ПОНЯТИЙНЫЙ АНАЛИЗ И ПОНЯТИЙНОЕ МОДЕЛИРОВАНИЕ

Выхованец В. С.[1]
(*ФГБУН Институт проблем управления им. В.А. Трапезникова РАН, Москва*)

Описываются понятийные модели, которые строятся на первичных ментальных абстракциях идентификации, обобщении и ассоциации. В процессе понятийного анализа строится понятийная модель предметной области, состоящая из понятийной структуры и содержания входящих в нее понятий. Понятийная структура определяет каждое понятие как результат обобщения или ассоциации других понятий. Содержание понятий задается перечислимыми или разрешимыми множествами сущностей предметной области. Основным отличием понятийной модели от других моделей знаний является отказ от описания ассоциации понятий в виде взаимосвязи. В понятийной модели ассоциации являются такими же понятиями, как и обобщения, что позволяет образовывать из ассоциаций новые понятия. Все это делает понятийную модель семантически инвариантной, т.е. не зависящий в своей интерпретации от знаний из предметной области. Еще одним отличием понятийных моделей является многоаспектное выражение понятий. Для обоснования предложенного подхода разработана формальная теория понятий. Семантическая часть теории (язык понятий) обосновывает методологию понятийного анализа, а синтаксическая ее часть (исчисление понятий) – технологию понятийного моделирования. Доказано, что язык понятий разрешим, полон и непротиворечив, в то время как исчисление понятий разрешимо и непротиворечиво на счетных моделях, а полно только на конечных моделях. Использование понятийного анализа и понятийного моделирования позволяет улучшить выразительность и наглядность представления знаний, повысить эффективность и надежность их обработки.

Ключевые слова: ментальные абстракции, понятие, концепт, понятийный анализ, понятийная модель, язык понятий, исчисление понятий, база знаний, вывод на знаниях, интеллектуальная система.

1. Введение

Нерешенные проблемы в области представления и обработки знаний затрудняют эффективную реализацию современных баз знаний и интеллектуальных информационных систем на их основе. В частности, извлечение знаний до сих пор не является

---

[1] *Валерий Святославович Выхованец, д.т.н., доцент* (valery@vykhovanets.ru).

полностью решенной задачей [16], формы представления знаний слабо адаптированы к психическим и психологическим особенностям человека [5, с. 31], обработка знаний занимает длительное время и подвержена возникновению скрытых противоречий [12].

Причина таких проблем заключается в том, что при моделировании человеческого мышления до сих пор используются традиции, заимствованные из логики, где не учитываются такие свойства человеческого мышления, как объективность (открытость к новому знанию, замена устаревшего знания новым), субъективность (перестройка мышления для решения конкретных прикладных задач), рефлексивность (самоорганизация познавательной деятельности), продуктивность (постоянная генерация нового знания) [1].

Некоторые из существующих проблем удалось избежать с помощью понятийного анализа и понятийного моделирования. Модели называются понятийными, чтобы отличать их от концептуальных моделей. В концептуальных моделях задаются понятия и различные взаимосвязи (отношения) между ними.

Например, в таких концептуальных моделях, как логические, продукционные, дескрипционные, грамматические, фреймовые, объектно-ориентированные, семантические сети и т.д. [11, с. 31], отношения между сущностями предметной области определяются как предикаты, которые интерпретируются как первичные семантические категории предметных знаний и не используются для образования других понятий. А если такие понятия все же требуется образовать, то выразительные возможности логического языка концептуальной модели необходимо повысить до логики второго порядка [10, с. 375]. Последнее порождает неразрешимые проблемы – в теоретическом плане, и непреодолимые трудности – в прикладном.

Наиболее близкой к понятийной модели является ER-модель, где абстракциями концептуального проектирования являются классификация, агрегация и обобщение [4, с. 41], которые используются для классификации примитивных объектов и построения из них сложных классов. Для задания взаимосвязи классов используется абстракция агрегации, которая не порождает ни объекта, ни класса, а рассматривается как семантическая категория, посредством которой производится интерпретация модели.

В понятийной модели отношения между понятиями являются обычными понятиями [15], понятийная модель строится путем указания абстракций, использованных для образования понятий, а также перечисления сущностей предметной области, им принадлежащих. Каждое понятие описываются в одном или нескольких аспектах, что позволяет учесть многовариантность и изменчивость предметных знаний.

2. 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 [14, с. 607]. Понятие соотносится с представлениями из предметной области – сущностями (англ. entities). В свою очередь концепт (англ. concept) – это общее понятие, абстрактная идея [14, с. 171]. Таким образом, в отличие от концепта, понятие не объективно, а субъективно по определению.

Концепт состоит из экземпляров (англ. instances), каждый экземпляр характеризуется совокупностью свойств (англ. properties), а свойства могут быть определенными (англ. defining) и неопределенными (англ. nondefining) [11, с. 38].

Понятия формируются (определяются) в процессе абстрагирования путем выполнения ментальных операций над сущностями. Каждое понятие имеет имя, которое обозначает некоторый набор сущностей из предметной области.

Существует три вида понятий и порождающих их ментальных абстракций (отвлечений): понятия-знаки (абстракция идентификации, отождествления), понятия-обобщения (абстракция обобщения) и понятия-ассоциации (абстракция ассоциации).

2.1. 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 интерпретации, при которой понятие-знак сопоставляется с некоторой уникальной сущностью из предметной области.

Пример 1. Примерами понятий-знаков могут служить понятия, обозначенные такими знаками (словами) как «Зеленый», «Длинный», «Первый», «Редко», «Много» и т.п. ♦

2.2. 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. Примером понятия-обобщения является понятие «Дерево», которое объединяет деревья различных пород: «Береза», «Бук», «Пихта», «Сосна», «Тополь» и т.д. В свою очередь «Береза» является специализацией понятия-обобщения «Дерево». ♦

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

Пример 3. Понятие-тип «Цвет» может быть образовано путем типизации таких понятий-знаков как «Красный», «Оранжевый» и т.д., а понятие-знак «Красный» является конкретизацией понятия-типа «Цвет». ♦

2.3. АССОЦИАЦИЯ

Понятие-ассоциация образуется при соединении 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 абстракции ассоциации является абстракция агрегации, при которой образовываемому понятию-агрегату принадлежат все комбинации сущностей агрегируемых понятий (полное декартово произведение множеств сущностей). Ассоциация (агрегация) имеет обратную абстракцию, называемую индивидуализацией (декомпозицией).

Пример 4. Примером понятия-ассоциации является понятие «Погода», которое соединяет такие понятия как «Место», «Дата», «Температура», «Влажность», «Ветер», «Облачность» и т.п. Сущностью понятия «Погода» может быть «г. Москва, 15 марта 2021 г., 8°C, 45 %, 3 м/с, Облачно, …». ♦

3. 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.

3.1. ПРОБЛЕМНЫЕ ОБЛАСТИ

Проблемы однозначного и непротиворечивого описания концептов, связанные с изменчивостью набора свойств у экземпляров, привели к появлению альтернативных теорий концептуального моделирования [11, 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 концепта, 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 [1]. Поскольку понятие предполагает субъективное отражение предметной области, 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, а обобщение одноименных понятий из различных проблемных областей может рассматриваться как концепт предметной области.

Пример 5. Концепт «Противоположность» является обобщением понятий «Противоположность» в аспектах «Свет», «Красота» и т.д., которые, в свою очередь, являются понятиями-ассоциациями, соединяющими понятия, сущности которых противоположны в каком-то смысле (ассоциация по контрасту). Для понятия «Противоположность» в аспекте «Свет» это сущности «Яркий, Неяркий», «Яркий, Тусклый», «Тусклый, Сильный», «Видимый, Невидимый», и т.д. Для понятия «Противоположности» в аспекте «Красота» это сущности «Красивый, Некрасивый», «Красивый, Уродливый», «Прекрасный, Некрасивый» и т.д. ♦

3.2. ПОНЯТИЙНАЯ СТРУКТУРА

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

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

К общим атрибутам понятий относится понятие-тип «Имя» (имя понятия), понятие-тип «Аспект» (имя проблемной области), а также понятие-тип «Абстракция» (способ образования понятия), состоящий из понятий-знаков «Знак», «Тип», «Обобщение» и «Ассоциация». Частные атрибуты понятия или просто атрибуты – это понятия, необходимые 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, ромбы – понятий-ассоциаций, а прямоугольники – понятий-обобщений. Стрелки на диаграмме соединяют понятий-атрибуты с образовываемыми ими понятиями.

