сети Петри издавна исследуются как в теоретическом, так и в практическом планах, методы их интерпретации продолжают интенсивно развиваться. Обширный обзор современных работ приведен в книге [5]. В настоящее время в недостаточной степени исследована проблема вложения сетей Петри в архитектуру распределенных сетевых приложений, используемых для реализации глобальных вычислений в смешанных облачных, грид и кластерных системах. Однако в современных исследо-

ваниях, например, на сайте [6] и в трудах конференций [5 – 11], сети Петри используются в основном при моделировании дискретных систем и процессов, а не в качестве основы для формализованных спецификаций при разработке распределенных приложений.

Модели, основанные на сетях Петри, обладают расширенными возможностями описания параллельно и асинхронно взаимодействующих подсистем сложных систем, в том числе и распределенных вычислительных систем с переменной структурой. Актуальной является интерпретация сетей Петри в приложениях к функциональной архитектуре распределенных вычислительных систем с переменной структурой, основанной на сетевом программном обеспечении промежуточного класса (класса *middleware*).

Основой для построения реконфигурируемых сетей Петри (РСП, или RPN – reconfigurable Petri net) являются обычные, живые и безопасные (1-ограниченные) сети Петри, не имеющие ограничений на конфигурации и кратность дуг в которых не превышает единицы. Основные сведения, касающиеся живости, безопасности и хорошей сформированности сетей РСП соответствуют принятым в работах [12, 13, 14].

В работах [14, 15] развиты технологии, основанные на применении логических сетей Петри для приложений в области микропрограммирования и промышленной автоматизации. Однако проблемы интерпретации сетей Петри для создания распределенных приложений в данных работах не затрагиваются. В отличие от данных работ, интерпретация РСП в дальнейшем может базироваться на продукционных и логико-алгебраических представлениях, пригодных для

использования бинарных и других типов сетей Петри как основы для технологии создания распределенных сетевых приложений.

Основные свойства сетей Петри были детально рассмотрены в работе [16], в том числе рассмотрены проблемы, связанные с построением графов достижимости и покрытия, с ограниченностью, живостью, консервативностью, *P*-инвариантами и *T*-инвариантами, параллелизмом, разрешением конфликтов и с решением других задач анализа свойств сетей Петри. Исследованию структурных свойств сетей Петри, посвящена работа [17].

В статье [18] рассматривается организация супервизорного управления в параллельных системах, поведение которых описано бинарными сетями Петри на основе использования структурного подхода. В этой статье рассматривается формальный подход к разработке средств, при помощи которых возможно автоматически проектировать супервизоры на основе предложенных формализованных спецификаций сетей Петри. Рассматриваются также проблемы обеспечения противодействия блокировкам и обеспечение живости сетей Петри.

В статье [19] авторы рассматривают различного рода модификации сетей Петри для обеспечения их возможностей моделирования самоорганизующихся сетей мобильной связи, и, в частности, модификации, позволяющие учитывать динамику моделируемых процессов. Особо отмечается, что решение данных проблем становится все более важным при организации современных коммуникационных сетей.

В статье [20] рассмотрены самомодифицируемые сети Петри, кардинальность дуг в которых зависит от марки-

ровки позиций. Цветные темпоральные, или временные, сети Петри с переменной структурой предложены в статье [21].

Управляемые сети Петри, рассмотренные в работе [22], представляют собой класс сетей Петри с внешними условиями включения, названными управляющими позициями, которые позволяют внешнему контроллеру управлять перемещением меток в сети. Впервые такие сети были введены в работах [23] и [24].

Авторы статьи [25] предложили аналогичную концепцию модифицированной бинарной сети Петри. Идея концепции состоит в использовании традиционных спусковых механизмов сетей Петри при ограничении, чтобы в каждой позиции сети было не более одной метки. Некоторые разновидности бинарных сетей Петри также рассматриваются в работе [26]. Однако в рассматриваемых сетях не используются ингибиторные и информационные дуги.

В статье [27] подтверждается тот факт, что моделирование адаптации системы к изменяющейся среде приобретает все большее значение. Области применения охватывают, например, скоординированную работу компьютеров в распределенной системе, многоагентные системы, динамические процессы поиска данных и сети с мобильными объектами. Основная идея сетевых преобразований в работах [27, 28] заключается в распространении классической теории сетей Петри на совокупность правил, которые позволяют моделировать изменения в сетевой структуре.

Работа [29] посвящена использованию системы Matlab при построении инструментального комплекса Pmediteur для исследования сетей Петри. Комплекс позволяет исследовать сети Петри на

ограниченность и недопущение тупиковых ситуаций. Метод, как и во многих других инструментальных комплексах, основан на построении графа достижимых состояний и анализе инвариантов в сетях Петри. Ряд полезных систем и инструментальных комплексов для исследования сетей Петри представлен на вебсайте [30]. Одним из апробированных инструментальных средств для работы с сетями Петри в настоящее время являются пакеты CPN Tools (URL: <http://cpntools.org>) и PIPE (URL: <http://pipe2.sourceforge.net>).

Недостатком методов, предложенных в перечисленных работах, является ограниченность интерпретации описанных модификаций сетей Петри для компьютерных сетей. В настоящей работе преследуется цель – предложить такие интерпретации реконфигурируемых сетей Петри, которые после этапа формализации могли бы служить исполнимыми спецификациями для создаваемого и перенастраиваемого сетевого программного обеспечения распределенных облачно-сетевых вычислительных систем.

В работах [31 – 39] уделяется большое внимание использованию аппарата сетей Петри в качестве языка спецификаций для создания программного обеспечения ЭВМ. Однако в данных работах практически не затрагиваются вопросы применения исполнимых спецификаций для создаваемого и перенастраиваемого сетевого программного обеспечения для современных распределенных облачно-сетевых вычислительных систем.

В книге [40] описаны ставшие уже традиционными вопросы генерации кодов программ на основе диаграмм языка UML 2. Целью настоящей работы является возможная генерация программ для распределенных приложений на основе концептуальных графов сетей Петри.

Предлагаемая в настоящей работе интеграция графических представлений концептуальных графов, семантических сетей, сценариев и сетей Петри позволяет создать эффективный инструментарий с графической поддержкой для проектирования функциональной архитектуры распределенных вычислительных систем с переменной структурой и, в частности, облачной архитектуры вида NCaaSOD – Network Computing as a Service on Demand (сетевые вычисления как облачный сервис по требованию пользователя).

1. Определение сетей Петри смешанного типа

Для многих приложений сетей Петри в области вычислительной техники необходимо определять несколько типов позиций. Например, возможно вводить позиции типа “очередь”, в которых может находиться несколько возможно упорядоченных меток, а также возможно использовать простые позиции, число меток в которых не превышает единицы. Такие позиции возможно использовать при необходимости отметки состояний или событий моделируемой системы. Поскольку далее будут использованы различные модификации подобных сетей; для систематизации представления дадим формальное определение смешанных сетей Петри (ССП или MPN-сети, англ. вариант: MPN – Mixed Petri Nets), основанное на отмеченных выше известных определениях обычных и бинарных, или логических, сетей Петри.

Смешанная (ординарная) сеть Петри представляется кортежем

$$MPN = (P, P_a, P_b, T, F_a, H_a, F_b, H_b, F_{Inh_a}, F_{Inh_b}, F_{Inf_a}, F_{Inf_b}, M_{0a}, M_{0b}), \quad (1)$$

где $P = P_a \cup P_b$ – конечное множество позиций двух типов;

$P_a = \{p_{a1}, p_{a2}, \dots, p_{am}\}$ – конечное множество позиций, в которых могут находиться по несколько меток (“арифметические” позиции);

$P_b = \{p_{b1}, p_{b2}, \dots, p_{bk}\}$ – конечное множество позиций, в которых могут находиться не более чем по одной метке (“логические” позиции);

$T = \{t_1, t_2, \dots, t_n\}$ – конечное множество переходов, причем $P \cup T \neq \emptyset$, и $P \cap T = \emptyset$;

$F_a: P_a \times T \rightarrow \{\text{true}, \text{false}\}$ – “входная” высказывательная функция инцидентности (бинарный предикат), определяющая дуги, ведущих от позиций из множества P_a к переходам;

$H_a: T \times P_a \rightarrow \{\text{true}, \text{false}\}$ – “выходная” высказывательная функция (бинарный предикат), определяющая дуги, ведущих от переходов к позициям из множества P_a ;

$F_b: P_b \times T \rightarrow \{\text{true}, \text{false}\}$ – “входная” высказывательная функция инцидентности (бинарный предикат), определяющая дуги, ведущих от позиций из множества P_b к переходам;

$H_b: T \times P_b \rightarrow \{\text{true}, \text{false}\}$ – “выходная” высказывательная функция (бинарный предикат), определяющая дуги, ведущих от переходов к позициям из множества P_b ;

$M_{0a}: P_a \rightarrow N$ – функция начальной разметки позиций из множества P_a , где N – множество неотрицательных целых чисел;

$M_{0b}: P_b \rightarrow \{\text{true}, \text{false}\}$ – функция начальной разметки позиций из множества P_b ;

$F_{Inh_a}: P_a \times T \rightarrow \{\text{true}, \text{false}\}$ – ингибиторная высказывательная функция инцидентности (бинарный предикат), определяющая ингибиторные (сдерживающие) дуги, ведущие от позиций из множества

P_a к переходам; данные дуги на графе сети Петри изображаются стрелками с кружками на концах; ингибиторная дуга, связывающая позицию $p_i \in P_a$ с переходом t_j считается активированной, то есть разрешающей срабатывание инцидентного ей перехода, при $M_a(p_i) = 0$; после срабатывания перехода t_j разметка $M_a(p_i)$ позиции p_i не изменяется;

$F_{Inh_b}: P_b \times T \rightarrow \{\text{true}, \text{false}\}$ – ингибиторная высказывательная функция инцидентности (бинарный предикат), определяющая ингибиторные (сдерживающие) дуги, ведущие от позиций из множества P_b к переходам; данные дуги на графе сети Петри, как и в предыдущем случае, изображаются стрелками с кружками на концах; ингибиторная дуга, связывающая позицию $p_m \in P_b$ с переходом t_j считается активированной, то есть разрешающей срабатывание инцидентного ей перехода, при $M_b(p_m) = \text{false}$; после срабатывания перехода t_j разметка $M_b(p_m)$ позиции p_m не изменяется;

$F_{Inf_a}: P_a \times T \rightarrow \{\text{true}, \text{false}\}$ – информационная высказывательная функция инцидентности (бинарный предикат), определяющая информационные дуги, ведущие от позиций из множества P_a к переходам; данные дуги на графе сети Петри изображаются стрелками с зачерненными кружками на концах; информационная дуга, связывающая позицию $p_k \in P_a$ с переходом t_j считается активированной, то есть разрешающей срабатывание инцидентного ей перехода, при $M_a(p_k) \neq 0$; после срабатывания перехода t_j разметка $M_a(p_k)$ позиции p_k не изменяется;

$F_{Inf_b}: P_b \times T \rightarrow \{\text{true}, \text{false}\}$ – информационная высказывательная функция инцидентности (бинарный предикат), определяющая информационные (сдерживающие) дуги, ведущие от позиций из

множества P_b к переходам; данные дуги на графе сети Петри, как и в предыдущем случае, изображаются стрелками с зачерненными кружками на концах; информационная дуга, связывающая позицию $p_n \in P_b$ с переходом t_j считается активированной, то есть разрешающей срабатывание инцидентного ей перехода, при $M_b(p_n) = \text{true}$; после срабатывания перехода t_j разметка $M_b(p_n)$ позиции p_n не изменяется.

