

Модель данных на гипердоменах: предпосылки, обоснование, формальное описание

Аннотация. Приведены характеристики и особенности моделей представления данных в наиболее распространенных системах управления базами данных. Выявлены ограниченность и проблематика указанных моделей для целей использования в методологии автоматизации интеллектуального труда. Приведено обоснование и формальное описание модели представления данных на основе гипердоменов.

Ключевые слова: характеристика моделей представления данных, недостатки, проблемы, использование в МАИТ, модель данных на гипердоменах, обоснование модели данных на гипердоменах, формальное описание данных на гипердоменах.

1. Проблематика методологии автоматизации интеллектуального труда

Одна из глобальных проблем в области информатики связана с тем, что потребители информации (организации, коллективы, персоны) испытывают огромные трудности с обработкой и своевременным использованием получаемой всеми способами и средствами информации - из устных источников, публикаций, средств массовой информации и компьютерных сетей.

В основе этой проблемы лежат противоречия [1], с одной стороны, между *объемом* информации и знаний, необходимым для принятия решений в разных областях деятельности и *возможностями (пропускной способностью) человека* по восприятию, обработке и оценке этой информации и знаний;

С другой стороны, между *«многомерностью»*, многоаспектностью, многоуровневостью системы знаний конкретных специалистов, зафиксированной в естественно-языковой и образной форме на соответствующих носителях, и *«одномерностью»* (в виде цепочки нулей и единиц) той же информации и знаний, представленных на формальном языке в компьютерной среде.

С учетом этого разрешение проблемы заключается не столько в применении нанотех-

нологий в современных компьютерных средах, сколько в переносе «центра тяжести» на исследование «многомерной» организации информации и знаний в памяти специалистов. На основе этой «многомерности» необходимо формировать принципиально новые пространственно-временные аспекты хранения и обработки информации и знаний в вычислительной среде и, соответственно, технологии создания автоматизированных систем.

Еще одним проблемным «узким местом» в развитии информационных технологий и создании прикладных автоматизированных систем – это взаимодействие предметных специалистов (заказчиков) и разработчиков этих систем. Информационное или семантическое моделирование предметной области, с точки зрения предметного специалиста, предполагает формирование максимального разнообразия информационных/понятийных элементов и их конструкций, а, с точки зрения специалиста в области баз данных, предполагает минимизацию количества информационных конструкций для обеспечения технологичности их обработки в вычислительной среде.

Опыт автоматизации проектной деятельности позволил установить некоторые особенности такого моделирования. Легче всего в вычислительную среду «погружаются» реляци-

онные отношения, представляющие совокупность характеристик объектов разной природы. А вот отношения, описывающие связи этих объектов, могут иметь сложную интерпретацию: от устойчивых статических связей объектов до описания совокупности сложных ситуаций, где объекты играют разные роли, а сами ситуации соотносятся как с однородными процессами, так и с разнородными.

Интересным примером, иллюстрирующим это, является описание ограничений на связи между конструктивными параметрами проектируемой детали и характеристиками обрабатывающего инструмента в возможном процессе обработки этой детали (то есть такое описание характеризует связь ситуаций в рамках жизненного цикла детали). «Разметить» такую интерпретацию в вычислительной среде практически невозможно и она остается в памяти разработчиков в виде системы умолчаний. К таким примерам можно отнести огромное число таблично представленных ограничений (зависимостей), характерных для машиностроительных задач, и оформленных в виде справочников и нормалей.

Для разрешения проблемы взаимодействия предметных специалистов и разработчиков автоматизированных систем была разработана методология автоматизации интеллектуального труда, изложенная в публикациях [2, 3], которая обеспечивает промышленный способ создания прикладных информационных и автоматизированных систем на основе **взаимоувязанных семантических и синтаксических модельных представлений** предметных задач.

Суть методологии автоматизации интеллектуального труда заключается в получении последовательности отображений прикладных задач в виде формализованных моделей (инфологической – ИЛМ и даталогической – ДЛМ) на основе первоначально формируемой семантической или концептуальной модели (КМ) предметной задачи на **трех уровнях абстрагирования**.

Следует отметить, что термин «концептуальная модель» в данном контексте имеет другое содержание, чем в теории и практике баз данных. Там он используется в другом смысле, а именно, для обозначения интегральной **модели данных**, объединяющей различные «взгляды» пользователей на этапе планирования реализации (или даталогического моделирования).

Многомодельное представление методологии автоматизации интеллектуального труда обусловило необходимость единообразного описания множества разнородных моделей, которые еще отражают и разную степень обобщения (абстрагирования) для прикладных задач. Унификация представлений разнородных моделей на разных уровнях абстрагирования по составу моделей, по механизмам порождения связей, по типологии связей обеспечивает определенную степень соответствия при установлении взаимосвязей между моделями.

Содержанием инфологического моделирования в рамках методологии автоматизации интеллектуального труда является разработка такого модельного представления предметной задачи, которое, с одной стороны, инвариантно к программно-технической среде и средствам ее реализации, а, с другой стороны, адекватно (в смысле синонимичности) естественно-языковому представлению этой задачи, так как увязано с ним на основе семантической модели.

Инфологическая модель (ИЛМ) может рассматриваться в различных, но взаимосвязанных аспектах, составляющих основу современных мультимедийных сред: вербальном, образном и звуковом представлении. Инфологическое моделирование в рамках МАИТ отражает вербальный аспект и рассматривается на нескольких уровнях абстрагирования: абстрактном, объектном, конкретном.