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

– «Разряд»: «Типизация», «1», «2», «3»;
– «Рабочий»: «Ассоциация», «Профессия», «Разряд», «ФИО»;
– «Инженер»: «Ассоциация», «Специальность», «ФИО»;
– «Руководитель»: «Ассоциация», «Инженер», «Бригада»;
– «Сотрудник»: «Обобщение», «Инженер», «Рабочий»;
– «Участник»: «Ассоциация», «Сотрудник», «Бригада»;

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

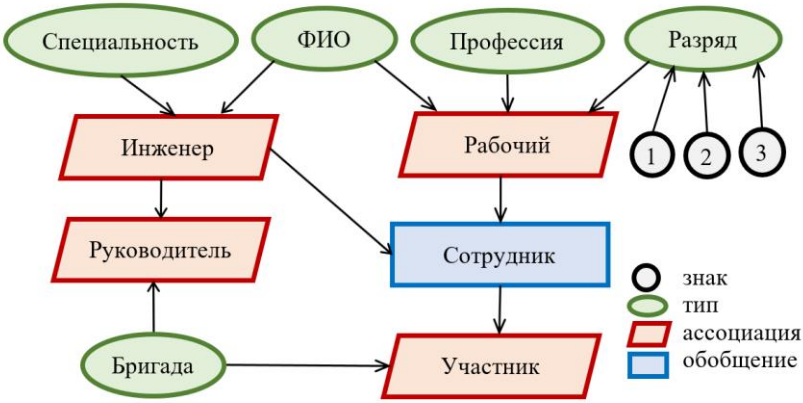

Рис. 1. Фрагмент понятийной диаграммы

Следует заметить, что по определению абстракций на понятийной диаграмме:

– 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. IsMemeberOf), а от ассоциируемого понятия к понятию-ассоциации – как взаимосвязь типа «часть – целое» (англ. sPartOf).

4. Понятийное моделирование

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

4.1. 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 «DateTime», 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].

Понятия-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) [11, с. 41, 59].

Помимо декларативных понятия 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].

Пример 7. 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, с. 163].

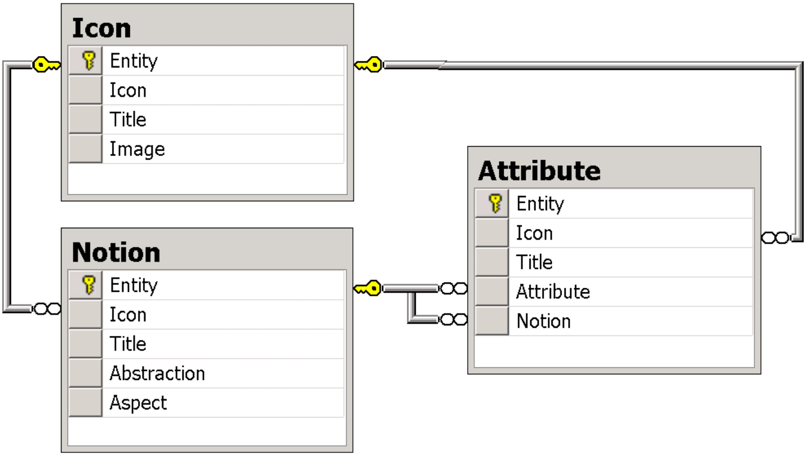

Рис. 2. Фрагмент диаграммы базы данных

Каждая запись в таблицах понятийной модели представляет собой сущность, имеющую следующие общие поля (общие атрибуты): Entity – уникальный идентификатор сущности, Tittle – наименование сущности (декоративное имя), Icon – иконка сущности (уникальный идентификатор изображения в таблице Icon). Поля Tittle 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 – *C*, типизация – *T*, ассоциация – *A* и обобщение – *G*, а в поля Aspect – строка имени аспекта (рис. 3).

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

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

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

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

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.

Пример 8. Пусть уникальным идентификатором иконки 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.

4.2. 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 делать умозаключения и получать новые утверждения (факты) об имеющихся или вновь выводимых фактах [5, с. 49].

4.2.1. ФАКТЫ

В понятийной базе знаний фактами являются истинные высказывания 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 $X[N] \div Y$, где $X[N]$ – функтор, возвращающий сущность понятия-атрибута N сущности X; $\div$ – знак отношения, допустимого между сущностями $X[N]$ и Y (равно =, больше >, и т.п.). Функтор $X[N]$ возвращает специальную сущность ε (пустое понятие), если сущность X не имеет атрибута N.

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.

Пример 9. Для понятийной модели, диаграмма которой показана на рис. 1, могут быть выражены следующие факты:

– Разряд(2) – 2 есть сущность понятия-типа «Разряд»;

– Рабочий(X) – X есть сущность понятия-ассоциации «Рабочий» или, если кратко, X – Рабочий;

– X[Разряд] = 3 – Рабочий X имеет Разряд, равный 3. ♦

4.2.2. ПРАВИЛА ВЫВОДА

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

– $\forall X$ Сотрудник(X) $\leftrightarrow$ Инженер(X) $\vee$ Рабочий(X);

– $\forall X$ Участник(X) $\rightarrow$ Сотрудник$\{X\}$ $\wedge$ Бригада$\{X\}$,

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

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.

4.2.3. ЗАПРОСЫ

Для поиска в базе знаний сущностей, удовлетворяющих заданным условиями, используются запросы.

Пример 11. Рассмотрим базу знаний с понятийной структурой, изображенной на рис. 1. Из рисунка видно, что в понятийной модели определены следующие предикаты и функторы:

– Инженер(X), X[Специальность], X[ФИО];
– Рабочий(Y), Y[ФИО], Y[Профессия], Y[Разряд];
– Руководитель(Z), Z[Инженер], Z[Бригада];

– Сотрудник(V), V[Инженер], V[Рабочий];
– Участник(W), W[Сотрудник], W[Бригада],

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

Следует обратить внимание на функторы V[Инженер] и V[Рабочий] понятия-обобщения «Сотрудник». Если V – сущность понятия «Инженер», то V[Инженер] = V, в противном случае V[Инженер] = ε. Аналогично ведет себя и функтор V[Рабочий].

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

$$\text{Участник}(X) \wedge X[\text{Бригада}] = \text{В} \wedge \forall Y\ (X[\text{Сотрудник}] = Y \wedge$$
$$\wedge\ \text{Рабочий}(Y) \wedge \neg\ Y[\text{Разряд}] = 3),$$

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

4.2.4. 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 сущностями 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.

Рассмотрим процедурный язык понятийной базы знаний (англ. knowledge language). Встроенными типами данных этого языка являются байты Byte, символы Char, натуральные числа Number, целые числа Integer, числа с плавающей запятой Float со встроенными операциями в стиле языка C [8].

В язык добавлен встроенный тип данных DateTime, 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: ″строка″, 0x12DE3A. Сущности понятия-типа «DateTime» – это дата и/или время, заключенные в апострофы, например: ′29.05.2021 14:32:56′, ′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) –> 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#.

Логическая часть языка стандартная:

– ==, != – отношения равенства и неравенства;
– >, <, <=, >= – отношения порядка;
– !, &&, || – логические связки НЕ, И, ИЛИ.

Для распараллеливания вычисления используются операции последовательного и параллельного вычисления:

– X, Y, …, Z – последовательное вычисление X, Y, …, Z;
– X :: Y :: … :: Z – параллельное вычисление X, Y, …, 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 ~ 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 абстракций:

$[N1 : X1, \ldots, ND : XD]$ – создание понятия-ассоциации с атрибутами Xj и именами Nj ($j = 1, 2, \ldots, D$),

$\{N1 : X1, \ldots, ND : XD\}$ – создание понятия-обобщения с атрибутами Xj и именами Nj ($j = 1, 2, \ldots, D$);

– операции интенсионала:

$X[]$ – число атрибутов понятия X,

$X[Y]$ – доступ к атрибуту Y сущности X,

$[X]Y$ – доступ к атрибуту Y понятия X,

$[]X$ – понятие сущности X;

– операции экстенсионала:

$X\{\}$ – 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,

$(X)Y$ – запрос Y со свободной переменной X,

$()X$ – обновление сущности X;

– операции отношений:

X =^ 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 ^^= Y – присваивание с исключением Y из X,

X ::= Y – присваивание с клонированием Y в X.

Операторы языка:

– 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, commit, rollback – операторы транзакций [6, с. 753].

