

# ЯЗЫК ОПИСАНИЯ ЗНАНИЙ

**Выхованец В.С.**

*Институт проблем управления им. В.А. Трапезникова РАН, Россия, г. Москва  
ул. Профсоюзная д.65  
valery@vykhovanets.ru*

*Аннотация. Рассматривается база знаний с понятийной моделью предметной области. Понятийная модель состоит из понятийной структуры и содержания входящих в нее понятий. Понятийная структура описывает каждое понятие как результат обобщения или ассоциации других понятий. Содержание понятий задается перечислимыми и разрешимыми множествами сущностей. Понятийная структура содержит правила вывода на знаниях и рассматривается как формальная теория предметной области, сохраняющая истинность всех выводимых в ней следствий. Декларативные знания представляются понятиями с перечислением принадлежащих им сущностей. Процедурные знания задаются на языке описания знаний, который используется для понятий с разрешимыми множествами сущностей, а также для выражения фактов и формирования запросов к базе знаний. На языке описания знаний фактами являются истинными высказываниями с логическим связками, круглыми скобками и элементарными выражениями: одноместными предикатами принадлежности сущности понятию, функторами извлечения сущностей из составных сущностей, а также отношениями между сущностями. Запросы выражаются в виде факта со свободным вхождением сущностной переменной, а выполнение запроса создает понятие с сущностями, для которых факт выполнен. Для изменения состояния базы знаний в языке предусмотрены операции создания, удаления и изменений сущностей, а также условные операторы и операторы циклов. Использование языка описания знаний позволяет повысить выразительность представления и эффективность обработки больших объемов знаний.*

Ключевые слова: абстракции понятий, понятийная модель, понятийная структура, представление знаний, база знаний, декларативные знания, процедурные знания, язык описания и манипулирования знаниями, вывод на знаниях, запросы к базе знаний.

## **Введение**

Нерешенные проблемы в области представления и обработки знаний создают значительные трудности при развертывании баз знаний и интеллектуальных информационных систем на их основе. В частности, задача извлечения знаний до сих пор не полностью решена [1], формы представления знаний слабо адаптированы к психическим и психологическим особенностям человека [2], обработка знаний длительна по времени и порождает скрытые противоречия [3].

Основная причина проблем заключается в том, что при моделировании человеческого мышления используются традиции, заимствованные из логики и философии, где не учитываются такие свойства человеческого мышления, как объективность (открытость к новому, замена устаревшего знания), субъективность (перестройка мышления при решении прикладных задач), рефлексивность (саморегуляция познавательной деятельности), продуктивность (постоянная генерация нового знания) [4].

Некоторые из перечисленных выше проблем удалось избежать с помощью понятийного моделирования. Модели называются понятийными, чтобы отличать их от концептуальных моделей. В концептуальных моделях задаются концепты и различные взаимосвязи между ними. В понятийных моделях взаимосвязи между понятиями являются полноценными понятиями и используются для образования других понятий. В концептуальных моделях взаимосвязи между концептами выражаются предикатами, которые несут основную семантическую нагрузку модели и не могут использоваться для образования других концептов. В понятийных моделях определяется множество понятий, на котором заданы способы их образования путем обобщения и ассоциации [1].

Основным теоретическим языком концептуальных моделей является язык исчисления предикатов первого порядка, который настолько выразителен, что порождает неразрешимые проблемы при описании предметных областей. Примерами таких проблем является неразрешимость самого исчисления, неполнота элементарной арифметики, противоречивость некоторых теорий множеств.

По этой причине для описания концептуальных моделей были предприняты попытки ограничить выразительность логического языка путем использования дескрипционных логик, которые оказались разрешимыми фрагментами логики предикатов первого порядка с относительно невысокой вычислительной сложностью задач представления и обработки знаний. Дескрипционные логики строятся на одноместных предикатах принадлежности экземпляра концепту и двуместных предикатах ролей (отношений между экземплярами).

Для понятийных моделей удалось еще более понизить выразительность логического языка до одноместного исчисления предикатов первого порядка с отношениями равенства и порядка на множестве натуральных чисел. Исчисление оказалось разрешимым и непротиворечивым на счетных и полным на конечных понятийных моделях.

Настоящая работа посвящена языку описания знаний, используемому при понятийном анализе и моделировании предметных областей. Из-за отсутствия в языке двуместных предикатов с предметной интерпретацией удалось улучшить выразительность представления знаний и повысить эффективность их обработки.

## 1 Понятия и концепты

Понятие – это вид мысли, которая соотносится с определенным набором уникальных представлений (сущностей) внутреннего или внешнего мира человека (предметной области). Понятие (англ. notion) – это простая идея, мнение, представление или понимание чего-либо; понятийный (англ. notional) – это гипотетический, воображаемый [5, с. 607]. Понятие соотносится с представлениями из предметной области – сущностями (англ. entities).

В свою очередь концепт (англ. concept) – это общее понятие, абстрактная идея [5, с. 171]. Концепт состоит из экземпляров (англ. instances), каждый экземпляр характеризуется совокупностью свойств (англ. properties), свойства могут быть определенными (англ. defining) и неопределенными (англ. nondefining) [3, с. 38]. Таким образом, в отличие от концепта, понятие не объективно, а субъективно по определению.

