

## Понятийная модель знаний

В. С. Выхованец

Московский государственный технический университет им. Н. Э. Баумана  
(национальный исследовательский университет), г. Москва, Россия

*valery@vykhovanets.ru*

**Аннотация.** В статье рассмотрены известные методы представления и обработки знаний. Предлагается использование новой модели, названной понятийной, отличающейся тем, что отношения (связи) между понятиями рассматриваются как обычные понятия. Понятие определено как форма мысли, выражаемая именованным множеством сущностей. Конкретные понятия идентифицируют одну сущность, а абстрактные образуются путём обобщения (объединения) или ассоциации (декартова произведения) других понятий. Декларативные знания задаются перечислимыми множествами сущностей, а процедурные – разрешимыми, где разрешающие процедуры выражаются формулами одноместного исчисления предикатов. Приведено описание прикладного языка представления и обработки знаний. Доказано, что запросы к понятийной модели выполняются за полиномиальное время от логарифма среднего числа сущностей у понятий. Обоснована возможность обучения понятийных моделей без учителя. Показано, что понятийная модель позволяет наглядно представлять и эффективно обрабатывать как декларативные, так и процедурные знания.

**Ключевые слова:** представление знаний, обработка знаний, модели знаний, понятийная модель, декларативные знания, процедурные знания, обучение без учителя, база знаний.

## Введение

Обработка знаний как задача искусственного интеллекта заключается в имитации техническими средствами приобретения, накопления, упорядочивания, адаптации, актуализации и использования знаний человеком [1].

Имеющиеся проблемы в области представления и обработки знаний выражаются в том, что ни одна из известных моделей знаний не позволяет реализовать и эффективно обрабатывать большие объемы знаний. Причина этих проблем в том, базовой частью всех моделей знаний является оргграф, вершинами которого являются понятия, а дуги задают отношения между ними. Отсюда любой запрос к базе знаний – это поиск в оргграфе подграфа с заданными свойствами (изоморфного графа), что в общем случае является NP-полной задачей [2].

В последнее время наметилась тенденция применения искусственных нейронных сетей для решения задач искусственного интеллекта. Однако искусственные нейронные сети также оказались непригодными для представления и обработки знаний т. к. при использовании таких сетей невозможно объяснить получаемые результаты, отсутствует гарантия получения верного (полного) решения задачи, наблюдается непредсказуемость прогностической способности сети, имеются трудности в решении задач по шагам и вычислительных задач, требуется длительное обучение сети и высокие затраты на получение обучающей выборки, имеет место низкое качество обучения при неполных или зашумленных обучающих данных, присутствует риск переобучения сети и зависимость строения сети от решаемой задачи, проявляется неадекватное

поведение сети в динамически изменяющихся предметных областях, и т. п.

Целью настоящей статьи является анализ известных моделей представления и обработки знаний и определение их достоинств и недостатков. Предлагается использование понятийной модели знаний, свободной от многих недостатков и отличающейся тем, что в ней отсутствует разделение на понятия и отношения между ними. По этой причине математическим языком понятийной модели служит одноместное исчисление предикатов, на котором выражаются как декларативные, так и процедурные знания. Обоснована возможность и эффективность использования понятийной модели для представления и обработки больших объемов знаний.

## **1. Структура знания**

В самом общем смысле знание – это психическое образование, направленное на адекватное отражение окружающей действительности в сознании человека, которое субъективно (идеально) по своей природе и не может быть непосредственно изучено [3]. Знания образуются в результате педагогического процесса, самообразования и жизненного опыта.

Знание проявляется как результат познавательной деятельности, который логически или фактически обоснован и допускает проверку на достоверность. Представление знаний – это отчуждение знания от его носителя в некоторой внешней (материальной) форме, а объективация знаний – признание знаний, представленных во внешней форме, объективно достоверными и социально значимыми.

Знание как результат абстрактно-логического мышления фиксируется в сознании человека в виде понятий, суждений, умозаключений, рассуждений, теорий и т. п.

Понятие – форма мышления, что-либо выделяющая и называющая.

Суждение – форма мышления, которая раскрывает связь между понятиями. Сложные суждения получаются в результате умозаключений и могут состоять из двух и более простых суждений.

Умозаключение – форма мышления, связывающее суждения-посылки с суждениями-следствиями. В зависимости от культурных традиций, уровня образования, психического состояния и т. п., правила выполнения умозаключений могут различаться. Умозаключения должны переносить достоверность (истинность, правдоподобность, вероятность, аналогичность, убедительность, аргументированность и т. п.) исходных суждений на результирующее [4]. Объективными считаются правила дедуктивных умозаключений, сформулированные в логике. Другим примером являются индуктивные и абдуктивные умозаключения, которые переносят на суждения-следствия правдоподобность, но не истинность [5].

Рассуждение – мыслительная деятельность, направленная на получение из суждений-посылок

суждений-следствий. В зависимости от используемых правил выполнения умозаключений различают дедуктивные, индуктивные и абдуктивные рассуждения, рассуждения на основе прецедентов, рассуждения по аналогии, вероятностные и немонотонные рассуждения, и др.

В свою очередь теория – это суждения-следствия, полученные в процессе рассуждения из некоторых суждений-посылок с помощью умозаключений заданного вида.

Таким образом знание – это результат рационального отражения действительности в сознании человека, представленный во внешней форме понятиями и суждениями, утвержденными некоторым множеством умозаключений.

## **2. Модели знаний**

Модель знаний – это средства формального описания знаний (язык знаний) и методы оперирования ими (вывод на знаниях). Известны несколько типов моделей знаний:

- логические модели (язык логики): исчисление предикатов [6], дескрипционные логики [7];
- продукционные модели (язык продукций) [8];
- сетевые модели (язык ориентированных графов): семантические сети [2], концептуальные графы [9];
- фреймовые модели (объектный язык): объектные (фреймовые) модели [10], реляционные модели [11];
- понятийные модели (язык понятий) [12].

Фундаментальными считаются логические модели знаний. Однако у логических моделей из-за монотонности вывода невозможен пересмотр промежуточных результатов, т. е. всё является фактами и нет гипотез. Также оказалось практически невозможным использование предикатов как предикатных переменных, что ограничивает выразительность логического языка. Неразрешимость формул логики первого порядка не позволяет надежно распознавать суждения о предметной области. Не представлены в языке логики средства для выражения процедурных знаний. Обучение логических моделей осуществляется только с учителем. Имеется высокая вычислительная сложность вывода в логике первого порядка, а также некоторая ограниченность и неестественность логического языка.

У продукционных моделей, основанных на правилах вида «если Условие, то Действие», имеются трудности проверки на непротиворечивость и полноту, наблюдается непредсказуемость результатов вывода из-за недетерминированности выбора продукций, при выводе необходимо использовать эвристики, отсутствуют средства для представления процедурных знаний, обучение моделей осуществляется только с учителем.

Фреймовая модель достаточно наглядна, однако в ней нет механизма управления выводом,

присутствует высокая структурная сложность данных, плохая модифицируемость знаний, наблюдается низкая скорость вывода, требующего применения эвристик. Обучаются фреймвые модели только с учителем.

