

УДК 004.9

doi: 10.21685/2072-3059-2023-2-3

**Представление и структурирование знаний
в семантико-ориентированной вычислительной среде.
Часть I. Интеграция концептуальных графов и логических сетей
на основе формализации структурированных ситуаций**

В. И. Волчихин¹, Н. С. Карамышева², М. А. Митрохин³, С. А. Зинкин⁴

^{1,2,3,4}Пензенский государственный университет, Пенза, Россия

¹cnit@pnzgu.ru, ²karamyshevans@yandex.ru, ³vt@pnzgu.ru, ⁴zsa49@yandex.ru

Аннотация. *Актуальность и цели.* Предложена методика глубокого структурирования знаний в семантико-ориентированной интеллектуальной вычислительной среде на основе расширения описательных возможностей сетей Петри путем их интеграции с концептуальными графами. Одной из основ для интеграции данных формализмов для описания ситуационных моделей систем являются не только удобные графические представления концептуальных графов и сетей Петри, но и их общая семантика, основанная на использовании свойств предметной области. Полученные в результате интеграции приложения могут использоваться далее в качестве прототипов при создании экспертных или справочных систем, а также экспертных систем имитационного моделирования для различных предметных областей. Актуальность исследований концептуальных графов и сетей Петри связана с широкой областью их применимости при моделировании дискретно-событийных систем. В концептуальных графах могут быть определены концепты-события и причинно-следственные связи между ними. Для участников событий могут быть определены исполняемые ими роли. Сети Петри фактически представляют собой модели, похожие на семантические сети и концептуальные графы, в которых определены отношения между объектами-позициями и объектами-переходами. Основой интеграции являются статическая и динамическая составляющие сети Петри с одной стороны, а с другой стороны – декларативность концептуальных графов. Целью работы является расширение области деятельности, основанной на интерпретации сетей Петри и концептуальных графов и их интеграции за счет выявления общей семантики, связанной не только семантикой сетей Петри и концептуальных графов как таковых, но и с семантикой и понятиями исследуемой предметной области. *Материалы и методы.* Интеграция и интерпретация концептуальных графов и логических, или бинарных, сетей Петри основана на использовании логики предикатов первого порядка, теории сетей Петри, исследовании применимости концептуальных моделей в различных предметных областях. *Результаты.* Предложена методика синтеза концептуальных логических сетей Петри на основе выявления общей семантики концептуальных графов и сетей Петри, в результате чего могут быть построены модели, обладающие декларативными, императивными и динамическими свойствами.

Ключевые слова: отношения, предикаты, концепты, продукционные правила, концептуальные графы, логические сети Петри, интеграция, интерпретация, структуризация и представление знаний, декларативность и императивность моделей, концептуальные сети Петри, динамика моделей

Для цитирования: Волчихин В. И., Карамышева Н. С., Митрохин М. А., Зинкин С. А. Представление и структурирование знаний в семантико-ориентированной вычисли-

тельной среде. Часть I. Интеграция концептуальных графов и логических сетей на основе формализации структурированных ситуаций // Известия высших учебных заведений. Поволжский регион. Технические науки. 2023. № 2. С. 24–51. doi: 10.21685/2072-3059-2023-2-3

**Representation and structuring of knowledge
in the semantic oriented computing environment.
Part 1. Integration conceptual graphs and logical networks
based on formalization of structured situations**

V.I. Volchikhin¹, N.S. Karamysheva², M.A. Mitrokhin³, S.A. Zinkin⁴

^{1,2,3,4}Penza State University, Penza, Russia

¹cnit@pnzgu.ru, ²karamyshevans@yandex.ru, ³vt@pnzgu.ru, ⁴zsa49@yandex.ru

Abstract. *Background.* A technique for deep knowledge structuring in a semantic-oriented intelligent computing environment based on the expansion of the descriptive capabilities of Petri nets by integrating them with conceptual graphs is proposed. One of the bases for integrating these formalisms for describing situational systems models is not only convenient graphical representations of conceptual graphs and Petri nets, but also their general semantics based on the use of domain properties. The applications obtained as a result of the integration can be further used as prototypes when creating expert or reference systems, as well as expert simulation systems for various subject areas. The relevance of research on conceptual graphs and Petri nets is associated with a wide area of their applicability in modeling discrete-event systems. Concept-events and causal relationships between them can be used in conceptual graphs. For participants in events, the roles they play can be defined. Petri nets are actually models similar to semantic nets and conceptual graphs that define relationships between position objects and transition objects. The integration is based on the static and dynamic components of the Petri net on the one hand, and on the other hand, the declarative nature of conceptual graphs. The purpose of the study is to expand the field of activity based on the interpretation of Petri nets and conceptual graphs and their integration by identifying the general semantics associated with the semantics and concepts of the studied subject area. *Materials and methods.* Integration and interpretation of conceptual graphs and logical, or binary, Petri nets is based on the use of first-order predicate logic, the theory of Petri nets, and the study of the applicability of conceptual models in various subject areas. *Results.* A technique for the synthesis of conceptual logical Petri nets is proposed based on the identification of the general semantics of conceptual graphs and Petri nets, as a result of which models with declarative, imperative and dynamic properties can be built.

Keywords: relations, predicates, concepts, production rules, conceptual graphs, logical Petri nets, integration, interpretation, structuring and representation of knowledge, declarative and imperative models, conceptual Petri nets, model dynamics

For citation: Volchikhin V.I., Karamysheva N.S., Mitrokhin M.A., Zinkin S.A. Representation and structuring of knowledge in the semantic oriented computing environment. Part 1. Integration conceptual graphs and logical networks based on formalization of structured situations. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences.* 2023;(2):24–51. (In Russ.). doi: 10.21685/2072-3059-2023-2-3

Введение

Важный подкласс интеллектуальных систем представляют семантико-ориентированные и логико-аналитические системы, которые способны из-

влекать из больших данных семантическую информацию, обрабатывать и осуществлять ее поиск по запросам. Семанτικο-ориентированные модели распределенных информационных систем применяются для организации обмена данными и знаниями между различными группами пользователей [1, 2]. Семантический анализ представляет собой важный этап при выполнении алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании семантического представления текстов [3]. Полученная информация часто подвергается структуризации – делению элементов предметной области на классы и подклассы. Между классами, подклассами и их элементами устанавливаются логические связи, что позволяет считать получаемые данные знаниями.

Обычно под концептуальной структуризацией знаний подразумевается ранний этап формализации знаний: разработка неформального описания знаний о предметной области в виде графа, таблицы, диаграммы или текста, которое отражает основные концепции и взаимосвязи между понятиями предметной области [4]. Концептуальная структуризация знаний в распределенных интеллектуальных системах достаточно сложна и может быть значительно упрощена, если ориентировать ее на последующее использование моделей предметной области, основанных на хорошо известных и изученных формализмах, некоторые из которых могут быть непосредственно не связаны с проблематикой интеллектуальных систем. При этом формализация, возможно первоначально слабая, постепенно усиливается вплоть до детальных непосредственно исполнимых формализованных спецификаций в терминах предметной области.

В настоящей работе предполагается, что подобный подход позволит снизить риски искажения знаний при формализации, поскольку формализация может выполняться постепенно, на основе пошаговой детализации и уточнении ситуаций, а также иерархической структуризации интеллектуальной системы, проектируемой для конкретной предметной области.

Действия в сложных системах, происходящие при возможном участии человека, и их взаимосвязи могут быть промоделированы сетями Петри [5–8]. Существует масса приложений аппарата и технологий данных сетей, разработанных для различных предметных областей [9–12]. Описательные возможности сетей Петри могут быть расширены путем их интеграции с концептуальными графами, семантика которых основана на формализме логики предикатов первого порядка. Основой для интеграции данных формализмов для описания моделей дискретных систем являются удобные графические представления объектов, или концептов. Концептуальные графы содержат графические представления классов, отношений, индивидуумов и квантификаторов [13–18]. Этот формализм основан на семантических сетях, к подклассу которых можно отнести и сети Петри, но имеет прямой выход на язык логики предикатов, из которого он берет свою семантику. Полученные в результате интеграции приложения могут использоваться далее в качестве прототипов при создании экспертных или справочных систем, а также экспертных систем имитационного моделирования для различных предметных областей.

Целью настоящей работы является расширение области деятельности, основанной на интерпретации сетей Петри и концептуальных графов и их интеграции за счет выявления общей семантики, связанной не только се-

мантикой сетей Петри и концептуальных графов как таковых, но, в большей степени, с семантикой и понятиями исследуемой предметной области.

Обоснование интеграции концептуальных графов и сетей Петри

Согласно классическим работам [13, 14] концептуальный граф представляет собой графическое представление логики, основанное на семантических сетях, используемых в моделях искусственного интеллекта, и на экзистенциальных графах Ч. С. Пирса. Несколько версий концептуальных графов были разработаны и внедрены в течение последних лет. При исследовании различных видов концептуальных графов разработаны новые методы проведения рассуждений, представления знаний и изучения семантики естественного языка.

Концептуальные графы представляют собой графическую интерпретацию формул в исчислении предикатов первого порядка. Согласно определению [13] семантические сети, называемые иногда концептуальными сетями, формируются как композиции концептуальных графов частного вида типа «звезда». Каждый концептуальный граф при этом содержит центральную вершину, или первичный концепт, связанный бинарными отношениями с несколькими вторичными концептами. Однако часто семантические сети и концептуальные графы с контекстным наполнением рассматриваются как одно и то же понятие, если речь не идет о точном формульном описании в рамках первопорядковой логики предикатов.

Концептуальные графы и сети Петри успешно применяются при моделировании дискретно-событийных систем. В концептуальных графах могут быть определены концепты-события и причинно-следственные связи между ними. Для участников событий могут быть определены исполняемые ими роли. Сети Петри фактически представляют собой модели, похожие на семантические сети и концептуальные графы, в которых определены отношения между объектами-позициями и объектами-переходами.

Известно [6–8], что сеть Петри, рассматриваемая в качестве модели дискретных процессов, имеет две составляющие – *статическую* и *динамическую*. Статическая составляющая представлена двудольным графом с начальной маркировкой позиций. Эта компонента порождает динамическую компоненту – модель, исследуемую в глобальных состояниях.

Определение концептуальных графов и сетей Петри показывает их сходство: графы обеих моделей являются двудольными и отношения инцидентности одинаковы по форме. Главное же различие заключается в том, что с позиций искусственного интеллекта концептуальными графами описывается декларативная составляющая модели представления знаний о предметной области, в то время как сетью Петри описывается как *декларативная*, так и *императивная* (процедурная) составляющая. Интеграция средств описания и системного анализа сетевых моделей обоих классов может повысить качество проектирования моделируемых систем. Важная разновидность сетей Петри – логические, или бинарные, сети Петри, рассматриваемые в целях последующей интеграции их с концептуальными графами, будут описаны далее в настоящей работе.