В качестве иллюстрирующего примера рассмотрим срабатывание перехода t_1 с входными связями всех шести типов (входные позиции $p_1 \in P_a$, $p_2 \in P_b$, $p_3 \in P_a$, $p_4 \in P_b$, $p_5 \in P_a$ и $p_6 \in P_b$) и с выходными связями двух типов (выходные позиции $p_7 \in P_a$ и $p_8 \in P_b$). Входные и выходные связи для перехода t_1 задаются истинностью следующих высказываний:

$$\begin{aligned} F_a(p_1, t_1) &= \text{true}, F_b(p_2, t_1) = \text{true}, \\ F_{Inh_a}(p_3, t_1) &= \text{true}, F_{Inh_b}(p_4, t_1) = \text{true}, \\ F_{Inf_a}(p_5, t_1) &= \text{true}, F_{Inf_b}(p_6, t_1) = \text{true}, \\ H_a(t_1, p_7) &= \text{true}, H_b(t_1, p_8) = \text{true}. \end{aligned}$$

Продукционное правило (2) срабатывания перехода t_1 имеет следующий вид:

$$\begin{aligned} r_{t_1}: (M_a(p_1) > 0) \& M_b(p_2) \& (M_a(p_3) = 0) \& \\ & \& M_b(p_4) \& (M_a(p_5) > 0) \& M_b(p_6) \rightarrow \\ & \rightarrow M_a(p_1) := M_a(p_1) - 1, M_b(p_2) := \text{false}, \\ & M_a(p_7) := M_a(p_7) + 1, M_b(p_8) := \text{true}. \quad (2) \end{aligned}$$

Определенные таким образом сети ССП возможно использовать при построении концептуальных и исполнимых моделей распределенных вычислительных систем с переменной структурой, а также различных моделей обмена данными между узлами в вычислительных сетях.

2. Концептуальные графы для распределенных сетей Петри

В отличие от известных работ, посвященных интерпретациям сетей Петри, в данной работе предлагается концептуальное представление сетей Петри, поз-

воляющее определить все отношения между переходами и позициями и роли данных объектов применительно к конкретной предметной области – функциональной архитектуре распределенных вычислительных систем с переменной структурой, реализуемой в рамках архитектуры NСaaS в сетевой компьютерной среде. В связи с тем, что ранее было дано определение сетей Петри смешанного типа, дальнейшее рассмотрение концептуальных представлений сетей Петри целесообразно рассмотреть на основе полного набора примеров.

В настоящей работе графика сетей Петри используется совместно с графикой концептуальных графов, что позволяет повысить информативность дискретных моделей вычислительных систем и процессов, протекающих в них. Расширяя понятия сетей Петри, возможно определять места размещения объектов, реализуемых в компьютерной сети, типы и свойства объектов. С другой стороны, в

семантических сетях естественно внедряется процедурная составляющая модели представления знаний, основанная на описании событий, ресурсов и процессов в сетях Петри логико-алгебраическими операционными выражениями. В процессе построения концептуальных графов использован графический редактор CharGer, нашедший широкое применение в международной практике [41, 42]. В литературе нередко термин «концептуальный граф» относят также и к семантической сети; строго говоря, семантическая сеть является суперпозицией нескольких концептуальных графов.

Концептуальные графы сетей Петри (КГ СП) построены на основе определения (1) для смешанных сетей Петри. Вначале рассмотрим простой пример концептуального графа CG_1 ординарной (без кратных дуг) «арифметической» сети Петри, представленного на рисунке 1.

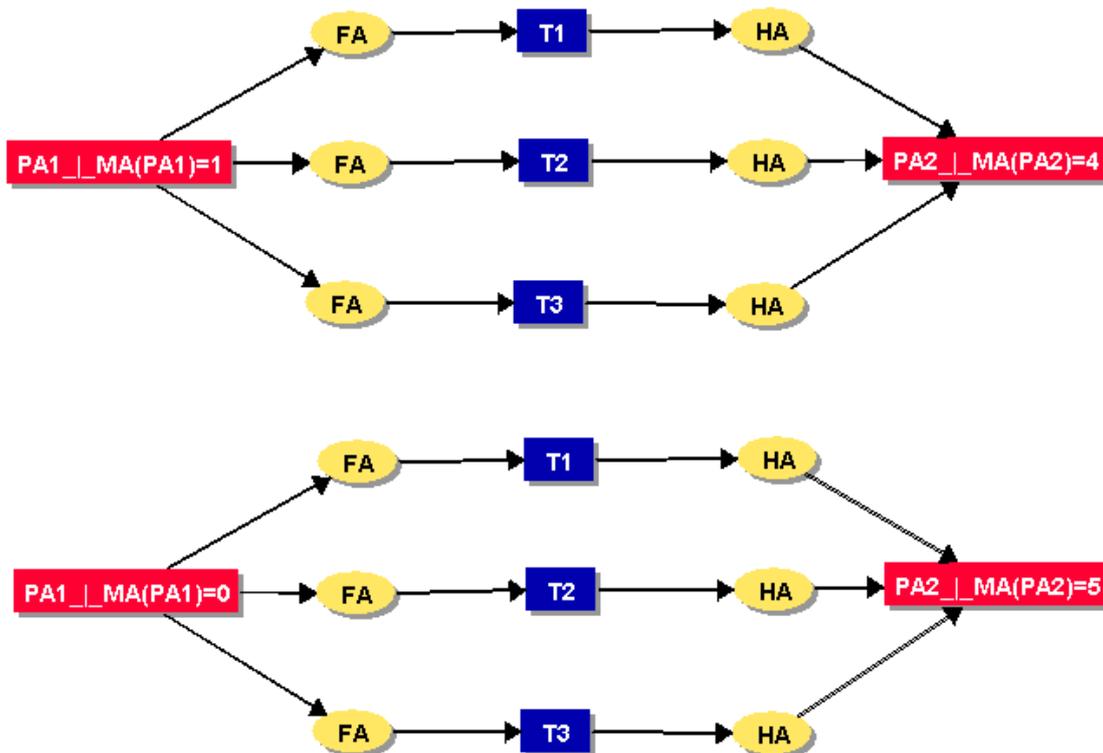


Рис. 1. Концептуальный граф CG_1 сети Петри до и после срабатывания одного из переходов

На данном рисунке концептами представлены “арифметические” позиции PA1, PA2 (с указанием текущей маркировки $MA(PA1)=1$ и $MA(PA2)=4$ до срабатывания одного из переходов в верхней части рисунка 1 и после срабатывания перехода $MA(PA1)=0$ и $MA(PA2)=5$ в нижней части рисунка 1) и переходы T1, T2, T3, связанные отношениями FA и HA с соответствующими позициями. Такие сети, в отличие от бинарных, или логических, условно названы “арифметическими”, поскольку областью значений функ-

ции разметки MA здесь является множество неотрицательных целых чисел. Вторая буква A в записи имен объектов – позиций и отношений, здесь и далее отражает этот факт.

Здесь и далее в концептуальных графах сетей Петри и в их описаниях по тексту используются имена объектов – позиций, переходов и отношений связности, соответствующие возможностям редактора CharGer. Как на рисунках, так и в формулах принято использовать одинаковый прямой шрифт.

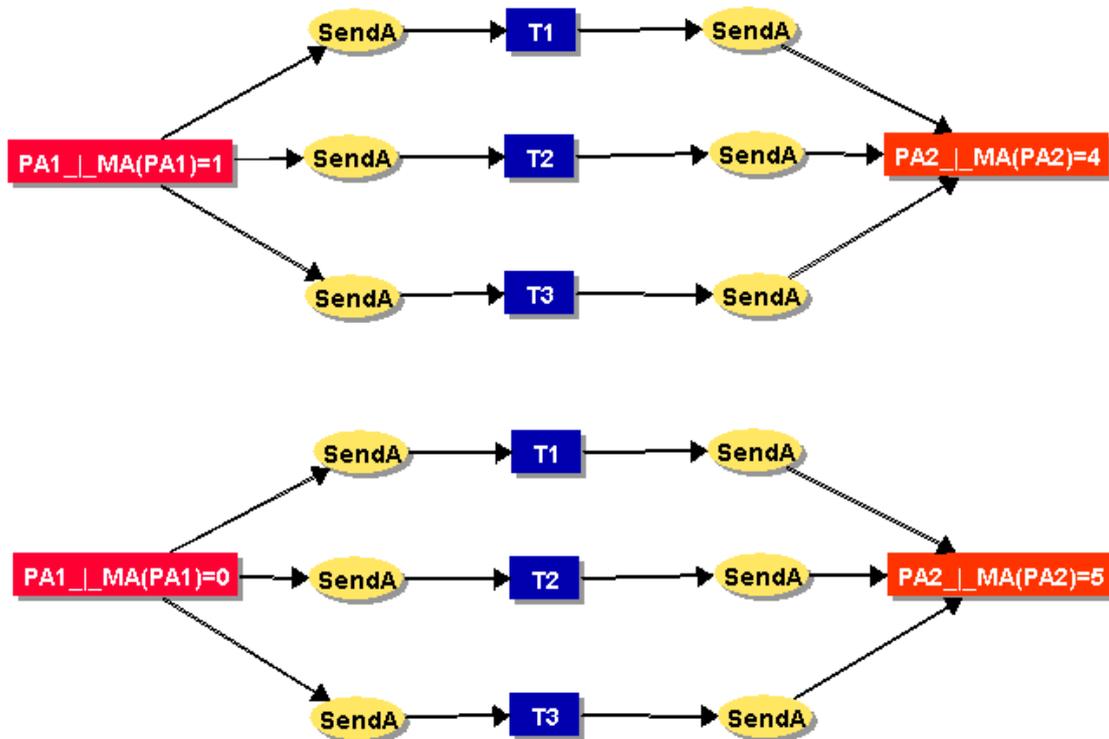


Рис. 2. Концептуальный граф CG_2 распределенной сети Петри, предназначенной для “вложения” в компьютерную сеть, до и после срабатывания одного из переходов

3. Методика отображения концептуальной сети Петри на архитектуру компьютерной сети

На рисунке 2 представлен концептуальный граф CG_2 распределенной сети Петри (КГ РСП), предназначенной для “вложения” в компьютерную сеть, до и после срабатывания одного из переходов. Его отличительной особенностью является

интерпретация отношения SendA, связывающего позиции и переходы. Например, истинность высказывания $SendA(PA1, T1)$ означает, что метка передается от позиции PA1 к переходу T1. В компьютерных сетях позиции и переходы представляются объектами соответствующих классов сетевого приложения, а метки представляются сообщениями.