Семантические сети наиболее распространены для представления знаний. Семантическая сеть состоит из большого числа понятий (узлов) и отношений между ними (дуг). Семантическая сеть непосредственно не содержит структуру предметной области, что приводит к сложности ее модификации и актуализации. Процедура вывода на сети не может гарантировать достоверность результата, а сам результат не имеет логического обоснования. В итоге вывод в сетевой модели возможен только приближенный, реализуемый при использовании эвристик. Так же, как и логические модели, семантические сети обучаются только с учителем и не имеют средств для представления процедурных знаний.

Многие из перечисленных недостатков моделей знаний отсутствуют у понятийных моделей.

### **3. Понятие и концепт**

Основным допущением, используемым в понятийных моделях, является различие понятия и концепта. Понятие (англ. notion) – это простая идея, мнение, представление или понимание чего-либо; понятийный (англ. notional) – это гипотетический, воображаемый [13, с. 538]. Понятие соотносится с представлениями из предметной области – сущностями (англ. entities). Понятие отличается от концепта.

Концепт (англ. concept) – это общее понятие, абстрактная идея, сформированная путем объединения всех аспектов некоторого класса объектов [13, с. 159]. Концепт состоит из экземпляров (англ. instances), каждый экземпляр характеризуется совокупностью свойств (англ. properties), а свойства могут быть определенными (англ. defining) и неопределенными (англ. nondefining) [14, с. 38]. Таким образом, концепт, по своему определению, объективирован, а понятие субъективно.

Проблемы однозначного и непротиворечивого описания концептов, связанные с изменчивостью набора свойств у экземпляров, привели к появлению альтернативных теорий концептуального моделирования [14, с. 39]. Часть таких теорий базируется на вероятностном подходе, при котором набор свойств расширяется и их появление у экземпляров концепта имеет вероятностную природу. Другие теории разделяют свойства экземпляров концепта на типичные, присущие всем экземплярам, и атипичные, которые могут появляться лишь у некоторых экземпляров. В предельном случае утверждается, что общего понимания концептов достичь не удастся по различным социальным, политическим и иным причинам [15, с. 334].

В понятийной модели проблемы, возникшие при описании концептов, решаются на основе

корпусного подхода – путем выделения в одной предметной области нескольких проблемных областей и формирования корпуса теорий [16]. Поскольку понятие субъективно, то необходимо предусмотреть возможность нескольких альтернативных описаний одного и того же понятия, в каждой проблемной области по-своему. Это позволит в последующем объективировать понятие в его коллективной (концептуальной) форме.

Проблемная область – это предметная область, рассматриваемая в некотором узком смысле (аспекте), с точки зрения некоторой активной проблемы, всякий раз по иному структурирующей анализируемую предметную область. Одно и то же понятие в разных аспектах (в разных проблемных областях) может иметь разные определения и описания. Для именованя понятий в различных аспектах к имени понятия добавляется имя аспекта, отделенное символом @. Если речь идет о концепте или о понятии в общем аспекте, то имя аспекта с символом @ к имени понятия не добавляется.

Например, концепт Проект является объективацией понятий Проект в аспектах Объект, План, Программа, Деятельность, Документ и т. д. Сущностями понятия Проект в аспекте Объект (понятие Проект@Объект) являются результаты целенаправленной практической деятельности; в аспекте План (понятие Проект@План) – разработанные планы (схемы) сооружений, механизмов, устройств, зданий и т. п.; в аспекте Программа (понятие Проект@Программа) – последовательность взаимосвязанных работ (задач) для достижения некоторой цели в заданное время; в аспекте Деятельность (понятие Проект@Деятельность) – временное предприятие, направленное на создание уникального продукта, услуги или результата; в аспекте Документ (понятие Проект@Документ) – предварительный текст какого-либо документа, резолюции, и т. д.

#### **4. Образование понятий**

При понятийном моделировании понятия формализуются как именованные множества сущностей предметной области и разделяются на конкретные и абстрактные. Конкретные понятия образуются на основе представлений, а абстрактные – путем выполнения операций над понятиями.

Конкретные понятия или понятия-знаки образуются путем идентификации (именования) уникальных представлений (сущностей) в той или иной проблемной области. В процессе интерпретации понятия-знака осуществляется обратный переход к обозначающим им сущностям.

Примерами понятий-знаков могут быть, например, сущности, идентифицируемые такими словами как Красный, Много, Иногда, Дружба и т. п. Обратный переход от понятий-знаков к

сущностям осуществляется при интерпретации этих слов.

Абстрактные понятия являются обобщениями или ассоциациями других понятий. Понятие-ассоциация образуется соединением (конкатенацией) сущностей ассоциируемых понятий в сущность понятия-ассоциации, а понятие-обобщение – объединением сущностей обобщаемых понятий.

Примерами понятий-обобщений может служить понятие Дерево, являющиеся обобщением понятий различных пород деревьев, или понятие Число, являющееся обобщением понятий различных видов чисел: Целых, Рациональных, Действительных, Комплексных.

Обратной операцией для обобщения является операция специализации, при которой из понятия-обобщения выделяются обобщенные в нем понятия.

Типизация – частный случай обобщения, при котором объединению подлежат понятия-знаки. Обратной операцией для типизации является операция конкретизации, при которой из понятия-типа выделяются типизированные в нем понятия.

Например, понятие Времени, является типизацией понятий-знаков, обозначающих различные моменты времени, а понятие Натуральное число – типизацией понятий-знаков натуральных чисел.

В свою очередь примером понятия-ассоциации может служить понятие Труба, соединяющее такие понятия как Материал, Диаметр и Длина, причем не все комбинации материала трубы, ее диаметра и длины являются допустимыми. Или понятие-ассоциация Сталь, соединяющее такие понятия как Углерод, Железо, Хром, Никель, Марганец и другие компоненты в заданных массовых долях для каждой марки стали. Как видно из примеров, ассоциация соединяет понятия, а не сущности, т. е. не является отношением «часть-целое» между экземплярами, как это имеет место в других моделях.

Обратной операцией для ассоциации является операция индивидуализации, при которой из понятия-ассоциации выделяются ассоциированные в ней понятия.

Частным случаем ассоциации является агрегация понятий. При агрегации сущности агрегированных понятий соединяются в сущности понятия-агрегата во всех возможных комбинациях. Обратной операцией для агрегации является операция декомпозиции, при которой из понятия-агрегата выделяются агрегированные понятия.

Примером понятия-агрегата является понятие Пространство, которое агрегирует три независимых понятия Координата; понятие Комплексное число, агрегирующее два независимых понятия числа; понятие Цвет, агрегирующее три понятия базовых цветов (Красный, Зеленый, Синий) различной интенсивности.

Следует заметить, что любая сущность предметной области в одних аспектах может

рассматриваться как понятие-знак, а в других – как различные абстрактные понятия, т. е. определяться как обобщения или ассоциации других понятий. Например, понятие Дерево в одном аспекте может быть обобщением пород деревьев, а в другом – ассоциацией корня, ствола, веток, листьев и т. д.