Понятия формируются (определяются) в процессе абстрагирования путем выполнения ментальных операций над сущностями. Существует три вида понятий и порождающих их ментальных абстракций (отвлечений): понятия-знаки (абстракция идентификации или отождествления), понятия-обобщения (абстракция обобщения) и понятия-ассоциации (абстракция ассоциации).

### 1.1 Идентификация

Понятия-знаки – это результат мысленного отбора уникальных представлений из предметной области и присвоения им имен. Понятия-знаки формируются для фиксации определенного состояния органов чувств или элементарных представлений. При формировании понятия-знака сущность мысленно заменяется знаком (именем), в результате чего между сущностями и знаками устанавливается взаимно однозначное соответствие. Противоположной абстракцией для идентификации является абстракция интерпретация, при которой понятие-знак сопоставляется с некоторой уникальной сущностью из предметной области.

**Пример 1.** Понятиями-знаками могут быть такие знаки (слова) как Красный, Мало, Последний, Часто, Внизу и т.п. В свою очередь знак Красный интерпретируется как красный цвет, и т.д. ♦

### 1.2 Обобщение

Понятие-обобщение образуется при объединении сущностей обобщаемых понятий (объединение множеств сущностей). Абстракция обобщения используется также для образования частного вида понятия-обобщения – понятия-типа, представляющего собой обобщение (типизацию) понятий-знаков. Обобщение (типизация) имеет обратную абстракцию, называемую специализацией (конкретизацией).

**Пример 2.** Понятием-обобщением является понятие Утка, которое объединяет утки различных видов: Кряква, Крохаль, Нырок, Гоголь, Морянка, Савка и т.д. В свою очередь понятие Кряква является специализацией понятия-обобщения Утка. ♦

Понятия-обобщения объединяют понятия, схожие в каком-либо смысле, а то время как понятия-типы используются для объединения однородных сущностей.

**Пример 3.** Понятие-тип Вкус может быть образовано путем типизации таких понятий-знаков как Солёный, Кислый, Сладкий, Горький, Терпкий, Металлический, Меловой, Жгучий и т.д., а понятие-знак Солёный является конкретизацией понятия-типа Вкус. ♦

### 1.3 Ассоциация

Понятие-ассоциация образуется при соединении (укрупнении) сущностей ассоциируемых понятий, когда каждая сущность понятия-ассоциации включает в себя по одной сущности ассоциируемых понятий (подмножество декартова произведения множеств сущностей). Не все комбинации сущностей ассоциированных понятий могут быть сущностями понятия-ассоциации. Частным случаем ассоциации является абстракция агрегации, при которой образуемому понятию-агрегату принадлежат все комбинации сущностей агрегируемых понятий (полное декартово произведение множеств сущностей). Ассоциация (агрегация) имеет обратную абстракцию,

называемую индивидуализацией (декомпозицией).

**Пример 4.** Понятие-ассоциация Воздух соединяет такие понятия как Температура, Влажность, Давление и т.п. Сущностью понятия Воздух может быть «8°C, 45 %, 101 325 Па, ...». ♦

## 2 Понятийный анализ

Понятийный анализ состоит в выявлении понятий и их абстракций, которые необходимы и достаточны для описания заданной предметной области и решаемых в ней задач.

Абстракции идентификации, ассоциации и обобщения, используемые при понятийном анализе, рассматриваются как мыслительные операции, необходимые и достаточные для выделения и превращения в понятия имеющихся представлений из описываемой предметной области.

Основная цель понятийного анализа – многовариантная структуризация предметной области в виде ее понятийной структуры.

### 2.1 Проблемные области

Проблемы однозначного и непротиворечивого описания концептов, связанных с изменчивостью набора свойств у экземпляров, привели к появлению альтернативных теорий концептуального моделирования [3, с. 39]. Часть таких теорий базируется на вероятностном подходе, при котором набор свойств экземпляров концепта расширяется и их появление имеет вероятностную природу. Другие теории разделяют свойства экземпляров на типичные, присущие всем экземплярам, и атипичные, которые могут появляться лишь у некоторых экземпляров.

При понятийном моделировании проблемы, возникшие при описании концептов, решаются на основе выделения в предметной области проблемных областей [4]. Поскольку понятие предполагает субъективное отражение предметной области, то предусматривается возможность нескольких альтернативных описаний одного и того же понятия в каждой проблемной области по-своему. Это позволяет 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, по другому структурирующей анализируемую предметную область. Одно и то же понятие в разных аспектах (в разных проблемных областях) может иметь разные определения и описания, а обобщение одноименных понятий из различных проблемных областей может рассматриваться как концепт предметной области.

**Пример 5.** Концепт Подобие является обобщением понятий Подобие в аспектах Геометрия, Биология, Одежда, Архитектура, и т.д., которые, в свою очередь, являются понятиями-ассоциациями, соединяющими понятия, сущности которых подобны в каком-либо смысле (ассоциация по сходству).