Методику непосредственного отображения фрагмента другой концептуальной сети Петри в архитектуру TCP/IP сети иллюстрирует рисунок 3. Пусть некоторая сеть содержит переход $nt1$ с тремя входными $np1$, $np2$, $np3$ и одной выходной позицией $np4$. Отношения F и H задают виртуальные связи позиций и пере-

ходов. В реальной сети этим отношениям соответствуют реальные маршруты передачи сообщений-меток. На рисунке 3 указаны также условные IP-адреса для узлов TCP/IP сети, на которые отображены позиции и переходы концептуальной сети Петри.

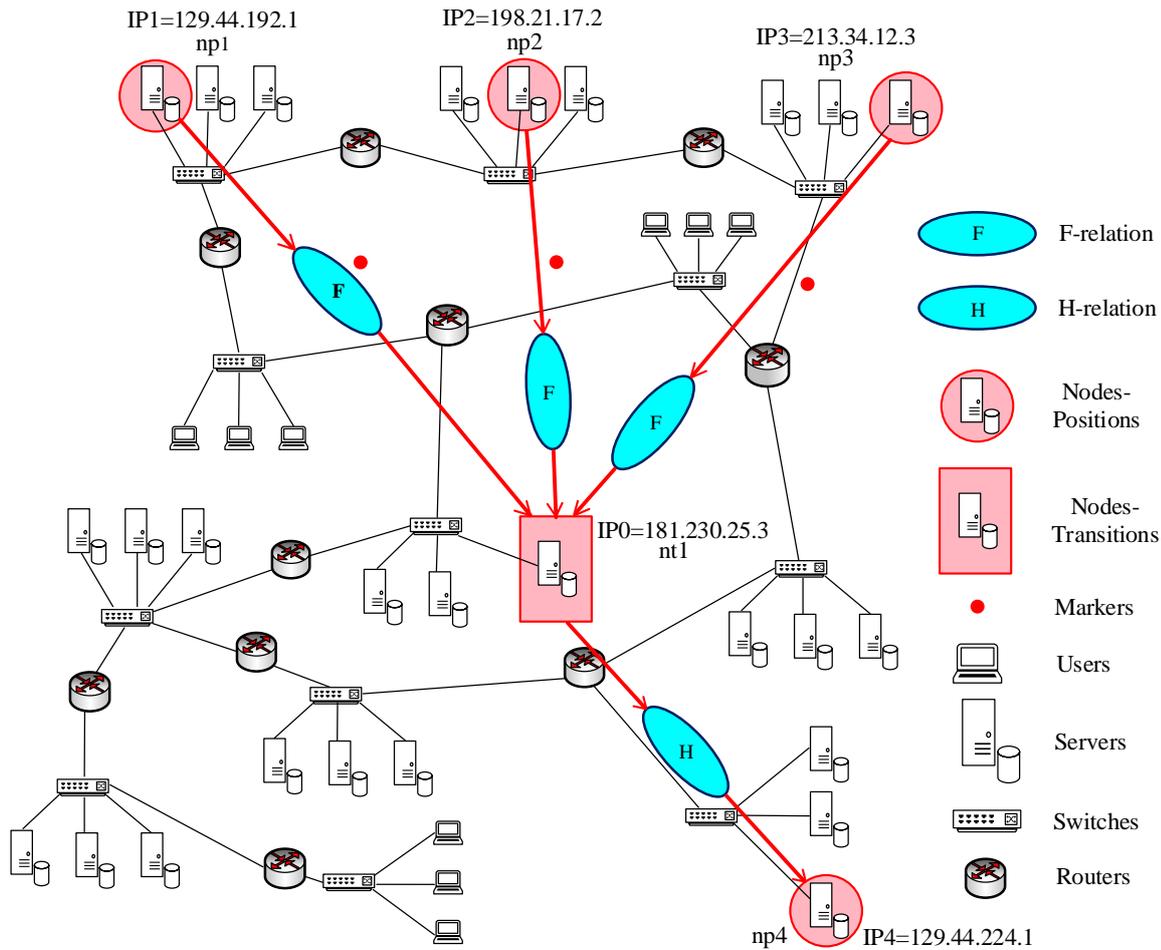


Рис. 3. Вложение концептуальной сети Петри в распределенную вычислительную систему на базе TCP/IP сети

Здесь стрелками представлены виртуальные дуги, связывающие позиции и переходы, а реальные пути перемещения меток в виде сообщений через маршрутизаторы и коммутаторы составной сети определяются сетевыми IP адресами узлов и таблицами маршрутизации. Размещение отдельных позиций и переходов, или целых фрагментов сети Петри на фи-

зических узлах компьютерной сети осуществляется аналогично размещению логических узлов распределенной вычислительной системы на реальной сетевой архитектуре.

Вложение логической структуры сети Петри в физическую осуществляется на основе выражений (3), (4), (5), (6) и (7) следующим образом. Провайдером об-

лачного сервиса, или автоматически, заранее формируется информационный объект на основе унарной функции

$$\text{Map}: P \cup T \rightarrow IPaddr, \quad (3)$$

где P – множество позиций размещаемой сети Петри, T – множество ее переходов; $IPaddr$ – множество IP-адресов физических узлов ТСП/IP сети, выделенных для реализации сервиса NaaS/D. Для перехода от логической структуры сети Петри к физической структуре сети Петри, “вложенной” в вычислительную сеть, определены следующие правила вывода:

$$(\forall p \in P)(\forall t \in T)[F(p, t) \rightarrow \rightarrow \text{PhSendPT}(\text{Map}(p), \text{Map}(t))]; \quad (4)$$

$$(\forall t \in T)(\forall p \in P)[H(t, p) \rightarrow \rightarrow \text{PhSendTP}(\text{Map}(t), \text{Map}(p))], \quad (5)$$

где $\text{PhSendPT}: IPaddr \times IPaddr \rightarrow \{\text{true}, \text{false}\}$ – бинарный предикат, область истинности которого задает отношение, связывающее по передаче сообщений узлы-позиции и узлы-переходы в физической вычислительной сети; $\text{PhSendTP}: IPaddr \times IPaddr \rightarrow \{\text{true}, \text{false}\}$ – бинарный предикат, область истинности которого задает отношение, связывающее по передаче сообщений узлы-переходы и узлы-позиции в физической вычислительной сети; предикаты $F: P \times T \rightarrow \{\text{true}, \text{false}\}$, $H: T \times P \rightarrow \{\text{true}, \text{false}\}$ задают логические связи между переходами и позициями в исходной ординарной безопасной сети Петри.

Данные правила реализуются программно на уровне сетевых приложений. В реальном приложении компьютером провайдера облачного сервиса должна быть реализована специальная программа-парсер (программа-разборщик), которая выявляет в исходном запросе клиента

на заданную архитектуру связей в сети Петри логические имена и заменяет их на IP-адреса исполнительной вычислительной сети.

Запросом клиента может быть также сетевое приложение с логическими адресами узлов. В этом случае программа-парсер выявляет в исходном запросе, то есть в сетевом приложении клиента логические имена и заменяет их на IP-адреса исполнительной вычислительной сети.

Для явного описания операционной семантики формирования отношения PhSendPT введем следующее логико-алгебраическое операционное выражение (ЛАОВ):

$$[(\exists_{\text{Sel_all}} (p, t) \in P \times T) F(p, t)](\text{PhSendPT}(\text{Map}(p), \text{Map}(t)) \Leftarrow \text{true} \vee E), \quad (6)$$

где $\exists_{\text{Sel_all}}$ – оператор выбора всех кортежей вида $(p, t) \in P \times T$ из области истинности предиката $F(p, t)$; $\text{PhSendPT}(\text{Map}(p), \text{Map}(t)) \Leftarrow \text{true}$ – оператор формирования области истинности предиката PhSendPT для всех выбранных пар (p, t) ; E – пустой оператор.

Аналогично, для формирования отношения PhSendTP введем следующее ЛАОВ:

$$[(\exists_{\text{Sel_all}} (t, p) \in T \times P) H(t, p)](\text{PhSendTP}(\text{Map}(t), \text{Map}(p)) \Leftarrow \text{true} \vee E), \quad (7)$$

где оператор $\exists_{\text{Sel_all}}$ выбирает все кортежи вида $(t, p) \in T \times P$ из области истинности предиката $H(t, p)$; $\text{PhSendTP}(\text{Map}(t), \text{Map}(p)) \Leftarrow \text{true}$ – оператор формирования области истинности предиката PhSendTP для всех выбранных пар (t, p) .

Символом дизъюнкции “ \vee ” здесь обозначена операция альтернативного выбора одного из действий, в зависимости от истинности или ложности условия в квадратных скобках. В данных выраже-

ниях эти условия заключаются в завершении выбора всех пар $(p, t) \in P \times T$ и $(t, p) \in T \times P$ из областей истинности соответствующих бинарных предикатов $F(p, t)$ и $H(t, p)$.

Данную методику можно распространить также и на неординарные сети Петри – для этого при реализации приложения необходимо указать вес каждой дуги, что должно соответствовать ее кратности и числу передаваемых меток.

4. Концептуальные графы для распределенных логических сетей Петри

Следующие рисунки 4 и 5 иллюстрируют формирование концептуальных графов CG_3 и CG_4 для бинарных, или логических, сетей Петри. Буква В в именах позиций означает логическую интерпретацию их разметки. Функция разметки

МВ принимает значения в двухэлементном множестве $\{true, false\}$. Вложение логических сетей Петри в архитектуру распределенную вычислительной системы осуществляется аналогично предыдущему примеру при учете того факта, что в узле-позиции может размещаться не более одной метки.

Отношение SendB означает возможность передачи метки, несущей логическое значение. Как и в предыдущем случае (см. рис. 2), на одном и том же узле компьютерной сети могут разворачиваться как отдельные переходы и позиции, так и целые фрагменты распределенной сети Петри. Такие сети удобны для реализации систем логического управления на уровне промежуточного программного обеспечения *middleware* в облачных сервисах системах типа NCaaSOD.