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

Пример 12. Рассмотрим пример процедуры, которая повышает разряды рабочим – участникам бригады В (см. пример 11).

```
// 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 Рабочие
Рабочие = (X)(Участник(X) != ′ ′ && X[Бригада] == В
      && Рабочий(X[Сотрудник]) != ′ ′);
// Цикл по сущностям 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

()Рабочие;

// Успешное завершение транзакции

commit 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] (см. пример 7).

Однако изменение наименований и иконок никак не сказывается на обработке знаний, так как операции над сущностями осуществляются с использованием уникальных идентификаторов Entity (см. пример 8). Если для обеспечения целостности обработки знаний требуется зафиксировать наименования или иконки сущностей, то применяется механизм транзакций.

Функции на процедурном языке могут использоваться для вычисления виртуальных атрибутов понятий по значениям других атрибутов.

Так, для встроенного понятия-типа «DateTime» предусматриваются такие виртуальные атрибуты как Date, 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 «DateTime» 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 задания статических функций может использоваться специальное понятие-ассоциация «Function», например, со следующими атрибутами:

Argument1: Notion, …, Argument16: Notion;
Arguments: Number;
Code: Program;
Return: Notion;
Text: String,

согласно которому сущность понятия «Function» состоит из аргументов с именами Argument1, …, Argument16, их числа Arguments, исполняемого кода Code, возвращаемого понятия Return и текста функции на процедурном языке Text. Примером такой функции может служить функция Function.Now(), которая возвращает текущую дату и время.

5. Формальная теория

Предыдущая часть статьи была посвящена разъяснению основных понятий и механизмов понятийного анализа и понятийного моделирования, а также описанию реализации базы знаний, в основе которой лежит понятийная модель. Рассмотрим теперь формальную сторону понятийного анализа и понятийного моделирования – теорию понятий.

Следует заметить, что иных более общих выразительных средств для формализации предметной области, кроме как ее описания в виде понятийной модели, не существует. Это объясняется тем, что понятийное мышление является предельно общим и единственным механизмом рационального познания окружающего мира, а понятийное моделирование – субъективной и, в конечном счете, объективной формой фиксации результатов понятийного мышления.

Очевидно, что предметная область является семантическим объектом теории понятий, а понятийная модель – ее синтаксическим объектом. Теория понятий будет полной, если каждая предметная область имеет свою понятийную модель, и каждая понятийная модель имеет свою предметную область, возможно, воображаемую.

По этой причине будем различать семантические и синтаксические понятийные модели, а также их теории. Семантические понятийные модели опишем формальной теорией, называемой языком понятий, а синтаксические понятийные модели формализуем в рамках аксиоматической теории, называемой исчислением понятий.

5.1. ОБЩИЕ ОПРЕДЕЛЕНИЯ

Теория понятий наследуется от теории множеств [7], которая, в свою очередь, включает в себя логику первого порядка [10, с. 41].

Определение 1. Алфавит Ω – это конечное упорядоченное множество без повторения элементов, состоящее из цифр и букв. ♦

Определение 2. Имя (строка) – это конечное упорядоченное множество с повторением элементов, принадлежащих алфавиту Ω. ♦

Определение 3. Пусть $A1$, $A2$, B, Dc и Dac – имена, тогда:

– $\{B, A2, A1, Dc, Dac\}$ – неупорядоченное множество без повторения элементов;

– $\langle B, A2, A1, A1, Dc, Dc, Dac\rangle$ – неупорядоченное множество с повторением элементов;

– $(B, A2, A1, Dc, A1, Dac, Dc)$ – упорядоченное множество с повторением элементов;

– $[A1, A1, A2, B, Dac, Dc, Dc]$ – упорядоченное множество с повторением и перечислением элементов в лексикографическом порядке, задаваемом алфавитом Ω;

– $Efg : \{B, A2, A1, Dc, Dac\}$ – именованное множество, где Efg – имя множества;

– $\{X \mid Y\}$ – множество, заданное порождающей процедурой, где X – конструктор элементов, а Y – условие включения элемента в множество. ♦

Определение 4. Пусть A – конечное множество имен проблемных областей, $A : \{Aj \mid j = 1, 2, ...\}$, где имя Aj называется аспектом предметной области или просто аспектом (англ. aspect). ♦

Определение 5. Пусть B – множество видов понятий (абстракций, используемых для их образования), $B : \{C, T, A, G\}$, где C обозначает понятие-знак (абстракция идентификации, англ. identification, comprehension), T – понятие-тип (абстракция типизации, англ. typification), A – понятие-ассоциацию (абстракция ассоциации, англ. association) и G – понятие-обобщение (абстракция обобщения, англ. generalization). ♦

Определение 6. Формальное понятие или просто понятие – это множество $N : (b, I, E)$, у которого:

– N – имя понятия, $N = a•c$;
– a – имя аспекта, $a = Aspect(N) \in A$;
– c – имя концепта, $c = Concept(N)$;
– b – абстракция, $b = Abs(N) \in B$;
– I – интенсионал, $I = Int(N) = [Ij \mid j = 1, 2, ..., D]$;
– Ij – атрибут, $Ij = Atr(N, j)$, $j = 1, 2, ..., D$;
– D – степень, $D = Degree(N)$;
– E – экстенсионал, $E = Ext(N) = \{Ek \mid k = 1, 2, ..., P\}$;
– Ek – сущность понятия, $Ek = Ent(N, k)$, $k = 1, 2, ..., P$;
– Ekj – сущность атрибута Ij сущности Ek, $Ekj = Ek[Ij]$;
– P – объем, $P = Scope(N)$;
– S – схема, $S = Schema(N) = (b, I)$,

где знак = обозначает равенство чисел, строк или множеств соответственно; • – разделитель аспекта и концепта; $\in$ – отношение принадлежности элемента множеству. ♦

Определение 7. Понятие $N1 : (B1, I1, E1)$ и понятие $N2 : (B2, I2, E2)$ равны тогда и только тогда, когда $B1 = B2$, $I1 = I2$, $E1 = E2$. ♦

Из определения 7 следует, что проверку понятий на равенство следует производить без учета имен понятий (с точностью до имен понятий).

Определение 8. Пусть $M : \{Nj : (Bj, Ij, Ej) \mid j = 1, 2, ..., P\}$ – множество понятий предметной области R, тогда M – понятийная модель предметной области R мощности P, а $S : \{Schema(Nj) \mid j = 1, 2, ..., P\}$ – ее понятийная структура. ♦

5.2. СЕМАНТИЧЕСКАЯ ТЕОРИЯ ПОНЯТИЙ

В семантической теории понятий изучаются понятия и операции над ними. Семантическая теория строится как формальный язык, предназначенный для первичного описания предметных областей в процессе понятийного анализа. Результатом понятийного анализа является именование понятий и выражение способов их образования путем задания операций над понятиями.

5.2.1. ОПЕРАЦИИ НАД ПОНЯТИЯМИ

Определение 9. Свободная переменная N обозначает понятие-знак $N : (C, (N), \{N\})$, которое имеет интенсионал и экстенсионал, определяемые именем N. ♦

Определение 10. Обобщение понятий $N : \{N1, ..., ND\}$ – это понятие $N : (G, I, E) = \oplus(N, N1, ..., ND)$ такое, что $I = [N1, ..., ND]$, $E = Ext(N1) \cup ... \cup Ext(ND)$, где D – степень понятия-обобщения, $D > 1$; $\oplus$ – операция обобщения; $\cup$ – объединение множеств. ♦

Определение 11. Ассоциация понятий $N : \langle N1, \ldots, ND\rangle$ с селектором σ – это понятие $N : (A, I, E) = \otimes(N, N1, \ldots, ND, \sigma)$ такое, что $I = [N1, \ldots, ND]$, $E \subseteq Ext(N1) \times ... \times Ext(ND)$, где D – степень понятия-ассоциации, $D > 1$; $\otimes$ – операция ассоциации, $\subseteq$ – отношение нестрогого включения множеств, $\times$ – декартово произведение множеств, σ – имя множества E. ♦

Операция обобщения не зависит от порядка операндов и их повторения, так как операнды задаются неупорядоченным множеством без повторения элементов, а операция ассоциации зависит от повторения операндов, так как операнды образуют неупорядоченное множество с повторением элементов (см. определение 3).

Интенсионалы понятий являются множествами с повторением элементов, которые перечислены в лексикографическом порядке. Последнее необходимо для повышения эффективности сравнения понятий при практической реализации понятийной модели (см. определение 7).

Использование имени экстенсионала (селектора) необходимо потому, что могут существовать понятия, заданные на одном и том же множестве атрибутов, но с разными экстенсионалами. В этом случае селектор позволяет различать (сравнивать) такие понятия и рассматривается как некоторое характеристическое условие включения сущностей из декартова произведения атрибутов в экстенсионал понятия.