В концептуальных графах аргументы, или концепты, обозначающие понятия предметной области, на рисунках часто обозначаются прямоугольниками, а отношения между этими понятиями обозначаются овалами. Овала-

ми также обозначаются унарные отношения, характеризующие свойства объектов. Концептуальный граф, таким образом, может быть описан простой логической формулой, представляющей собой конъюнкцию бинарных и унарных предикатов. Принят международный стандарт ISO/IEC 24707, специфицирующий синтаксис и семантику моделей концептуальных графов общего вида – CGIF (англ. *Conceptual Graphs Interchange Format*) [19, 20]. Исследования в области концептуальных логических моделей в настоящее время сопровождаются разработкой и использованием множества формальных и неформальных расширений.

При выполнении настоящей работы были исследованы следующие программные средства для работы с концептуальными графами и связанными с ними приложениями (источник – сайт *A World of Conceptual Graphs (CG)* [19]):

Amine – многоуровневая платформа, предназначенная для разработки интеллектуальных и многоагентных систем;

CharGer – концептуальный графический редактор, реализованный на языке Java;

CG Mars Lander – инструмент быстрого поиска концептуальных графов и ответов на вопросы, доступный для совместной разработки и промышленного финансирования;

CoGITaNT – несколько полезных утилит: набор библиотечных подпрограмм на C++ для концептуального моделирования, несколько баз знаний, представленных в форме концептуальных графов, и спецификация XML для CG XML;

CPE – среда, которая содержит модули для предоставления пользователю функциональности без необходимости использования всей среды; в настоящее время существуют редакторы CGIF, ARCEdit и другие средства, поддерживающие операции компьютерной графики;

GoGui – визуальный инструмент на основе графов, разработанный на Java для построения баз знаний, основанных на концептуальных графах, представленных в формате COGXML и совместимых с форматом CoGITaNT;

Prolog+CG – объектно-ориентированное расширение языка Prolog, основанное на CG, где CG – это базовая структура данных; Prolog+CG реализован на языке Java;

WebKB – инструменты для поиска информации и представления знаний.

Список программных средств для работы с концептуальными графами периодически обновляется на сайте *International Conference on Conceptual Structures*¹.

Для дальнейшего использования в настоящей работе выбран интеллектуальный графический редактор CharGer [21, 22], интегрированный с языком SWI Prolog [23].

Логические (бинарные) сети Петри

Сеть Петри по определению, данному в классических работах [6–8], представляет собой двудольный ориентированный граф, содержащий позиции, переходы, дуги и метки. Дуги связывают позиции с переходами и пере-

¹ <https://iccs-conference.org/>

ходы с позициями. Переходы, их входные и выходные позиции могут представлять события обработки данных с входной и выходной логикой. При анализе динамического поведения системы, моделируемой сетью Петри, исследуется процесс смены ее состояний, моделируемый путем запуска, или срабатывания, переходов. При срабатывании какого-либо перехода изменяются состояния его входных и выходных позиций, что сопровождается перемещением меток, или маркеров, из одних позиций в другие. Метка – это примитивное понятие для сетей Петри и вводится в дополнение к позициям и переходам.

Официально история сетей Петри начинается с 1962 г., когда К. Петри защитил диссертацию и опубликовал ее [5]. Современная литература, связанная с сетями Петри, обширна и разнообразна [6–12]. Было продемонстрировано, что сети Петри являются мощным инструментом моделирования различных типов дискретных динамических систем, управляемых событиями. Исследование проводилось во многих отраслях, каждая из которых изучала важный и перспективный аспект применения этого формализма.

Сеть Петри [6, 7] называется размеченный маркированный двудольный ориентированный граф $PN = (P, T, F, W, M_0)$, где $P = \{p_0, p_1, \dots, p_m\}$ – конечное множество позиций; $T = \{t_1, t_2, \dots, t_n\}$ – конечное множество переходов; $P \cap T = \emptyset$; $P \cup T \neq \emptyset$; $F \subseteq (P \times T) \cup (T \times P)$ – отношение инцидентности, или отношение потока, определяющее множество дуг; $W: F \rightarrow \{1, 2, 3, \dots\}$ – функция кратности дуг; $M_0: P \rightarrow \{0, 1, 2, \dots\}$ – функция начальной маркировки позиций. В процессе срабатывания переходов происходит перемещение меток и, как следствие, смена маркировок. Подробное описание правил срабатывания переходов и связанных с ними смен маркировок дано в работах [6–8]. Для интеграции сетей Петри с концептуальными логическими моделями наиболее интересен случай, когда $W: F \rightarrow \{1\}$, т.е. кратность всех дуг равна единице и, как следствие, по каждой дуге может передаваться только одна метка; $M_0: P \rightarrow \{0, 1\}$ – при любой достижимой в сети маркировке в каждой позиции смогут находиться не более чем по одной метке (такая сеть Петри называется безопасной, или 1-ограниченной, и может быть легко трансформирована в логическую, или бинарную, сеть).

Для практического применения аппарата сетей Петри, как и многих других поведенческих формализмов, необходима программная поддержка имитационного моделирования и построения графа достижимых состояний. Для сетей Петри это графы достижимых разметок позиций. В ряде случаев может быть полезен анализ сетей Петри на основе матричных уравнений [24].

Другая сторона приложений сетей Петри, в отличие от имитационного моделирования, связана с реализацией аппаратного и программного обеспечения для параллельных и распределенных вычислительных систем. Одной из первых работ в области исследования связей сетей Петри с программированием является монография [7]. В частности, в этой работе большое внимание уделено семантике структур управления в программах, реализованных на основе формализма сетей Петри. Рассмотрена семантика и логика реализации событийного, потокового и динамического управления, причем условия готовности формулируются как логическая функция от некоторых событий.

Известно множество модификаций сетей Петри [25]: простые, автоматные, маркированные, свободного выбора, ингибиторные, правильно сформированные WF (англ. *Well Formed nets*), временные, цветные, иерархические, нечеткие, стохастические, функциональные. К комбинированным сетям Петри относятся следующие: иерархические цветные, временные цветные, иерархические цветные временные, цветные нечеткие, иерархические цветные временные нечеткие и др. Сети Петри моделируют очень широкий класс логических задач. Логические методы используются для анализа нечетких и логических (бинарных) сетей Петри, что позволяет создавать модели систем на основе нечеткой логики, исчислений высказываний и предикатов.

В работе [26] исследованы операционная семантика декларативных и процедурных знания в задаче логического программирования сетей Петри. Предлагаются сети Петри логического программирования (LPPN), создаваемые на основе двух семантик: гибридной операционной семантики собственно сети Петри и денотационной семантики на основе исчисления событий. Сети LPPN имеют интерфейс со средствами логического программирования. Полученные результаты применены при решении задачи управления реактивными интеллектуальными агентами. Предлагается особое внимание обращать на метки (маркеры), которые в простых логических сетях Петри (SLPN) не передают данные, но в целом такие сети пригодны для проверки агентских программ. В других приложениях сетей Петри требуется, чтобы метки переносили данные (сети DPN).

В работах [27, 28] исследуются методы оптимального управления в асинхронных булевских сетях, моделируемых сетями Петри. Предложены методы трансформации булевских сетей в сети Петри, а затем сети Петри трансформируются в сети логических динамических систем. Результаты работы используются при моделировании биологических систем.

Достаточно подробный обзор средств редактирования и моделирования сетей Петри представлен на сайте *Welcome to the Petri Nets World*¹. Для внутреннего представления сетей Петри в памяти компьютера часто используется язык разметки PNML (англ. *The Petri Net Markup Language*) – это основанный на языке разметки XML формат для большинства редакторов и симуляторов сетей Петри. Язык PNML в настоящее время стандартизирован (ISO/IEC-15909).

Приведенные далее определения логических сетей Петри в целом соответствуют описаниям из работ [29–34] и приведены для уточнения используемых обозначений и терминологии. В дальнейшем эти уточнения понадобятся при интеграции концептуальных графов с логическими сетями Петри.

Логическая, или бинарная, сеть Петри LN (*Logical Net*) определяется как следующий кортеж из пяти элементов [35–37]:

$$LN = (P, T, F, H, R_0), \quad (1)$$

где $P = \{P_1, P_2, \dots, P_m\}$ – конечное множество позиций; $T = \{T_1, T_2, \dots, T_n\}$ – конечное множество переходов, $P \cup T \neq \emptyset$, и $P \cap T = \emptyset$; $F(p, t): P \times T \rightarrow \{\mathbf{true}, \mathbf{false}\}$, $p \in P, t \in T$, – входной по отношению к переходу бинарный предикат, при интерпретации которого определяются дуги, ведущие от позиций к переходам; $H(t, p): T \times P \rightarrow \{\mathbf{true}, \mathbf{false}\}$, $p \in P, t \in T$, – выходной по

¹ <https://www2.informatik.uni-hamburg.de/TGI/PetriNets/tools/>

отношению к переходу бинарный предикат, при интерпретации которого определяются дуги, ведущие от переходов к позициям; $R_0(p): P \rightarrow \{\mathbf{true}, \mathbf{false}\}$, $p \in P$, – начальная интерпретация унарного предиката разметки $R(p)$.

Логическая сеть Петри с дополнительными ингибиторными (сдерживающими) [6–8] и информационными дугами *LNII* (*Logical net with additional Inhibitory and Information arcs*) определяется как кортеж с семью элементами [35–37]:

$$LNII = (P, T, F, F_{Inh}, F_{Inf}, H, R_0), \quad (2)$$

где бинарные предикаты F_{Inh} и F_{Inf} задают дуги с дополнительными свойствами:

$F_{Inh}(p, t): P \times T \rightarrow \{\mathbf{true}, \mathbf{false}\}$, $p \in P$, $t \in T$, – входной по отношению к переходу бинарный предикат, при интерпретации которого определяются ингибиторные, или сдерживающие, дуги, ведущие от позиций к переходам; например, ингибиторная дуга (P_i, T_j) , определенная истинным высказыванием $F_{Inh}(P_i, T_j)$, является разрешающей по отношению к переходу T_j в том случае, если $R(P_i) = \mathbf{false}$, т.е. в случае, если в позиции P_i отсутствует метка;

$F_{Inf}(p, t): P \times T \rightarrow \{\mathbf{true}, \mathbf{false}\}$, $p \in P$, $t \in T$, – входной по отношению к переходу бинарный предикат, при интерпретации которого определяются информационные дуги, ведущие от позиций к переходам; например, информационная дуга (P_i, T_j) , определенная истинным высказыванием $F_{Inf}(P_i, T_j)$, является разрешающей по отношению к переходу T_j в том случае, если $R(P_i) = \mathbf{true}$, т.е. в случае, если в позиции P_i присутствует метка; после срабатывания перехода T_j метка остается в прежней позиции.