В итоге имеем, что материал, данный в представлениях, преобразуется в понятия с помощью следующих способов его логико-теоретической переработки: операций идентификации-интерпретации, обобщения-специализации (в частности типизации-конкретизации), ассоциации-индивидуализации (в частности агрегации-декомпозиции) [17].

## 5. Понятийная модель

Понятийная модель – это понятийная структура предметной области с описанием состава входящих в нее понятий. В понятийной структуре перечисляются понятия и указываются способы их образования, т. е. задаются интенционалы (содержания понятий). Описание состава понятий осуществляется заданием их экстенционалов (объемов понятий), т. е. перечислением или разрешением множеств сущностей, в них входящих.

Для задания интенционала понятия-обобщения используются фигурные скобки, в которых перечисляются обобщаемые понятия, а для задания интенционала понятия-ассоциации – квадратные скобки, в которых перечисляются ассоциируемые понятия:

$$\text{Int } G = \{G1 \ G2 \ \dots \ GK\}; \quad (1)$$

$$\text{Int } A = [A1 \ A2 \ \dots \ AL], \quad (2)$$

где  $\text{Int } G$  – интенционал понятия-обобщения с именем  $G$ ;  $G1, G2, \dots, GK$  – имена обобщаемых понятий;  $\text{Int } A$  – интенционал понятия-ассоциации с именем  $A$ ;  $A1, A2, \dots, AL$  – имена ассоциируемых понятий; знак равенства используется для установления эквивалентности между интенционалом образуемого понятия и выражением, его определяющим. Традиционно фигурные скобки указывают на неупорядоченное множество без повторения элементов, а квадратные скобки – на упорядоченное множество с повторением элементов.

На рис. 1 показан фрагмент понятийной структуры, в которой перечислены понятия-обобщения Специальность, ФИО, Разряд, Сотрудник, Проект (показаны прямоугольниками) и понятия-ассоциации Инженер, Рабочий, Руководитель, Исполнитель (показаны параллелограммами). С целью упрощения примера аспекты понятий на диаграмме не указаны. Понятиями, не определенными на диаграмме, являются Специальность, ФИО, Разряд и Проект. Образование остальных понятий показано подходящими к ним стрелками. Так понятие Сотрудник является обобщением понятий Инженер и Рабочий, а понятие Руководитель –

ассоциацией понятий Проект и Инженер.

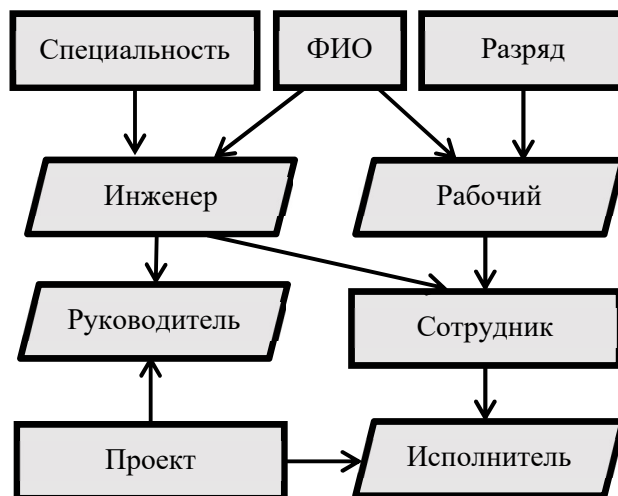


Рис. 1. Фрагмент понятийной структуры

Описание объема понятий осуществляется путем задания принадлежащих им перечислимых или разрешимых множеств сущностей. Перечислимые множества сущностей применяется для описания понятий небольшого объема, а для понятий большого объема – разрешимые множества.

Например, понятие-обобщение Разряд (рис. 1) является понятием-типом, обобщающим разряды рабочих – понятия-знаки 1, 2 и 3. В свою очередь понятие-обобщение ФИО является типизацией строк, состоящих из фамилий, имен и отчеств сотрудников. Аналогично описывается и понятие-тип Специальность путем перечисления инженерных специальностей проблемной области. Все эти понятия описываются перечислимыми множествами сущностей. Однако понятия ФИО и Специальность являются специализациями понятия Строка, множество сущностей которого велико и задается соответствующей разрешающей процедурой.

Формулы (1-2) косвенно задают объемы понятий. Так понятие-обобщение  $G$  из (1) состоит из всех сущностей, принадлежащих понятиям  $G1, G2, \dots, GK$ ,

$$\text{Ext } G = \text{Ext } G1 \cup \text{Ext } G2 \cup \dots \cup \text{Ext } GK, \quad (3)$$

в то время как понятие-ассоциация  $A$  из (2) состоит из сущностей, получаемых соединением сущностей понятий  $A1, A2, \dots, AL$  в сущность понятия  $A$ ,

$$\text{Ext } A \subseteq \text{Ext } A1 \times \text{Ext } A2 \times \dots \times \text{Ext } AL, \quad (4)$$

где  $\text{Ext } X$  – экстенционал понятия  $X$ ,  $\subseteq$  – отношение нестрогого включения множеств,  $\cup$  – операция объединения множеств, а  $\times$  – операция декартова произведения множеств.



Формула (3) описывает экстенционал понятия обобщения, а формула (4) задает условие, накладываемое на экстенционал понятия-ассоциации. Если в формуле (4) справедливо равенство левой и правой частей, то имеем частный случай ассоциации – агрегацию понятий.

Для определения принадлежности сущности  $E$  понятию-ассоциации  $A$  используются формулы вида

$$A\{E\} \leftrightarrow \varphi(E, A1, A2, \dots, AL), \quad (5)$$

где  $\varphi$  – некоторое логическое высказывание с вхождением свободных переменных  $E, A1, A2, \dots, AL$ , которое ограничивает вхождения сущностей ассоциируемых понятий  $A1, A2, \dots, AL$  в сущность понятия-ассоциации  $A$ ;  $\leftrightarrow$  – логическая связка «равносильно».

Формулы логических высказываний (5) конструируются по следующим правилам:

1) предикат принадлежности сущности  $E$  понятию  $N$  вида  $N\{E\}$  является формулой (равносильно  $E \in \text{Ext } N$ );

2) отношение между сущностями вида  $(X[i] \bullet E)$  является формулой, где  $X[i]$  – функтор, возвращающий сущность понятия с номером  $i$  ( $i > 0$ ) из состава сущности  $X$ ,  $\bullet$  – символ отношения, которое определено для сущностей  $X[i]$  и  $E$  (отношение «равно» – для любых сущностей; отношения «больше» и «меньше» – для упорядоченных сущностей);

3) выражение с логической связкой «не» вида  $(\neg \varphi)$  является формулой, если  $\varphi$  – формула;

4) выражение с логическими связками «и», «или», «следует» и «равносильно» вида  $(\varphi \wedge \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \rightarrow \psi)$  и  $(\varphi \leftrightarrow \psi)$  являются формулами, если  $\varphi$  и  $\psi$  – формулы;

5) выражения с квантором всеобщности и квантором существования вида  $(\forall X \varphi)$  и  $(\exists X \varphi)$  являются формулами, если  $\varphi$  – формула, а  $X$  – свободная переменная в формуле  $\varphi$ ;

6) скобки в формулах могут быть опущены при учете следующего порядка интерпретации отношений, логических связок и кванторов:  $\bullet, \neg, \wedge, \vee, \forall, \exists, \rightarrow, \leftrightarrow$ .