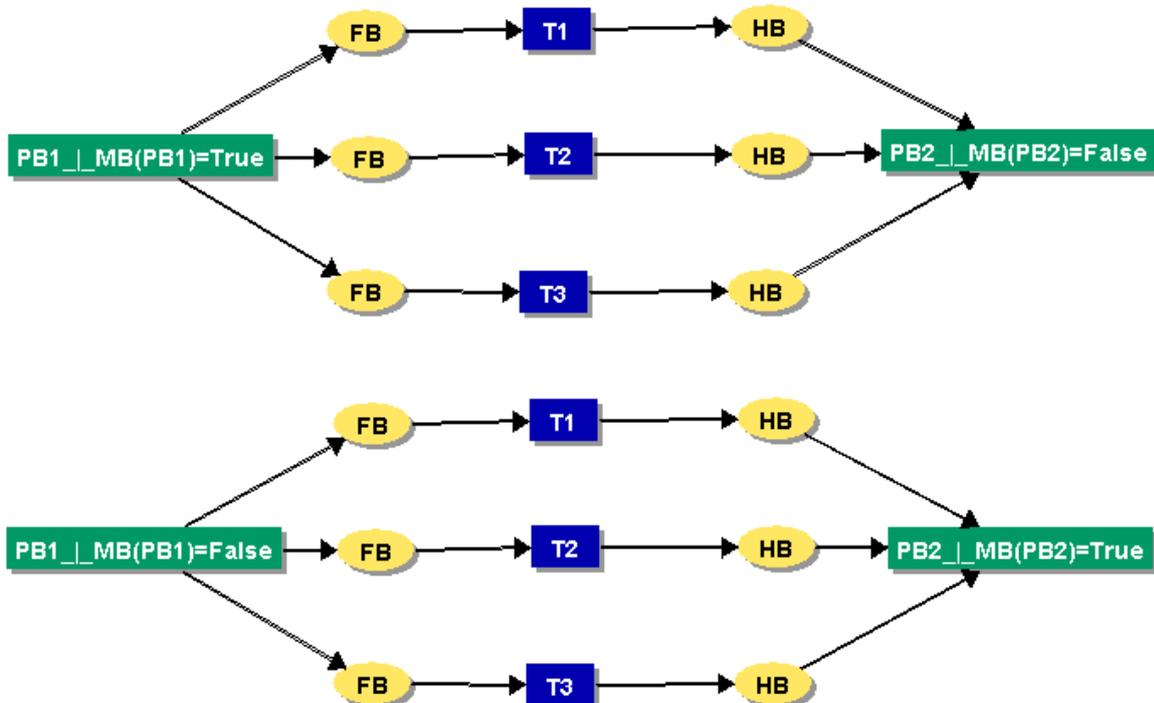


Рис. 4. Концептуальный граф CG_3 логической сети Петри до и после срабатывания одного из переходов

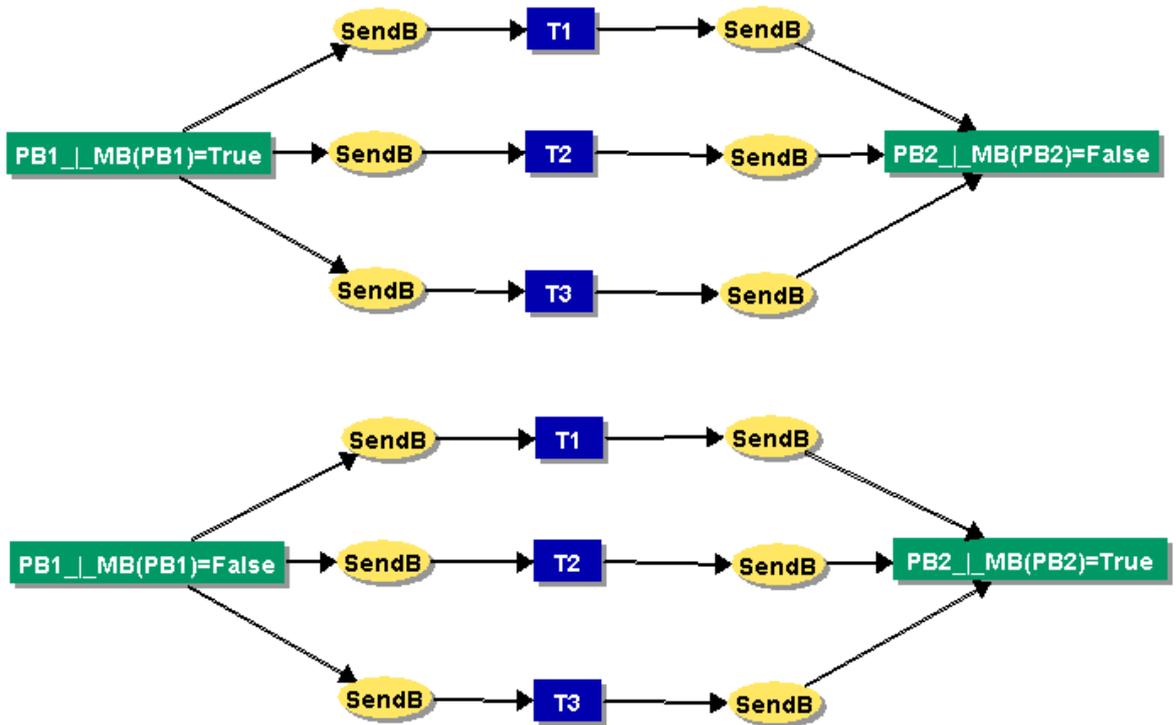


Рис. 5. Концептуальный граф CG_4 распределенной логической сети Петри до и после срабатывания одного из переходов

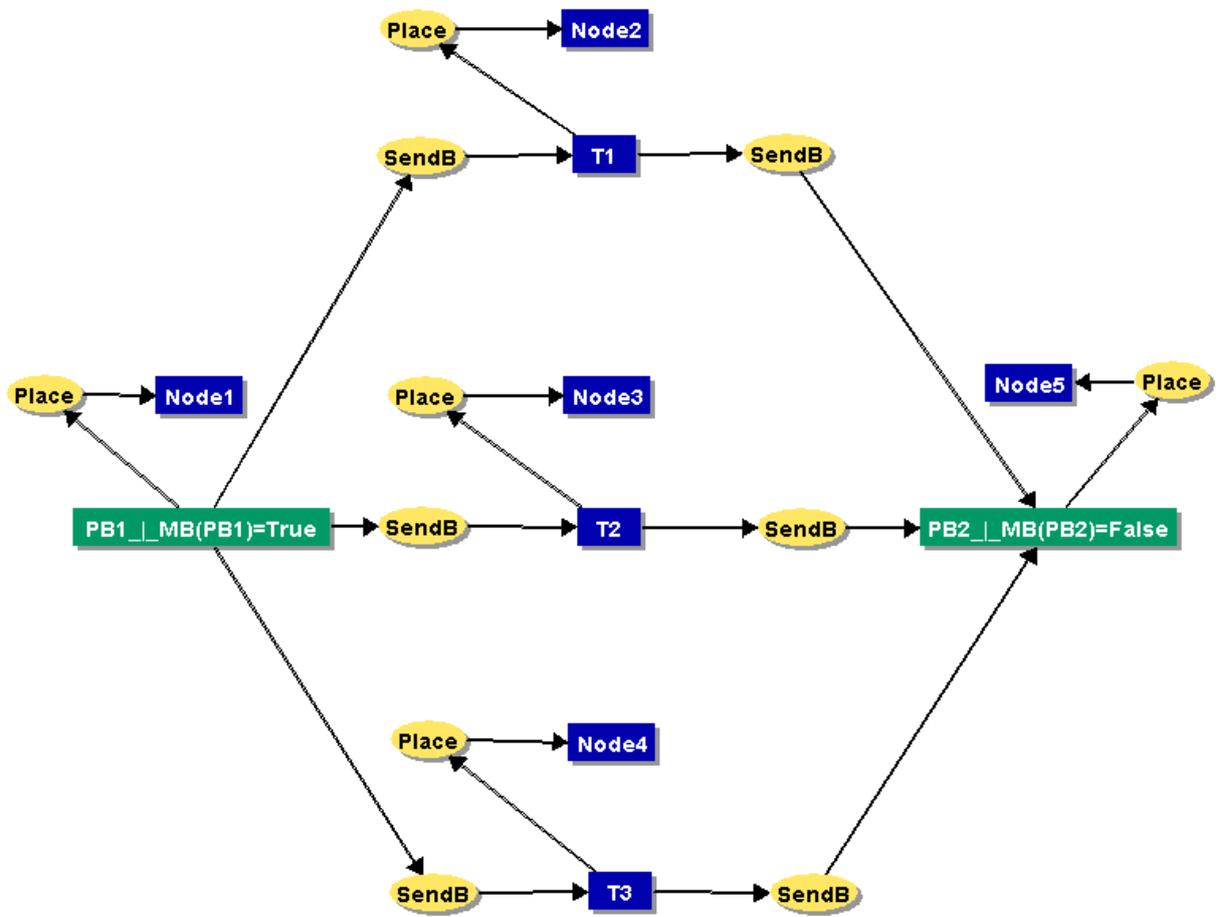


Рис. 6. Концептуальный граф CG_5 распределенной логической сети Петри

Как видно из предыдущих примеров, при вложении распределенных сетей Петри в логическую или физическую архитектуру распределенной вычислительной системы необходимо учитывать месторасположение объектов – переходов, позиций и меток (маркеров). Концептуальное представление сети Петри возможно дополнить новыми концептами – например, концептами, представляющими логические или физические узлы вычислительной системы. Полученный таким образом объект похож на семантическую сеть с событиями и его можно назвать “концептуальной распределенной сетью Петри” (КР СП). Пример такой сети с новыми дополнительными концептами Node1, Node2, ..., Node5 и новым отношением Place представлен на рисунке 6. Например, истинность высказывания Place(T1, Node2) означает, что переход T1 размещен на логическом узле Node2 при реализации сети Петри в вычислительной среде NCaaSOD. Здесь и далее в определениях, касающихся КР СП, будет использован прямой шрифт для лучшего соответствия графическим представлениям.

Развивая принципы построения КР СП, рассмотрим пример вложения сети Петри в вычислительную среду компьютерной сети с возможностью перераспределения позиций и переходов. Введем новое отношение NewPlace для перераспределения позиций и переходов на новые места и построим новый граф CG_6 на основе графа CG_5 (рис. 7). Особенностью графа CG_6 является то, что в нем представлено как старое отношение развертывания Place, так и новое NewPlace. Для перераспределения объектов необходимо разорвать старое отношение и конкретизировать новое, то есть необходимо ис-

пользовать процедуру перераспределения объектов.

Эта процедура сначала должна разорвать старые связи:

Place(PB1, Node1) ← false;
 Place(T1, Node2) ← false;
 Place(T2, Node3) ← false;
 Place(T3, Node4) ← false;
 Place(PB2, Node5) ← false,

а затем установить новые связи:

NewPlace(PB1, Node6) ← false;
 NewPlace(T1, Node7) ← false;
 NewPlace(T2, Node8) ← false;
 NewPlace(T3, Node9) ← false;
 NewPlace(PB2, Node10) ← false.

Указанные действия выполняются программой компьютера – провайдера облачных сервисов.

Следующий пример иллюстрирует не менее важное свойство концептуальных сетей Петри: свойство реконфигурируемости структуры самой сети. Реконфигурация в общем случае может затронуть отношения, связывающие позиции и переходы.