Для инфологического моделирования были определены и выявлены методологические, теоретические и практические основания, состав, структура инфологических представлений.

Одним из принципиальных отличий от существующих подходов информационного моделирования является наличие в инфологических моделях в рамках МАИТ:

- дополнительного типа структурной единицы, аналогичного базе данных, и названного «информационный модуль»;
- производной конструкции на структурных единицах, названной «схема структурных единиц»;
- закона цикличности знаковых вербальных представлений;
- установленных связей между схемами структурных единиц на основе закона цикличности знаковых вербальных представлений;
- совокупности закономерностей формирования инфологических моделей.

Одной из проблем методологии автоматизации интеллектуального труда является наличие определенного разрыва между модельными представлениями, инвариантными к программно-технической среде реализации прикладных автоматизированных систем (т.е. инфологическими), и модельными представлениями, ориентированными на конкретную среду реализации (т.е. даталогическими). Суть такого разрыва в том, что статические структуры инфологических моделей формируются как многослойные конструкции, организованные на основе закона цикличности, а существующие специализированные программные комплексы изначально не поддерживают такие регулярные многослойно – многомерные конструкции.

Решение указанной проблемы возможно по трем направлениям:

- адаптация инфологического модельного представления предметной задачи к даталогическим конструкциям (компонентам) существующих программно-технических средств и сред (т.е. к замене многоуровневых конструкций одноуровневыми);

- разработка «надстройки», обеспечивающей возможность использования существующих программно-технических средств и сред для организации многоуровневых конструкций и алгоритмов на них;

- разработка принципиально новой модели организации данных, обеспечивающей представление многоуровневых и взаимоувязанных информационных конструкций и параллельную их обработку.

Решение в рамках первого направления, по сути, является усложнением процедур составления «заголовочных» описаний информационных конструкций при сохранении процедур обработки в существующих СУБД.

Решение в рамках второго направления связано с созданием «мета- СУБД», позволяющей описывать, хранить и обрабатывать многоуровневые конструкции данных в совокупности «одноуровневых» СУБД.

Решение в рамках третьего направления предполагает первоначальные исследования и анализ наиболее известных из существующих моделей данных, характеристика их особенностей и областей использования с целью выявления возможностей их применения в методологии автоматизации интеллектуального труда.

Кроме того, особое место в проблематике информационных технологий занимает сдерживающий фактор, связанный отсутствием новых методологических подходов в моделировании информации и данных. В широко используемой методологии Чена преломление реального мира через достаточно простую семантическую конструкцию – «сущность-связь-атрибут» – порождает колоссальные размерности информационного представления (количество и разнообразие отношений) в памяти компьютера и, как следствие, усложняет и обработку, и интерфейс, несмотря на введение новых формализмов.

Краеугольным камнем при моделировании информации и данных является требование элементарности значений доменов, что также является определенным тормозом для развития информационных технологий.

Проанализируем существующие модели представления данных в современных СУБД.

2. Обзор моделей представления данных

Системы управления базами данными (СУБД) составляют важнейшую часть современных программных комплексов. Каждая СУБД поддерживает определенные способы описания данных и связей между ними, составляющих модель представления данных в конкретной реализации. Понятие модели данных СУБД, как правило, отражает особенности представления информации по двум направлениям:

- способ структурирования данных, которые рассматриваются как некоторая абстракция в отрыве от предметной области;

- инструмент описания концептуальной модели предметной области и динамики ее изменения в виде базы данных.

В данном исследовании главный акцент сделан на первую часть этого понятия, то есть модель данных представляется в качестве формальной теории описания и обработки данных в СУБД, которая включает в себя три основных элемента:

- структура представления данных: методы отображения их типов и логических построений при описании предметной области;

- аспект целостности: методы описания и обеспечения в базе данных согласованной и непротиворечивой информации;

- методы манипулирования данными.

Кроме теоретических характеристик моделей представления данных в исследование включены практические пояснения и интерпретации особенностей построения СУБД каждого вида с учетом специфики их организации, вариантов конкретной реализации и прикладного использования.

Рассмотрим наиболее востребованные и перспективные на текущий момент времени технологии моделирования информации и данных:

- реляционная модель данных (РМД);
- постреляционная модель данных (ПМД);
- объектно-ориентированная модель данных (ООМД);
- многомерная модель данных (ММД).

3. Особенности реляционной модели данных

Структуры данных в реляционной модели основываются на плоских нормализованных отношениях. Ограничения целостности в этих моделях выражаются с помощью средств логики первого порядка, а манипулирование данными осуществляется на основе реляционной алгебры или равносильного ей реляционного исчисления. Разработчики любой конкретной реляционной модели показывают соответствие своей конкретной модели данных общей реляционной модели и требованиям реляционности по Ф. Кодду [4-9].

Ключевыми понятиями реляционных баз данных являются: отношение, кортеж, атрибут и домен. Их характеристики необходимы для анализа и оценки достоинств и недостатков данной модели. Реляционная модель данных рассматривает только две структурные единицы, формально увязанные между собой – отношение и домен – на двух уровнях абстрагирования.

Несколько важных свойств отношений определяют специфику работы с реляционной моделью данных: отсутствие кортежей-дубликатов; отсутствие упорядоченности кортежей; отсутствие упорядоченности атрибутов; атомарность значений атрибутов; использование принципа замыкания.

Эти характеристики в большей степени относятся к структурной составляющей реляционной модели.

В целостной части реляционной модели фиксируются два основополагающих принципа, которые должны поддерживаться в любой

реляционной СУБД. Первый принцип называется требованием целостности сущностей. Это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений. Для соблюдения целостности сущности достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа.