Примером описания экстенционала понятия-ассоциации Инженер из демонстрационной понятийной модели (см. рис. 1) может служить логическое высказывание

$$\begin{aligned} \text{Инженер}\{E\} \leftrightarrow & (E[\text{ФИО}] = \text{«Иванов Петр Сидорович»} \vee \\ & \vee E[\text{ФИО}] = \text{«Петров Иван Ильич»}) \wedge \\ & \wedge E[\text{Специальность}] = \text{«Технология машиностроения»}, \end{aligned}$$

утверждающее, что составная сущность  $E$  является Инженером тогда и только тогда, когда ее часть ФИО или «Иванов Петр Сидорович», или «Петров Иван Степанович», а часть Специальность – «Технология машиностроения».

Помимо формульного описания сущностей, принадлежащих понятиям, используется их табличное перечисление. В этом случае заголовок таблицы задает интенционал понятия (его содержание), а тело таблицы – экстенционал (его объем).

Например, для описанного выше понятия Инженер может использоваться следующая таблица (табл. 1).

Табл. 1. Табличная форма понятия Инженер

ФИО	Специальность
Иванов Петр Сидорович	Технология машиностроения
Петров Иван Ильич	Технология машиностроения

Однако для предотвращения дублирования объемных сущностей их вхождения в состав других сущностей заменяются на краткие имена. Для сопоставления сущности и ее имени в таблицах понятий предусматривается уникальное ключевое поле с именем понятия (табл. 2). Аналогичным образом создаются таблицы понятий ФИО и Специальность, в которых определяются сущности с именами ФИО-1, ФИО-2, Специальность-1 и т. п.

Табл. 2. Таблица понятия Инженер

Инженер	ФИО	Специальность
Инженер-1	ФИО-1	Специальность-1
Инженер-2	ФИО-2	Специальность-1

Универсальным отношением, заданным для всех сущностей и понятий, является их равенство. Две сущности равны тогда и только тогда, когда они соединяют в себе одни и те же сущности. Например, сущности Инженер-1 и Инженер-2 не равны, так как составляющие их сущности ФИО разные (см. табл. 2). Аналогично определяется равенство понятий: два понятия равны тогда и только тогда, когда они состоят из одних и тех же сущностей, т. е. когда их экстенционалы равны.

Логические формулы, используемые для выражения экстенционалов, являются формулами одноместного исчисления предикатов [18, теорема 5]. Известно, что одноместное исчисление предикатов (англ. pure monadic predicate calculus) имеет много полезных свойств, делающих его похожим на исчисление высказываний [19, с. 24]: непротиворечивость (англ. consistent), полнота на конечных моделях (англ. completeness), разрешимость формул (англ. decidability), низкая выразительность (англ. expressiveness), эффективная вычислимость (англ. computability).

Для логических моделей также были предприняты попытки ограничить выразительность логического языка путем использования дескрипционных логик. Дескрипционные логики строятся на одноместных предикатах принадлежности экземпляра концепту и двуместных

предикатах ролей (отношений между экземплярами). Для сведения дескрипционной логики к одноместному исчислению предикатов использование ролей ограничено конструкциями  $\forall R.C$  и  $\exists R.C$ , которые сводят применение двуместных предикатов к одноместным, связывая вторую переменную роли  $R$  экземплярами концепта  $C$  [7, с. 50, 150]. Это позволило добиться разрешимости формул и относительно невысокой вычислительной сложности решения основных задач представления и обработки знаний в дескрипционных логиках.

В понятийных моделях удалось еще более понизить выразительность логического языка благодаря отказу от использования двуместных предметных предикатов (ролей) и получить большую эффективность обработки знаний по сравнению с дескрипционными моделями.

## 6. Вывод на знаниях

Использование объективированных знаний подразумевает наличие форм представления знаний и методов манипулирования ими с целью имитации рассуждений человека. Обычно для представления знаний используются факты (суждения), а для обработки знаний – правила вывода, позволяющие на основе имеющихся фактов делать умозаключения и получать новые утверждения (факты).

В понятийной модели понятия-обобщения моделируют понятийный состав знаний предметной области, а интенционалы понятий-обобщений задают правила вывода, согласно которым сущность  $E$  принадлежит понятию-обобщению  $G$  тогда и только тогда, когда она принадлежит хотя бы одному обобщенному понятию  $G_1, G_2, \dots, G_K$ :

$$G\{E\} \leftrightarrow G_1\{E\} \vee G_2\{E\} \vee \dots \vee G_K\{E\}. \quad (6)$$

В свою очередь понятия-ассоциации формализуют суждения относительно понятий, а интенционалы и экстенционалы ассоциированных понятий задают правила вывода, согласно которым сущность  $E$  принадлежит понятию-ассоциации  $A$ , тогда и только тогда, когда она соединяет в себе сущности  $E[1], E[2], \dots, E[L]$  ассоциированных понятий  $A_1, A_2, \dots, A_L$ , а множество этих сущностей удовлетворяет условиям, заданным логическим выражением  $\varphi$ :

$$A\{E\} \leftrightarrow A_1\{E[1]\} \wedge A_2\{E[2]\} \wedge \dots \wedge A_L\{E[L]\} \wedge \varphi(E, A_1, A_2, \dots, A_L). \quad (7)$$

Первая часть условий формулы (7) задает принадлежность сущностей  $E[1], E[2], \dots, E[L]$  из состава проверяемой сущности  $E$  соответствующим ассоциированным понятиям  $A_1, A_2, \dots, A_L$  понятия-ассоциации  $A$ , а вторая часть условий ограничивает произвольные вхождения сущностей ассоциируемых понятий  $A_1, A_2, \dots, A_L$  в сущность понятия-ассоциации  $A$ .

Фрагмент понятийной структуры, диаграмма которой показана на рис. 1, порождает следующие правила вывода, но не все:

$$\text{Сотрудник}\{E\} \leftrightarrow \text{Инженер}\{E\} \vee \text{Рабочий}\{E\};$$

$$\text{Руководитель}\{E\} \leftrightarrow \text{Инженер}\{E[1]\} \wedge \text{Проект}\{E[2]\} \wedge \varphi(E, \text{Инженер}, \text{Проект}),$$

где  $\varphi$  – некоторая формула, которая ограничивает вхождение пар сущностей вида Инженер  $X$  понятия Инженер и Проект  $Y$  понятия Проект в экстенционал понятия Руководитель, например так:

$$\begin{aligned} E[1] = \text{Инженер-1} \wedge (E[2] = \text{Проект-1} \vee E[2] = \text{Проект-3}) \vee \\ \vee E[1] = \text{Инженер-2} \wedge E[2] = \text{Проект-2}. \end{aligned}$$

При выводе на знаниях требуется установить или опровергнуть некоторый факт путем поиска сущностей, удовлетворяющих заданным условиям, для чего используются запросы к понятной модели. В этом случае правила вывода вида (6) задают обобщенные понятия, подлежащие обработке, а правила вывода вида (7) позволяют установить составляющие сущности обрабатываемого понятия, а также проверить принадлежность обрабатываемому понятию произвольной сущности.