Для понятия Подобие в аспекте Геометрия – это сущности понятия-ассоциации подобных треугольников вида «Треугольник А, Треугольник Б», сущности понятия-агрегации подобных окружностей вида «Окружность 1, Окружность 2» и т.д.

Для понятия Подобие в аспекте Биология это сущности «Легкие позвоночных, Легкие моллюсков», «Крылья птиц, Крылья насекомых», «Усики винограда, Усики гороха» и т.д. ♦

### 2.2 Понятийная структура

Результатом понятийного анализа является понятийная структура предметной области. В понятийной структуре перечисляются понятия и их аспекты, а также способы их образования.

Понятия, использованные для образования других понятий, называются атрибутами, а само понятие рассматривается как ассоциация своих атрибутов.

К общим атрибутам понятий относится понятие-тип Имя и понятие-тип Аспект, а также понятие-тип Абстракция. Частные атрибуты понятия – это понятия, использованные для его образования (абстрагирования).

Понятийная структура состоит из схем понятий. Схема понятия – это упорядоченный набор атрибутов, первые три элемента которого – общие атрибуты, остальные – частные атрибуты понятия.

**Пример 6.** На рис. 1 показан фрагмент понятийной диаграммы некоторой предметной области. Диаграмма состоит из геометрических фигур, внутри которых указаны имена понятий и их аспекты. Окружности используются для задания понятий-знаков, овалы – понятий-типов, прямоугольники – понятий-ассоциаций, а трапеции – понятий-обобщений. Стрелки на диаграмме соединяют понятий-атрибуты с образующими ими понятиями.

Из диаграммы могут быть получены схемы понятий, входящие в ее понятийную структуру:

- Разряд: Типизация, 1, 2, 3;
- Врач: Ассоциация, Специальность, Разряд, ФИО;

- Медсестра: Ассоциация, Профессия, Разряд, ФИО;
- Заведующий: Ассоциация, Врач, Отделение;
- Медработник: Обобщение, Врач, Медсестра;
- Штат: Ассоциация, Отделение, Медработник,

где имя понятия отделено от других атрибутов двоеточием. Понятия-типы Специальность, ФИО, Профессия, Отделение на диаграмме не определены, как и не указаны аспекты понятий. ♦

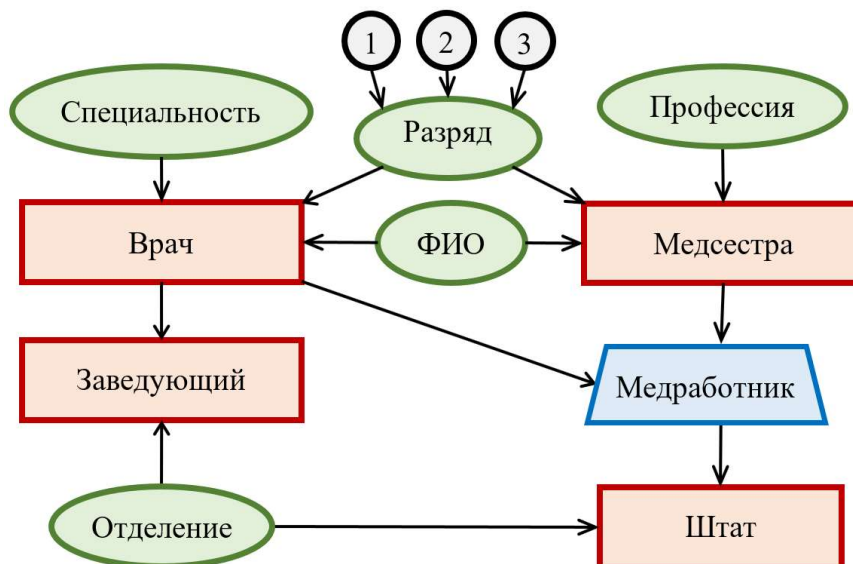


Рис. 1. Фрагмент понятийной диаграммы

Следует заметить, что на понятийной диаграмме по определению абстракций:

- к каждому понятию-обобщению и понятию-типу может идти только одна стрелка от каждого обобщаемого (типизируемого) понятия;
- к каждому понятию-ассоциации может идти одна и более стрелок от каждого ассоциируемого понятия.

Стрелка, идущая от обобщаемого (типизируемого) понятия и понятию-обобщению (понятию-типу) может интерпретироваться как взаимосвязь типа «член-группа» (англ. IsMemberOf), а от ассоциируемого понятия к понятию-ассоциации – как взаимосвязь типа «часть-целое» (англ. IsPartOf).

### 3 Понятийное моделирование

Понятийная модель состоит из понятийной структуры и описаний сущностей ее понятий, выполненных с использованием средств перечисления и разрешения множеств.

#### 3.1 Понятийная модель

В современных информационных системах понятийная модель предметной области может быть реализована с использованием системы управления базами данных реляционного типа.