Второй принцип называется требованием целостности по ссылкам и состоит в том, что для каждого значения внешнего ключа, имеющегося в исходном отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным.

Манипуляционная составляющая реляционной модели определяет два базовых механизма обработки данных – реляционная алгебра и реляционное исчисление. Реляционная алгебра и реляционное исчисление обладают большой выразительной мощностью: очень сложные запросы к базе данных могут быть выражены с помощью одного алгебраического выражения или одной формулы исчисления. Механизмы реляционной алгебры и реляционного исчисления эквивалентны, однако оба этих механизма различаются уровнем процедурности. Выражения реляционной алгебры имеют процедурную интерпретацию. Для формулы реляционного исчисления однозначная интерпретация, вообще говоря, отсутствует. Поэтому языки реляционного исчисления являются более непроцедурными или декларативными.

Определены пять основных операций реляционной алгебры, выполняющих большинство действий по извлечению данных: объединение (union); разность (difference); декартово произведение (Cartesian product); проекция (projection); селекция (selection).

С помощью комбинаций использования основных операций можно вывести три дополнительные операции: пересечение (intersection); соединение (join); деление (division).

Кроме того, в состав реляционной алгебры дополнительно включаются операция присваивания, позволяющая сохранить в базе данных результаты вычисления алгебраических выражений, и операция переименования атрибутов, дающая возможность корректно сформировать заголовок (схему) результирующего отношения.

Практические шаги формирования реляционной модели представления данных также содержат определенный набор рекомендаций, обеспечивающих приведение структуры базы данных к виду с минимальной избыточностью хранимой информации. На начальной стадии проектирования основной задачей является максимально полное и корректное описание предметной области. Отправной точкой является некоторая структура в виде одного или нескольких отношений. Моделирование представляет собой последовательный процесс приближения исходных данных к удовлетворительному набору схем отношений, когда на каждом следующем шаге формируется измененный набор отношений, обладающих лучшими свойствами.

Этот процесс преобразований отношений называется **нормализацией**, так как исходные данные приводятся к виду нормальных форм. Целью нормализации является уменьшение противоречивости хранимой информации за счет исключения данных с повторяющейся и зависимой информацией. Нормализация существенно снижает риск нарушения целостности данных, то есть возникновения ошибок, противоречий, некорректных связей отношений и аномалий изменения данных.

Выделяется следующая последовательность нормальных форм от первой до пятой нормальной формы. Каждая последующая форма расширяет предыдущую, заимствуя все ее ограничения. На практике наиболее важны три первых формы. К отношению в третьей нормальной форме могут быть применены операции реляционной алгебры. Дальнейшая нормализация может понадобиться в случае сложной семантической организации данных.

Широкое распространение реляционных СУБД и их использование в самых разнообразных приложениях показывает, что реляционная модель данных достаточна для моделирования многих предметных областей.

Однако проектирование реляционной базы данных в терминах отношений на основе механизма нормализации часто представляет собой очень сложный и неудобный для проектировщика процесс. При этом проявляется **ограниченность реляционной модели данных** в следующих моментах:

- модель не предоставляет достаточных средств для представления смысла данных – семантика реальной предметной области долж-

на независимым от модели способом представляться в голове проектировщика;

- для многих приложений трудно моделировать предметную область на основе плоских таблиц;

- хотя весь процесс проектирования происходит на основе учета функциональных зависимостей как аксиом вне формализма реляционной модели;

- переход от представления предметной задачи к реляционной модели осуществляется посредством модели Чена «сущность-связь».

Примерами коммерческих реляционных СУБД являются Microsoft SQL Server, IBM DB2, Ingres, Informix, Oracle и др.

4. Постреляционная модель данных

Исследования показали, что существует ряд случаев, когда ограничения классической реляционной модели серьезно мешают эффективной реализации приложений. В основе проблемы лежат три аспекта:

- работа с полями переменной длины и группами записей;

- управление связями между таблицами и полями;

- отражение подлинно семантического содержания реальных структур, которые будут смоделированы в базе данных.

Оказалось, что самая сильная сторона реляционного подхода, где сложные записи разбиваются на простые таблицы – атомарность данных, является одновременно и самой большой его слабостью. Опыт разработки прикладных информационных систем показал, что отказ от этой установки ведет к качественно полезному расширению свойств модели представления данных [10-12].

Постреляционная модель снимает ограничение атомарности. В качестве значений атрибутов допускается использование многозначных, сложно структурированных объектов, значения которых состоят из нескольких подзначений. Это требует от СУБД поддержки сложных структур – таких как вложенные таблицы или массивы, а также возможности динамической нормализации этих структур. Представление данных в такой системе приобретает дополнительную иерархию: выделяются основные и вложенные таблицы. При условии, что вложенная таблица удовлетворяет общим реляцион-

ным критериям (например, имеет уникальный ключ), естественным образом происходит расширение операторов реляционной алгебры.

О таблицах, содержащих многозначные поля, говорят, что они находятся в «не первой» нормальной форме или NF2 (Non First Normal Form). Форма NF2 почти не нарушает принципы реляционной алгебры. Более того, такая информация полностью доступна, так как расширенные операторы, которые работают с таблицами NF2, позволяют извлекать встроенные таблицы и рассматривать данные как информацию, поступившую из таблиц 1NF. В большинстве случаев гораздо более эффективно осуществлять доступ к многозначным полям одновременно с остальными данными, зная, что их всегда можно извлечь и рассматривать как отдельную таблицу в тех случаях, когда это может понадобиться.