Функционирование определенных выше логических сетей Петри, определенных кортежами (8) и (9), как и обычных сетей Петри, описывается формально при помощи логических продукционных правил срабатывания переходов, задающих множество последовательностей срабатываний и множество достижимых разметок позиций. Дополнительные ингибиторные дуги в логических сетях Петри приходится вводить по той причине, что при моделировании часто неизвестно, истинно ли, например, высказывание $R(P_j)$ для позиции P_j , являющейся выходной для некоторого перехода T_i и связанной с ним дугой (T_i, P_j) . При обычном, без «участия» ингибиторной дуги (P_j, T_i) , срабатывании перехода T_i и при истинности высказывания $R(P_j)$ поступление новой метки вызовет лишь подтверждение истинности этого высказывания. Эта ситуация является ошибочной, так как эквивалентна потере метки. Введение ингибиторной дуги (P_j, T_i) , что эквивалентно выполнению присваивания $F_{Inh}(P_j, T_i) := \mathbf{true}$, позволит избежать этой ошибки путем запрета на срабатывание перехода T_i в данной ситуации.

Ранее в работах [38–40] был предложен вариант интеграции моделей двух классов: концептуальных графов и сетей Петри в виде интегрированной концептуальной сети Петри (КСП). В работах [38, 39] определена «слабая» интеграция концептуальных графов и сетей Петри, выражающаяся в определении свойств переходов и позиций. В отличие от предлагаемой в настоящей работе методологии, в работе [40] сеть Петри не рассматривается как часть концептуального графа в том смысле, что для нее не определяются автоматически правила срабатывания переходов в рамках собственно концептуальной модели.

Методика построения концептуальных графов для логических сетей Петри

Несмотря на наличие ряда работ, связанных с проблематикой сетей Петри и концептуальных графов, проблемы их интеграции в рамках одного формализма полностью не решены, хотя очевидна необходимость сочетания процедурной операционной семантики сетей Петри, широко используемых при спецификации программ и моделировании широкого класса систем, с моделями представления знаний, описываемых концептуальными графами.

Оба направления в развитии информационных технологий успешно развиваются и взаимно дополняют друг друга, но требуют дальнейших интеграционных усилий.

Представление унарных и бинарных предикатов в концептуальных графах

Основная используемая в настоящей работе терминология, касающаяся определения и графического представления концептуальных графов, иллюстрируется на рис. 1. Унарные предикаты могут характеризовать свойства объектов, представленных концептами-позициями, а бинарные предикаты могут задавать логические связи между этими объектами. Правила срабатывания концептов-переходов в концептуальных графах (КГ) для логических сетей Петри (ЛСП) далее в настоящей работе будут представлены в форме продукционных правил, содержащих символы « \rightarrow », разделяющие посылки (условия, при истинности которых происходит срабатывание переходов) и следствия (логические выражения, становящиеся истинными после срабатывания переходов).

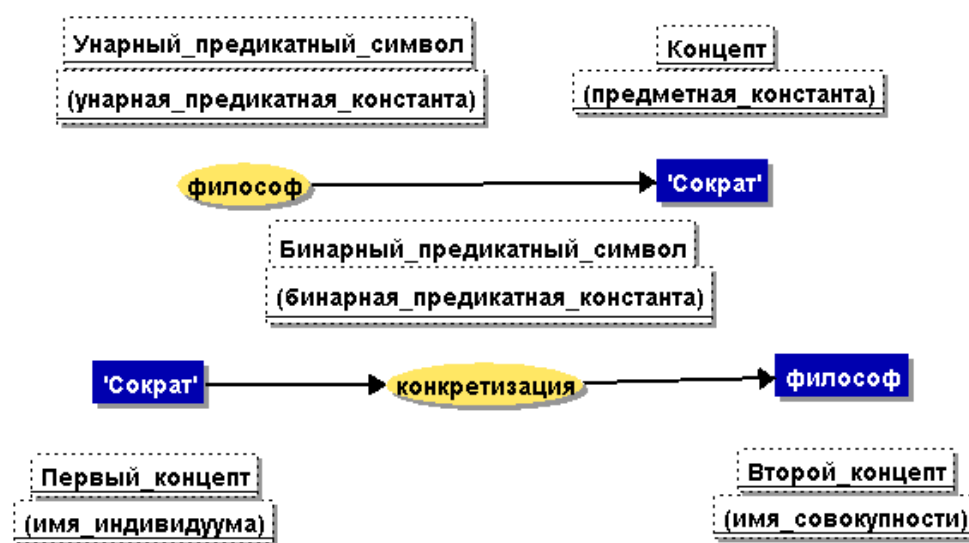


Рис. 1. Представление фразы «Сократ – философ» концептуальными графами: унарным и бинарным предикатами

В связи с тем, что вводимый в настоящей работе формализм основан на логике высказываний и логике предикатов первого порядка, необходимо

уточнить некоторые используемые понятия и определения. Рассмотрим различные способы логического и концептуального описания некоторой фразы, например утверждения «Сократ – философ», отдавая дань уважения великому древнегреческому философу [41, 42]. На верхней части рис. 1 дано графическое представление следующего высказывания:

$$\text{философ}('Сократ'), \quad (3)$$

где *философ* – унарное предикатное имя, или предикатная константа; '*Сократ*' – предметная константа; на нижней части рис. 1 дано представление той же самой фразы следующим высказыванием:

$$\text{конкретизация}('Сократ', \text{философ}). \quad (4)$$

Здесь для бинарного предиката *конкретизация* определены: в качестве значения первого аргумента – имя индивидуума, а в качестве значения второго аргумента – имя совокупности, или класса. В концептуальном графе для предиката первый и второй его аргументы представлены *концептами* (обозначены прямоугольниками), или *сущностями* предметной области. Овалом обозначен сам *предикат* или одноименное *отношение*, рассматриваемое как область истинности указанного предиката [13].

Выражения (3), (4) и соответствующие им концептуальные графы на рис. 1, естественно, различны, однако их можно считать эквивалентными в том смысле, что они представляют одну и ту же фразу.

Переинтерпретация концептуальных моделей

Модели на основе логики предикатов первого порядка для различных предметных областей в общем случае могут содержать выражения с n -арными предикатами, где $n > 2$ и конечно. Построение же обычных концептуальных графов и семантических сетей основано на использовании бинарных предикатов. Известная методика подобной переинтерпретации рассмотрена в работе [13]. Применение данной методики переинтерпретации модели с n -арным предикатом в модель на основе бинарных предикатов иллюстрирует следующий пример из работ [41, 42]. На рис. 2 изображен концептуальный граф G_1 для следующего высказывания:

$$\text{event}('Nadezhda', 'Gleb', '11h_40m_01_05_2023', e_mail, \text{letter_26}), \quad (5)$$

формализующего фразу

“ ‘Nadezhda’ sent to ‘Gleb’ at ‘11h_40m_01_05_2023’ by e_mail the letter_26” (русс. “Надежда послала Глебу в 11 часов 40 минут

$$1 \text{ мая } 2023 \text{ г. по электронной почте письмо_26}”). \quad (6)$$

Номера в ромбах (ромбами здесь обозначены не акторы, а простые нумерованные бирки) означают порядок следования предметных констант в выражении (5), которое записано в нотации факта языка Prolog.

Высказывание (3) получено путем фиксации предметных переменных следующего 5-арного предиката:

$$\text{event}(\text{sender}, \text{receiver}, \text{time}, \text{medium}, \text{letter}), \quad (7)$$

его концептуальный граф G_2 представлен на рис. 3, из которого следуют что для события *event* определены отправитель *sender*, получатель *receiver*, время

выполнения события *time*, среда передачи информации *medium* и пересылаемое письмо *letter*.

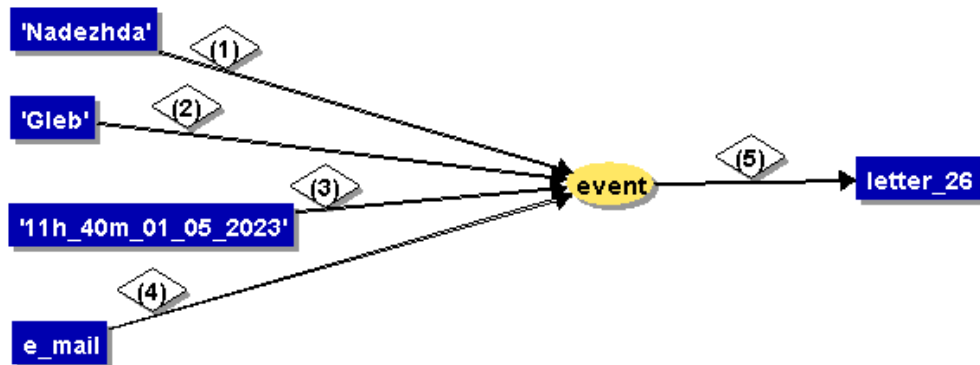


Рис. 2. Концептуальный граф G_1 для высказывания (5)

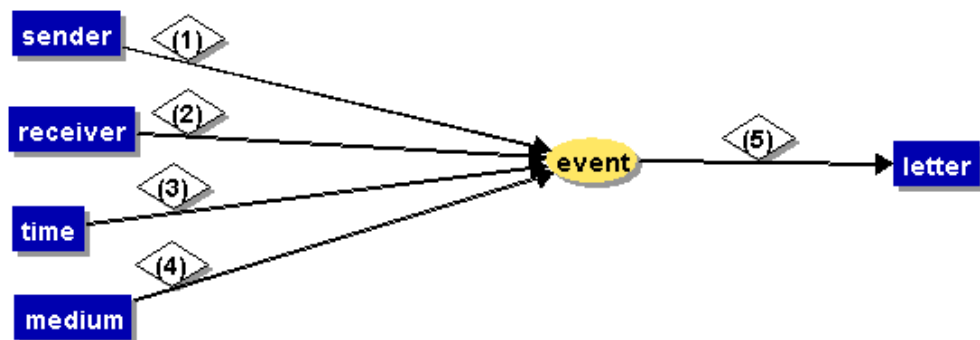


Рис. 3. Концептуальный граф G_2 для 5-арного предиката (7)

Фраза (6) представляется составным высказыванием – конъюнкцией высказываний (8) с именами бинарных предикатов и аргументами – предметными константами 'Nadezhda', 'Gleb', '11h_40m_01_05_2023', *e_mail* и *letter_26*, значениями соответствующих предметных переменных *sender*, *receiver*, *time*, *medium* и *letter*):

```
send(message_8, 'Nadezhda')&
accept(message_8, 'Gleb')&
date(message_8, '11h_40m_01_05_2023')&
means(message_8, e_mail)&
object(message_8, letter_26),
```

(8)

и далее переходим к составному предикату, представляющему собой конъюнкцию бинарных предикатов с аргументами – предметными переменными *sender*, *receiver*, *time*, *medium*, *letter*:

```
send(event, sender)&
accept(event, receiver)&
date(event, time)&
means(event, medium)&
object(event, letter).
```

(9)

Здесь и далее текст, выделенный прямым шрифтом, получен автоматически по внутреннему XML-представлению концептуального графа. Соответствующие выражения (8) и (9) концептуальные графы G_3 и G_4 представлены на рис. 4 и 5.