В качестве понятий-знаков используются значения простых типов данных, а сами простые типы данных рассматриваются как встроенные понятия-типы, например, бит Bit, целое число Integer, число с плавающей запятой Float, дата и время Date, символ Char, строка Varchar, двоичные данные Binary и т.д. [6, с. 51].

Специфические для предметной области понятия-типы создаются путем ограничения простых типов данных и реализуются в виде таблиц базы данных, в которых перечисляются сущности понятия-типа, или в виде формул, задающих разрешающие процедуры для множества сущностей понятия-типа [6, с. 268].

Понятие-ассоциации реализуются в виде таблиц базы данных (англ. tables), соединяющих в одной записи сущности ассоциированных понятий, а понятия-обобщения представляются как запросы (англ. queries) или представления (виртуальные таблицы, англ. views), объединяющие записи из таблиц обобщаемых понятий [6, с. 180, 188, 228].

Понятия-ассоциации по своей реализации близки к типам отношений (англ. relation types), а понятия-обобщения – к типам сущностей (англ. entity types) [3, с. 41, 59].

Понятия помимо декларативных могут иметь процедурные описания и реализовываться,

например, в виде процедур (функций) на языке базы данных. Процедурные средства позволяют генерировать таблицы понятий, а также преобразовывать сущности-аргументы в сущности-результаты [6, с. 335].

Для отслеживания и обработки событий, возникающих в понятийной модели (создание, изменение и удаление сущностей), может использоваться механизм триггеров [6, с. 226].

### 3.2 База данных

Фрагмент диаграммы базы данных с понятийной моделью представлена на рис. 2, где показаны таблицы трех понятий: Notion, Attribute и Icon, а также ключевые поля, хранящие первичные ключи записей (англ. primary keys), и связи между таблицами, реализованные в виде полей с внешними ключами (англ. foreign keys) [6, p. 163].

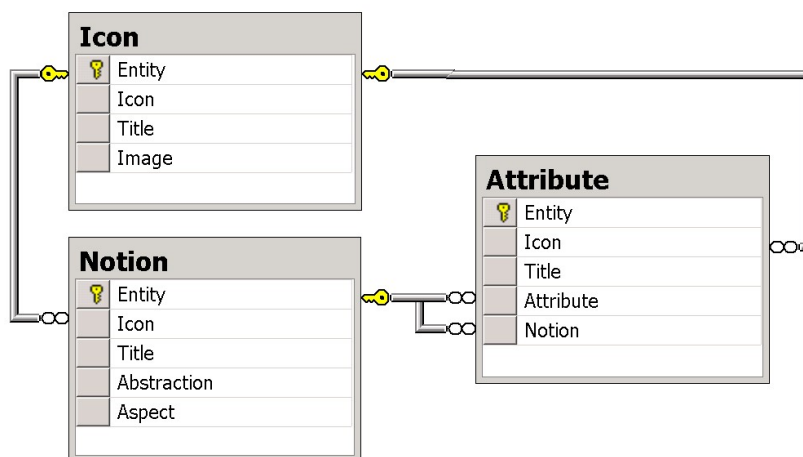


Рис. 2. Фрагмент диаграммы базы данных

Каждая запись в таблицах понятийной модели представляет собой сущность, имеющую следующие общие поля (общие атрибуты): Entity – уникальный идентификатор сущности, Title – наименование сущности (декоративное имя), Icon – иконка сущности (уникальный идентификатор изображения в таблице Icon). Поля Title и Icon используются для отображения сущностей в интерфейсе пользователя.

Таблица Notion имеет частные атрибуты Abstraction и Aspect, которые представляются полями встроенных типов данных с сущностями соответствующих понятий-типов: в поле Abstraction записывается один из символов, обозначающих абстракцию понятия: идентификация – С, типизация – Т, ассоциация – А и обобщение – G, а в поля Aspect – строка имени аспекта (рис. 3).

Entity	Icon	Title	Abstraction	Aspect
61000000060	63000000054	Attribute	A	Common
61000000061	63000000020	Notion	A	Common
61000000062	63000000018	Model	A	Common
61000000063	63000000019	Icon	A	Common
61000000064	63000000021	Abstraction	A	Common
61000000065	63000000034	Aspect	A	Common

Рис. 3. Фрагмент таблицы понятия Notion

Таблица Icon имеет частный атрибут Image, который является встроенным типом двоичных данных и содержит изображение иконки (рис. 4).

Entity	Icon	Title	Image
63000000018	63000000018	Model	0x424D3605000000...
63000000019	63000000019	Icon	0x424D3605000000...
63000000020	63000000020	Notion	0x424D3605000000...
63000000021	63000000021	Abstract	0x424D3605000000...
63000000022	63000000022	Abstraction	0x424D3605000000...
63000000023	63000000023	Tag	0x424D3605000000...

Рис. 4. Фрагмент таблицы понятия Icon

Таблица Attribute перечисляет частные атрибуты понятий и имеет два частных атрибута Attribute и Notion. Каждая запись в таблице Attribute описывает соответствующие поле в таблицах понятий, в том числе и поля самой таблицы Attribute (рис. 5).