Постреляционная модель данных поддерживает ассоциированные многозначные поля, которые часто называют множественными группами. Это обеспечивает возможность увязки нескольких столбцов с множественными значениями в единое целое, называемое ассоциацией. При этом в строке первое значение одного столбца ассоциации соответствует первым значениям всех других столбцов ассоциации, в такой же связи находятся все вторые значения столбцов и т.д. Однако многозначные поля и ассоциации не могут вкладываться друг в друга, но в целом многозначность полей является очень полезным свойством при создании коммерческих приложений, где информация нередко представлена в виде списков элементов.

За расширение реляционной модели для лучшего отражения семантики предметной области приходится расплачиваться возникновением проблем обеспечения целостности и непротиворечивости хранимых данных. Эти проблемы решаются за счет включения в СУБД механизмов неявного поддержания ограничений целостности и согласованности данных. То есть система выполняет не только те действия, которые явно указывает пользователь, но и проводит дополнительные операции в соответствии с заложенными в нее правилами. Реализация такого аппарата в СУБД сложна, накладна и не всегда полностью очевидна. Так среди главных вопросов, на которые нет четко определенного ответа, можно выделить следующие:

- как эффективно определить набор вспомогательных операций, вызываемых прямым действием пользователя;
- каким образом распознавать циклы в цепочке «действие-условие-реакция...», приводящие к каскадно-вложенным операциям, и что делать при возникновении таких циклов;
- в рамках какой транзакции выполнять дополнительные условные операции и за счет каких механизмов компенсировать возникающие накладные расходы.

Рассмотренная постреляционная модель данных поддерживается СУБД uniVerse. К числу других СУБД, основанных на постреляционной модели представления данных, относятся также системы Adabas, Pick, Cache.

5. Объектно-ориентированная модель данных

Возникновение объектно-ориентированной модели данных определялось, прежде всего, потребностями практики: необходимостью разработки сложных информационных прикладных систем, для которых предшествующие технологии были недостаточно удовлетворительными.

Соответствующий методологический подход обеспечило развитие языков программирования с абстрактными типами данных и ориентацией на объектно-ориентированную технологию описания предметной области. Особое влияние на идеи и концепции объектно-ориентированной базы данных (ООБД) оказал семантический подход к моделированию [13-20], в котором была введена новая терминология, являющаяся теперь наиболее распространенной и в объектно-ориентированном программировании.

При наличии большого количества экспериментальных проектов и коммерческих систем отсутствует общепринятая объектно-ориентированная модель данных по причине отсутствия общего согласия о принятии какой-либо модели. Имеются и практические проблемы, связанные с разработкой декларативных языков запросов, их выполнением и оптимизацией, формулированием и поддержанием ограничений целостности, синхронизацией доступа, управлением транзакциями.

Тем не менее, характеризуя объектно-ориентированный подход в наиболее общей и классической постановке, можно выделить сле-

дующие базовые понятия и концепции: объект и идентификатор объекта; атрибуты и методы; классы; иерархия и наследование классов.

Каждый объект имеет состояние и поведение. Состояние и поведение объекта **инкапсулированы** в объектах различных классов, взаимодействие между которыми производится на основе **передачи сообщений** и **выполнении соответствующих методов**.

Множество объектов с одним и тем же набором атрибутов и методов образует **класс объектов**. Объект, как правило, должен принадлежать только одному классу. Допускается порождение нового класса на основе уже существующего класса – наследование. В этом случае новый класс, называемый потомком исходного класса (предка) наследует все атрибуты и методы суперкласса.

Наиболее важным новым качеством ООБД, которое позволяет получить объектно-ориентированный подход, является поведенческий или функциональный аспект объектов. В системах с традиционной организацией представления данных между структурной (статической) и функциональной (динамической) частями существовал **принципиальный разрыв**. Структурная часть системы поддерживалась всем аппаратом баз данных, ее можно было моделировать, верифицировать и т.д., а функциональная часть создавалась изолированно. В частности, отсутствовали формальный аппарат и системная поддержка совместного моделирования и гарантирования согласованности этих статической и динамической частей. В среде ООБД проектирование, разработка и сопровождение прикладной системы становится процессом, в котором интегрируются структурный и функциональный аспекты. Конечно, для этого нужны специальные языки, позволяющие определять объекты и создавать на их основе прикладную систему.

Выделяют **три аспекта**, отсутствующие в традиционной парадигме, но **требующиеся в ООБД**. Первый аспект касается потребности в средствах спецификации знаний при определении класса (ограничений целостности, правил дедукции и т.п.) Второй аспект – потребность в механизме определения семантических связей между объектами разных классов. Фактически это означает требование полного распространения на ООБД средств семантического моделирования данных. Третий аспект связан с пере-

смотром понятия класса. В контексте ООБД оказывается более удобным рассматривать класс как множество объектов данного типа, то есть одновременно поддерживать понятия и типа и класса объектов.

Основные трудности объектно-ориентированного моделирования данных связаны с тем, что такого развитого математического аппарата, какой определен для реляционных моделей, не существует. При описании предметной области выделяются **два уровня моделирования объектов**: нижний (структурный) и верхний (функциональный). База данных – это набор элементов данных, связанных отношениями «входит в класс» или «является атрибутом». Таким образом, ООБД может рассматриваться как ориентированный граф, узлами которого являются объекты. Немаловажным моментом является поддержание наряду с понятием объекта понятия значения.