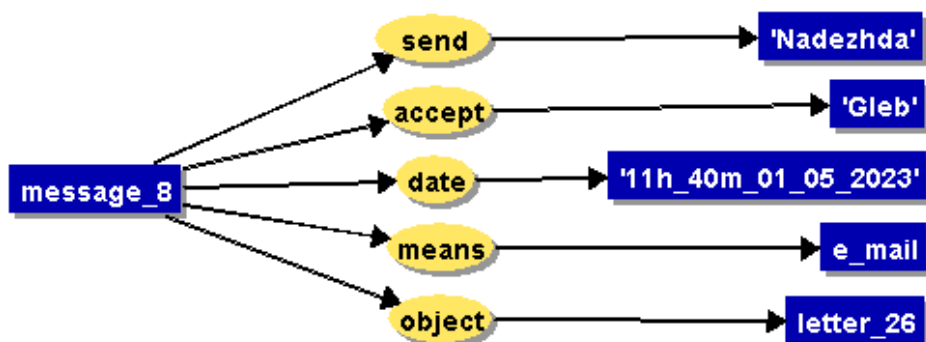


Рис. 4. Концептуальный граф G_3 , соответствующий выражению (8)

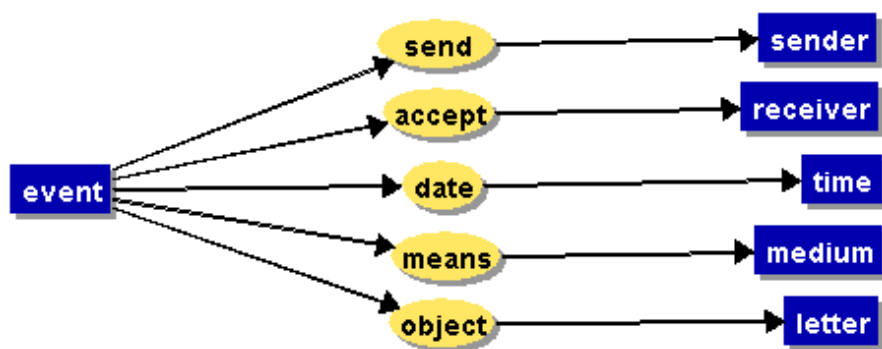


Рис. 5. Концептуальный граф G_4 , соответствующий выражению (9)

Концептуальные графы G_3 и G_4 , изображенные на рис. 4 и 5, при совмещении соответствующих концептов и отношений представляют в графической форме *событийный функциональный фрейм*, где имена совокупностей, или предметные переменные, *sender*, *receiver*, *time*, *medium* и *letter* могут интерпретироваться как имена слотов, а соответствующие им предметные константы *'Nadezhda'*, *'Gleb'*, *'11h_40m_01_05_2023'*, *e_mail* и *letter_26* – как значения слотов. В исходном виде концептуальный граф G_3 на рис. 4 соответствует *фрейму-экземпляру*, а граф G_4 на рис. 5 – *фрейму-образцу*, или *фрейму-прототипу* [13].

При структуризации знаний о предметной области путем перехода от графической формы к табличной необходимо подвергнуть конвертированию в текстовую форму графы G_3 и G_4 по раздельности и объединить полученные таблицы в одну, получив следующий результат – факты базы знаний [41, 42]:

```

accept(event, receiver).
accept(message_8, 'Gleb').
date(event, time).
date(message_8, '11h_40m_01_05_2023').
means(event, medium).
  
```

```

means(message_8, e_mail).
object(event, letter).
object(message_8, letter_26).
send(event, sender).
send(message_8, 'Nadezhda').
when(message_8, past).

```

Средствами языка программирования с использованием логики Prolog и языка запросов к реляционным базам данных SQL в прямой форме нельзя определить бинарные отношения, связывающие концепты, в силу того, что при создании данных языков было недопустимо использование логик порядков выше первого, так называемых логик HOL (англ. *High Order Logics* – логики высших порядков). В ряде случаев возможна переинтерпретация моделей, созданных на основе HOL, в модели на основе первопорядковой логики FOL (*First Order Logic*) и последующее использование приемлемых программных средств. Переинтерпретация не является эквивалентным преобразованием моделей, но сохраняет их семантику.

Перенеся предикатные символы-константы в кортежи с предметными константами и определив новое тернарное отношение *triada* для полученных троек (триад), получим выражения для следующих фактов базы знаний:

```

triada(accept, event, receiver).
triada(accept, message_8, 'Gleb').
triada(date, event, time).
triada(date, message_8, '11h_40m_01_05_2023').
triada(means, event, medium).
triada(means, message_8, e_mail).
triada(object, event, letter).
triada(object, message_8, letter_26).
triada(send, event, sender).
triada(send, message_8, 'Nadezhda').
triada(when, message_8, past).

```

Объединяя бинарные и тернарные отношения, получим факты общей базы знаний, представляющей структурированную информацию для фразы о посылке письма: “*Nadezhda sent a letter to Gleb by e-mail, and Gleb accepted this letter*” (“Надежда послала письмо Глебу по электронной почте, а Глеб получил это письмо”):

```

accept(event, receiver).
accept(message_8, 'Gleb').
date(event, time).
date(message_8, '11h_40m_01_05_2023').
means(event, medium).
means(message_8, e_mail).
object(event, letter).
object(message_8, letter_26).
send(event, sender).
send(message_8, 'Nadezhda').
when(message_8, past).
triada(accept, event, receiver).

```

```

triada(accept, message_8, 'Gleb').
triada(date, event, time).
triada(date, message_8, '11h_40m_01_05_2023').
triada(means, event, medium).
triada(means, message_8, e_mail).
triada(object, event, letter).
triada(object, message_8, letter_26).
triada(send, event, sender).
triada(send, message_8, 'Nadezhda').
triada(when, message_8, past).

```

Примечание 1. По дате '11h_40m_01_05_2023' отправки письма по электронной почте можно определить, что данное событие *message_8* состоялось в прошлом. Уточнить это возможно путем добавления факта *when(message_8, past)* к множеству фактов. Другая возможность уточнения этой семантической информации состоит в явном указании временных форм глаголов: *sent* и *accepted*, которое, однако, усложняет машинное восприятие событий.

Примечание 2. Полученные факты базы знаний "Почта" представлены в форме, готовой для включения в программы на языке SWI Prolog. При составлении многих программ необходимо различать концепты, представляющие объекты – элементы классов или множеств, и концепты, представляющие классы и множества объектов. Для этого следует добавить к базе знаний новые факты. Например, рассмотрим факты *accept(event, receiver)* и *accept(message_8, 'Gleb')*, где концепты с именами *event* и *receiver* представляют классы, а концепты с именами *message_8* и 'Gleb' представляют сами элементы соответствующих классов. Вводя факты *is-a(message_8, event)* и *is-a('Gleb', receiver)*, где *is-a* – бинарный предикатный символ, определяющий отношение принадлежности элемента к классу, получим возможность составлять правила, позволяющие учитывать различие между элементами классов и классами. Другая возможность состоит в определении свойств концептов при помощи унарных предикатов, непосредственно указывающих на классы и на элементы классов. Например, факт *class(event)* обозначает, что концепт *event* представляет какой-либо класс, а факт *object(message_8)* – что концепт *message_8* представляет элемент, или объект, какого-либо класса.

Дальнейшая структуризация информации о состоявшемся событии отправки почтового отправления может быть произведена путем разбиения события *event* на два события – *event_1* и *event_2*, из которых второе следует за первым – определено отношение *next*. Для аналитического или графического представления этого факта потребовалась логика предикатов второго порядка (HOL2 – *High Order Logic 2*) [43], которая позволила бы задать отношение *next* над отношениями *send* и *accept*. Этот факт описывается следующей формулой:

$$\text{next}(\text{send}(\text{event_1}, \text{sender}), \text{accept}(\text{event_2}, \text{receiver})), \quad (10)$$

где *next* – бинарный предикатный символ второго порядка. Воспользовавшись тем фактом, что концептами могут быть простые и составные высказывания, представим последнюю формулу графом G_5 (рис. 6).

Для дальнейшего использования факта (10) в программах на основе первопорядковой логики следует перейти к семантически эквивалентному выражению для факта на основе следующего 7-арного предиката:

$$\text{letter7}(\text{next}, \text{send}, \text{event_1}, \text{sender}, \text{accept}, \text{event_2}, \text{receiver}). \quad (11)$$

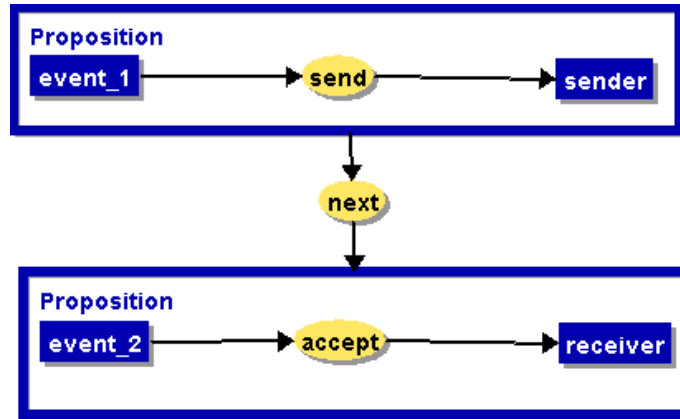


Рис. 6. Представление концептуальным графом G_5 выражения (10)

В следующей формуле для указания на то, что оба события состоялись в прошлом, используется унарный предикат *past* третьего порядка, примененный к предикату второго порядка:

$$past(next(send(event_1, sender), accept(event_2, receiver))). \quad (12)$$

Данная формула представима графом G_6 (рис. 7).