На рисунке 8 представлена концептуальная распределенная сеть Петри CG_7 с переменной структурой. При замене отношений FB1 и HB1 на отношения FB2 и HB2 сеть будет функционировать по-другому. Данным концептуальным графом описывается ситуация, когда необходимо перейти от барьерной синхронизации трех процессов к их независимому выполнению. Начальная структура сети задается конкретизацией отношения FB1:

FB1(PB1, Gather) ← true;
 FB1(PB2, Gather) ← true;
 FB1(PB3, Gather) ← true;
 HB1(Gather, PB4) ← true;
 HB1(Gather, PB5) ← true;
 HB1(Gather, PB6) ← true.

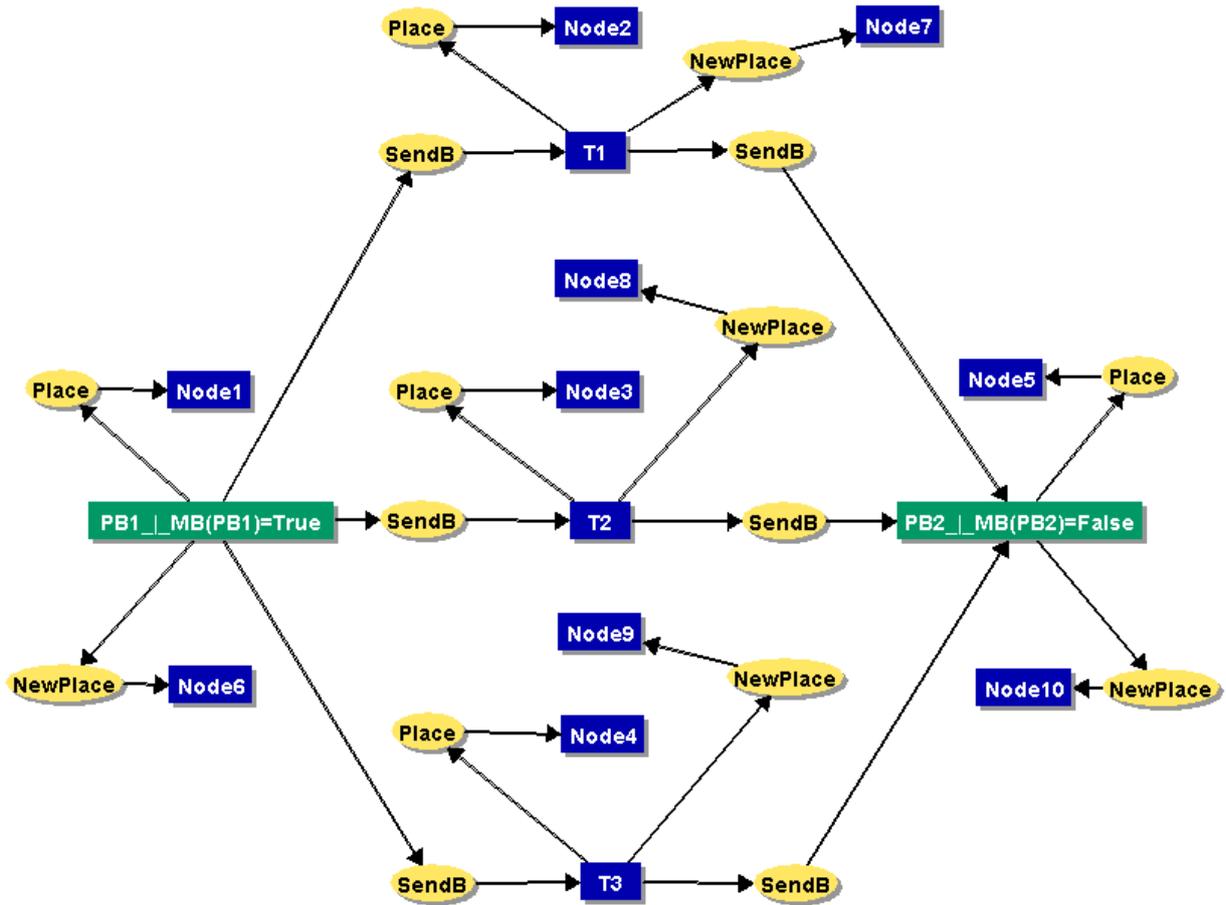


Рис. 7. Концептуальный граф CG_6 концептуальной распределенной сети Петри с возможным перерасположением позиций и переходов

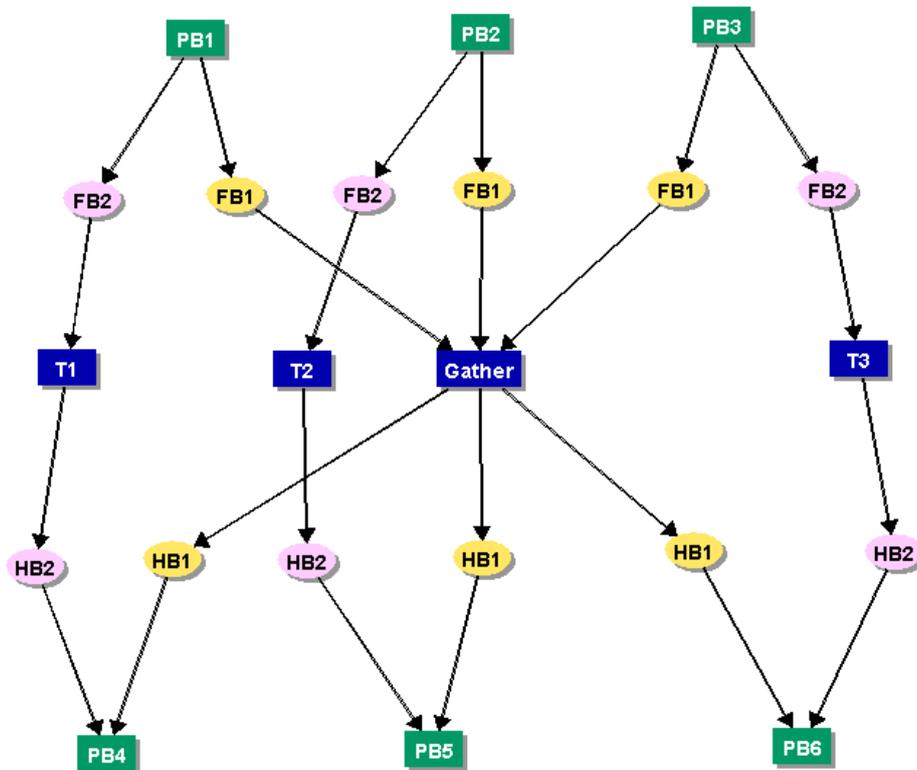


Рис. 8. Концептуальная распределенная сеть Петри CG_7 с переменной структурой

Для разрыва данных связей необходимо выполнить следующие действия:

FB1(PB1, Gather) \leftarrow false;
 FB1(PB2, Gather) \leftarrow false;
 FB1(PB3, Gather) \leftarrow false;
 HB1(Gather, PB4) \leftarrow false;
 HB1(Gather, PB5) \leftarrow false;
 HB1(Gather, PB6) \leftarrow false.

Для создания новых связей необходимо конкретизировать отношения FB2 и HB2:

FB2(PB1, T1) \leftarrow true;
 FB2(PB2, T2) \leftarrow true;
 FB2(PB3, T3) \leftarrow true;
 HB2(T1, PB4) \leftarrow true;
 HB2(T2, PB5) \leftarrow true;
 HB2(T3, PB6) \leftarrow true.

5. Концептуальные графы для распределенных сетей Петри смешанного типа

На рисунках 9 и 10 представлены примеры концептуальных графов CG_8 и CG_9 для распределенных сетей Петри (до и после срабатывания перехода T1), соответствующих определению (1) смешанной сети с простыми, ингибиторными и информационными дугами. Соответствующие связи представлены на рисунках 9 и 10. Сначала рассмотрим связи, представленные в графе CG_8 на рисунке 9. В общем случае при наличии связи типа FA_INH (ингибиторной) утверждается, что для срабатывания перехода необходимо, чтобы в его входной позиции не было ни одной метки. При наличии связи типа FA_INF (информационной) утверждает, что для срабатывания перехода необходимо, чтобы в его входной позиции находилось не менее одной метки. Связь типа FA была охарактеризована ранее. Указанные отношения связывают “арифметические” позиции смешанной сети Петри с переходом. Остальные от-

ношения FB_INH, FB_INF и FB в смешанной сети Петри связывают “логические” позиции с переходом. При наличии связи вида типа FB_INH (ингибиторной) утверждается, что для срабатывания перехода необходимо, чтобы в его входной позиции не находилась метка, то есть необходимо выполнение условия $MB(PB4)=false$. При наличии связи вида типа FB_INF (информационной) утверждается, что для срабатывания перехода необходимо, чтобы в его входной позиции находилась метка, то есть необходимо выполнение условия $MB(PB5)=true$; кроме того, эта метка должна остаться в указанной позиции и после срабатывания перехода. Связь типа FB была охарактеризована ранее.

Концептуальный граф CG_9 распределенной сети Петри смешанного типа, представленный на рисунке 10, в отличие от концептуального графа CG_8 на рисунке 9, содержит отношения SendA_INH, SendA_INF, SendA, SendB_INH, SendB_INF, SendB, задаваемые путем передачи соответствующих сообщений от позиций к переходу. Эти сообщения “информируют” переход о выполнении условий, необходимых для его срабатывания. В остальном сеть CG_9 функционирует аналогично предыдущей сети CG_8 .

Рисунок 11 иллюстрирует работу распределенной концептуальной сети Петри CG_{10} (построенной на основе сетей CG_8 и CG_9) после ее вложения в логическую архитектуру компьютерной сети. При реализации распределенного приложения необходимо учесть, что в ряде случаев необходимая информация должна запрашиваться. Этот факт не был отражен в примерах графов CG_8 и CG_9 . Поэтому в концептуальный граф CG_{10} введены новые отношения Q1 и Q2.