Отличительным аспектом является четкое разделение схемы базы данных и самой базы данных. В качестве первичных концепций схемного уровня ООБД выступают типы и классы. Отмечается, что во всех системах, использующих только одно понятие (либо тип, либо класс) это понятие неизбежно перегружено: тип предполагает наличие некоторого множества значений, определяемого структурой данных этого типа; класс же предполагает наличие множества объектов, но это множество определяется пользователем. Таким образом, типы и классы играют разную роль, и для строгости и недвусмысленности требуются одновременное поддержание обоих понятий.

На **функциональном уровне** для универсальной модели данных может быть предложен только обобщенный подход к требуемому для этого логическому аппарату. При этом для точного определения ООБД требуется сформировать мета-описание, содержимое которого определяет виды объектов и их связей, допустимых на схемном уровне моделирования. Такое описание должно играть для ООБД такую же роль, какую играет структурная часть реляционной модели данных для схем реляционных баз данных.

Резюмируя основные характеристики ООБД и реализованной в них модели представления данных, необходимо отметить следующие важные особенности:

- поддержка сложных объектов – возможность создания агрегированных объектов за счет применения конструкторов;
- поддержка индивидуальности объектов – все объекты должны иметь уникальный идентификатор, который не зависит от значений их атрибутов;
- поддержка инкапсуляции – внешние процедуры обладают правом доступа только к спецификации интерфейса методов, а данные и реализация методов скрыты внутри объектов;
- поддержка типов и классов – требуется, чтобы в ООБД поддерживалась концепция различия между типами и классами;
- поддержка наследования типов и классов от их предков – потомки наследуют атрибуты и методы исходных типов или классов;
- перегрузка в сочетании с полным связыванием – методы применяются к объектам разных типов, а связывание имен методов в системе не должно выполняться до времени выполнения программы;
- вычислительная полнота – язык манипулирования данными должен быть языком программирования общего назначения.
- расширяемый набор типов данных – возможность создания новых типов данных на основе набора предопределенных системных типов.

Представителями объектно-ориентированным СУБД являются Jasmine, O2, Orion, Postgres.

6. Многомерная модель данных

Средства реализации и концепция построения информационных систем, ориентированных на аналитическую обработку данных с помощью СУБД, основаны на многомерном подходе [21, 22]. Построение многомерных баз данных (МБД) базируется на **трех основных аспектах**: агрегируемость, историчность и прогнозируемость данных. **Агрегируемость** данных означает рассмотрение информации на различных уровнях ее обобщения. В информационных системах степень детальности представления информации для пользователя зависит от его уровня: аналитик, пользователь-оператор, управляющий, руководитель. **Историчность** данных предполагает обеспечение высокого уровня статичности (неизменности) собственно данных и их взаимосвязей, а также обязательность привязки данных ко времени. Это позволяет использовать при обработке спе-

циализированные методы загрузки, хранения, индексации и выборки данных. Временная привязка необходима для частого выполнения запросов, имеющих значения времени и даты в составе выборки. Такая необходимость упорядочения данных по времени накладывает определенные требования и ограничения на механизмы хранения и доступа к информации. **Прогнозируемость** данных подразумевает задание функций прогнозирования и их применение к различным временным интервалам.

Многомерность модели данных означает **многомерное логическое представление структуры информации** при описании и в операциях манипулирования данными. Для МБД в настоящее время не существует единых общепринятых методов организации среды хранения данных. Однако может быть выделена совокупность наиболее характерных ее признаков:

- для обеспечения эффективного поиска запрашиваемых данных используется некоторая разновидность техники индексирования;
- данные хранятся не в виде таблиц с индексированными записями, а в форме логически упорядоченных блоков, состоящих из групп записей одной или нескольких хранимых таблиц;
- индексная часть МБД не отделяется в виде самостоятельных структур от самих хранимых данных, при этом собственно индекс может составлять не более нескольких процентов от общего объема базы данных;
- модификация данных не приводит к модификации индекса, отсюда – быстрота выполнения множественных операций обновления МБД;
- слабовыраженная зависимость времени обработки запросов от объема хранимых данных;
- необходимость заранее (на этапе загрузки) предусматривать возможные способы доступа к хранимым данным, при этом количество индексированных измерений хранимой таблицы не может быть динамически изменено без ее перезагрузки.

Основными понятиями многомерной модели данных являются: гиперкуб данных; измерение; метка; ячейка; мера.

В существующих МБД используются две **основных схемы организации данных**: поликубическая и гиперкубическая. В **поликубической** схеме предполагается, что в базе данных может быть определено несколько гиперкубов с различной размерностью и с различными измерениями

в качестве граней. В случае **гиперкубической** схемы предполагается, что все ячейки определяются одним и тем же набором измерений. Это означает, что при наличии в базе данных нескольких гиперкубов, все они имеют одинаковую размерность и совпадающие измерения.

Для получения доступа к данным пользователю необходимо указать одну или несколько ячеек путем выбора значений измерений, которым соответствуют необходимые ячейки. **Процесс выбора** значений измерений будем называть **фиксацией меток**, а множества выбранных значений измерений – множеством фиксированных меток.

В многомерной модели данных определяется **ряд операций**, позволяющих наиболее эффективно использовать достоинства многомерной модели данных. Среди таких можно выделить операции «среза», «вращения», «свертки и детализации» и «агрегации». При формировании **среза** пользователю предоставляется некоторое подмножество гиперкуба, полученное в результате фиксации меток одного или более измерений. Эта операция проводится с целью получения требуемого подмножества ячеек и отсеечения лишних значений путем последовательной фиксации меток. Срез представляет собой двумерный массив или таблицу.