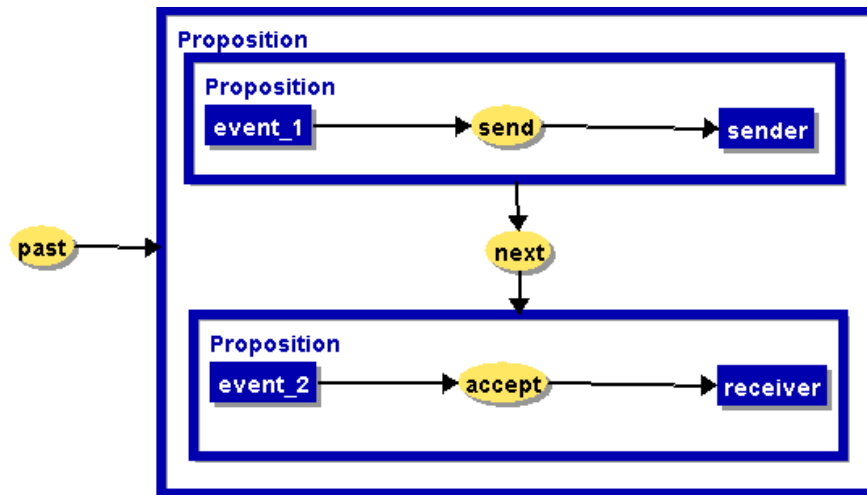


Рис. 7. Представление концептуальным графом G_6 выражения (12)

В результате перехода к первопорядковой логике получится следующее выражение для факта на основе 8-арного предиката:

$$letter8(past, next, send, event_1, sender, accept, event_2, receiver). \quad (13)$$

Для указания на определенные времена свершения для обоих событий вводятся концепты (атрибуты) *time_1* и *time_2*, чему соответствует следующее выражение:

$$next((date(send(event_1, sender), time_1), date(accept(event_2, receiver), time_2))). \quad (14)$$

Здесь *date* – предикатный символ второго, а *next* – третьего порядка. Данная формула представима графом G_7 (рис. 8). На основе выражения (14) получено выражение для факта на основе 11-арного предиката первого порядка:

$$\text{letter11}(\text{next}, \text{date_1}, \text{send}, \text{event_1}, \text{sender}, \text{time_1}, \text{date_2}, \text{accept}, \text{event_2}, \text{receiver}, \text{time_2}). \quad (15)$$

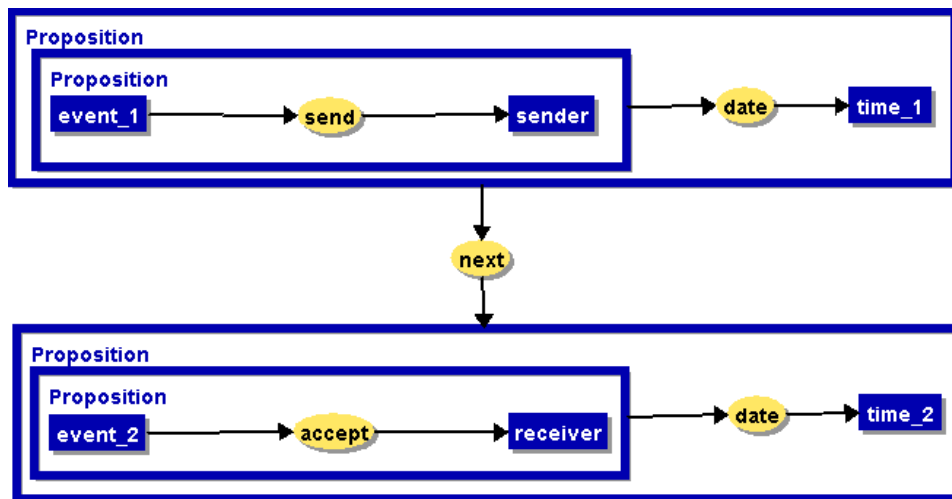


Рис. 8. Представление концептуальным графом G_7 выражения (14)

Проведенная таким образом структуризация знаний, первоначально представленных фразой на естественном языке, позволяет использовать язык программирования Prolog, созданный на основе первопорядковой логики, для дальнейшего анализа знаний. Использование логик высших порядков целесообразно в случае обеспечения соответствующей программной поддержки.

При структуризации знаний, проводимой экспертом, могут быть определены отношения между объектами и классами (понятиями предметной области), в том числе и такие, которые явно не заданы фразой на естественном языке.

Графические концептуальные и логические представления сетей Петри

Интеграция декларативного подхода на основе графической формы логической модели представления знаний с исполнимой моделью на основе сети Петри, также представимой в виде графического образа, позволит расширить состав моделей для различных предметных областей. На начальном этапе интеграции определяется логическая сеть Петри, которая может быть получена путем простой трансформации из безопасной, или 1-ограниченной, сети Петри, проблема безопасности для которой разрешима [7, 8].

На основе данных определений (1) и (2) возможно представление логических сетей Петри LN и $LNII$ в форме концептуальных графов. На рис. 9 и рис. 10 представлены примеры концептуальных сетей Петри. Как концепты-позиции, так и концепты-переходы обозначены прямоугольниками и отлича-

ются друг от друга только именами и, возможно, цветом. Это связано с особенностями реализации используемого редактора концептуальных графов. Предикатам F , F_{inh} , F_{inf} , H соответствуют овалы. Для указания начальной разметки можно воспользоваться графическим представлением унарного предиката разметки R , как показано на рис. 9 для позиций P_1 и P_2 . При реализации симулятора логической сети Петри разметку можно отмечать цветом графического объекта.

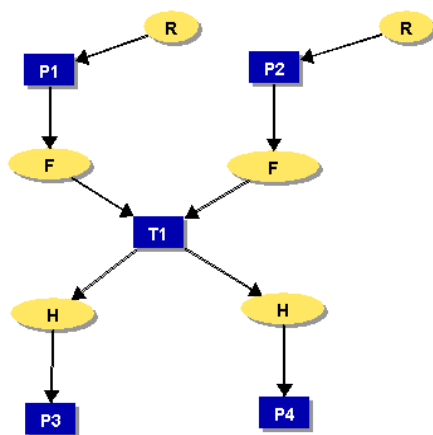


Рис. 9. Концептуальный граф для простой логической сети Петри

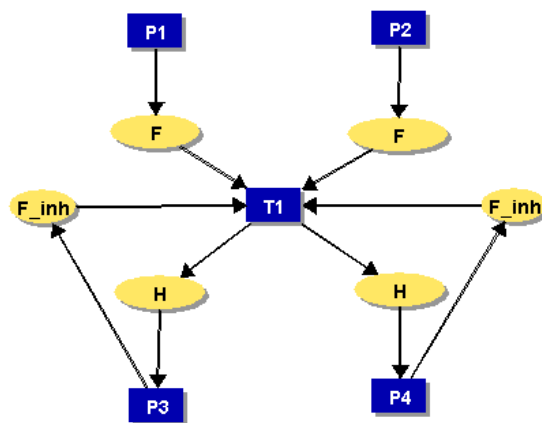


Рис. 10. Концептуальный граф для логической сети Петри с обычными и ингибиторными дугами

Редактор концептуальных графов позволяет получить факты, или высказывания, соответствующие структуре КСП на рис. 9, которые автоматически включены в конъюнкцию высказываний, содержащих имена бинарных и унарных предикатов и предметных констант, обозначающих концепты-переходы и концепты-позиции:

There is a Proposition where:

H of T1 is P3 and
H of T1 is P4 and
F of P1 is T1 and
F of P2 is T1 and
R is P1 and
R is P2.

The Final Form of the Compound Proposition is:

$$F(P1, T1) \& F(P2, T1) \& H(T1, P3) \& H(T1, P4) \& R(P1) \& R(P2). \quad (16)$$

Здесь первая фраза означает, что далее следует составное высказывание, а вторая фраза – что сформировано составное высказывание в окончательной форме. Истинность выражения (16) свидетельствует о том, что граф КСП сформирован.

Примечание. Здесь и далее прямым шрифтом выделены факты, правила и формулы в том виде, в котором они сформированы программой-редактором концептуальных графов по их внутреннему представлению

в формате языка разметки XML. Другие формулы, представленные в обычном курсивном математическом стиле, сформированы вручную.

Правило срабатывания концепта-перехода T_1 при начальной разметке $R(P_1) = \mathbf{true}$ и $R(P_2) = \mathbf{true}$ имеет следующий вид:

$$T_1: R(P_1) \& R(P_2) \rightarrow \neg R(P_1) \& \neg R(P_2) \& R(P_3) \& R(P_4). \quad (17)$$

Предполагается, что концепты-позиции P_3 и P_4 до срабатывания концепта-перехода T_1 не были размечены, т.е. были ложны высказывания $R(P_3) = \mathbf{false}$ и $R(P_4) = \mathbf{false}$.

Если отсутствие меток в выходных позициях нельзя гарантировать, то в концептуальный граф следует добавить ингибиторные дуги $F_{inh}(P_3, T_1)$ и $F_{inh}(P_4, T_1)$, связывающие выходные концепты-позиции с концептом-переходом. Составные высказывания, описывающие граф КСП на рис. 10, формируются редактором концептуальных графов автоматически:

There is a Proposition where:

F_inh of P3 is T1 and

F_inh of P4 is T1 and

F of P1 is T1 and

F of P2 is T1 and

H of T1 is P3 and

H of T1 is P4.

The Final Form of the Compound Proposition is:

$F_{inh}(P_3, T_1) \& F_{inh}(P_4, T_1) \& F(P_1, T_1) \& F(P_2, T_1) \& H(T_1, P_3) \& H(T_1, P_4)$.

Правило срабатывания перехода T_1 в этом случае записывается следующим образом:

$$T_1: R(P_1) \& R(P_2) \& \neg R(P_3) \& \neg R(P_4) \rightarrow \neg R(P_1) \& \neg R(P_2) \& R(P_3) \& R(P_4). \quad (18)$$

В выражении для правила (18) учтено, что в графе КСП выходные позиции P_3 и P_4 связаны с переходом T_1 ингибиторными дугами, как и показано на рис. 10. От правила (17) правило (18) отличается тем, что в его левую часть добавлена конъюнкция $\neg R(P_3) \& \neg R(P_4)$.

При наличии нескольких типов дуг – обычных, ингибиторных и информационных, составные высказывания, описывающие граф КСП на рис. 11, формируются редактором концептуальных графов автоматически и имеют следующий вид:

There is a Proposition where:

F of P1 is T1 and

F of P2 is T1 and

F_inh of P6 is T1 and

F_inh of P5 is T1 and

F_inf of P4 is T1 and

F_inh of P3 is T1 and

H of T1 is P5 and

H of T1 is P6.

The Final Form of the Compound Proposition is:

$F(P_1, T_1) \& F(P_2, T_1) \& F_inh(P_6, T_1) \& F_inh(P_5, T_1) \& F_inf(P_4, T_1) \& F_inh(P_3, T_1) \& H(T_1, P_5) \& H(T_1, P_6)$.