Используя имена сущностей и их переменные формулируются запросы к понятийной модели, например, такой: «Кто из исполнителей проекта А является рабочим без третьего разряда?». На языке одноместного исчисления предикатов этот запрос будет иметь следующее выражение:

$$\begin{aligned} \text{Исполнитель}\{X\} \wedge X[\text{Проект}] = \text{Проект-1} \wedge \\ \wedge \forall Y (\text{Рабочий}\{Y\} \wedge X[\text{Сотрудник}] = Y \wedge \neg(Y[\text{Разряд}] = 3)) \end{aligned}$$

в котором  $X$  является свободной переменной, а  $Y$  – связанной. Для наглядности в выражении номера понятий составляющих сущностей заменены на их имена:  $X[\text{Проект}]$ ,  $X[\text{Сотрудник}]$ ,  $Y[\text{Разряд}]$ . Запрос имеет глубину 2 так как для каждой сущности  $X$  обрабатывается вложенный подзапрос с переменной  $Y$ .

Время выполнения простых запросов к понятийной модели имеет линейную или логарифмическую оценку, т. к. поиск сущности в таблице понятия с числом записей  $N$  осуществляется за время с асимптотической оценкой  $N$  [20, с. 667], а в случае использования индекса – то за время с асимптотической оценкой  $\log N$  [20, с. 601]. Если используются сложные вложенные запросы с глубиной  $D$  и средним числом записей в просматриваемых таблицах  $N$ , то выполнение запроса к понятийной модели осуществляется за время с асимптотической оценкой  $N^D$  или  $\log^D N$  в зависимости от отсутствия или наличия индексов у таблиц понятий.

## 7. Процедурные знания

До сих пор рассматривалось представление декларативных знаний, которые задавались

различными способами образования понятий и перечислением сущностей в их экстенционалах. Однако помимо декларативных знаний на практике используются знания другого вида – процедурные знания.

В искусственном интеллекте процедурные знания – это тип знаний, которыми может обладать интеллектуальный агент, который для достижения своих целей воспринимает свое окружение и действует разумно, а в процессе своей деятельности приобретает новые знания [15, с. 66]. Такие знания представляются в виде конечного автомата или компьютерной программы, которые реализуют функцию агента, отображая восприятия агентом окружающей среды на действия, воздействующие на эту среду и изменяющие ее восприятие.

В теоретическом аспекте процедурные знания в понятийной модели выражены на языке одноместного исчисления предикатов и используются для задания разрешимых множеств сущностей. В прикладном аспекте процедурные знания представим как именованные последовательности действий (процедуры), позволяющие из некоторых исходных сущностей и понятий получить результирующие.

Процедуры, как и все в понятийных моделях, представим сущностями соответствующих понятий. Элементарные действия, из которых строится процедура – это операции создания, изменения и удаления сущностей понятий. Причем сами понятия будем рассматривать как сущности некоторого понятия понятий, что позволяет создавать, изменять и удалять сами понятия [21].

Для повышения выразительности языка представления и обработки знаний предусмотрим оператор присваивания, а также условный оператор и операторы циклов. Условия у операторов задаются логическими выражениями, истинность которых проверяется при выполнении оператора.

Особенностью пространства имен описываемого языка представления и обработки знаний является интерпретация всех переменных как понятий. Любая переменная – это одномерный массив сущностей. Если в массиве один элемент, то такая переменная является сущностью, но одновременно и понятием с одной принадлежащей ему сущностью.

Для кодирования процедурных знаний введем процедурный тип данных, сущности которого – это последовательности команд некоторой виртуальной машины, выполняющей операции над сущностями понятий. Синтаксически процедурные знания описываются как выражения вида

$$(X, Y, \dots, Z) \Rightarrow W, \quad (8)$$

где  $X, Y, \dots, Z$  – формальные понятия-аргументы,  $W$  – текст процедуры на языке представления и обработки знаний с вхождением в него формальных понятий-аргументов. Результатом

выполнения процедуры (8) является возврат некоторого результирующего понятия, т. е. любую процедуру можно запустить на выполнение, передав ей исходные понятия, и получить результат, в том числе и процедурный. Побочным эффектом выполнения процедуры может быть изменение используемой понятийной модели.

Встроенными операциями языка представления и обработки знаний являются операции над простыми типами данных, а именно: числами ограниченной разрядности, датами в различных форматах, строками конечной длины, двоичными данными, процедурами и т. п.

В качестве примера рассмотрим процедуру, которая повышает разряды рабочим, участвующим в Проекте-1 (программа 1).

Программа 1. Определение и вызов анонимной процедуры с аргументом **Имя Проект**

```
(Проект) => {  
    // Определение внутренней функции повышения разряда  
    Повышение = (Разряд) => Разряд < 3 ? 1 : 0;  
    // Создание индексной переменной и ее инициализация  
    Индекс = 0;  
    // Открытие транзакции с именем Разряды  
    begin Разряды;  
    // Запрос сущностей X временного понятия Рабочие  
    Рабочие = {X: Исполнитель} (X[Проект] == Проект && Рабочий {X[Сотрудник]});  
    // Цикл по сущностям временного понятия Рабочие  
    while (Индекс < Рабочие {})  
    {  
        // Создание локальной сущности Рабочий  
        Рабочий = Рабочие {Индекс};  
        // Увеличение атрибута Разряд сущности Рабочий  
        Рабочий[Разряд] += Повышение(Рабочий[Разряд]);  
        // Приращение индекса сущности понятия Рабочие  
        Индекс = Индекс + 1;  
    }  
    // Обновление измененных сущностей в понятийной модели  
    update Рабочие;  
    // Успешное завершение транзакции  
    commit Разряды;  
}(Проект-1);
```

В тексте программы конструкция вида  $Y\{X\}$  является функтором извлечения сущности с именем (номером)  $X$  из экстенционала понятия  $Y$ . В логическом контексте  $Y\{X\}$  – предикат принадлежности сущности  $X$  понятию  $Y$ . Если в  $Y\{X\}$  аргумент опущен, то  $Y\{\}$  – число сущностей в экстенционале понятия  $Y$ .

Следует обратить внимание на то, что операторы присваивания создают только временные переменные. Однако связь с понятийной моделью сохраняется для всех извлеченных из нее сущностей, что позволяет вносить изменения в понятийную модель оператором обновления *update*. В языке также присутствуют операторы для создания и удаления сущностей понятий.

## 8. Обучение понятийных моделей

Обучение понятийных моделей с учителем представляет собой процесс, при котором пользователь (учитель) определяет понятия и перечисляет принадлежащие им сущности, т. е. задает интенционалы и экстенционалы понятий. Аналогичные подходы используются для наполнения предметным содержанием других моделей знаний.