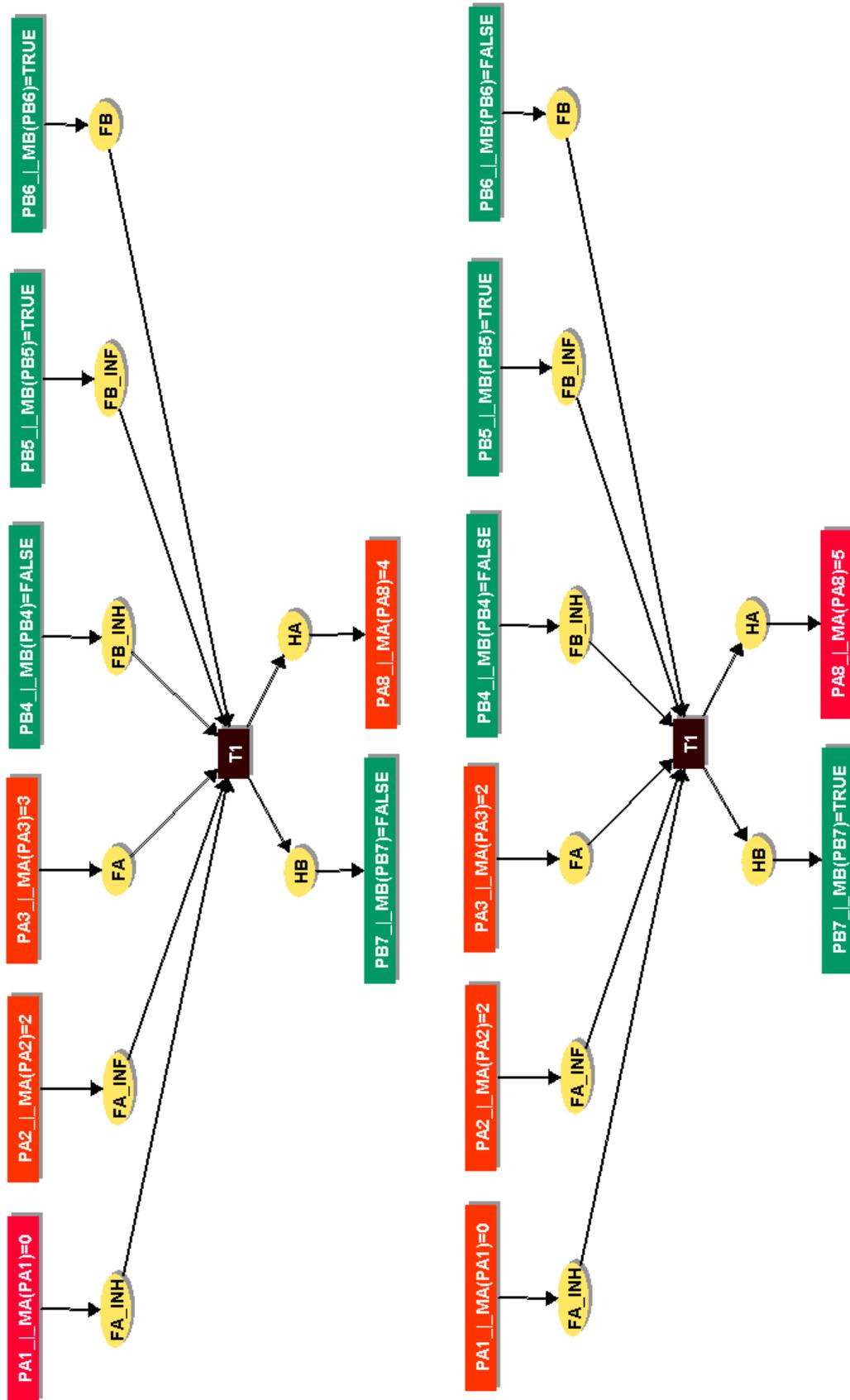


Рис. 9. Концептуальная смешанная сеть Петри CG_8

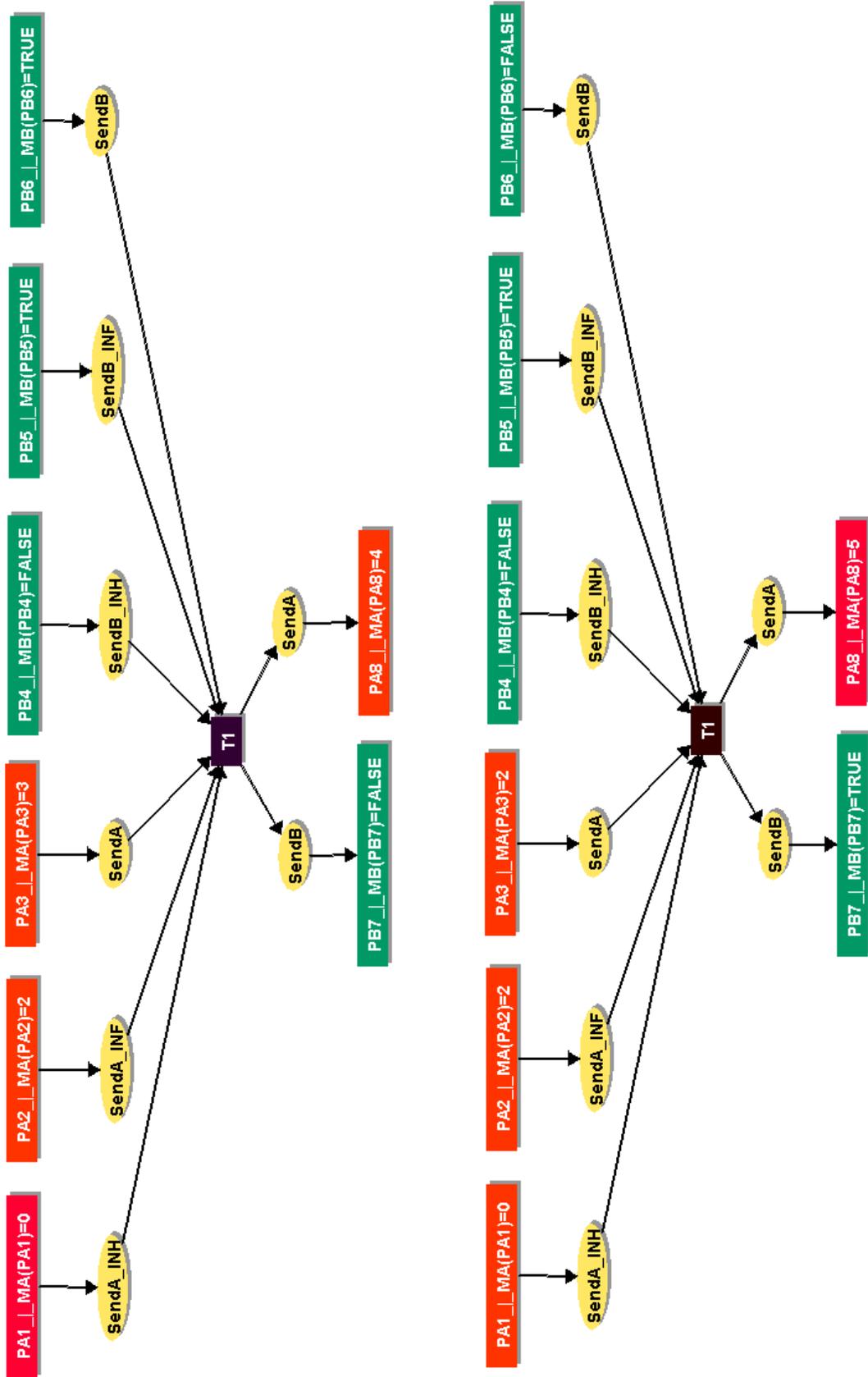


Рис. 10. Концептуальная распределенная смешанная сеть Петри SG_9

Кроме того, введено отношение Place и концепты-узлы, задающее размещение позиций и переходов в логической структуре компьютерной сети. До срабатывания перехода T1 реализующая его процедура запрашивает информацию о состоянии входных позиций. На абстрактном уровне этот факт сопровождается конкретизацией отношения Q1.

Получение ответов переходом T1 на абстрактном уровне сопровождается конкретизацией отношений SendA_INH, SendA_INF, SendA, SendB_INH, SendB_INF, SendB, а на физическом уровне – путем передачи сообщений от размещенных в компьютерной сети позиций к переходу. Конкретизация дополнительного отношения Q2 соответствует запросам процедуры перехода T1 о состоянии выходных позиций P_{В7} и P_{А8} (“способны” ли они принять метки). Таким образом, концептуальный граф смешанной распределенной сети Петри представляет собой семантическую сеть с процедурами, способную описать работу распределенного сетевого приложения на прикладном и транспортном уровнях, а также на уровне промежуточного программного обеспечения *middleware*.

Заключение

1. Предложены новые концептуально-поведенческие модели на основе концептуальных графов распределенных сетей Петри (КГ РСП) для определения системной и функциональной архитектур распределенных вычислительных систем с переменной структурой, предоставляемой пользователю по его требованию в качестве гибридного облачно-сетевого сервиса; данные модели отличаются возможностью оперативной реконфигурации и непосредственной исполнимостью.

2. На основе известных определений бинарных (логических) и обычных ординарных сетей Петри дано определение и концептуально-поведенческое представление сетей Петри смешанного типа, позволяющие синтезировать сети Петри, вложимые в архитектуру компьютерных сетей.

3. Предложена методика вложения концептуальных сетей Петри в архитектуру облачно-сетевых компьютерных сетей типа NCaaSOD – сетевые (облачные) вычисления как сервис, организуемый по запросу пользователя. Предложены правила получения отношений связности между позициями и переходами сети Петри, размещаемыми на узлах физической компьютерной сети.

Список литературы

1. Petri C.A. Kommunikation mit Automaten. English Translation, 1966: Communication with Automata, Technical Report RADG-TR-65-377, Rome Air Dev. Center, New York.
2. Wang Jiacun. Petri Nets for Dynamic Event-Driven System Modeling / Published in Handbook of Dynamic System Modeling. Department of Software Engineering. Monmouth University, West Long Branch, NJ 07764, 2007. P. 1-17.
3. Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE. April 1989. Vol. 77, No. 4. P. 541-580.
4. Peterson J. L. Petri Net Theory and the Modeling of Systems. N.J.: Prentice-Hall, 1981. 288 p.
5. Дубинин В.Н., Зинкин С.А. Сетевые модели распределенных систем обработки, хранения и передачи данных. Пенза: Изд-во «Приволжский Дом Знаний», 2013. 452 с.
6. Welcome to the Petri Nets World. [Web resource]. URL: <http://www.informatik.uni->