Операция «**вращения**» определяет изменение порядка представления (визуализации) измерений, обеспечивая возможность вывода данных в форме, наиболее комфортной для их восприятия. В терминах рассматриваемой модели данных вращение означает смену последовательности фиксации меток при построении среза. Результатом вращения для двумерного среза (таблицы) будет замена столбцов на строки, а строк на столбцы.

Операции «**свертки и детализации**» осуществляются благодаря наличию иерархической структуры измерений. Значения измерений могут объединяться в иерархии, состоящие из одного или нескольких уровней. Например, метки измерения времени могут объединяться в иерархию с уровнями: год, квартал, месяц, день и т.п. Принципиально операции «свертки и детализации» не отличаются от операции построения среза гиперкуба данных, однако их выделяют для описания работы с агрегированными данными.

Операция «**агрегация**» обеспечивает переход к более общему представлению информации из гиперкуба за счет наличия иерархиче-

ской структуры измерений. Это означает возможность получения значений, соответствующих меткам одного уровня иерархического измерения на основе значений другого уровня. Например, суммарное значение некоторого показателя по всем значениям заданного параметра. Операции «свертки и детализации» данных в этом случае представляют собой построение среза гиперкуба данных, соответствующего смене меток различных уровней агрегации.

Эти особенности демонстрируют область наиболее предпочтительного использования многомерных баз данных как **узкоспециализированных СУБД**, предназначенных для интерактивной аналитической обработки информации.

Особенно ярко достоинства МБД проявляются при выполнении многокритериальных запросов над практически неограниченными объемами данных в системах с критичным временем отклика. При организации обработки аналогичной информации на основе реляционной модели происходит нелинейный рост трудоемкости операций в зависимости от размерности базы данных и существенное увеличение затрат оперативной памяти на индексацию.

Главным **недостатком многомерной модели данных** является ее громоздкость при решении задач обычной оперативной обработки информации. В то время, как для многомерных баз данных, в настоящее время отсутствуют единые стандарты на интерфейс, языки описания и манипулирования данными. Многомерные СУБД не поддерживают репликацию данных, наиболее часто используемую в качестве механизма загрузки.

Примерами систем, реализующими многомерные модели данных, является Essbase, Media Multi-matrix, Oracle Express Server.

7. Выделение критериев сравнения моделей представления данных

Выполненный обзор моделей представления данных позволил сформировать набор критериев для их сравнительного анализа. Целью этого анализа наряду с определением достоинств и недостатков каждого способа описания данных является необходимость сформулировать список требований, предъявляемых текущим уровнем развития методов отображения знаний о предметной области, к **современным базам**

данных и СУБД следующего поколения с учетом особенностей организации модельных представлений в методологии автоматизации интеллектуального труда [23].

Предлагаемые критерии сравнения моделей данных и СУБД, реализующих конкретную технологию, можно разделить на две группы:

теоретические аспекты:

- представление данных – особенности формы организации информации об объектах предметной области и связях между ними;
- математический аппарат – теоретическая проработанность формального описания модели и ее отдельных характеристик;
- обработка данных – набор предусмотренных операций, позволяющий осуществлять манипулирование данными;

прикладные характеристики:

- предметная область – перечень и особенности специфики задач, для решения которых использование выбранной модели наиболее эффективно;
- явные ограничения – правила формирования модели и способов обработки хранимой информации для достижения оптимальных результатов;
- неявные ограничения – проблематика применения моделей для предметных областей со сложной семантической структурой.

Наибольший акцент в сравнении сделан на форме представления и организации данных о предметной области, как ключевой характеристике каждой модели.

8. Сравнительный анализ моделей по выделенным критериям

Представление данных. Практическое использование ранних моделей данных (таких, как сетевые и иерархические) достаточно быстро выявило их главные недостатки, уйти от которых и предполагалось при создании реляционной модели данных. Таким образом, в число основных требований попали следующие:

- ориентация на пользователя, не имеющего специфических навыков в программировании;
- появление новых возможностей использования данных и необходимость введения новых связей не должны приводить к реструктуризации всей модели в целом;
- различие между непосредственно моделью предметной области и ее реализацией в СУБД,

то есть из модели исключаются все элементы, относящиеся к физическому уровню хранения данных.

К достоинствам реляционной модели данных можно отнести:

- простота представления и понимания модели для конечного пользователя: наличие небольшого числа абстракций, которые позволяют сравнительно быстро моделировать различные предметные области, используя точные формальные определения, но оставаясь интуитивно понятными;
- удобство физической реализации в ЭВМ с помощью двумерных плоских таблиц, являющихся основными и единственными элементами для непосредственного хранения информации.

Недостатки реляционной модели являются прямым следствием ее достоинств. При использовании ее для автоматизации проектной деятельности, где требуется описание предельно сложных структур данных и их взаимосвязей проявляется множество проблем. К ним можно отнести принципиальную невозможность адекватно полного отражения семантики предметной области с помощью простой логической схемы: сущность – связь – атрибут. Другими словами, возможности представления знаний о семантической специфике предметной области в реляционных системах ограничены.