5.2.2. ФОРМУЛЫ

Определение 12. Формула на языке понятий задается следующими рекурсивными правилами:

– свободная переменная N (имя) является формулой N;

– если $F1, ..., FD$ – формулы, а N – имя, то $\oplus(N, F1, ..., FD)$ – формула;

– если $F1, ..., FD$ – формулы, N – имя и σ – селектор (имя), то $\otimes(N, F1, ..., FD, \sigma)$ – формула. ♦

Пример 13. Часть понятий понятийной диаграммы на рис. 1 выражаются следующими формулами:

– элементарные формулы – это 1, 2, 3;

– формула понятия «Разряд» – $\oplus$(Разряд, 1, 2, 3);

– формула понятия «Инженер» – $\otimes$(Инженер, $F1$, $F2$, Инженеры), где $F1$ и $F2$ – формулы, выражающие понятия «Специальность» и «ФИО», а Инженеры – селектор, который именует (характеризует) экстенсионал понятия «Инженер»;

– формула понятия «Сотрудник» – $\oplus$(Сотрудник, $F3$, $F4$), где $F3$ и $F4$ – формулы, выражающие понятия «Инженер» и «Рабочий». ♦

5.2.3. ОТНОШЕНИЕ ЭКВИВАЛЕНТНОСТИ

Определение 13. Две формулы $F1$ и $F2$ эквивалентны, если они выражают одно и то же понятие. ♦

Из определений 10–13 непосредственно следует то, что в семантической теории понятий справедливы следующие эквивалентности:

(1) $\oplus(N, \ldots, Fj, \ldots, Fj, \ldots) \Leftrightarrow \oplus(N, \ldots, Fj, \ldots)$;

(2) $\oplus(N, \ldots, Fj, \ldots, Fk, \ldots) \Leftrightarrow \oplus(N, \ldots, Fk, \ldots, Fj, \ldots)$;

(3) $\otimes(N, \ldots, Fj, \ldots, Fk, \ldots, \sigma) \Leftrightarrow \otimes(N, \ldots, Fk, \ldots, Fj, \ldots, \sigma)$,

где N – имя, Fj и Fk – формулы, σ – селектор, $\Leftrightarrow$ – отношение эквивалентности формул.

Определение 14. Формула находится в канонической форме, если она преобразована с помощью эквивалентностей вида (1) – удаление лишних подформул, и эквивалентностей вида (2) и (3) – перестановки подформул в лексикографическом порядке. ♦

***Утверждение 1**. Понятия равны тогда и только тогда, когда их канонические формы равны с точностью до имен понятий.*

Доказательство. Пусть понятия $N1$ и $N2$ равны (определение 7). Тогда они могут быть выражены эквивалентными формулами (определение 12), канонические формы которых равны с точностью до имен понятий (определения 7 и 14).

Верно и обратное. Пусть $F1$ и $F2$ – канонические формы, равные с точностью до имен понятий. Эти формы являются эквивалентными формулами (определения 7 и 14). Следовательно, выражаемые ими понятия равны (определение 13). ♦

Утверждение 1 позволяет выявить и удалить из понятийной модели равные понятия. Следует обратить внимание на игнорирование имен понятий при их сравнении, кроме имен понятий-знаков – первичных понятий предметной области. Последнее означает, что понятия сравниваются структурно, т.е. по порядку их образования из других понятий и с учетом используемых для этого абстракций.

Определение 15. Понятийная модель называется приведенной, если все ее формулы находятся в канонической форме и в модели нет формул равных понятий. ♦

Очевидно, модель в приведенной форме эквивалентна исходной понятийной модели.

Пример 14. Пусть понятийная модель задана следующими формулами:

$\oplus(A, C, \otimes(B, D, C, \sigma), C)$;

$\oplus(V, \otimes(Y, X, D, \delta), C)$;

$\oplus(X, C, \otimes(Y, D, C, \sigma))$,

где свободные переменные обозначают первичные (не раскрываемые) понятия предметной области.

Используя определения 13 и 14 находим канонические формы этих формул:

$\oplus(A, \otimes(B, C, D, \sigma), C)$;

$\oplus(V, \otimes(Y, D, X, \delta), C)$;

$\oplus(X, \otimes(Y, C, D, \sigma), C)$.

Из утверждения 1 следует, что понятия A и X равны. Оставляем формулу понятия A и удаляем формулу понятия X, а все вхождения имени X заменяем именем A. В итоге имеем приведенную модель:

$\oplus(A, \otimes(B, C, D, \sigma), C)$;

$\oplus(V, \otimes(Y, D, A, \delta), C)$. ♦

Заметим, что свободные переменные уникальны и выражают по определению 9 различные понятия.

5.2.4. СЕМАНТИЧЕСКАЯ РАЗРЕШИМОСТЬ

Важным свойством любой формальной теории является ее разрешимость (англ. decidability), заключающаяся в возможности определить, принадлежит ли теории некоторая формула или нет.

Для доказательства разрешимости семантической теории понятий построим формальную грамматику, порождающую формулы языка понятий (см. определение 12).

Определение 16. Формальная грамматика языка понятий – это множество $G : (V, W, P, F)$, у которого:

– $V : (Nj, \sigma j, \oplus, \otimes, (,), ; \mid j = 1, 2, ...)$ – терминальный алфавит грамматики;

– $W : (F, L)$ – нетерминальный алфавит, где нетерминальный символ F означает «Первая формула списка», а нетерминальный символ L – «Последняя формула списка»;

– $P : \{F \Rightarrow Nj,\ \ F \Rightarrow \oplus(Nj; F; L),\ \ F \Rightarrow \otimes(Nj; F; L; \sigma j),\ \ L \Rightarrow F;L,\ L \Rightarrow F\}$ – продукции грамматики;

– F – начальный нетерминальный символ, $F \in W$;

где Nj – понятийные переменные, σj – селекторы, ; – разделитель формул в списке (вместо запятой), $\Rightarrow$ – разделитель правых и левых частей продукций, а также знак грамматического вывода. ♦

Язык, порожденный грамматикой из определения 16, является бесконечным контекстно-свободным формальным языком [9, с. 11].

Пример 15. Выведем формулы из примера 13 в пустом аспекте, когда разделитель • не используется для построения имен понятий.

Из начального символа F и продукции $F \Rightarrow Nj$ получаем терминальные понятия:

$F \Rightarrow Nj \Rightarrow 1$, $F \Rightarrow Nj \Rightarrow 2$, $F \Rightarrow Nj \Rightarrow 3$.

Далее выведем формулу ⊕(Разряд, 1, 2, 3):

$F \Rightarrow \oplus(Nj; F; L) \Rightarrow \oplus(Nj; F; F; L) \Rightarrow \oplus(Nj; F; F; F) \Rightarrow$

$\Rightarrow \oplus(\text{Разряд}; F; F; F) \Rightarrow \oplus(\text{Разряд}; 1; F; F) \Rightarrow$

$\Rightarrow \oplus(\text{Разряд}; 1; 2; F) \Rightarrow \oplus(\text{Разряд}; 1; 2; 3)$.

Аналогично выводятся другие формулы. ♦

Для грамматики из определения 16 существует эффективная процедура, с помощью которой можно проверить, принадлежит ли формула языку, порождаемому этой грамматикой, или нет [9, с. 21]. Однако выводимость формулы языка понятий оказывается недостаточной для проверки принадлежности этой формулы семантической теории.

Каждое формальное понятие, выраженное на языке семантической теории, должно иметь интерпретацию в предметной области. При понятийном мышлении недопустимы понятия, которые прямо или косвенно определяются через самих себя, а также не имеющие принадлежащих им сущностей. Следовательно, понятия с пустым экстенсионалом и понятия, выраженные через самих себя, не имеют интерпретации в любой предметной области. Для семантической теории это означает, что в формулах языка понятий не должно быть понятийных циклов, а также не должно быть формул пустых понятий (понятий с пустым экстенсионалом).

Определение 17. Формальная теория понятий разрешима, если существует эффективная процедура, распознающая понятийные модели с понятийными циклами и пустыми понятиями. ♦

Утверждение 2. *Язык понятий разрешим*.

Доказательство. Пусть M – семантическая понятийная модель, $M : \{Fj \mid j = 1, 2, ..., P\}$, где Fj – формула, выражающая понятие с именем Nj, P – мощность модели.

Рассмотрим распознавание формул модели на пустоту выражаемых ими понятий. Из определения 9 следует, что понятия-знаки не пусты. Образование понятий-обобщений также не приводит к определению пустых понятий, если обобщаемые понятия не пусты (определение 10). Однако при образовании понятий-ассоциаций возможно появление пустых понятий, если именованный экстенсионал образуемого понятий пуст (определение 11). Однако появление таких понятий в процессе понятийного анализа считается невозможным, так как понятия с пустым экстенсионалом не могут мыслится. Следовательно, формулы языка понятий не выражают пустых понятий.

Для нахождения понятийных циклов в модели применим следующую процедуру. В начале удаляются все формулы из M, которые выражают равные понятия (см. утверждение 1).

Для каждого оставшегося понятия Nj, выраженного формулой Fj, ищется вхождение этого понятия в формулу Fj. Если вхождение найдено, то понятийный цикл обнаружен и процедура завершается. В противном случае понятие Nj ищется в других формулах Fk, имена понятий которых Nk встречаются в формуле Fj. Если все формулы просмотрены и имя понятия Nj не обнаружено, то понятийные циклы с понятием Nj отсутствуют.