Entity	Icon	Title	Attribute	Notion
6000000003	63000000020	Notion	61000000061	61000000060
6000000004	63000000023	Attribute	61000000061	61000000060
6000000001	63000000021	Abstract	61000000064	61000000061
6000000002	63000000034	Aspect	61000000065	61000000061
6000000019	63000000019	Image	61000000045	61000000063

Рис. 5. Фрагмент таблицы понятия Attribute

Для уникальной идентификации сущностей понятийной модели ключевые поля состоят из двух частей: первая часть – это номер записи сущности в таблице понятия, а вторая часть – это номер записи понятия как сущности в таблице понятий.

Например, уникальным идентификатором иконки «Понятие» является число 63000000020. Из анализа идентификатора следует, что иконка как сущность имеет номер 20 в таблице иконок Icon (рис. 4), а иконка как понятие Icon имеет номер 63 в таблице понятий Notion (рис. 3). •

Следует обратить внимание на то, что в понятийной модели понятие само является понятием и описывается теми же средствами, что и другие понятия.

### 3.3 База знаний

Понятийная база знаний – это база данных, содержащая понятийную модель, а также механизм вывода, позволяющий обрабатывать предметные знания.

Обработка 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 выполнять умозаключения и получать новые утверждения (факты) об имеющихся или вновь выводимых фактах [2, с. 49].

В понятийной базе знаний фактами являются истинными высказываниями с логическим связками И ( $\wedge$ ), ИЛИ ( $\vee$ ), НЕ ( $\neg$ ), круглыми скобками и двумя видами элементарных выражений: одноместными предикатами принадлежности сущности  $X$  понятию  $N$  вида  $N(X)$ , а также отношениями вида  $X[M] \div Y$ , где  $X[M]$  – функтор, возвращающий сущность понятия-атрибута  $N$  для сущности  $X$ ,  $\div$  – знак отношения, допустимого между сущностями  $X[M]$  и  $Y$  (равно  $=$ , больше  $>$ , и т.п.). Функтор  $X[M]$  возвращает специальную сущность  $\varepsilon$  (пустое понятие), если сущность  $X$  не имеет атрибута  $N$ .

**Пример 7.** Для понятийной модели, диаграмма которой показана на рис. 1, могут быть выражены следующие факты:

- Разряд(2) – 2 есть сущность понятия-типа Разряд;
- Медсестра( $X$ ) –  $X$  есть сущность понятия-ассоциации Медсестра или, если кратко,  $X$  – Медсестра;
- $Y[\text{Разряд}] = 3$  – Врач  $Y$  имеет Разряд, равный 3. ♦

Универсальное отношение равенства сущностей определяется рекурсивно: две сущности равны тогда и только тогда, когда значения (сущности) атрибутов этих сущностей равны. Другие отношения могут быть унаследованы от встроенных типов данных, используемых в понятийной модели.

В понятийной модели схемы понятий задают правила вывода, а понятийная структура рассматривается как формальная теория предметной области, которая сохраняет истинность всех выводимых в ней следствий.

Из схемы понятия-обобщения следует правило вывода, согласно которому сущность  $X$  принадлежит понятию  $N$  тогда и только тогда, когда эта сущность принадлежит хотя бы одному понятию-атрибуту понятия  $N$ . Из схемы понятия-ассоциации следует правило вывода, согласно которому если сущность  $X$  принадлежит понятию  $N$ , то атрибуты сущности  $X$  являются атрибутами понятия  $N$ .

**Пример 8.** Понятийная структура, диаграмма которой показана на рис. 1, порождает следующие правила вывода (но не все):

- $\forall X \text{ Медработник}(X) \leftrightarrow \text{Врач}(X) \vee \text{Медсестра}(X)$ ;
- $\forall X \text{ Штат}(X) \rightarrow \text{Медработник}\{X\} \wedge \text{Отделение}\{X\}$ ,

где  $\forall$  – квантор всеобщности;  $\rightarrow$  ( $\leftrightarrow$ ) – импликация (двухсторонняя);  $N\{X\}$  – предикат наличия у сущности  $X$  атрибута  $N$ ,  $N\{X\} \leftrightarrow \neg X[N] = \varepsilon$ ;  $\varepsilon$  – пустое понятие. ♦

При выводе на знаниях требуется установить или опровергнуть некоторый факт путем поиска сущностей, удовлетворяющих заданным условиям. В этом случае правила, следующие из схемы понятия-обобщения, задают понятия, подлежащие обработке, а правила, следующие из схемы понятия-ассоциации, позволяют установить наличие атрибутов у сущностей этого понятия и извлечь значения атрибутов, если это необходимо. Для поиска в базе знаний сущностей, удовлетворяющих заданным условиям, используются запросы.