Современные исследования в сфере создания постреляционных СУБД главным образом посвящены устранению именно этих недостатков. С точки зрения представления данных, постреляционная модель данных очень схожа с реляционной. Главным отличием является допущение использовать в качестве значений атрибутов не только атомарные данные, но и многозначные поля, которые для унифицированного описания считаются представленными в виде вложенных отношений. Применение вложенных таблиц связано с попыткой расширить возможности реляционной модели для улучшения отражения семантики, сохранив при этом простоту логической организации данных. При сравнении отдельных структур, представленных в реляционном и постреляционном виде, преимущества последних кажутся весьма существенными:

- высокая наглядность представления информации;
- описание данных в форме, близкой к семантике предметной области;

- повышение эффективности выполнения и упрощение синтаксиса запросов на выборку данных.

В тоже время, приобретая подобные преимущества, постреляционная модель данных теряет в аспекте, связанном с управлением хранимой информацией. Поддержание целостности и непротиворечивости данных, исключение аномалий вставки, обновления и удаления данных – большинство вопросов, решаемых в реляционных СУБД за счет механизма нормализации, нуждаются теперь в дополнительном внимании и обработке. Насколько велико влияние этих побочных эффектов на функциональность модели и являются ли достигнутые улучшения достаточной компенсацией за повышение накладных общесистемных расходов, определяется спецификой конкретной предметной области.

Объектно-ориентированное представление данных формируется на основе идей, заимствованных большей частью из концепции объектно-ориентированного подхода к программированию. Объектно-ориентированная модель предполагает, что при проектировании базы данных должны быть сформированы классы используемых объектов, построены их описания, затем созданы экземпляры необходимых объектов и определено взаимодействие между ними.

Как и в объектно-ориентированном программировании, используются **три основополагающих принципа**: инкапсуляция, наследование и полиморфизм. Инкапсуляция означает, что в объекте реализован механизм защиты (скрытия) данных и методов от внешнего вмешательства или неправильного использования. Достоинство этой схемы в том, что разработчик может использовать класс, зная его интерфейс и не вдаваясь в детали реализации. Механизм наследования позволяет строить иерархию классов, переходя от общих описаний к более конкретным. Классы-потомки содержат все свойства и операции исходного класса, которые можно переопределять, а также обладают дополнительными атрибутами и методами. Полиморфизм – способность объектов выбирать внутреннюю процедуру (метод), исходя из типа данных, принятых в вызывающем сообщении. Это означает, что для разных вариантов реализации может быть использовано одно имя, хотя конкретные действия в каждом случае будут различны.

Формулируя перечень достоинств и недостатков объектно-ориентированной модели пред-

ставления данных, следует еще раз подчеркнуть, что общепринятого определения такой модели не существует. На сегодняшний день можно говорить лишь о некотором «объектном» подходе к логическому представлению данных и о различных объектно-ориентированных способах его реализации. Общим является то, что в ООБД, в отличие от реляционных, хранятся не записи, а объекты. Механизм наследования предоставляет более совершенные средства отображения реального мира, позволяя рассматривать элементы на разных уровнях абстракции. Кроме того, имеется возможность определения новых типов данных и операций с ними.

К числу недостатков объектно-ориентированной модели относят высокую понятийную сложность используемых описаний и отсутствие непроцедурных средств извлечения объектов из базы данных: все запросы приходится реализовывать на процедурных языках, а проблема их оптимизации возлагается на разработчика. Вместо декларативных ограничений целостности (например, явного объявления первичных и внешних ключей) или полудекларативных описаний для обеспечения внутренней связности объектов приходится писать программный код. Очевидно, что оба эти недостатка связаны с **отсутствием развитых средств манипулирования данными**. Эта задача решается двумя способами – расширение объектно-ориентированных языков в сторону управления данными, либо добавление объектных свойств в реляционные СУБД.

Многомерная модель данных, как и другие, основана на расширении свойств реляционного подхода для использования в специализированных предметных областях, в число которых входит интерактивная статистическая обработка информации при проведении анализа и принятии оперативных управленческих решений. Гиперкуб является обобщением электронных таблиц, увязанных в произвольное число измерений. Набор соответствующих гиперкубов составляет многомерную базу данных (или хранилище данных). В зависимости от того, как формируются значения некоторой ячейки гиперкуба, она может быть переменной (значения изменяются и могут быть загружены из внешнего источника данных или сформированы программно) либо формулой (значения, подобно формульным ячейкам электронных таблиц, вычисляются по заранее заданным формулам).

Достоинства многомерной модели данных проявляются в трех важных областях применения, связанных с проблематикой анализа информации:

- хранилища данных интегрируют для анализа информации из нескольких источников;
- системы оперативной аналитической обработки позволяют оперативно получить ответы на запросы, охватывающие большие объемы данных в поисках общих тенденций.
- приложения добычи данных служат для выявления знаний за счет полуавтоматического поиска ранее неизвестных шаблонов и связей в базах данных.

Среди недостатков многомерной модели данных выделяют следующие:

- для проведения анализа уровень агрегации данных должен быть достаточно высок, при этом объем полезной информации не велик (приводятся оценки, что за счет денормализации и предварительно выполненной агрегации, 20 гигабайт в многомерной базе, эквивалентны 1 гигабайту исходных данных);
- набор информационных измерений должен быть стабилен (так как любое изменение в их структуре почти всегда требует полной перестройки гиперкуба);
- время ответа системы на нерегламентированные запросы является наиболее критичным параметром;
- требуется широкое использование сложных встроенных функций для выполнения кроссмерных вычислений над ячейками гиперкуба, в том числе возможность написания неформализованных пользовательских функций.