Правило срабатывания перехода T_1 примет следующий вид:

$$T_1: R(P_1) \& R(P_2) \& \neg R(P_3) \& R(P_4) \& \neg R(P_5) \& \neg R(P_6) \rightarrow \neg R(P_1) \& \neg R(P_2) \& R(P_5) \& R(P_6). \quad (19)$$

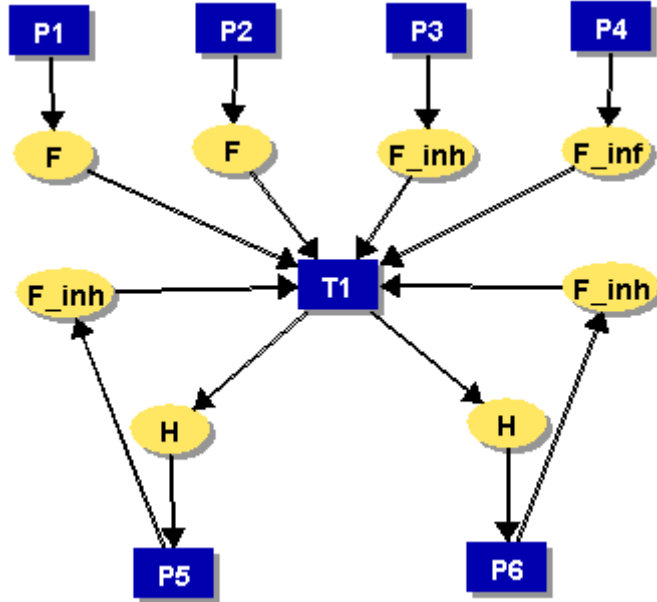


Рис. 11. Концептуальный граф для логической сети Петри с обычными, ингибиторными и информационными дугами

Из выражения (13) для этого правила следует, что после срабатывания перехода T_1 разметка позиций P_3 и P_4 , связанных с данным переходом ингибиторной и информационной дугами соответственно, не изменится.

Рассматривая приведенные правила срабатывания переходов, можно заметить, что в выражениях отсутствуют высказывания с обозначениями имен предикатов H , F , F_inh и F_inf . В приведенном ниже выражении для правила срабатывания перехода T_1 на рис. 11 включение этих новых высказываний позволит учесть наличие или отсутствие связей позиций с переходами:

$$T_1: R(P_1) \& F(P_1, T_1) \& R(P_2) \& F(P_2, T_1) \& \neg R(P_3) \& \neg R(P_5) \& \neg R(P_6) \& F_inh(P_3, T_1) \& R(P_4) \& F_inf(P_4, T_1) \& F_inh(P_6, T_1) \& F_inh(P_5, T_1) \rightarrow \neg R(P_1) \& \neg R(P_2) \& R(P_5) \& H(T_1, P_5) \& R(P_6) \& H(T_1, P_6). \quad (20)$$

Отсутствие любой из связей $F(P_1, T_1)$, $F(P_2, T_1)$, $F_inh(P_3, T_1)$, $F_inf(P_4, T_1)$, $H(T_1, P_5)$ или $H(T_1, P_6)$ в КСП сделает невозможной передачу метки далее по сети, что эквивалентно невозможности срабатывания перехода T_1 .

Приведенный пример наглядно демонстрирует двойственный декларативный и процедурный (исполнимый) характер сетей Петри. Предикатами

F , H , F_{inh} и F_{inf} задаются структурные связи между понятиями предметной области, моделируемой сетью Петри. Эти связи определяют декларативную часть модели представления знаний о предметной области. Другую, процедурную часть, определяют правила срабатывания переходов, модифицирующие унарный предикат R разметки позиций. Отсюда следует обоснование возможности интеграции концептуальных графов и логических сетей Петри в рамках одной модели.

Иерархические концептуальные сети Петри

Структура многих сложных систем является иерархической и состоит из взаимосвязанных подсистем. Важные характеристики сложной системы определяются не только свойствами ее элементов, но и взаимосвязями этих элементов в некоторой иерархии. Использование иерархических сетей Петри предоставляет возможность понижения сложности большой модели путем разделения ее на ряд подмоделей. Подмодели могут повторно использоваться далее при синтезе иерархических сетей Петри.

Одними из первых работ, в которых были предложены методы исследования иерархических сетей, называемых также структурированными сетями, являются работы [7, 44], в которых введено понятие составного перехода и отмечено, что срабатывание структурного перехода является не мгновенным событием, а составным действием. В работе [45] структурный подход использован при проектировании мультипроцессорной вычислительной системы управления базой данных. В статьях [46, 47] предложены некоторые способы трансформации иерархических сетей Петри. Использование методики концептуальной структуризации, предложенной в настоящей работе, позволяет осуществлять преобразования, в отличие от [45–47], при помощи средств логического программирования, что упрощает проведение семантического анализа моделей для заданной предметной области. Использование интегрированных иерархических концептуальных сетей Петри (ИКСП) способно вывести исследование иерархических систем на новый уровень абстракции, что поможет облегчить их синтез.

Пример фрагмента логической ИКСП приведен на рис. 12. Эта сеть содержит составной переход Ts , который запускается при истинности составного высказывания $R(P_1) \& R(P_2)$. Запуск сопровождается появлением метки в позиции P_3 , что в логической сети соответствует истинности высказывания $R(P_3)$. Затем продолжается выполнение составного события – срабатывания составного перехода до появления метки в позиции P_7 , что отмечается истинностью высказывания $R(P_7)$. Выполнение фрагмента завершается появлением меток в позициях P_8 и P_9 .

Составные высказывания, описывающие структурные связи в графе ИКСП на рис. 12, формируются редактором концептуальных графов автоматически и имеют следующий вид:

There is a Proposition Ts where:
 F of P_3 is T_1 and
 H of T_1 is P_4 and
 H of T_1 is P_5 and
 H of T_1 is P_6 and
 F of P_4 is T_2 and

F of P5 is T2 and
F of P6 is T2 and
H of T2 is P7;

There is a Proposition where:

F of P1 is Ts and
F of P2 is Ts and
H of Ts is P8 and
H of Ts is P9.

The Final Form of the Compound Proposition Ts is:

$Ts = F(P3, T1) \& H(T1, P4) \& H(T1, P5) \& H(T1, P6) \& F(P4, T2) \& F(P5, T2) \& F(P6, T2) \& H(T2, P7).$

The Final Form of the Compound Proposition is:

$F(P1, Ts) \& F(P2, Ts) \& H(Ts, P8) \& H(Ts, P9).$

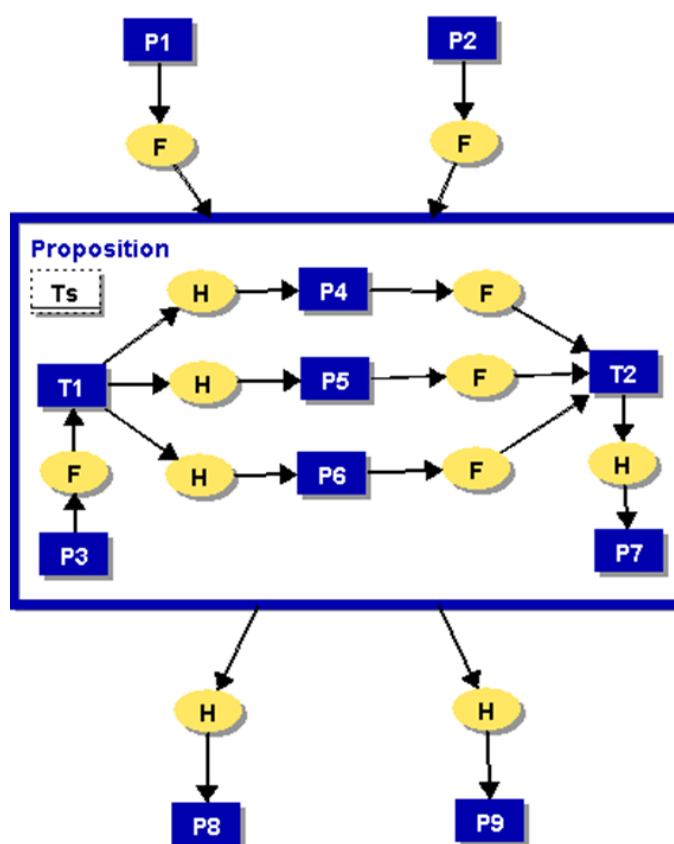


Рис. 12. Пример фрагмента иерархической концептуальной логической сети Петри с составным переходом

Правила срабатывания составного перехода Ts при выполнении составного события имеют следующий вид:

$Ts_{in}: R(P_1) \& R(P_2) \rightarrow \neg R(P_1) \& \neg R(P_2) \& S(P_3);$
 $T_1: S(P_3) \rightarrow \neg S(P_3) \& S(P_4) \& S(P_5) \& S(P_6);$

$$T_2: S(P_4) \& S(P_5) \& S(P_6) \rightarrow \neg S(P_4) \& \neg S(P_5) \& \neg S(P_6) \& S(P_7);$$

$$T_{Sout}: S(P_7) \rightarrow \neg S(P_7) \& R(P_8) \& R(P_9),$$

где S – унарный предикатный символ разметки внутренних позиций, принадлежащих составному переходу Ts ; Ts_{in} – вход перехода Ts ; Ts_{out} – выход перехода Ts .

Заключение

Предложены методы структурирования знаний в семантико-ориентированной интеллектуальной информационно-вычислительной среде, основанные на глубокой интеграции сетей Петри и концептуальных графов и их интеграции за счет выявления общей семантики. Основами для интеграции данных формализмов при разработке моделей дискретных систем являются не только удобные графические представления концептуальных графов и сетей Петри, но и общая семантика, тесно связанная с семантикой предметной области, в результате чего могут быть построены модели, обладающие декларативными, императивными (процедурными) и динамическими свойствами. Показано, что интеграция и интерпретация концептуальных графов и логических, или бинарных, сетей Петри может быть основана на использовании логики предикатов первого порядка, теории сетей Петри, исследовании применимости концептуальных моделей в различных предметных областях. Продемонстрирована возможность структурирования знаний о предметной области при помощи логики предикатов первого и высших порядков.

Определены концептуальные графы для представления логических сетей Петри с обычными, ингибиторными и информационными дугами. Предложена методика автоматизированного синтеза концептуальных логических сетей Петри. Полученные результаты позволяют расширить области деятельности, основанные на интерпретации сетей Петри и концептуальных графов и их интеграции за счет выявления общей семантики, связанной не только семантикой сетей Петри и концептуальных графов как таковых, но, в большей степени, с семантикой и понятиями исследуемой предметной области.