**Пример 9.** Рассмотрим базу знаний с понятийной структурой, изображенной на рис. 1. Из рисунка видно, что в понятийной модели определены следующие предикаты и функторы:

- Врач( $X$ ),  $X$ [ФИО],  $X$ [Специальность],  $X$ [Разряд];
- Медсестра( $Y$ ),  $Y$ [ФИО],  $Y$ [Профессия],  $Y$ [Разряд];
- Заведующий( $Z$ ),  $Z$ [Врач],  $Z$ [Отделение];
- Медработник( $V$ ),  $V$ [Врач],  $V$ [Медсестра];
- Штат( $W$ ),  $W$ [Медработник],  $W$ [Отделение],

где ФИО, Специальность, Профессия, Разряд и Отделение являются понятиями-типами, заданными перечислением принадлежащих им сущностей, как, например, это сделано для понятия-типа Разряд.

Следует обратить внимание на функторы  $V$ [Врач] и  $V$ [Медсестра] понятия-обобщения Медработник. Если  $V$  – сущность понятия Врач, то  $V$ [Врач] =  $V$ , в противном случае  $V$ [Медсестра] =  $\varepsilon$ . Аналогично ведет себя и функтор  $V$ [Врач].

Используя имена сущностей и их переменные, а также перечисленные выше предикаты и функторы, можно формулировать запросы к базе знаний, например, такой: «Кто из штата отделения Хирургия не является медсестрой с третьим разрядом?». На языке формальной логики этот запрос будет иметь выражение

$$\begin{aligned} \text{Штат}(X) \wedge X[\text{Отделение}] = \text{Хирургия} \wedge \forall Y (X[\text{Медработник}] = Y \wedge \\ \wedge \text{Медсестра}(Y) \wedge \neg Y[\text{Разряд}] = 3), \end{aligned}$$

в котором  $X$  является свободной переменной, а  $Y$  – связанной. ♦

## 4 Процедурные знания

Декларативные знания в понятийной базе знаний представляются понятиями и принадлежащими им сущностями. Для представления процедурных знаний предусмотрены процедуры и функции, которые, как и все в понятийных моделях, являются сущности соответствующих понятий.

Без ограничения общности операциями, через которые выражаются процедурные знания, являются операции создания, изменения и удаления сущностей. Для повышения уровня процедурного языка до универсальных языков программирования предусматривается операция присваивания, а также условные операторы и операторы циклов. Условия у этих операторов выражаются в виде фактов, выполнимость которых проверяется. В свою очередь результат выполнения запроса может быть присвоен внутренней переменной для дальнейшей обработки извлеченных знаний.

### 4.1 Встроенные типы данных

Рассмотрим процедурный язык понятийной базы знаний (англ. knowledge language). Встроенными типами данных этого языка являются байты Byte, символы Char, целые числа Integer, числа с плавающей запятой Float со встроенными операциями в стиле языка C.

В язык добавлен встроенный тип данных Date, а также строковый String и двоичный Binary типы с двуместными операциями конкатенации `` (двойной обратный апостроф, англ. grave accent) и присваивания с конкатенацией ``=. Сущностями последних двух типов являются последовательности символов и байтов конечной длины, например, "это строка", 0x1F2D45E3DAF102A8AB92. Сущности понятия-типа Date – это день и/или время, заключенные в апострофы, например: '29.05.2021 14:32:56', '29.05.2021', '14:32:56'.

Для представления процедурных знаний в язык введен встроенный тип данных Program, сущности которого – последовательности команд некоторой виртуальной машины, генерируемые лямбда-выражениями в стиле языка Java ( $X$ )  $\rightarrow$   $Y$ , где  $X$  – аргументы (могут отсутствовать), а  $Y$  – оператор тела функции (процедуры).

Единственной встроенной понятийной константой является пустое понятие ' ' (двойной апостроф) с двуместными операциями объединения с пустым понятием ?? и присваивания с объединением с пустым понятием ??= в стиле языка C#.

## 4.2 Операции

Логическая часть языка стандартная:

- $=$ ,  $!$  – отношения равенства и неравенства;
- $>$ ,  $<$ ,  $>=$ ,  $<=$  – отношения порядка;
- $!$ ,  $\&\&$ ,  $||$  – логические связи НЕ, И, ИЛИ.

Для распараллеливания вычислений используются операции последовательного и параллельного вычисления:

- $X, Y, \dots, Z$  – последовательное вычисление  $X, Y, \dots, Z$ ;
- $X :: Y :: \dots :: Z$  – параллельное вычисление  $X, Y, \dots, Z$ .

Для улучшения выразительности языка и повышения компактности программ расширен состав тернарных операций:

- $X ? Y : Z$  – условное вычисление (если  $X$ , то  $Y$ , иначе  $Z$ );
- $X \sim Y : Z$  – цикл (пока  $X$ , вычислять  $Y$ , потом  $Z$ );
- $X ! Y : Z$  – блокировка  $X$  пока вычисляется  $Y$ , потом  $Z$ ;
- $X ` Y : Z$  – поиск в  $X$  вхождений  $Y$  и замена  $Z$ .