Если все понятия просмотрены и понятийные циклы не обнаружены, то процедура завершается. Число понятий P конечно, как и конечна длина формул модели. Следовательно, найдена эффективная процедура определения циклов в понятийных моделях.

В итоге разрешимость языка понятий доказана. ♦

Интерпретируемость понятийный модели может быть также проверена по понятийной диаграмме предметной области (см. рис. 1). В этом случае на диаграмме, рассматриваемой как ориентированный граф, не должно быть циклов.

Следует обратить внимание на то, что разрешимость языка понятий в теории понятий – это то же самое, что регулярность множеств в теории множеств (см. аксиому основания [7, с. 17]).

5.2.5. НЕПРОТИВОРЕЧИВОСТЬ И ПОЛНОТА

Другими важными свойством любой формальной теории является ее непротиворечивость (англ. consistent) и полнота (англ. completeness). Для доказательства полноты и непротиворечивости синтаксической теории понятий нам потребуется утверждение о полноте и непротиворечивости семантической теории.

Противоречием (англ. contradiction) в теории понятий будем называть ситуацию, при которой одна и та же сущность мыслится (описывается) как принадлежащая, так и не принадлежащая одному и тому же понятию в одно и то же время. Следует заметить, что логическое противоречие является частным случаем понятийного. В логике противоречием называется ситуация, при которой некоторое высказывание принадлежит как множеству истинных высказываний, так и множеству ложных.

Утверждение 3. *Язык понятий непротиворечив.*

Доказательство. Язык понятий непротиворечив по своему построению, так как нет никаких выразительных средств для выражения противоречий, нет ни одного способа указать непринадлежность какой-либо сущности какому-либо понятию, нет операции отрицания. Это служит основанием для утверждения о непротиворечивости языка понятий. ♦

Вывод о семантической полноте любой теории можно сделать только относительно класса моделей, для описания которых теория предназначена. Семантическая теория понятий полна, если любая предметная область может быть описана в этой теории, т.е. имеет описывающую ее формулу на языке понятий.

Утверждение 4. *Язык понятий является полной теорией.*

Доказательство. Пусть имеется предметная область с принадлежащими ей множеством первичных сущностей – понятий-знаков $C = \{c1, c2, \ldots\}$. Будем считать множество C разрешимым, так как оно как-то выделено в предметной области и предъявлено в процессе понятийного анализа.

Как доказано в утверждении 1 и показано на примере 14, выводимые в семантической теории понятия сравнимы с точностью до имен понятий-знаков. Используя процедуру генерации типов из теории типов [2] определим множество сущностей предметной области E на основе первичных понятий-знаков C так:

– если $\varphi \in C$, то φ – сущность (выделение);
– если φ – сущность, то $\{\varphi\}$ – сущность (группировка);
– если φ и ψ – сущности, то $\varphi \bullet \psi$ – сущность (соединение).

Множество E является счетным и содержит все мыслимые сущности предметной области, т.к. ему принадлежат все строки конечной длины $\{c1, c2, \ldots, c1\bullet c1, c1\bullet c2, \ldots, c2\bullet c1, c2\bullet c2, \ldots, c1\bullet c1\bullet c1, c1\bullet c1\bullet c2, \ldots, c1\bullet c2\bullet c1, \ldots\}$ со всеми вариантами правильно расставленных скобок, по-разному структурирующих исходную строку. Так как множество C разрешимо, то разрешимым будет и множество E.

Для конструирования всех мыслимых понятий образуем булеан $B(E)$ – множество всех подмножеств множеств E. Существование булеана следует из аксиомы степени теории множеств [7, с. 17].

Каждый элемент $B(E)$ является объединением конечного числа элементов из множества E. Так как разрешающая процедура для элементов множества E существует, то существует и разрешающая процедура для элементов множества $B(E)$, так как объединение конечного числа множеств разрешимо.

Таким образом, удалось показать, что с помощью выделения (идентификации), соединения и группировки (ассоциации), а также объединения (обобщения) сущностей предметной области могут быть конструктивно построены все мыслимые понятия. Осталось только выбрать те понятия, которые интерпретируются в предметной области, и включить их в понятийную модель (абстракция ограничения, игнорирования, англ. restriction).

Очевидно, верно и обратное. Для любого множества понятий M, выведенных в семантической теории из C, в $B(E)$ содержатся прототипы всех понятий из M. ♦

Аналогично доказательству утверждения 4 выполнено обоснование полноты логики высказываний через функциональную полноту логических операций [10, с. 18], т.е. путем доказательства возможности выразить любое сложное высказывание из простых с помощью логических связок.

Доказательство утверждения 4 также близко технологии концептуализации предметных областей [3, с. 47], где основными шагами концептуального проектирования являются:

1) выделение базисных концептов X (идентификация);
2) построение булеана $B(X)$ (синтез обобщений);
3) построение декартиана $B(X \times B(X))$ (синтез ассоциаций);
4) выделение из булеана и декартиана множества концептов X, имеющих интерпретацию в предметной области (ограничение).
5) если концептов X недостаточно, то переход на шаг 2.

В итоге имеем, что предметом теории понятий являются понятийные модели, выводимые в этой теории, а ее объектом – предметные области, рассматриваемые как совокупности ментальных представлений и иерархии понятий, абстрагированных от них.

5.3. СИНТАКСИЧЕСКАЯ ТЕОРИЯ ПОНЯТИЙ

Семантическая теория понятий ответственна за первичное описание предметной области в виде ее понятийной структуры, выраженной на языке понятий.

Синтаксическая теория понятий, также называемая исчислением понятий, используется для вывода понятийных моделей, включающих в себя не только понятийную структуру предметной области, но и перечисление сущностей из экстенсионалов понятий.

Исчисление определяется алфавитом, аксиомами и правилами вывода.

5.3.1. АЛФАВИТ

Алфавит исчисления понятий включает:

– имена понятий (аспектов, селекторов), состоящие из букв и цифр (см. определение 2);

– разделитель имен • (точка), отношение именования : (двоеточие), операции над понятиями $\oplus$ (обобщение) и $\otimes$ (ассоциация);

– символы логического вывода, логические связки и кванторы: $\vDash$, $\vdash$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$ [10, с. 447];

– предикаты, операции и отношения над множествами: $\varnothing$, $\in$, $\notin$, $-$, $\cup$, $\cap$, $\times$, $\subset$, $\supset$, $\subseteq$, $\supseteq$, $<$, $>$, $=$, $\neq$ [7, с. 49];

– скобки (круглые, квадратные, угловые, фигурные): (,), [,], $\langle$, $\rangle$, {, } (см. определение 3).

5.3.2. АКСИОМЫ

Аксиомы исчисления понятий – это аксиомы логики первого порядка [10, с. 62], аксиомы теории множеств [7, с. 15], а также следующие четыре аксиомы.

Аксиома 1 (ограничение). Существует пустое понятие $\varepsilon : (C, \varnothing, \varnothing)$, не принадлежащее интенсионалу и экстенсионалу любого понятия:

$$\exists \varepsilon : (C, \varnothing, \varnothing)\ \forall N\ (\varepsilon \notin Int(N) \wedge \varepsilon \notin Ext(N)).\ \blacklozenge$$

Аксиома 2 (идентификация). Существует понятие-знак $N : (C, (N), \{N\})$ такое, что его интенсионал и экстенсионал состоит только из этого понятия:

$$\exists N : (C, (N), \{N\}).\ \blacklozenge$$

Аксиома 3 (обобщение). Для любых $D > 1$ понятий $N : \{N1, \ldots, ND\}$ существует понятие-обобщение $\oplus(N, N1, \ldots, ND)$:

$$\forall N : \{N1, \ldots, ND\}$$

$$\exists N : (G, [N1, \ldots, ND], Ext(N1) \cup \ldots \cup Ext(ND)).\ \blacklozenge$$

Аксиома 4 (ассоциация). Для любых $D > 1$ понятий $N : \langle N1, \ldots, ND\rangle$ и селектора σ существует понятие-ассоциация $\otimes(N, N1, \ldots, ND, \sigma)$:

$$\forall N : \langle N1, \ldots, ND\rangle$$

$$\exists N : (A, [N1, \ldots, ND],$$

$$\{Ej : (E1, \ldots, ED) \mid E1 \in Ext(N1) \wedge \ldots \wedge ED \in Ext(ND) \wedge \sigma(Ej)\}),$$

где $\sigma(Ej)$ – логическая формула принадлежности сущности Ej экстенсионалу понятия-ассоциации. ♦