Однако понятийные модели могут эффективно обучаться и без учителя. Обучение понятийных моделей без учителя осуществляется путем «восприятия» понятийной моделью «окружающего мира» в виде поступающего на вход потока сущностей и решения задачи распознавания понятий этих сущностей с учетом знаний, хранящихся в понятийной модели.

Для воспроизведения имеющихся знаний построим формальную грамматику  $F = [N, M, T, P]$ , порождающую сущности заданной понятийной модели, где  $N$  – конечное множество основных нетерминальных символов, совпадающих с именами понятий;  $M$  – конечное множество вспомогательных нетерминальных символов, являющихся именами сущностей;  $T$  – конечное множество терминальных символов, используемых для выражения сущностей;  $P$  – конечное множество продукций вида  $S \Rightarrow \omega$ ;  $S$  – нетерминальный символ грамматики,  $S \in N \cup M$ ;  $\omega$  – строка конечной длины, состоящая из терминальных и нетерминальных символов;  $\Rightarrow$  – знак операции грамматического вывода, разделяющий левую и правую часть продукций.

Правила вывода (5), задающие принадлежность сущности  $E$  понятию-ассоциации  $A$ , порождают продукции вида

$$A \Rightarrow M1 \circ M2 \circ \dots \circ ML, \quad (9)$$

такие, что  $\varphi(E, A1, A2, \dots, AL)$ , где  $A$  – нетерминальный символ грамматики;  $M1, M2, \dots, ML$  – вспомогательные нетерминальные символы, имена сущностей ассоциированных понятий  $A1, A2, \dots, AL$  из состава сущности  $E$ ;  $\circ$  – символ конкатенации, разделяющий имена сущностей.

Продукции (9) порождают множество строк, которыми выражаются сущности  $E$  понятия-

ассоциации  $A$ . Вспомогательные нетерминальные символы  $M$  используются в продукциях именованя сущностей,

$$E \Rightarrow \mu, \quad (10)$$

где  $E$  – имя сущности,  $E \in M$ , а  $\mu$  – ее выражение в объединенном алфавите терминальных и нетерминальных символов грамматики.

Приведем пример формальной грамматики для демонстрационной понятийной модели (см. рис. 1). Принадлежность сущностей понятийной модели понятиям-ассоциациям задаются следующими продукциями вида (9):

ФИО  $\Rightarrow$  ФИО-1 | ФИО-2 | ФИО-3 | ФИО-4;

Специальность  $\Rightarrow$  Специальность-1 | Специальность-2;

Разряд  $\Rightarrow$  Разряд-1 | Разряд-2 | Разряд-3;

Проект  $\Rightarrow$  Проект-1 | Проект-2;

Рабочий  $\Rightarrow$  ФИО-3  $\circ$  Разряд-3 | ФИО-4  $\circ$  Разряд-1;

Инженер  $\Rightarrow$  ФИО-1  $\circ$  Специальность-1 | ФИО-2  $\circ$  Специальность-2;

Исполнитель  $\Rightarrow$  Рабочий-1  $\circ$  Проект-1 | Рабочий-2  $\circ$  Проект-2;

Руководитель  $\Rightarrow$  Инженер-2  $\circ$  Проект-1 | Инженер-1  $\circ$  Проект-2,

где | – символ, разделяющий альтернативные варианты в правой части продукции, а вспомогательные нетерминальные символы (10) выражаются так:

ФИО-1  $\Rightarrow$  «Иванов Петр Сидорович»;

ФИО-2  $\Rightarrow$  «Петров Иван Ильич»;

ФИО-3  $\Rightarrow$  «Сидоров Илья Иванович»;

ФИО-4  $\Rightarrow$  «Ильин Сидор Петрович»;

Специальность-1  $\Rightarrow$  «Технология машиностроения»;

Специальность-2  $\Rightarrow$  «Литье под давление»;

Проект-1  $\Rightarrow$  «Проект А»;

Проект-2  $\Rightarrow$  «Проект Б»;

Разряд-1  $\Rightarrow$  1;

Разряд-2  $\Rightarrow$  2;

Разряд-3  $\Rightarrow$  3.

Формальная грамматика с продукциями вида (9) и (10) является контекстно-свободной. Для контекстно-свободных грамматик разрешимой является задача определения принадлежности произвольной строки любому нетерминальному символу грамматики [22, с. 352].



Отсюда следует, что все сущности понятийной модели могут быть распознаны и определены их понятия. Если понятие некоторой сущности из входного потока не распознано, то при обучении без учителя решается задача пополнения экстенционалов соответствующих понятий новыми сущностями, при котором входная сущность распознается. Если путем пополнения экстенционалов сущность распознать не удалось, то создается новое понятие, в экстенционал которого включается нераспознанная сущность.

Новые сущности понятийной модели должны выводиться из продукций расширенной грамматики  $F$ , следующих из правил вывода (6) и (7):

$$G \Rightarrow G1 | G2 | \dots | GK, \quad (11)$$

$$A \Rightarrow A1 \circ A2 \circ \dots \circ AL, \quad (12)$$

где  $G$  – нетерминальный символ понятия-обобщения;  $G1, G2, \dots, GK$  – нетерминальные символы обобщенных понятий;  $A$  – нетерминальный символ понятия-ассоциации;  $A1, A2, \dots, AL$  – нетерминальные символы ассоциированных понятий.

Продукции (11) порождают множество строк  $G$ , которыми выражаются сущности понятия-обобщения, если известны множества строк  $G1, G2, \dots, GK$ , которыми выражаются сущности соответствующих обобщенных понятий. В свою очередь продукции (12) порождают множества строк  $A$ , которыми могут быть выражены сущности понятия-ассоциации, если известны множества строк  $A1, A2, \dots, AL$ , которыми выражаются сущности соответствующих ассоциированных понятий.

Рассмотрим пример обучения понятийной модели без учителя при поступлении следующих обучающих данных: «Сидоров Илья Иванович» 3 «Проект А». На рис. 2 приведено дерево грамматического разбора входной строки, из которого следует, что сущность Рабочий-3  $\circ$  Проект-1 не распознана (обведена пунктиром).

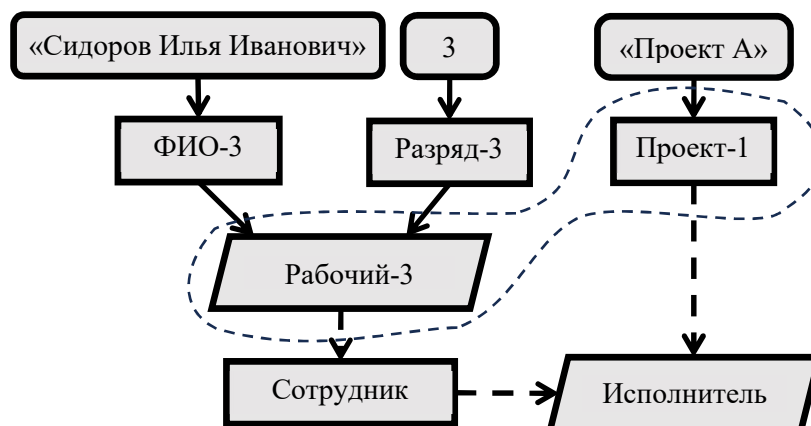


Рис. 2. Дерево грамматического разбора

Далее для распознавания новых сущностей грамматический разбор продолжается в грамматике, расширенной следующими продукциями:

Инженер  $\Rightarrow$  Специальность  $\circ$  ФИО;

Рабочий  $\Rightarrow$  ФИО  $\circ$  Разряд;

Руководитель  $\Rightarrow$  Инженер  $\circ$  Проект;

Сотрудник  $\Rightarrow$  Рабочий | Инженер;

Исполнитель  $\Rightarrow$  Сотрудник  $\circ$  Проект.