Понятийные операции языка:

– операции абстракции:

- $[N1 : X1, \dots, ND : XD]$  – создание понятия-ассоциации с атрибутами  $Xj$  и именами  $Nj$  ( $j = 1, \dots, D$ ),
- $\{N1 : X1, \dots, ND : XD\}$  – создание понятия-обобщения с атрибутами  $Xj$  и именами  $Nj$  ( $j = 1, \dots, D$ );

– операции интенционала:

- $X[]$  – число атрибутов понятия  $X$ ,
- $X[Y]$  – доступ к атрибуту  $Y$  сущности  $X$ ,
- $[X]Y$  – доступ к атрибуту  $X$  понятия  $Y$ ,
- $[]X$  – понятие сущности  $X$ ;

– операции экстенционала:

- $X\{\}$  – число сущностей понятия  $X$ ,
- $X\{Y\}$  – доступ к сущности  $Y$  понятия  $X$ ,
- $\{X\}Y$  – создание сущности  $X$  понятия  $Y$ ,
- $\}X$  – удаление сущности  $X$ ;

– процедурные операции:

- $X.Y$  – функция (процедура)  $Y$  понятия  $X$ ,
- $X()$  – выполнение процедуры  $X$ ,
- $X(Y)$  – выполнение функции  $X$  с аргументами  $Y$ ,
- $(X)Y$  – определение функции с аргументами  $X$  и телом  $Y$ ,
- $()X$  – определение процедуры с телом  $X$ ;

– запросы к базе знаний:

- $\langle X \rangle Y$  – запрос  $Y$  со свободной переменной  $X$ ,
- $\langle \rangle X$  – обновление сущности  $X$ ;

– операции отношений:

- $X \wedge = Y$  – понятие  $X$  включено в понятие  $Y$  (операция in),
- $X ? = Y$  – понятие  $X$  есть понятие  $Y$  (операция is),
- $X := Y$  – понятие  $X$  как понятие  $Y$  (операция as),
- $X ` = Y$  – понятие  $X$  похоже на понятие  $Y$  (операция like);

– операции присваивания:

- $X = Y$  – присваивание сущности  $X$  сущности  $Y$ ,
- $X || = Y$  – присваивание с объединением  $X$  и  $Y$ ,
- $X \&\& = Y$  – присваивание с пересечением  $X$  и  $Y$ ,
- $X \wedge \wedge = Y$  – присваивание с исключением  $Y$  из  $X$ ,
- $X :: = Y$  – присваивание с клонированием  $Y$  в  $X$ .

## 4.3 Операторы

Операторы языка стандартные, за исключением операторов транзакций:

- операция присваивания и точка с запятой;
- составной оператор – операторы в фигурных скобках;
- goto – оператор перехода;
- switch – оператор выбора;
- if-then-else – условный оператор;



- while, do, for – операторы циклов;
- try-catch, throw – операторы исключений;
- begin, end, undo – операторы транзакций [6, с. 753].

**Пример 10.** Ниже приведена процедура, которая повышает разряды медсестрам – медперсоналу отделения Хирургия (см. пример 9).

```
// Определение 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 локального понятия Медсестры
Медсестры = <X>(Штат(X) != ' ' && X[Отделение] == Хирургия
&& Медсестра(X[Медработник]) != ' ');
// Цикл по сущностям понятия Медсестры
while (Индекс < Медсестры{ })
{
    // Создание локальной сущности Медсестра
    Медсестра = Медсестры{Индекс};
    // Увеличение атрибута Разряд сущности Медсестра
    Медсестра[Разряд] += Повышение(Медсестра[Разряд]);
    // Приращение индекса сущности понятия Медсестры
    Индекс = Индекс + 1;
}
// Обновление измененных сущностей
< > Медсестры;
// Успешное завершение транзакции
end Разряды; ♦
```

Следует обратить внимание на то, что в программах операции присваивания создают только локальные переменные. Однако связь с базой знаний сохраняется для всех извлеченных из нее сущностей. Для внесения изменений в базу знаний после изменения сущностей используется оператор обновления.

Операция доступа к атрибуту позволяет извлекать и изменять не только частные атрибуты сущности  $X$  по их номерам или именам, но и общие атрибуты по их именам, например, наименование сущности  $X$ [Title], ее иконку  $X$ [Icon]. Однако изменение наименований и иконок никак не сказывается на обработку знаний, так как операции над сущностями осуществляются с использованием уникальных идентификаторов Entity. Если для обеспечения целостности обработки знаний требуется зафиксировать наименования или иконки сущностей, то применяется механизм транзакций.

#### 4.4 Методы

Функции на процедурном языке могут использоваться для вычисления виртуальных атрибутов понятий по значениям других атрибутов. Так для встроенного понятия-типа Date предусматриваются такие виртуальные атрибуты как Year, Month, Day, Time, Hour, Minute, Second, Millisecond, а для понятия-типа String (Binary) – виртуальный атрибут длины Length.

Помимо виртуальных атрибутов могут определяться функции и процедуры понятий в стиле методов классов объектно-ориентированных языков программирования. Например, для понятия-типа Date таким методом является  $X.Diff(W, Y)$  – разница дат  $X$  и  $Y$  в неделях, а для понятий-типов String и Binary – индексаторы, возвращающие соответственно символ и байт с индексом  $Y$ :  $X.Char(Y)$  и  $X.Byte(Y)$ .