Пустое понятие ε используется для обозначения сущностей, отсутствующих в модели. Очевидно, такое понятие не может использоваться при выводе других понятий, так как обозначаемые им сущности исключаются из рассмотрения. По своей сути пустое понятие реализует абстракцию ограничения.

Как и в семантической теории, операции $\oplus$ и $\otimes$ обозначают обобщение и ассоциацию понятий, а селектор σ – это логическая формула со свободным вхождением одной переменной, которая используется для разрешения сущностей из экстенсионалов понятий-ассоциаций.

Возможность построения понятий-обобщений с помощью аксиомы 3 следует из аксиомы объемности теории множеств, а построение понятий-ассоциаций с помощью аксиомы 4 – из аксиомы выделения [7, с. 16].

5.3.3. ПРАВИЛА ВЫВОДА

Синтаксическая модель предметной области состоит из понятий (определение 8), каждое из которых должно быть выведено с помощью аксиом исчисления понятий. Исчисление понятий, как и теория множеств, наследует правила вывода из логики первого порядка: правило отделения (modus ponens) и правило обобщения [10, с. 62].

Определение 18. Вывод понятийной модели $M : \{N1, \ldots, NL\}$ – это упорядоченное множество понятий ($N1$, ..., Nj, Nk, ..., NL) таких, что предыдущие понятия $N1$, ..., Nj (базовая модель) используются для вывода следующих понятий Nk, ..., NL (наследуемой модели) с помощью аксиом исчисления понятий: $\vDash (N1, \ldots, NL)$ – вывод модели M; $\{N1, ..., Nj\} \vdash (N1, ..., NL)$ – вывод наследуемой модели M из базовой модели $\{N1, ..., Nj\}$, где $\vDash$ ($\vdash$) – символ абсолютной (относительной) выводимости. ♦

Пример 16. Выведем фрагмент понятийной модели из примера 13:

– $\vDash 1 : (C, [1], \{1\})$, $\vDash 2 : (C, [2], \{2\})$, $\vDash 3 : (C, [3], \{3\})$ (аксиома 2);

– 1, 2, 3 $\vdash$ Разряд : (G, [1, 2, 3], {1, 2, 3}) (аксиома 3);

– ФИО, Специальность $\vdash$ Инженер : (A, [Специальность, ФИО], {X | Инженеры(X)}) (аксиома 4).

В последнем выводе селектор «Инженеры» характеризует экстенсионал понятия «Инженер» путем разрешения соединения сущностей понятий «Специальность» и «ФИО», например, так:

Инженеры(X) $\leftrightarrow$ X[Специальность] = Металлург $\wedge$
$\wedge$ X[ФИО] = Иванов Иван Иванович $\vee$
$\vee$ X[Специальность] = Программист $\vee$
$\vee$ X[ФИО] = Петров Петр Петрович,

где X – свободная переменная, обозначающая сущность понятия «Инженер».

Селектор «Инженеры» включает в экстенсионал понятия «Инженер» сущности (Металлург, Иванов Иван Иванович), (Программист, *) и (*, Петров Петр Петрович), где * – произвольные вхождения сущностей «ФИО» и «Специальность». ♦

Заметим, что правила вывода, полученные из понятийной модели, являются правилами вывода в предметной области и используются для построения предметной формальной теории (см. пример 10). Здесь же используются правила вывода исчисления понятий, с помощью которых строится сама понятийная модель.

5.3.4. ТИПЫ И КЛАССЫ

Теорема 1 (типизация). *Для множества понятий-знаков* $T : \{Nj \mid j = 1, \ldots, D\}$ *существует понятие-тип* $T : (G, I, E)$ *такое, что* $I = [N1, \ldots, ND]$, $E = \{N1, \ldots, ND\}$.

Доказательство. Существование понятий-знаков следует из аксиомы 2, а существование искомого понятия-типа – из аксиомы 3. ♦

Теорема 1 позволяет вывести все конечные понятия-типы, используемый на практике, как-то: число с ограниченной разрядностью представления, символ, строка и т.п.

Однако в исчислении понятий справедлива и теорема, в которой доказывается существование понятия-класса – абстрактного понятия-типа со счетным объемом.

Теорема 2 (классификация). *Для множества понятий-знаков* $C : \{Nj \mid j = 1, 2, \ldots\}$ *существует понятие-класс* $C : (G, I, E)$ *такое, что* $I = [N1, N2, N3, \ldots]$, $E = \{N1, N2, N3, \ldots\}$.

Доказательство. Существование счетного числа понятий-знаков следует из аксиомы 2 и аксиомы бесконечности теории множеств [7, с. 16].

Выведем семейство понятий-типов $\{Cj \mid j = 1, 2, \ldots\}$, используя математическую индукцию и аксиому 3, добавляя по одному понятию-знаку на каждом шаге:

$$\{\oplus(Cj, Ck, Nk) \mid C1{=}N1 \wedge j = k{+}1 \wedge k = 1, 2, \ldots\}.$$

В пределе (при индуктивном переходе) получаем существование искомого понятия-класса C счетного объема. ♦

5.3.5. АКТУАЛИЗАЦИЯ МОДЕЛЕЙ

Теорема 3 (насыщение и усечение). *Для любого понятия-ассоциации* $N : (A, I, E)$ *существует насыщенное (усеченное) понятие-ассоциация* $Nx : (A, I, Ex)$ *такое, что* $E \subseteq Ex$ ($E \supseteq Ex$).

Доказательство. Из аксиомы 4 следует, что понятие-ассоциация может быть выведено с тем же интенсионалом, но с расширенным (усеченным) экстенсионалом, поскольку экстенсионал – это любое подмножество декартова произведения атрибутов понятия. Для этого случая существует селектор $\sigma \vee \delta$ ($\sigma \wedge \gamma$), где δ (γ) – логическая формула, которая выбирает новые (исключает имеющиеся) сущности экстенсионала Ex. ♦

Из теоремы 3 следует, что сущность предметной области является полноправным понятием. Действительно, усекая экстенсионал понятия до состояния, когда в нем остается всего одна сущность Ex, имеем понятие-сущность $Nx : (A, I, \{Ex\})$.

Теорема 3 позволяет актуализировать понятийную модель с помощью таких операций, как удаление и добавление сущностей в экстенсионалы понятий. Кроме того, теорема 3 позволяет создавать новые понятия с помощью запросов, выбирающих сущности из экстенсионала одного понятия-ассоциации для включения их в экстенсионал подобного понятия.

5.3.6. РАВЕНСТВО СУЩНОСТЕЙ

Определение 19. Сущности X и Y равны, если и только если равны сущности их атрибутов, $X = Y \leftrightarrow \forall j\ (X[j] = Y[j])$. ♦

Для обеспечения выполнимости определения 19 и повышения эффективности реализации понятийных моделей:

– интенсионалы понятий отсортированы в лексикографическом порядке (определение 6);

– для отсутствующего атрибута у сущности функтор возвращает пустое понятие (определение 22).

Определение 20. Пусть X – сущность некоторого понятия. Тогда $Name(X)$ – функтор, возвращающий имя сущности X, определяемый рекурсивно:

– если X – сущность понятия-знака, понятия-типа или понятия-класса, то $Name(X) = X$;

– если X – сущность понятия-ассоциации, то $Name(X) = \{Name(X[1])\bullet\ldots\bullet Name(X[D])\}$, где $D = Degree(N)$ – степень понятия сущности;

– если X – сущность понятия-обобщения, то имя X равно имени сущности понятия-знака, понятия-типа или понятия-ассоциации, которому принадлежит X. ♦

Из определения 20 видно, что имя сущности задается атрибутами понятия-ассоциации или понятия-знака, поскольку интенсионал понятия-обобщения не описывает структуру понятия, а задает лишь множество обобщаемых понятий (аксиома 3).

Пример 17. Именем некоторой сущности понятия-ассоциации «Участник» (см. рис. 1) может быть {В•{Металлург•3•Иванов Иван Иванович}}, где В – имя сущности понятия-типа «Бригада», Металлург – имя сущности понятия-типа «Профессия», 3 – имя сущности понятия-типа «Разряд», Иванов Иван Иванович – имя сущности понятия-типа ФИО. В свою очередь {Металлург•3•Иванов Иван Иванович} – имя сущности понятия-ассоциации «Рабочий». Порядок перечисления имен – как в интенсионалах соответствующих понятий-ассоциаций. ♦

Из аксиомы 4 следует, что существуют понятия с одинаковым интенсионалом и различными экстенсионалами. Тогда у таких понятий могут быть равные сущности.

Теорема 4 (равенство сущностей). *Сущности равны тогда и только тогда, когда они имеют одинаковые имена:*

$$X1 = X2 \leftrightarrow Name(X1) = Name(X2).$$

Доказательство. Из определения 20 следует, что имена сущностей состоят из имен понятий-знаков, которые уникальны (аксиома 2) и могут быть пронумерованы, а благодаря использованию разделителя имен и фигурных скобок неоднозначностей в структурировании составных имен не возникает.

Тогда из определения 19, аксиом 3 и 4 следует, что имена сущностей можно сравнивать, а равные сущности имеют одинаковые имена. Верно и обратное: если сущности имеют одинаковые имена, то они равны. ♦

Теорема 4 позволяет реализовать эффективный поиск равных сущностей в базе знаний. Для такого поиска достаточно сравнить имена сущностей без доступа к их атрибутам. Более того, если пронумеровать понятия-знаки предметной области, то сравнение имен сводится к сравнению последовательностей натуральных чисел, разделенных точками и структурированных фигурными скобками.