- hamburg.de/TGI/PetriNets/index.php. Free admission. Access date: 10.11.2018.
7. Applications and Theory of Petri Nets and Concurrency // Proc. 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 2014 / Ed. by Ciardo G., Kindler E. Lecture Notes in Computer Science. Vol. 8489. Berlin.: Springer. 2014. 395 p.
 8. Application and Theory of Petri Nets and Concurrency: Proc./36th International Conference, PETRI NETS 2015, Brussels, Belgium, June 2015 / Ed. by Devillers R., Valmari A. Lecture Notes in Computer Science. Vol. 9115. Berlin.: Springer, 2015. 351 p.
 9. Application and Theory of Petri Nets and Concurrency // Proc. 37th International Conference, PETRI NETS 2016, Torun, Poland, June 2016 / Ed. by Kordon F., Moldt D. Lecture Notes in Computer Science. – Vol. 9698. Berlin.: Springer, 2016. 345 p.
 10. Application and Theory of Petri Nets and Concurrency // Proc. 38th International Conference, PETRI NETS 2017, Zaragoza, Spain, June 2017 / Ed. by Van Der Aalst W., Best E. Lecture Notes in Computer Science. Vol. 10258. Berlin.: Springer, 2017. 351 p.
 11. Application and Theory of Petri Nets and Concurrency // Proc. 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, June 2018 / Ed. by Khomenko V., Roux O. H. Lecture Notes in Computer Science. Vol. 10877. Berlin.: Springer, 2018. 427 p.
 12. Котов В.Е. Сети Петри. М.: Наука. Главная редакция физико-математической литературы, 1984. 160 с.
 13. Peterson J. L. Petri Nets // ACM Computing Surveys. 1977. 9 (3). P. 223–252.
 14. Методы параллельного микропрограммирования / под ред. О. Л. Бандман. Новосибирск: Наука, 1981. 182 с.
 15. Юдицкий С. А., Магергут В. З. Логическое управление дискретными процессами. Модели, анализ и синтез. М.: Машиностроение, 1987. 178 с.
 16. W. M. P. van der Aalst. A class of Petri nets for modeling and analyzing business processes // Computing Science Reports 95/26, Eindhoven University of Technology, Eindhoven, 1995. P. 1-25.
 17. Hruz B., Zhou M. C. Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tools. Springer-Verlag, London, 2007. 351 p.
 18. Iordache M. V., Antsaklis P. J. Supervisory Control of Concurrent Systems. A Petri Net Structural Approach. Boston: Birkhauser, 2006. 281 p.
 19. On Modification in Petri Nets / R. Jahns, M. Becker, T. Bessey, H. Szczerbicka // Journal Proc. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2002. 5 p.
 20. Valk R. Self-modifying Nets // Report Bericht Nr. 34, IFI-HH-B-34/77, from the Institute of Informatics, University of Hamburg, Schluterstrasse 70, Hamburg 13, D-2000, West Germany, July 1977. 36 p.
 21. Temporized Colored Petri Nets with Changeable Structure (TCPN-CS) for the Performance Modeling of Dynamic Production Systems / Jiang Z., Zuo M. J., Fung R. Y. K., Tu P. Y. // International Journal of Production Research 2000. 38, No. 8. P. 1917-1945.
 22. Holloway L. E., Krogh B. H., Giua A. A Survey of Petri Net Methods for Controlled Discrete Event Systems // Discrete Event Dynamic Systems: Theory and Applications, 7, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands, 1997. P. 151-190.
 23. Krogh B. H. Controlled Petri nets and maximally permissive feedback logic // Proc. 25th Annual Allerton Conference, University of Illinois, Urbana, 1987. P. 317-326.
 24. Ichikawa A., Hiraishi K. Analysis and control of discrete event systems represented by

Petri nets // Discrete Event Systems: Models and Applications. Varaiya P., and Kurzhanski A. B. eds. Lecture Notes in Control and Information Sciences. Vol. 103, New York: Springer Verlag, 1988. P. 115-134.

25. Omar Yaqub, Lingxi Li. Modeling and Analysis of Connected Traffic Intersections Based on Modified Binary Petri Nets // Hindawi Publishing Corporation, International Journal of Vehicular Technology. Vol. 2013, Article ID 192516, 2013. 10 p.

26. Jianxin Liao, Yuting Zhang, Xiaomin Zhu. An Inter-Working Petri Net Model between SIMPLE and IMPS for XDM Service // Chapter from the book "Petri Net, Theory and Applications", pp. 73-90. Downloaded from URL: http://www.intechopen.com/books/petri_net_theory_and_applications. [Web resource]. Source: Petri Net, Theory and Applications. Book edited by: Vedran Kordic, ISBN 978-3-902613-12-7, 534 p., February 2008, I-Tech Education and Publishing, Vienna, Austria.

27. Petri Net Transformations / H. Ehrig, K. Hoffmann, J. Padberg, C. Ermel, U. Prange, E. Biermann, T. Modica // Chapter from the book "Petri Net, Theory and Applications", pp. 1 – 16. Downloaded from URL: http://www.intechopen.com/books/petri_net_theory_and_applications. [Web resource]. Free admission. Access date: 10.11.2018. Source: Petri Net, Theory and Applications, Book edited by: Vedran Kordic, ISBN 978-3-902613-12-7, 534 p., February 2008, I-Tech Education and Publishing, Vienna, Austria.

28. Petri Nets Fundamental Models, Verification and Applications / Edited by Michel Diaz. John Wiley & Sons, 2009. 585 p.

29. Svadova M., Hanzalek Z. Matlab Toolbox for Petri Nets. URL1: <https://simonline.akademitelkom.ac.id/dosen/riset/smartgrid/Bukan/Hanzalek01a.pdf>. URL2: <http://www.academia.edu/7645464/>

MATLAB_TOOLBOX_FOR_PETRI_NET S. [Web resource]. Free admission. (Access date: 10.11.2018).

30. Petri Nets Tools Database Quick Overview. URL: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick>. [Web resource]. Free admission. (Access date: 10.11.2018).

31. Component-based synthesis for complex APIs. / Yu Feng, Ruben Martins, Yuepeng Wang, Isil Dillig, Thomas W. Reps // Newsletter, ACM SIGPLAN Notices - POPL '17. Vol. 52. Issue 1, January 2017. P. 599-612.

32. Samokhvalov D.I., Dworzanski L.W. Automatic Code Generation from Nested Petri nets to Event-based Systems on the Telegram Platform // Trudy ISP RAN / Proc. RAS. 2016. Vol. 28. Is. 3. P. 65-84.

33. Petri Nets and Ladder Logic for Fully-Automating and Programmable Logic Control of Semi-Automatic Machines and Systems / Mahmoud A. Barghash, Osama M. Abuzeid, Anas N. Al-Rabadi, Ahmad M. Jaradat // American J. of Engineering and Applied Sciences 2011. №4 (2). P. 252-264.

34. Conway C., Cheng-Hong Li, Megan Pengelly M. Pencil: A Petri Net Specification Language for Java. 3 December 2002 [Web resource]. URL: <http://www.cs.columbia.edu/~sedwards/classes/2003/w4115/conway-report.pdf>. Free admission. (Access date: 10.11.2018).

35. PetriNetExec: Java Embeddable Petri Nets: PetriNetExec a library for embedding Petri Nets into Java applications. Last Update: 2013-05-30. [Web resource]. URL: <https://sourceforge.net/projects/petrinetexec/> Free admission. (Access date: 10.11.2018).

36. Jai Manral. Automated Test Case Generation using Petri Nets. [Web resource]. URL: <https://arxiv.org/ftp/arxiv/papers/1509/1509.08401.pdf>. Free admission. (Access date: 10.11.2018).

37. Wang L., Jiesong Y., Xiaofeng Y., Jun H., Xuandong L., Guoliang Z. Generating test cases from UML activity diagram based on Gray-box method // Software Engineering Conference. Nov. 2004. P. 284-291.

38. Masahiro Osogami, Teruya Yamanishi, Katsuji Uosaki. Input-Output Conditions for Automatic Program Generation Using Petri Nets // International Conference on Knowledge-Based and Intelligent Information and Engineering Systems / Lecture Notes in Computer Science. Vol. 6881. Berlin.: Springer, 2011. P. 296-305.

39. A Guide to Modelling and Control with Modules of Signal Nets. Jörg Desel, Hans-Michael Hanisch, Gabriel Juhás, Robert Lorenz, Christian Neumair // Lecture Notes in

Computer Science. Vol. 3147. Berlin, Heidelberg: Springer, 2004. P. 270-300.

40. Arlow J., Neustadt I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. 2nd Edition. New Jersey: Pearson Education Inc., Addison-Wesley Professional. 2005. 624 p.

41. CharGer Manual v3.5b1 2005-11-30. P. 1–58. URL: [http:// charger.sourceforge.net/](http://charger.sourceforge.net/). [Web resource]. Free admission. (Access date: 10.11.2018).

42. Delugah H. CharGer – A Conceptual Graph Editor written by Harry Delugah. URL: [http://www.cs.uah.edu/~delugach/ CharGer/](http://www.cs.uah.edu/~delugach/CharGer/). [Web resource]. Free admission. (Access date: 10.11.2018).

Поступила в редакцию 26.11.18

UDC 004.832.32

S.A. Zinkin, Doctor of Engineering Science, Professor, Penza State University (Russia, 440026, Penza, Krasnaya str., 40) (e-mail: zsa49@yandex.ru)

Mustafa Sadeq Jaafar, Post-Graduate Student, Penza State University (Russia, 440026, Penza, Krasnaya str., 40) (e-mail: mustafajaafar221087@gmail.com)

N.S. Karamysheva, Candidate of Engineering Sciences, Associated Professor, Penza State University (Russia, 440026, Penza, Krasnaya str., 40) (e-mail: karamyshevans@yandex.ru)

CONCEPTUAL REPRESENTATIONS AND MODIFICATIONS OF PETRI NETS FOR APPLICATIONS IN THE AREA OF SYNTHESIS OF A FUNCTIONAL ARCHITECTURE OF DISTRIBUTED COMPUTATIONAL SYSTEMS WITH VARIABLE STRUCTURE

Background. *The object of the research is the functional architecture of distributed computing systems with a variable (reconfigurable) structure characteristic of hybrid systems of cloud-network (grid) type. Despite the fact that Petri nets have long been studied both theoretically and practically, the methods of their interpretation continue to develop intensively. At present, the problem of embedding Petri nets in the architecture of distributed network applications used to implement global computing in modern mixed cloud, grid and cluster systems has not been sufficiently studied. It is shown that in modern studies, Petri nets are used mainly in the simulation of discrete systems and processes, and not as the basis for formalized specifications in the development of distributed applications. In this regard, the interpretation of Petri nets in applications to the functional architecture of distributed computing systems with a variable structure based on the network software of the intermediate class (middleware class) is relevant. The aim of the work was the integration of graphical representations of conceptual graphs, semantic networks, scenarios and Petri nets, which made it possible to create effective tools with graphical support for designing a functional architecture of distributed computing systems with a variable structure and, in particular, a cloudy architecture of the NCaaS type - Network Computing as a Service on Demand (network computing as a cloud service at the request of the user).*

Materials and methods. *The conceptual models of distributed processes that are a graphical interpretation of the first-order predicate calculus are used. Conceptual graphs for distributed Petri nets of mixed type have been proposed, which allow describing computation processes in global computational networks with a view to their subsequent implementation.*

Results. Based on the integration of graphical representations of conceptual graphs, semantic networks, scenarios and Petri nets, conceptual representations of distributed reconfigurable Petri nets are proposed, allowing them to be directly integrated into the architecture of the computer network.

Results. New conceptual-behavioral models based on conceptual graphs of distributed Petri nets have been proposed to determine the system and functional architectures of distributed computing systems with a variable structure provided to the user as a hybrid cloud-based network service; these models are distinguished by the possibility of operational reconfiguration and immediate execution.