Из последних двух продукций выводится нераспознанная сущность Рабочий-3  $\circ$  Проект-1, которая включается в экстенционал понятия Исполнитель (пунктирные стрелки на рис. 2), а для распознавания добавленной сущности в грамматику добавляется продукция Исполнитель  $\Rightarrow$  Рабочий-3  $\circ$  Проект-1.

При неоднозначных грамматиках, выражающих сущности с альтернативными интерпретациями, могут встречаться несколько вариантов грамматического разбора обучающих данных. В этом случае каждый из этих вариантов проверяется и выбирается тот, который вносит, например, наименьшие изменения в понятийную модель.

## Заключение

Модель знаний названа понятийной, чтобы отличить ее от концептуальных моделей. Понятие отличается от концепта. Концепт объективен, а понятие субъективно. По этой причине существует множество формальных понятий, имеющих одно и то же имя, но разный объем и содержание в различных проблемных областях (аспектах). В этом случае концепт может быть определен как результат объективации одноименных понятий в различных аспектах.

С помощью операций идентификации, ассоциации и обобщения могут быть конструктивно построены все мыслимые понятия, а также задана принадлежность построенным понятиям перечислимых или разрешимых множеств сущностей. В понятийной модели экстенционалы понятий описываются на языке одноместного исчисления предикатов, что делает понятийные модели разрешимыми и непротиворечивыми на счетных предметных областях, и полными – на конечных [18]. Благодаря понижению выразительности понятийного языка уменьшена вычислительная сложность задач представления и обработки знаний по сравнению другими моделями знаний.

Трудности представления и обработки знаний в логических, продукционных, сетевых и фреймовых моделях не позволяют эффективно реализовать большие базы знаний. Однако при использовании понятийной модели знаний удалось получить некоторые преимущества:

- низкая вычислительная сложность вывода на знаниях, нет необходимости в использовании эвристик;

- запросы к понятийной модели выполняются за полиномиальное время от логарифма среднего числа сущностей в экстенционалах понятий, а результаты запроса имеют строгое логическое обоснование и содержательное объяснение;

- возможность выполнять сужение области знания путем использования понятий в различных аспектах;

- использование процедурных знаний, формализуемых на развитом процедурном языке с операциями создания, изменения, сравнения и удаления сущностей понятий, где сами понятия также являются сущностями некоторого понятия – понятия понятий;

- обучение понятийных моделей без учителя путем грамматического разбора потока входных понятий-знаков, выделяя в нем неизвестные сущности для пополнения экстенционалов существующих понятий новыми сущностями или для образования новых понятий;

- наглядность понятийной модели из-за непосредственного воспроизведения ею представлений, понятий и суждений предметной области знания.

Высокая эффективность и выразительность понятийных моделей, их легкая актуализация, не приводящая к возникновению скрытых противоречий, позволяет надеяться на широкое внедрение понятийного представления знаний в практику создания и использования современных интеллектуальных информационных систем.

Следует также заметить, что иных более общих выразительных средств для представления знаний, кроме как их описание в виде понятийной модели, не существует [23]. Это объясняется тем, что понятийное мышление является предельно общим и единственным механизмом рационального познания, а понятийный анализ – субъективной и, в конечном счете, объективной

формой фиксации результатов понятийного мышления.

Однако обладание знаниями и умение мыслить (осуществлять вывод на знаниях) не является достаточным для сильного искусственного интеллекта (англ. artificial general intelligence). Имеется много задач, которые не решаются рациональными методами, например, синтез и анализ алгоритмов и программ, доказательство теорем, комбинаторные задачи большой размерности, глобальная оптимизация и т.п. Для реализации сильного искусственного интеллекта необходимо нечто большее, например, использование не только рациональных, но и иррациональных формы познания, а также эмоций.

## Литература

1. Handbook of Knowledge Representation / Eds. F. Harmelen, V. Lifschitz, B. Porter. Elsevier, 2008. 1031 p.
2. Chein M., Mugnier M.-L. Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. Springer, 2009. 425 p.
3. Gough B., Madill A. Subjectivity in Psychological Science: from Problem to Prospect // Psychological Methods. 2012. Vol. 7(3). P. 374-384. DOI 10.1037/a0029313.
4. Polya G. Mathematics and Plausible Reasoning. Vol. II: Patterns of Plausible Inference. Princeton, New Jersey: Princeton University Press, 1954. 464 p.
5. Walton D. Abductive Reasoning. Tuscaloosa: The University of Alabama Press, 2005. 320 p.
6. Brachman R. J., Levesque Y.J. Knowledge Representation and Reasoning. Elsevier, 2014. 413 p.
7. The description Logic Handbook: Theory, Implementation and Applications / Eds. F. Baader, D. Calvanese, D. L. McGuinness, et al. New York: Cambridge University Press, 2003. 573 p.
8. Klahr D., Langley P., Neches R. Production System Models of Learning and Development. Cambridge: The MIT Press, 1987. 466 p.
9. Sowa J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Thomson Learning, 2000. 610 p.
10. Object-oriented Design Knowledge: Principles, Heuristics, and Best Practices / Eds. J. Garzas, M. Piattini. Idea Group Publishing, 2007. 376 p.
11. Gardarin G., Valduriez P. Relational databases and knowledge bases. Addison-Wesley, 1989. 448 p.
12. Vykhoivanets V.S. The Notional Model of Knowledge Representation // 13th Multiconference on Control Problems (MCCP). Journal of Physics: Conference Series. 2021. 1864 012058. DOI: 10.1088/1742-6596/1864/1/012058.
13. The Oxford Dictionary of Current English / Ed. D. Thompson. New York, NY: Oxford University Press, 1999. 996 p.
14. Olive A. Conceptual Modeling of Information Systems. New York, NY: Springer, 2007. 471 p.
15. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. Hoboken: Pearson, 2022. 1167 p.
16. Выхованец В. С. О существенной неполноте формального метода // Философия, математика, лингвистика: аспекты взаимодействия: Материалы Международной научной конференции. СПб: Международный математический институт им. Л. Эйлера, 2009. С. 234-240.
17. Vykhoivanets V.S. Intelligent Information Systems based on Notional Models without Relationships // Z. Hu et al. (Eds.): CSDEIS 2020. Advances in Intelligent Systems, Computer Science and Digital Economics II, AISC 1402, p. 20–28, 2021. DOI 10.1007/978-3-030-80478-7.
18. Выхованец В. С. Понятийный анализ и понятийное моделирование // Управление большими системами. Вып. 92. М.: ИПУ РАН, 2021. С. 64-109. DOI 10.25728/ubs.2021.92.4.
19. Mendelson E. Introduction to Mathematical Logic. New York, NY: CRC Press, 2010. 496 p.
20. Elmasri R., Navathe S. B. Fundamentals of Database Systems. Hoboken, NJ: Pearson, 2016. 413 p.
21. Vykhoivanets V.S. The Knowledge Description Language // 14th International Conference Management of large-scale system development (MLSD). Moscow, 2021. P. 1-5. DOI: 10.1109/MLSD52249.2021.9600103.
22. Reghizzi S. C., Breveglieri L., Morzenti A. Formal Languages and Compilation. Springer, 2019. 499 p. DOI: 10.1007/978-3-030-04879-2.
23. Выхованец В. С. Методика понятийного анализа сложных предметных областей // Наукоемкие