Список литературы

1. Кузнецов И. П., Мацкевич А. Г. Семантико-ориентированные системы на основе баз знаний. М. : МТУСИ, 2007. 173 с.
2. Шарнин М. М., Кузнецов И. П. Особенности семантического поиска информационных объектов на основе технологии баз знаний // Информатика и ее применение. 2012. Т. 6, № 2. С. 49–57.
3. Корешкова Т. Семантический анализ для автоматической обработки естественного языка. URL: https://rdc.grfc.ru/2021/09/semantic_analysis/ (дата обращения: 31.01.2023).
4. Авдеева З. К., Коврига С. В. Эвристический метод концептуальной структуризации знаний при формализации слабоструктурированных ситуаций на основе когнитивных карт // Управление большими системами. 2010. № 31. С. 6–34.
5. Petri C. A. Kommunikation mit Automaten. English Translation: Communication with Automata : preprint (Report RADG-TR-65-377). New York, 1966.
6. Питерсон Дж. Теория сетей Петри и моделирование систем. М. : Мир, 1984. 254 с.
7. Котов В. Е. Сети Петри. М. : Наука, 1984. 160 с.
8. Мараховский В. Б., Розенблюм Л. Я., Яковлев А. В. Моделирование параллельных процессов. Сети Петри. СПб. : Профессиональная литература, 2014. 400 с.

9. Розенблум Л. Я. Сети Петри // Известия АН СССР. Техническая кибернетика. М., 1983. № 5. С. 12–40.
10. Кулагин В. П. Моделирование структур параллельных вычислительных систем на основе сетевых моделей : учеб. пособие. М. : МГИРЭА(ТУ), 2008. 102 с.
11. Petri Nets. Fundamental Models, Verification and Applications / ed. by M. Diaz. John Wiley and Sons, 2009. 613 p.
12. Petri Nets. Theory and Applications / ed. by V. Kordic. Vienna, Austria : I-TECH Education and Publishing, 2008. 544 p.
13. Тейз А., Грибомон П., Луи Ж. [и др.]. Логический подход к искусственному интеллекту: от классической логики к логическому программированию / пер. с фр. П. Пермякова ; под ред. Г. П. Гаврилова. М. : Мир, 1990. 429 с.
14. Sowa J. F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA : Brooks Cole Publishing Co., 2000. 594 p.
15. Sowa J. F. Conceptual Graphs / Handbook of Knowledge Representation / ed. by F. van Harmelen, V. Lifschitz, B. Porter. Elsevier, 2008. P. 213–237.
16. Sowa J. F. Cognitive Architectures for Conceptual Structures // Proc. 19th International Conference on Conceptual Structures / ed. by S. Andrews, S. Polovina, R. Hill, B. Akhgar, LNAI 6828. Heidelberg : Springer, 2011. P. 35–49.
17. Polovina S. An Introduction to Conceptual Graphs. Conceptual Structures: Knowledge Architectures for Smart Applications / ed. by U. Priss, S. Polovina, R. Hill // Lecture Notes in Artificial Intelligence (LNAI 4604). Springer, 2007. P. 1–15.
18. Палагин А. В., Кривой С. Л., Петренко Н. Г. Концептуальные графы и семантические сети в системах обработки естественно-языковой информации // Математические машины и системы. 2009. № 3. С. 67–79.
19. A World of Conceptual Graphs. URL: <http://conceptualgraphs.org/> (дата обращения: 31.01.2023).
20. Власенко С. В., Варчев Д. А. Использование аппарата концептуальных графов в задачах представления и обработки знаний : учеб. пособие. СПб. : Изд-во СПбГЭТУ «ЛЭТИ», 2009. 64 с.
21. CharGer Manual v3.5b1 2005-11-30. P. 1–58. URL: <http://charger.sourceforge.net/> (дата обращения: 31.01.2023).
22. Delugach H. CharGer – A Conceptual Graph Editor written by Harry Delugach. URL: <http://www.cs.uah.edu/~delugach/CharGer/> (дата обращения: 31.01.2023).
23. SWI-Prolog. URL: <https://www.swi-prolog.org/> (дата обращения: 31.01.2023).
24. Murata T. Petri nets: properties, analysis and applications // Proceedings of the IEEE. 1989. Vol. 77 (4). P. 541–580.
25. Ganzhura M. A., Smirnova O. V., Zakharova A. A., Romanov D. L., Dyachenko N. V. Methods of Modeling Objects Using Petri Nets // XV International Scientific-Technical Conference “Dynamic of Technical Systems” (DTS-2019) AIP Conf. Proc. AIP Publishing, 2019. Vol. 2188. P. 1–8. doi: 10.1063/1.5138450
26. Sileno G. Operationalizing Declarative and Procedural Knowledge: a Benchmark on Logic Programming Petri Nets (LPPNs) // International Conference on Logic Programming (ICLP2020) Workshop on Causal Reasoning and Explanation in Logic Programming (CAUSAL2020). 2020. P. 1–13.
27. Kobayashi K., Hiraishi K. Optimal Control of Asynchronous Boolean Networks Modeled by Petri Nets // Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011). 2011. Vol. 724. P. 7–20. URL: <http://ceur-ws.org/>
28. Akutsu T., Hayashida M., Ching W.-K., Ng M. K. Control of Boolean networks: Hardness results and algorithms for tree structured networks // Journal of Theoretical Biology. 2007. Vol. 244. P. 670–679.
29. Yaqub O., Li L. Modeling and Analysis of Connected Traffic Intersections Based on Modified Binary Petri Nets // International Journal of Vehicular Technology. Hindawi Publishing Corporation, 2013. Vol. 2013. 10 p.

30. Iordache M. V., Antsaklis P. J. Supervisory Control of Concurrent Systems. A Petri Net Structural Approach. Boston : Birkhauser, 2006. 281 p.
31. van der Aalst W. M. P. A class of Petri nets for modeling and analyzing business processes // Computing Science Reports 95/26. Eindhoven : Eindhoven University of Technology, 1995. P. 1–25.
32. Du Y., Jiang C. Formal Representation and Analysis of Batch Stock Trading Systems by Logical Petri Net Workflows // Lecture Notes in Computer Science. 2002. Vol. 2495. P. 221–225.
33. Du Y., Jiang C., Guo Y. Towards a Formal Model for Grid Architecture via Petri Nets // Information Technology Journal. 2006. Vol. 5. P. 833–841. doi: 10.3923/itj.2006.833.841
34. Du Y., Qi L., Zhou M. Analysis and Application of Logical Petri Nets to E-Commerce Systems // IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2014. Vol. 44, № 4. P. 468–481. doi: 10.1109/TSMC.2013.2277696
35. Волчихин В. И., Карамышева Н. С., Зинкин С. А., Гурин Е. И. Алгоритмика, логика и моделирование агентно-базированных метакомпьютерных систем с повышенным уровнем параллельности // Известия высших учебных заведений. Поволжский регион. Технические науки. 2022. № 2. С. 5–25. doi: 10.21685/2072-3059-2022-2-1
36. Zinkin S. A., Volchihin V. I., Karamysheva N. S., Jaafar M. S. Synthesis of the Functional Architecture of Agent-based Computing Systems when Using Conceptual Models of Artificial Intelligence // 2021 International Conference on Engineering Management of Communication and Technology, EMCTECH 2021. Austria, Vienna, 2021. P. 1–6. doi: 10.1109/EMCTECH53459.2021.9619174
37. Zinkin S. A., Volchihin V. I., Karamysheva N. S., Jaafar M. S. Dynamic Topology Transformation of Cloud-Network Computer Systems: Conceptual Level // 2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020. Austria, Vienna, 2020. P. 1–10. doi: 10.1109/EMCTECH49634.2020.9261554
38. Зинкин С. А., Пашенко Д. В., Пучкова У. Н., Джафар М. С. Интеграция методов концептуального и поведенческого моделирования дискретно-событийных систем: I. Синтез и анализ концептуальной модели // Кибернетика и программирование. 2016. № 6. С. 83–95.
39. Зинкин С. А., Пашенко Д. В., Пучкова У. Н., Джафар М. С. Интеграция методов концептуального и поведенческого моделирования дискретно-событийных систем: II. Логико-алгебраические операционные модели и инфокоммуникационные технологии // Кибернетика и программирование. 2017. № 1. С. 75–93.
40. Зинкин С. А., Джафар М. С., 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.
41. Митрохин М. А., Карамышева Н. С., Егоров В. Ю., Зинкин С. А. Интеллектуальная обработка информации : учеб. пособие. Пенза : Изд-во ПГУ, 2023. 320 с.
42. Карамышева Н. С., Зинкин С. А., Митрохин М. А. Методы и модели интеллектуализации вычислительных и киберфизических систем : учеб. пособие. Пенза : Изд-во ПГУ, 2022. 176 с.
43. Miller D., Nadathur G. Programming with Higher-Order Logic. Cambridge University Press, 2012. 249 p. doi: 10.1017/CBO9781139021326
44. Котов В. Е., Черкасова Л. А. Структурированные сети // Кибернетика. 1981. № 4. С. 33–41.
45. Вашкевич Н. П., Зинкин С. А., Кулагин В. П. Структурный подход к проектированию мультипроцессорной вычислительной системы управления базой данных // Известия вузов. Приборостроение. 1983. Т. 24, № 9. С. 15–24.

46. Lorbeer J.-U., Padberg J. Hierarchical, Reconfigurable Petri Nets / Workshops at Modellierung 2018 / ed. by I. Schaefer, L. Cleophas, M. Felderer // Petri Nets and Modeling 2018 (PeMod18). 2018. P. 167–186.
47. Palacz W. Algebraic hierarchical graph transformation // Journal of Computer and System Sciences. 2004. Vol. 68 (3). P. 497–520.