Пример 18. Пусть понятия-знаки некоторой предметной области – это *Abc*, 12,3, 0x2359AD, 73, 15.05.2021. Пронумеруем их в лексикографическом порядке: 0x2359AD ~ 1, 12,3 ~ 2, 15.05.2021 ~ 3, 73 ~ 4, *Abc* ~ 5, где ~ – отношение соответствия. Пусть также образованы понятия с сущностями, имена которых и их числовые эквиваленты такие:

73 ~ 4,
{15.05.2021•0x2359AD} ~ {3•1},
{15.05.2021•0x2359AD•73} ~ {3•1•4},
{*Abc*•{12,3•15.05.2021}} ~ {5•{2•3}},
{*Abc*•0x2359AD•{12,3•73}} ~ {5•1•{2•4}}.

Тогда сущности {3•1} и {3•1} равны, сущность {3•1•4} больше сущности {3•1}, а сущности {5•{2•5}} и {5•1•{2•4}} несравнимы, так как сущность {2•3} не сравнима с сущностью 1. ♦

В прикладном плане отношение равенства сущностей проверяется сравнением числовых идентификаторов сущностей (см. пример 8) и, если идентификаторы не равны, то сравнением их имен.

5.3.7. СИГНАТУРА МОДЕЛИ

Селекторы, используемые в исчислении понятий, отличаются от селекторов языка понятий. В языке понятий селекторы – это имена подмножеств (определение 11), а в исчислении понятий – это логические формулы (аксиома 4).

Сигнатура понятийной модели – это множество предикатов, функторов и отношений, которые используются в логических формулах для выражения экстенсионалов, а также в запросах к базе знаний (см. пример 11).

Определение 21. Пусть N – понятие. Тогда $N(X)$ – предикат принадлежности сущности X экстенсионалу понятия N, $N(X) \leftrightarrow X \in Ext(N)$. ♦

Определение 22. Пусть X – сущность. Тогда $X[j]$ – функтор, возвращающий сущность (значение) j-го атрибута сущности X или пустое понятие ε, если у сущности X нет атрибута с номером (именем) j. ♦

Из практических соображений помимо функторов, извлекающих значения простых типов данных или идентификаторы сущностей атрибутов понятия, могут использоваться и другие функторы, общие для всех понятий, например: функторы наименования $Title(X)$, иконки $Icon(X)$ (см. пример 7), имени $Name(X)$, аспекта $Aspect(X)$, концепта $Concept(X)$, числа атрибутов $Degree(X)$ и т.д. (см. определение 6).

При создании исторических баз знаний отражение времени в понятийной модели может быть реализовано добавлением общих функторов времени, возвращающих время появления $Descent(X)$ и исчезновения $Ascent(X)$ сущности X.

Определение 23. Отношения понятийной модели – это отношения равенства = и порядка <, заданные на множестве натуральных чисел. ♦

Основанием для включения в сигнатуру понятийной модели только двух отношений и их достаточности для разрешения экстенсионалов понятий непосредственно следует из теоремы 4, где сравнение сущностей сведено к сравнению последовательностей натуральных чисел – в теоретическом плане, и сравнению значений встроенных типов данных, которые кодируются теми же последовательностями натуральных чисел, – в прикладном плане.

Однако из-за наличия фигурных скобок сущности оказались упорядоченными только частично (см. пример 18). Но и частичная упорядоченность – достаточно сильное свойство, повышающее эффективность обработки знаний.

В итоге имеем, что сигнатура понятийном модели состоит из одноместных предикатов, одноместных функторов, а также отношений равенства и порядка, заданных на множестве натуральных чисел.

Известно, что одноместное исчислении предикатов (англ. pure monadic predicate calculus) имеет много полезных свойств, делающих его похожим на исчисление высказываний [10, с. 24].

Но наличие функторов и отношений в сигнатуре модели не позволяет сделать заключение о том, что в исчислении понятий логические формулы являются формулами одноместного исчисления предикатов.

Теорема 5 (одноместное исчисление предикатов). *Логические формулы, используемые в исчислении понятий для выражения экстенсионалов, являются формулами одноместного исчисления предикатов.*

Доказательство. Все вхождения подформул $\psi(X, Y)$ с функторами и отношениями в логическую формулу φ могут иметь следующий вид (см. пример 11):

$X \div Y$; $X[j] \div Y$; $X \div Y[k]$; $X[j] \div Y[k]$,

где X и Y – сущности понятий, $X[j]$ и $Y[k]$ – функторы, возвращающий значение j-го атрибута сущности X и k-го атрибута сущности Y; $\div$ – отношения между сущностями.

Так как формула φ используется для разрешения экстенсионала понятия, то не менее чем одна переменная из двух в $\psi(X, Y)$ является связанной. Без потери общности выберем в φ подформулу $QY\ \psi(X, Y)$, где Q – связывающий квантор для Y. Пусть P – свободный предикатный символ, не входящий в формулу φ, и ϕ – формула, полученная из φ заменой вхождений $QY\ \psi(X, Y)$ предикатом $P(X)$. Тогда формулы φ и $\phi \wedge \forall X\ P(X) \leftrightarrow QY\ \psi(X, Y)$ эквивалентны.

Проделав аналогичные подстановки для всех вхождений $\psi(X, Y)$ в φ и добавив предикаты $P(X)$ в сигнатуру модели, получим формулу ϕ, не содержащую вхождений функторов и отношений, а составленную только из одноместных предикатов. Что требовалось доказать. ♦

Следует заметить, что в одноместном исчислении предикатов нет ограничений на мощность сигнатуры модели. Однако сама сигнатура значительно ограничивает выразительность логического языка (англ. expressivity of logic).

Для концептуальных 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 для разрешения множеств экземпляров к одноместным, связывая вторую переменную роли R экземплярами концепта C [13, с. 50, 150]:

$\forall R.C = \{a|\forall b\ R(a, b) \to C(b)\}$;

$\exists R.C = \{a|\exists b\ R(a, b) \wedge C(b)\}$.

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. Отсутствие в языке двуместных предметных предикатов (ролей) позволяет надеяться на большую эффективность обработки знаний в формализме понятийных моделей по сравнению с моделями дескрипционных логик.

5.3.8. ПРИВЕДЕННАЯ МОДЕЛЬ

Приведенная понятийная модель не содержит равных понятий (определение 15).

Теорема 6 (приведенная модель). *Для любой понятийной модели существует эквивалентная ей приведенная понятийная модель.*

Доказательство. Для семантической теории существует процедура приведения понятийных моделей (см. пример 14). В отличие от семантической теории в исчислении понятий сравнению подлежат не уникальные имена селекторов, а логические формулы, их выражающие.

Логические формулы в исчислении понятий выражаются в одноместном исчислении предикатов (теорема 5), которое разрешимо [10, с. 221]. Отсюда следует и разрешимость проблемы эквивалентности логических формул с одноместными предикатами.

Действительно, доказательство разрешимости формул в одноместном исчислении предикатов сводится к проверке выполнимости формулы на конечном множестве точек области определения.

Применив аналогичную процедуру для двух формул, заданных на общем множестве предикатов, можно сделать вывод об эквивалентности формул, если они выполнимы на одном и том же подмножестве точек области определения и только на нем.

Следовательно, существует процедура приведения понятийных моделей и в синтаксической теории. ♦

Для уменьшения избыточности данных, хранимых в базе знаний, и устранения потенциальных неоднозначностей, связанных с именованием одного и того понятия разными именам, при понятийном моделировании используются только приведенные модели.

5.3.9. СИНТАКСИЧЕСКАЯ РАЗРЕШИМОСТЬ

Исчисление понятий разрешимо, если существует эффективная процедура распознавания формул исчисления, содержащих понятийные циклы и пустые понятия (определение 17).

Теорема 7 (синтаксическая разрешимость). *Исчисление понятий разрешимо.*

Доказательство. Формулы, порождаемые исчислением понятий, отличаются от формул языка понятий только селекторами, которые в исчислении понятий являются логическими формулами (аксиома 4), а на языке понятий – именами множеств (определение 11).

Отсюда следует, что в исчислении понятий наличие в понятийной модели циклов проверяется той же процедурой, которая используется и в семантической теории (утверждение 2).

Теперь осталось доказать разрешимость в исчислении понятий проблемы пустоты экстенсионала. Ранее показано, что язык понятий разрешим (утверждение 7). Следовательно, доказательство разрешимости исчисления понятий сводится к доказательству выполнимости логических формул (англ. satisfiability), используемых для выражения экстенсионалов.

Логические формулы, используемые в исчислении понятий, принадлежат одноместному исчислению предикатов (теорема 5), которое разрешимо на счетных моделях [10, с. 221]. Это позволяет проверить селектор на выполнимость процедурой, аналогичной процедуре из доказательства теоремы 6. В этом случае выполнимость селектора хотя бы в одной точке гарантирует непустоту соответствующего понятия.