## Об авторе

**Выхованец Валерий Святославович.** Доктор технических наук, доцент. Профессор Московского государственного технического университета им. Н. Э. Баумана. Области исследований: логические вычисления, теория и технология программирования, представление и обработка знаний. E-mail: [valery@vykhovanets.ru](mailto:valery@vykhovanets.ru).

## The Notional Model of Knowledge

V. S. Vykhoanets  
Bauman Moscow State Technical University  
[valery@vykhovanets.ru](mailto:valery@vykhovanets.ru)

**Abstract.** The article discusses well-known methods of knowledge representation and processing. A new model, called notional, is proposed, characterized in that the relations (links) between notions are considered as ordinary notions. The notion is considered as a form of thought expressed by a named set of entities. A concrete notion identifies one entity. An abstract notion is formed from other notions by generalization (union) and association (Cartesian product) them. Declarative knowledge is given by enumerable sets of entities, and procedural knowledge is given by solvable ones, where resolving procedures are expressed by formulas of the pure monadic predicate calculus. The description of the applied language of knowledge representation and processing is given. It is proved that queries to the notional model are executed in polynomial time from the logarithm of the average number of entities in notions. The possibility of unsupervised learning notional models is substantiated. It is shown that the notional model makes it possible to visually represent and effectively process both declarative and procedural knowledge.

**Keywords:** knowledge representation, knowledge processing, knowledge models, notional model, declarative knowledge, procedural knowledge, unsupervised learning, intelligent information system.

## References

1. Handbook of Knowledge Representation / Eds. F. Harmelen, V. Lifschitz, B. Porter. Elsevier, 2008. 1031 p.
2. Chein M., Mugnier M.-L. Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. Springer, 2009. 425 p.
3. Gough B., Madill A. Subjectivity in Psychological Science: from Problem to Prospect // Psychological Methods. 2012. Vol. 7(3). P. 374-384. DOI 10.1037/a0029313.
4. Polya G. Mathematics and Plausible Reasoning. Vol. II: Patterns of Plausible Inference. Princeton, New Jersey: Princeton University Press, 1954. 464 p.
5. Walton D. Abductive Reasoning. Tuscaloosa: The University of Alabama Press, 2005. 320 p.
6. Brachman R. J., Levesque Y.J. Knowledge Representation and Reasoning. Elsevier, 2014. 413 p.
7. The description Logic Handbook: Theory, Implementation and Applications / Eds. F. Baader, D. Calvanese, D. L. McGuinness, et al. New York: Cambridge University Press, 2003. 573 p.
8. Klahr D., Langley P., Neches R. Production System Models of Learning and Development. Cambridge: The MIT Press, 1987. 466 p.
9. Sowa J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Thomson Learning, 2000. 610 p.
10. Object-oriented Design Knowledge: Principles, Heuristics, and Best Practices / Eds. J. Garzas, M. Piattini. Idea Group Publishing, 2007. 376 p.
11. Gardarin G., Valduriez P. Relational databases and knowledge bases. Addison-Wesley, 1989. 448 p.
12. Vykhoanets V.S. The Notional Model of Knowledge Representation // 13th Multiconference on Control Problems (MCCP). Journal of Physics: Conference Series. 2021. 1864 012058. DOI: 10.1088/1742-6596/1864/1/012058.
13. The Oxford Dictionary of Current English / Ed. D. Thompson. New York, NY: Oxford University Press, 1999.

996 p.

14. Olive A. Conceptual Modeling of Information Systems. New York, NY: Springer, 2007. 471 p.
15. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. Hoboken: Pearson, 2022. 1167 p.
16. Vykhovanets V. S. O sushchestvennoj nepolnote formal'nogo metoda [On the essential incompleteness of the formal method] // *Filosofiya, matematika, lingvistika: aspekty vzaimodejstviya: Materialy Mezhdunarodnoj nauchnoj konferencii [International Interdisciplinary Conference «Philosophy, Mathematics, Linguistics: Aspects of Interaction»]*. SPb: Mezhdunarodnyj matematicheskij institut im. L. Ejlera [St. Petersburg: Euler International Mathematical Institute], 2009. P. 234-240.
17. Vykhovanets V.S. Intelligent Information Systems based on Notional Models without Relationships // Z. Hu et al. (Eds.): CSDEIS 2020. Advances in Intelligent Systems, Computer Science and Digital Economics II, AISC 1402, p. 20–28, 2021. DOI 10.1007/978-3-030-80478-7.
18. Vykhovanets V.S. Ponyatijnyj analiz i ponyatijnoe modelirovanie [The notional analysis and notional modelling] // *Upravlenie bol'shimi sistemami [Large-Scale Systems Control]*. Issue 92. Moscow: IPU RAN, 2021. P. 64-109. DOI 10.25728/ubs.2021.92.4.
19. Mendelson E. Introduction to Mathematical Logic. New York, NY: CRC Press, 2010. 496 p.
20. Elmasri R., Navathe S. B. Fundamentals of Database Systems. Hoboken, NJ: Pearson, 2016. 413 p.
21. Vykhovanets V.S. The Knowledge Description Language // 14th International Conference Management of large-scale system development (MLSD). Moscow, 2021. P. 1-5. DOI: 10.1109/MLSD52249.2021.9600103.
22. Reghizzi S. C., Breviglieri L., Morzenti A. Formal Languages and Compilation. Springer, 2019. 499 p. DOI: 10.1007/978-3-030-04879-2.
23. Vykhovanets V.S. Metodika ponyatijnogo analiza slozhnyh predmetnyh oblastej [The methodology of notional analysis of complex subject domains] // *Naukoemkie tekhnologii [Journal Science Intensive Technologies]*. 2022. Vol. 23. № 8. P. 60–68. DOI: 10.18127/j19998465-202208-07.

### **About author**

**Vykhovanets Valeriy S.** Doctor of technical sciences, docent. Professor, Bauman Moscow State Technical University. Research areas: logical computing, theory and technology of programming, knowledge representation and processing. E-mail: [valery@vykhovanets.ru](mailto:valery@vykhovanets.ru).