Conclusion. A method was proposed and formalized for embedding conceptual Petri nets into the architecture of cloud-networked computer systems such as NCaaS — network (cloud) computing as a service organized at the user's request. The rules for obtaining relations of connectivity between the positions and transitions of the Petri net, placed on the nodes of the physical computer network, are proposed.

Key words: distributed network computing; functional architecture; conceptual Petri nets of mixed type; reconfiguration of network models; embedding Petri nets into the computer network architecture.

DOI: 10.21869/2223-1560-2018-22-6-143-167

For citation: Zinkin S.A., Mustafa Sadeq Jaafar, Karamysheva N.S. Conceptual Representations and Modifications of Petri Nets for Applications in the Area of Synthesis of a Functional Architecture of Distributed Computational Systems with Variable Structure. Proceedings of the Southwest State University, 2018, vol. 22, no. 6(81), pp. 143-167 (in Russ.).

Reference

1. Petri C.A. Kommunikation mit Automaten. English Translation, 1966: Communication with Automata, Technical Report RADC-TR-65-377, Rome Air Dev. Center, New York.
2. Wang Jiacun. Petri Nets for Dynamic Event-Driven System Modeling. Published in Handbook of Dynamic System Modeling. Department of Software Engineering. Monmouth University, West Long Branch, NJ 07764, 2007, pp. 1-17.
3. Murata T. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*. April 1989, vol. 77, no. 4, pp. 541-580.
4. Peterson J. L. Petri Net Theory and the Modeling of Systems. N.J.: Prentice-Hall, 1981, 288 p.
5. Dubinin V.N., Zinkin S.A. Setevye modeli raspredelennyh sistem obrabotki, hranenija i peredachi dannyh. Privolzhskij Dom Znaniy Publ., 2013, 452 p.
6. Welcome to the Petri Nets World. [Web resource]. URL: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/index.php>. Free admission. (Access date: 10.11.2018).
7. Applications and Theory of Petri Nets and Concurrency. Proc. 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 2014; ed. by Ciardo G., Kindler E. Lecture Notes in Computer Science. Vol. 8489. Berlin, Springer Publ., 2014, 395 p.
8. Application and Theory of Petri Nets and Concurrency: Proc./36th International Conference, PETRI NETS 2015, Brussels, Belgium, June 2015; ed. by Devillers R., Valmari A. Lecture Notes in Computer Science. Vol. 9115, Berlin, Springer Publ., 2015. 351 p.
9. Application and Theory of Petri Nets and Concurrency. Proc. 37th International Conference, PETRI NETS 2016, Torun, Poland, June 2016; ed. by Kordon F., Moldt D. Lecture Notes in Computer Science. Vol. 9698, Berlin, Springer Publ., 2016, 345 p.
10. Application and Theory of Petri Nets and Concurrency // Proc. 38th International Conference, PETRI NETS 2017, Zaragoza, Spain, June 2017; ed. by Van Der Aalst W., Best E. Lecture Notes in Computer Science, Vol. 10258, Berlin, Springer Publ., 2017, 351 p.
11. Application and Theory of Petri Nets and Concurrency. Proc. 39th Interna-

tional Conference, PETRI NETS 2018, Bratislava, Slovakia, June 2018. Ed. by Khomenko V., Roux O. H. Lecture Notes in Computer Science, vol. 10877, Berlin: Springer., 2018, 427 p.

12. Kotov V.E. Seti Petri. Moscow, Nauka Publ.. Glavnaja redakcija fiziko-matematicheskoy literatury, 1984, 160 p.

13. Peterson J. L. Petri Nets. ACM Computing Surveys, 9 (3), 1977, pp. 223–252.

14. Metody parallel'nogo mikro-programmirovaniya, ed. by Bandman O. L. Novosibirsk, Nauka Publ., 1981, 182 p.

15. Judickij S. A., Magergut V. Z. Logicheskoe upravlenie diskretnymi procesami. Modeli, analiz i sintez. Moscow, Mashinostroenie Publ., 1987, 178 p.

16. W. M. P. van der Aalst. A class of Petri nets for modeling and analyzing business processes. Computing Science Reports 95/26, Eindhoven University of Technology, Eindhoven, 1995, pp. 1-25.

17. Hruz B., Zhou M. C. Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tools. Springer-Verlag. London, 2007, 351 p.

18. Iordache M. V., Antsaklis P. J. Supervisory Control of Concurrent Systems. A Petri Net Structural Approach. Boston: Birkhauser Publ., 2006, 281 p.

19. Jahns R., Becker M., Bessey T., Szczerbicka H. On Modification in Petri Nets. *Journal Proc. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 2002, 5 p.

20. Valk R. Self-modifying Nets. Report Bericht Nr. 34, IFI-HH-B-34/77, from the Institute of Informatics, University of Hamburg, Schluterstrasse 70, Hamburg 13, D-2000, West Germany, July 1977, 36 p.

21. Jiang Z., Zuo M. J., Fung R. Y. K., Tu P. Y. Temporized Colored Petri Nets with Changeable Structure (TCPN-CS) for the Performance Modeling of Dynamic Production Systems. *International Journal of Production Research*, 2000, 38, no. 8, pp. 1917-1945.

22. Holloway L. E., Krogh B. H., Giua A. A Survey of Petri Net Methods for Controlled Discrete Event Systems. *Discrete Event Dynamic Systems: Theory and Applications*, 7, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands, 1997. pp. 151-190.

23. Krogh B. H. Controlled Petri nets and maximally permissive feedback logic. Proc. 25th Annual Allerton Conference, University of Illinois, Urbana, 1987. pp. 317-326.

24. Ichikawa A., Hiraishi K. Analysis and control of discrete event systems represented by Petri nets. *Discrete Event Systems: Models and Applications*. Varaiya P., and Kurzhanski A. B. eds. Lecture Notes in Control and Information Sciences, vol. 103, New York, Springer Verlag, 1988, pp. 115-134.

25. Omar Yaqub, Lingxi Li. Modeling and Analysis of Connected Traffic Intersections Based on Modified Binary Petri Nets. Hindawi Publishing Corporation, International Journal of Vehicular Technology. Vol. 2013, Article ID 192516, 2013, 10 p.

26. Jianxin Liao, Yuting Zhang, Xiaomin Zhu. An Inter-Working Petri Net Model between SIMPLE and IMPS for XDM Service. Chapter from the book “Petri Net, Theory and Applications”, pp. 73-90. Downloaded from URL: http://www.intechopen.com/books/petri_net_theory_and_applications. [Web resource]. Source: Petri Net, Theory and Applications. Book edited by: Vedran Kordic, ISBN 978-3-902613-12-7, 534 p., February 2008, I-Tech Education and Publishing, Vienna, Austria.

27. Ehrig H., Hoffmann K., Padberg J., Ermel C., Prange U., Biermann E., Modica T. Petri Net Transformations. Chapter from the book “Petri Net, Theory and Applications”, pp. 1 – 16. Downloaded from URL: http://www.intechopen.com/books/petri_net_theory_and_applications. [Web resource]. Free admission. Access date: 10.11.2018. Source: Petri Net, Theory and Applications, Book edited by: Vedran Kordic, ISBN 978-3-902613-12-7, 534 p., Feb-

ruary 2008, I-Tech Education and Publishing, Vienna, Austria.

28. Petri Nets Fundamental Models, Verification and Applications. ed. by Michel Diaz. John Wiley & Sons, 2009. 585 p.

29. Svadova M., Hanzalek Z. Matlab Toolbox for Petri Nets. URL1: <https://simononline.akademitelkom.ac.id/dosen/riset/smartgrid/Bukan/Hanzalek01a.pdf>. URL2: http://www.academia.edu/7645464/MATLAB_TOOLBOX_FOR_PETRI_NETWORKS. [Web resource]. Free admission. (Access date: 10.11.2018).

30. Petri Nets Tools Database Quick Overview. URL: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick>. [Web resource]. Free admission. (Access date: 10.11.2018).

31. Component-based synthesis for complex APIs. / Yu Feng, Ruben Martins, Yuepeng Wang, Isil Dillig, Thomas W. Reps. Newsletter, ACM SIGPLAN Notices - POPL '17, Vol. 52. Issue 1, January 2017, pp. 599-612.

32. Samokhvalov D.I., Dworzanski L.W. Automatic Code Generation from Nested Petri nets to Event-based Systems on the Telegram Platform. Trudy ISP RAN. Proc. RAS, 2016, vol. 28, is. 3, pp. 65-84.

33. Mahmoud A. Barghash, Osama M. Abuzeid, Anas N. Al-Rabadi, Ahmad M. Jaradat Petri Nets and Ladder Logic for Fully-Automating and Programmable Logic Control of Semi-Automatic Machines and Systems. *American J. of Engineering and Applied Sciences*, 2011, no. 4 (2), pp. 252-264.

34. Conway C., Cheng-Hong Li, Megan Pengelly M. Pencil: A Petri Net Specification Language for Java. 3 December 2002 [Web resource]. URL: <http://www.cs.columbia.edu/~sedwards/classes/2003/w4115/conway-report.pdf/>. Free admission. (Access date: 10.11.2018).

35. PetriNetExec: Java Embeddable Petri Nets: PetriNetExec a library for embed-

ding Petri Nets into Java applications. Last Update: 2013-05-30. [Web resource]. URL: <https://sourceforge.net/projects/petrinetexec/>. Free admission. (Access date: 10.11.2018).

36. Jai Manral. Automated Test Case Generation using Petri Nets. [Web resource]. URL: <https://arxiv.org/ftp/arxiv/papers/1509/1509.08401.pdf>. Free admission. (Access date: 10.11.2018).

37. Wang L., Jiesong Y., Xiaofeng Y., Jun H., Xuandong L., Guoliang Z. Generating test cases from UML activity diagram based on Gray-box method. Software Engineering Conference, Nov. 2004, pp. 284-291.

38. Masahiro Osogami, Teruya Yamashita, Katsuji Uosaki. Input-Output Conditions for Automatic Program Generation Using Petri Nets. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Lecture Notes in Computer Science, vol. 6881, Berlin, Springer Publ., 2011, pp. 296-305.

39. A Guide to Modelling and Control with Modules of Signal Nets. Jörg Desel, Hans-Michael Hanisch, Gabriel Juhás, Robert Lorenz, Christian Neumair. Lecture Notes in Computer Science, vol. 3147. Berlin, Heidelberg, Springer Publ., 2004, pp. 270-300.

40. Arlow J., Neustadt I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. 2nd Edition. New Jersey: Pearson Education Inc., Addison-Wesley Professional, 2005, 624 p.

41. CharGer Manual v3.5b1 2005-11-30. P. 1-58. URL: <http://charger.sourceforge.net/>. [Web resource]. Free admission. (Access date: 10.11.2018).

42. Delugach H. CharGer – A Conceptual Graph Editor written by Harry Delugach. URL: <http://www.cs.uah.edu/~delugach/CharGer/>. [Web resource]. Free admission. (Access date: 10.11.2018).