В итоге разрешимость исчисления понятий доказана. ♦

5.3.10. НЕПРОТИВОРЕЧИВОСТЬ И ПОЛНОТА

Теорема 8 (синтаксическая непротиворечивость). *Исчисление понятий непротиворечиво на конечных и счетных моделях.*

Доказательство. Язык понятий непротиворечив (утверждение 3). Чтобы доказать непротиворечивость исчисления понятий, достаточно доказать непротиворечивость теории множеств, формулы которой на языке одноместного исчисления предикатов используется для выражения экстенсионалов понятий (теорема 5). Доказано, что теория натуральных чисел непротиворечива [7, с. 27]. Тогда и исчисление понятий непротиворечиво на конечных и счетных моделях. ♦

Теорема 9 (синтаксическая полнота). *Исчисление понятий является полным на конечных моделях.*

Доказательство. Семантическая теория полна (утверждение 4). Чтобы доказать полноту исчисления понятий относительно семантической теории, достаточно доказать полноту теории множеств, которая используется в исчислении понятий для выражения экстенсионалов. Полнота теория множеств доказана только для конечных моделей, а на счетных моделях теория множеств существенно неполна [7, с. 27]. Следовательно, на счетных моделях неполным будет и исчисление понятий. ♦

6. 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:

– именование 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.

Формальная теория понятий состоит из языка понятий (семантическая теория) и исчисления понятий (синтаксическая теория). Язык понятий используется для описания предметных областей в процессе понятийного анализа. Результатом понятийного анализа является понятийная структура, содержащая интенсионалы понятий (множества понятий, использованных для их абстрагирования), но без детального описания экстенсионалов понятий (множеств сущностей, принадлежащих понятиям). Семантическая теория понятий оказалась разрешимой, непротиворечивой и полной.

Исчисление понятий используется для понятийного моделирования. Результатом понятийного моделирования является понятийная модель, состоящая из понятийной структуры, у которой экстенсионалы понятий выражены в виде формулы одноместного исчисления предикатов с отношениями равенства и порядка на множестве натуральных чисел. Синтаксическая теория понятий оказалась разрешимой и непротиворечивой на счетных и полной на конечных моделях. Благодаря понижению выразительности логического языка, используемого в исчислении понятий, снижена вычислительная сложность задач представления и обработки знаний по сравнению дескрипционными логиками.

Формальная теория понятий является формальной теорией понятийных моделей, а сами понятные модели – формальными теориями предметных областей. Это позволяет рассматривать понятийную модель как форму представления знаний о предметной области, а также осуществлять вывод предметных 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 реализуемой модели знаний, ее легкая актуализация и адаптация, не приводящая к возникновению скрытых противоречий, позволяет надеяться на широкое внедрение понятийного моделирования в практику создания и использования современных интеллектуальных информационных систем.

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

Еще одним направлением развития понятийного анализа и понятийного моделирования видится разработка методов и средств 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 экстенсионалов существующих понятий или путем образования новых. К этому направлению относится и большой класс задач по распознаванию образов.

Литература

1. ВЫХОВАНЕЦ В.С. *О существенной неполноте формального метода* // International Interdisciplinary Conference «Philosophy, Mathematics, Linguistics: Aspects of Interaction». – СПб.: Международный математический институт им. Л. Эйлера, 2009. – С. 79-87.
2. ВЫХОВАНЕЦ В.С., КРЫЖАНОВСКАЯ А.В. *Совмещенные сети управления и данных* // Управление большими системами. – 2015. – Вып. 58. – С. 41–66.
3. НИКАНОРОВ С.П. *Концептуализация предметных областей*. – М.: Концепт, 2009. – 268 с.
4. BATINI C., CERI S., NAVATHE S.B. *Conceptual Database Design: An Entity-Relationship Approach*. – Redwood City, CA: The Benjamin/Cummings Publishing Company, 1992. – 496 p.
5. BRACHMAN R.J., LEVESQUE H.J. *Knowledge Representation and Reasoning*. – San Francisco, CA: Morgan Kaufmann, 2004. – 381 p.
6. ELMASRI R., NAVATHE S.B. *Fundamentals of Database Systems*. Hoboken. – NJ: Pearson, 2016. – 1273 p.
7. HAO W., NAUGHTON R. *Les Systemes Axiomatiques de la Theorie des Ensembles*. – Paris, France: Gauthier-Villars, 1953. – 54 p.
8. *ISO/IEC 9899:2018. – Information technology. – Programming language – C.* – URL: https://www.iso.org/standard/74528.html.

9. KALLMEYER L. *Parsing Beyond Context-Free Grammars*. – New York, NY: Springer, 2010. – 260 p.
10. MENDELSON E. *Introduction to Mathematical Logic.* – New York, NY: CRC Press, 2010. – 496 p.
11. OLIVÉ A. *Conceptual Modeling of Information Systems*. – New York, NY: Springer, 2007. – 471 p.
12. TALENS G., BOULANGER D., SÉGURAN M. *Domain Ontologies Evolutions to Solve Semantic Conflicts* // Ontologies-Based Databases and Information Systems / Ed. M. Collard. – Berlin, Heidelberg: Springer, 2007. – P. 51–67.
13. *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.
14. *The Oxford Dictionary of Current English* / Ed. D. Thompson. – New York, NY: Oxford University Press, 1993. – 1091 p.
15. VYKHOVANETS V.S. *Large-Scale Information Systems based on Conceptual Models* // Proc. of 19th Int. Conf. on Management of Large-Scale System Development. – Moscow, Russia: V.A. Trapeznikov Institute of Control Sciences, October 1-3, 2019. – Режим доступа: https://ieeexplore.ieee.org/document/8911106.
16. ZHAO Y., ZHANG C., CAO L. *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*. – Hershey, New York: Information science reference, 2009. – P. 81–149.

THE NOTIONAL ANALISYS AND NOTIONAL MODELLING

Valeriy Vykhovanets, Institute of Control Sciences of RAS, Moscow, Dr. Sc., associative professor (valery@vykhovanets.ru).

Abstract: The article describes notional models that are based on primary mental abstractions: identification, generalization and association. In the process of the notional analysis, a notional model is constructed, which consists of a notional structure and contents of its notions. The notional structure defines each notion as the result of generalization or association of the other notions. The content of the notion is described by an enumerable or solvable set consisting of subject domain entities. The main difference between the notional model and the other knowledge models is the refusal to describe the association of the notions in the form of a *relationship. In the notional model, the association are the same notion as the generalization, which makes it possible to form the other notions from the associations. All this makes the notional model semantically invariant, i.e. independent in its interpretation from the knowledge of the subject domain. Another difference between notional and conceptual models is the multi-aspect expression of the notions. In order to prove the proposed approach to the representation and processing of knowledge, a formal theory of notions is given. The semantic part of the theory (notional language) proves the methodology of the notional analysis, and its syntactic part (notional calculus) proves the technology of notional modeling. It is proved that the notional language is decidable, complete and consistent, while the notional calculus is decidable and consistent on countable models, but complete only on finite models. The use of the notional analysis improves the expressiveness and clarity of knowledge representation, and the use of the notional models increases the efficiency and reliability of knowledge processing.*

Keywords: primary mental abstractions, notion, concept, notional analysis, notional model, language of notions, notion calculus, knowledge base, knowledge inference, intelligent system.

УДК 510.8+004.89
ББК 22.128+32.972
DOI: 10.25728/ubs.2021.92.4

Статья представлена к публикации
членом редакционной коллегии Э.Ю. Калимулиной.

Поступила в редакцию 12.05.2021.
Опубликована 31.07.2021.