References

1. Kuznetsov I.P., Matskevich A.G. *Semantiko-orientirovannye sistemy na osnove baz znaniy = Semantically oriented systems based on knowledge bases*. Moscow: MTUSI, 2007:173. (In Russ.)
2. Sharnin M.M., Kuznetsov I.P. Features of the semantic search for information objects based on knowledge base technology. *Informatika i ee primeneniye = Computer science and its application*. 2012;6(2):49–57. (In Russ.)
3. Koreshkova T. *Semanticheskii analiz dlya avtomaticheskoy obrabotki estestvennogo yazyka = Semantic analysis for automatic natural language processing*. (In Russ.). Available at: https://rdc.grfc.ru/2021/09/semantic_analysis/ (accessed 31.01.2023).
4. Avdeeva Z.K., Kovriga S.V. Heuristic method of conceptual structuring of knowledge in the formalization of semi-structured situations based on cognitive maps. *Upravlenie bol'shimi sistemami = Large systems management*. 2010;(31):6–34. (In Russ.)
5. Petri C.A. *Kommunikation mit Automaten. English Translation: Communication with Automata: preprint (Report RADC-TR-65-377)*. New York, 1966.
6. Piterson Dzh. *Teoriya setey Petri i modelirovanie system = Petri net theory and system modeling*. Moscow: Mir, 1984:254. (In Russ.)
7. Kotov V.E. *Seti Petri = Petri nets*. Moscow: Nauka, 1984:160. (In Russ.)
8. Marakhovskiy V.B., Rozenblyum L.Ya., Yakovlev A.V. *Modelirovanie parallel'nykh protsessov. Seti Petri = Simulation of parallel processes. Petri nets*. Saint Petersburg: Professional'naya literatura, 2014:400. (In Russ.)
9. Rozenblyum L.Ya. Petri nets. *Izvestiya AN SSSR. Tekhnicheskaya kibernetika = Proceedings of the Academy of Sciences of the USSR. Technical cybernetics*. Moscow, 1983;(5):12–40. (In Russ.)
10. Kulagin V.P. *Modelirovanie struktur parallel'nykh vychislitel'nykh sistem na osnove setevykh modeley: ucheb. posobie = Modeling the structures of parallel computing systems based on network models: textbook*. Moscow: MGIREA(TU), 2008:102. (In Russ.)
11. Diaz M. (ed.). *Petri Nets. Fundamental Models, Verification and Applications*. John Wiley and Sons, 2009:613.
12. Kordic V. (ed.). *Petri Nets. Theory and Applications*. Vienna, Austria: I-TECH Education and Publishing, 2008:544.
13. Teyz A., Gribomon P., Lui Zh. et al. *Logicheskii podkhod k iskusstvennomu intellektu: ot klassicheskoy logiki k logicheskomu programmirovaniyu = Logic approach to artificial intelligence: from classical logic to logic programming*. Translated from French by P. Permyakova. Moscow: Mir, 1990:429. (In Russ.)
14. Sowa J.F. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co., 2000:594.
15. Sowa J.F. *Conceptual Graphs / Handbook of Knowledge Representation*. Elsevier, 2008:213–237.
16. Sowa J.F. Cognitive Architectures for Conceptual Structures. *Proc. 19th International Conference on Conceptual Structures*. Ed. by S. Andrews, S. Polovina, R. Hill, B. Akhgar, LNAI 6828. Heidelberg: Springer, 2011:35–49.
17. Polovina S. An Introduction to Conceptual Graphs. *Conceptual Structures: Knowledge Architectures for Smart Applications. Lecture Notes in Artificial Intelligence (LNAI 4604)*. Springer, 2007:1–15.
18. Palagin A.V., Krivoy S.L., Petrenko N.G. Conceptual graphs and semantic networks in natural language information processing systems. *Matematicheskie mashiny i sistemy = Mathematical machines and systems*. 2009;(3):67–79. (In Russ.)

19. *A World of Conceptual Graphs*. Available at: <http://conceptualgraphs.org/> (accessed 31.01.2023).
20. Vlasenko S.V., Varchev D.A. *Ispol'zovanie apparata kontseptual'nykh grafov v zadachakh predstavleniya i obrabotki znaniy: ucheb. posobie = Using the apparatus of conceptual graphs in the tasks of knowledge representation and processing: textbook*. Saint Petersburg: Izd-vo SPbGETU «LETI», 2009:64. (In Russ.)
21. *CharGer Manual v3.5b1 2005-11-30*. P. 1–58. Available at: <http://charger.sourceforge.net/> (accessed 31.01.2023).
22. Delugach H. *CharGer – A Conceptual Graph Editor written by Harry Delugach*. Available at: <http://www.cs.uah.edu/~delugach/CharGer/> (accessed 31.01.2023).
23. *SWI-Prolog*. Available at: <https://www.swi-prolog.org/> (accessed 31.01.2023).
24. Murata T. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*. 1989;77(4):541–580.
25. Ganzhura M.A., Smirnova O.V., Zakharova A.A., Romanov D.L., Dyachenko N.V. Methods of Modeling Objects Using Petri Nets. *XV International Scientific-Technical Conference “Dynamic of Technical Systems” (DTS-2019) AIP Conf. Proc.* AIP Publishing, 2019;2188:1–8. doi: 10.1063/1.5138450
26. Sileno G. Operationalizing Declarative and Procedural Knowledge: a Benchmark on Logic Programming Petri Nets (LPPNs). *International Conference on Logic Programming (ICLP2020) Workshop on Causal Reasoning and Explanation in Logic Programming (CAUSAL2020)*. 2020:1–13.
27. Kobayashi K., Hiraishi K. Optimal Control of Asynchronous Boolean Networks Modeled by Petri Nets. *Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011)*. 2011;724:7–20. Available at: <http://ceur-ws.org/>
28. Akutsu T., Hayashida M., Ching W.-K., Ng M.K. Control of Boolean networks: Hardness results and algorithms for tree structured networks. *Journal of Theoretical Biology*. 2007;244:670–679.
29. Yaqub O., Li L. Modeling and Analysis of Connected Traffic Intersections Based on Modified Binary Petri Nets. *International Journal of Vehicular Technology*. Hindawi Publishing Corporation, 2013;2013:10.
30. Iordache M.V., Antsaklis P.J. *Supervisory Control of Concurrent Systems. A Petri Net Structural Approach*. Boston: Birkhauser, 2006:281.
31. van der Aalst W.M.P. A class of Petri nets for modeling and analyzing business processes. *Computing Science Reports 95/26*. Eindhoven: Eindhoven University of Technology. 1995:1–25.
32. Du Y., Jiang C. Formal Representation and Analysis of Batch Stock Trading Systems by Logical Petri Net Workflows. *Lecture Notes in Computer Science*. 2002;2495:221–225.
33. Du Y., Jiang C., Guo Y. Towards a Formal Model for Grid Architecture via Petri Nets. *Information Technology Journal*. 2006;5:833–841. doi: 10.3923/itj.2006.833.841
34. Du Y., Qi L., Zhou M. Analysis and Application of Logical Petri Nets to E-Commerce Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2014;44(4):468–481. doi: 10.1109/TSMC.2013.2277696
35. Volchikhin V.I., Karamysheva N.S., Zinkin S.A., Gurin E.I. Algorithms, logic and modeling of agent-based metacomputer systems with a high level of parallelism. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences*. 2022;(2):5–25. (In Russ.). doi: 10.21685/2072-3059-2022-2-1
36. Zinkin S.A., Volchikhin V.I., Karamysheva N.S., Jaafar M.S. Synthesis of the Functional Architecture of Agent-based Computing Systems when Using Conceptual Models of Artificial Intelligence. *2021 International Conference on Engineering Management of Communication and Technology, EMCTECH 2021*. Austria, Vienna, 2021:1–6. doi: 10.1109/EMCTECH53459.2021.9619174

37. Zinkin S.A., Volchihin V.I., Karamysheva N.S., Jaafar M.S. Dynamic Topology Transformation of Cloud-Network Computer Systems: Conceptual Level. *2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020*. Austria, Vienna, 2020:1–10. doi: 10.1109/EMCTECH49634.2020.9261554
38. Zinkin C.A., Pashchenko D.V., Puchkova U.N., Dzhafer M.S. Integration of methods of conceptual and behavioral modeling of discrete-event systems: 1. Synthesis and analysis of the conceptual model. *Kibernetika i programmirovaniye = Cybernetics and programming*. 2016;(6):83–95. (In Russ.)
39. Zinkin C.A., Pashchenko D.V., Puchkova U.N., Dzhafer M.S. Integration of methods of conceptual and behavioral modeling of discrete-event systems: 2. Logic-algebraic operating models and infocommunication technologies. *Kibernetika i programmirovaniye = Cybernetics and programming*. 2017;(1):75–93. (In Russ.)
40. Zinkin S.A., Dzhafer M.S., Karamysheva N.S. Conceptual representations and modifications of Petri nets for applications in the field of synthesis of the functional architecture of distributed computing systems with a variable structure. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwestern State University*. 2018;22(6):143–167. (In Russ.)
41. Mitrokhin M.A., Karamysheva N.S., Egorov V.Yu., Zinkin S.A. *Intellektual'naya obrabotka informatsii: ucheb. posobie = Intelligent Information Processing: textbook*. Penza: Izd-vo PGU, 2023:320. (In Russ.)
42. Karamysheva N.S., Zinkin S.A., Mitrokhin M.A. *Methods and models of intellectualization of computing and cyber-physical systems: textbook*. Penza: Izd-vo PGU, 2022:176. (In Russ.)
43. Miller D., Nadathur G. *Programming with Higher-Order Logic*. Cambridge University Press, 2012:249. doi: 10.1017/CBO9781139021326
44. Kotov V.E., Cherkasova L.A. Structured networks. *Kibernetika = Cybernetics*. 1981;(4):33–41. (In Russ.)
45. Vashkevich N.P., Zinkin S.A., Kulagin V.P. Structural approach to designing a multi-processor computing database management system. *Izvestiya vuzov. Priboroostroenie = University proceedings. Instrument engineering*. 1983;24(9):15–24. (In Russ.)
46. Lorbeer J.-U., Padberg J. Hierarchical, Reconfigurable Petri Nets / Workshops at Modelierung 2018 / ed. by I. Schaefer, L. Cleophas, M. Felderer. *Petri Nets and Modeling 2018 (PeMod18)*. 2018:167–186.
47. Palacz W. Algebraic hierarchical graph transformation. *Journal of Computer and System Sciences*. 2004;68(3):497–520.

Информация об авторах / Information about the authors

Владимир Иванович Волчихин

доктор технических наук, профессор,
президент Пензенского государственного
университета (Россия, г. Пенза,
ул. Красная, 40)

E-mail: cnit@pnzgu.ru

Vladimir I. Volchikhin

Doctor of engineering sciences, professor,
president of Penza State University
(40 Krasnaya street, Penza, Russia)

Надежда Сергеевна Карамышева

кандидат технических наук, доцент,
доцент кафедры вычислительной
техники, Пензенский государственный
университет (Россия, г. Пенза,
ул. Красная, 40)

E-mail: karamyshevans@yandex.ru

Nadezhda S. Karamysheva

Candidate of engineering sciences, associate
professor, associate professor of the
sub-department of computer
engineering, Penza State University
(40 Krasnaya street, Penza, Russia)

Максим Александрович Митрохин

доктор технических наук, доцент,
заведующий кафедрой вычислительной
техники, Пензенский государственный
университет (Россия, г. Пенза,
ул. Красная, 40)

E-mail: vt@pnzgu.ru

Maksim A. Mitrokhin

Doctor of engineering sciences,
associate professor, head of the
sub-department of computer
engineering, Penza State University
(40 Krasnaya street, Penza, Russia)

Сергей Александрович Зинкин

доктор технических наук, доцент,
профессор кафедры вычислительной
техники, Пензенский государственный
университет (Россия, г. Пенза,
ул. Красная, 40)

E-mail: zsa49@yandex.ru

Sergey A. Zinkin

Doctor of engineering sciences, associate
professor, professor of the sub-department
of computer engineering, Penza State
University (40 Krasnaya street,
Penza, Russia)

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.

Поступила в редакцию / Received 21.04.2023

Поступила после рецензирования и доработки / Revised 28.05.2023

Принята к публикации / Accepted 19.06.2023