Для хранения методов как специального типа данных может использоваться понятие-ассоциация Method, например, со следующим интенционалом:

```
This: Notion;
Argument1: Notion, ..., Argument16: Notion;
Arguments: Number;
Code: Program;
Return: Notion;
Text: String;
```

согласно которому сущность понятия Method связывается с понятие This и состоит из понятий-аргументов с именами Argument1, ..., Argument16, их числа Arguments, исполняемого кода Code, возвращаемого понятия Return и текста функции на процедурном языке Text.

**Пример 11.** Метод Empty.Now() является статическим методом (привязан к пустому понятию Empty) и возвращает текущую дату и время. Другой пример – ранее упомянутый метод Date.Diff(Part: String, End: Date), который является методом понятия Date и вычисляет разницу дат Date и End в единицах Part. ♦

## 5 Заключение

Принципиальным отличием рассмотренного подхода к анализу и моделированию предметных областей, представлению и обработке знаний является использование помимо формальной логики еще одного семантического инварианта – формальной теории понятий. Формальная логика основана на трех логических связках (НЕ, И, ИЛИ), которые используются для построения формальных высказываний, а формальная теория понятий – на трех ментальных абстракциях (идентификация, ассоциация, обобщение), которые используются для построения формальных понятий.

Семантическими средствами формальной теории понятий, делающими ее инвариантной предметным областям, являются:

- именованье сущностей предметной области при идентификации и образование первичных семантических категорий;
- объединение в каком-либо смысле сущностей предметной области при обобщении;
- соединение сущностей предметной области при ассоциации так, что смысл такого соединения заключается в создании составных (укрупненных) сущностей;
- перечисление сущностей предметной области при обобщении и ассоциации в смысле ограничения существования подобных сущностей.

Анализ и моделирование названы понятийными, чтобы отличить их от концептуального анализа и концептуального моделирования. Понятие отличается от концепта. Концепт – это абстрактное объективное понятие, а понятие – конкретный субъективный концепт. По этой причине существует множество формальных понятий, имеющих одно и то же имя, но разные структуру и содержание в различных аспектах (проблемных областях). В этом случае концепт может быть определен как обобщение одноименных понятий в различных аспектах.

Для представления декларативных знаний в понятийной базе знаний используется перечисление сущностей понятий в таблицах базы данных. Важным свойством понятийной базы знаний является хранение процедурных знаний также, как и декларативных. Это позволяет отказаться от сложных программных средств, запускаемых на стороне пользователя, а использовать в качестве основного хранилища программ сервер баз знаний.

Основная трудность использования понятийных моделей видится в необходимости использования новой методологии анализа и моделирования предметных областей, а также новой технологии представления и обработки знаний. Однако высокая эффективность и выразительность реализуемых моделей знаний, их легкая актуализация и адаптация, позволяют надеяться на широкое внедрение понятийного моделирования в практику создания и использования современных интеллектуальных информационных систем.

Следующим шагом в исследовании формальной теории понятий видится оптимизация алгоритмов выполнения запросов к понятийной базе знаний, встроенных в язык описания знаний, и получение оценок их вычислительной сложности. Однако уже сейчас можно сказать, что время выполнения запроса будет иметь хорошую полиномиальную оценку, т.к. поиск сущности в таблице понятия с числом записей  $n$  осуществляется за время с асимптотической оценкой  $n$  [6, с. 667], а в случае использования индекса – то за время с асимптотической оценкой  $\log n$  [6, с. 601], где  $\log$  – логарифмическая функция.

Другим направлением развития понятийного анализа и понятийного моделирования видится разработка на языке описания знаний методов обучения понятийных моделей на основе восприятия окружающего мира в виде поступающего потока неструктурированных данных. Понятийная модель позволяет структурировать этот поток, выделяя в нем сущности известных и неизвестных понятий. Найденные неизвестные сущности могут использоваться для изменения понятийной модели путем пополнения экстенционалов существующих понятий или путем образования новых. К этому направлению относится и большой класс задач по распознаванию образов.

## Литература

1. *Vykhovanets V. S.* The Notional Model of Knowledge Representation // Journal of Physics: Conference Series, Vol. 1864, No 012058. – 2021. – URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1864/1/012058>.
2. *Brachman R. J., Levesque H. J.* Knowledge Representation and Reasoning. – San Francisco, CA: Morgan Kaufmann, 2004. – P. 31-45.
3. *Olive A.* Conceptual Modeling of Information Systems. – New York, NY: Springer, 2007. – 471 p.
4. *Выхованец В. С.* О существенной неполноте формального метода // International Interdisciplinary Conference «Philosophy, Mathematics, Linguistics: Aspects of Interaction» (PML'2009). – СПб: Международный математический институт им. Л. Эйлера, 2009. – С. 79-87.
5. The Oxford Dictionary of Current English / Ed. D. Thompson. – New York, NY: Oxford University Press, 1993. – 1091 p.
6. *Elmasri R., Navathe S. B.* Fundamentals of Database Systems. Hoboken. – New Jersey: Pearson, 2016. – 413 p.