Математический аппарат. С позиции формального описания и проработанного математического аппарата реляционной модели данных нет равных. Заложенный Коддом более 40 лет назад теоретический базис во многом и обеспечил реляционным системам их широкое распространение. Похожий подход демонстрируют многомерные базы данных, где во главу угла ставится математическое пространственное описание данных, а манипуляции с данными формулируются в терминах теории множеств и логики предикатов первого порядка. Такой способ позволяет осуществить абстрагирование объектов модели и операций над ними от специфики предметной области. В свою очередь, это ведет к более универсальным методам представления и обработки информации.

Объектно-ориентированный подход в этом смысле представляет практически полную противоположность. Его основу, напротив, составляют характеристики и особенности моделируемой предметной области и решаемых задач. В отрыве от них объектно-ориентированное представление теряет большую часть своей эффективности, представляя лишь самые общие описания и рекомендации, затрудняющие их восприятие и воплощение на практике. Отсутствие же единого подхода к использованию, например, механизма наследования, приводит к тому, **что формируемая модель данных во многом зависит от опыта и квалификации разработчика**, являясь, по большому счету, частным отражением его понимания взаимодействующих объектов и их связей.

Обработка данных. Этот аспект описания моделей продолжает характеристику различий между ориентацией на инфологическую и прикладную организацию данных. Модели, поддерживающие формальный математический аппарат для представления структурной части, имеют и развитые предопределенные функции обработки данных. Так, для реляционной модели заданы операции реляционной алгебры и эквивалентного ей реляционного исчисления, независимые от конкретной физической организации баз данных во внешней памяти. Для многомерной модели определен свой перечень методов для выборки, поиска, свертки и детализации данных, имеющих как математическое выражение, так и содержательное наполнение. Объектно-ориентированная модель данных, предоставляя богатые функциональные возможности, оставляет формирование списка подбора операций за пользователем.

Предметная область. В этом аспекте каждая из моделей данных имеет свои плюсы и минусы – области, где модель работает оптимально, и, напротив, те предметные задачи, специфика которых не позволяет достичь максимального эффекта. Реляционные СУБД на текущий момент времени предназначаются, главным образом, для информационных систем оперативной обработки данных. В этом направлении реляционная модель обладает, кроме общепризнанных повышения продуктивности работы программистов и простоты использования, рядом дополнительных преимуществ. Реляционные СУБД хорошо пригодны к применению в архитектуре «клиент-сервер», па-

параллельной обработке и графических пользовательских интерфейсах.

Универсальный интерфейс SQL (Structured Query Language — язык структурированных запросов) минимизирует коммуникации между клиентом и сервером. Сегодня многие клиент-серверные средства строятся на основе протокола Open DataBase Connectivity (ODBC), который обеспечивает для клиента стандартный механизм запросов высокого уровня к серверу.

Параллельная обработка баз данных связана с принципом замкнутости реляционной модели, когда результатом некоторых манипуляций над отношениями является новое отношение. Реляционные операции естественным образом предоставляют возможности конвейерного параллелизма путем направления вывода одной операции на вход следующей. Задачи интеллектуального анализа данных (data mining), для решения которых могли бы потребоваться недели или месяцы поиска, при использовании параллелизма выполняются за часы. К тому же этот параллелизм полностью автоматизирован. Разработчики только предоставляют данные СУБД, а система сама разделяет и индексирует информацию. Пользователи предоставляют системе запросы (на основе ODBC), а система автоматически обеспечивает параллельный план выполнения запроса и его реализацию.

Реляционные данные также хорошо приспособлены к графическим пользовательским интерфейсам (GUI). Очень легко представлять отношение как множество записей, к которым пригодна метафора электронных таблиц. Имеется достаточно много инструментальных средств, позволяющих перемещать данные между документами, электронными таблицами и базами данных. Явно и единообразно представленные данные, связи и метаданные делают это возможным. Реляционные системы, объединенные с GUI, позволяют формулировать сложные запросы к базам данных, превращая создание программного кода в задачу автоматического программирования. Для заданного непроцедурного запроса реляционные СУБД могут сформировать наиболее эффективные способы его выполнения.

Вместе с тем все чаще возникает потребность в программных средствах, характеристики которых существенно отличаются от характеристик традиционных СУБД, и которые применяются в приложениях, где реляционные

СУБД слишком тяжеловесны или недостаточно функциональны и эффективны. Кроме того, и в самих реляционных системах появляется все больше совсем нетрадиционных возможностей, предназначенных для расширения областей их применения. Это направление поддерживают постреляционные СУБД, заимствующие ряд методов, присущих объектно- и многомерно-му моделированию, при сохранении общей ориентации на реляционную модель представления данных.

За счет использования подобной промежуточной схемы предметные области, в которых у постреляционной модели дела обстоят достаточно хорошо, включают управление графической, аудио-, видео- и текстовой информацией, временными рядами, геопространственными данными, а также web-приложения. Для управления медиа-информацией требуется применение идейно простых методов к неструктурированным данным. В отличие от этого, управление временными рядами и геопространственными данными требует наличия относительно сложных структур данных и применения аналитических методов. Web-приложения занимают промежуточную позицию, основываясь на неструктурированных данных для представления шаблонов динамических страниц и усложненных методов управления переменными, логикой страниц и запросами к базам данных. Принятие для использования расширяемого языка разметки (eXtensible Markup Language – XML) поможет добиться большей структуризации шаблонов Web-страниц и других документов.

До появления постреляционных СУБД управление временными рядами, текстовыми и геопространственными данными обычно производилось с помощью специализированного программного обеспечения с нестандартизованными интерфейсами и языками запросов. Медиа-данные (а часто и тексты) хранились в файловых системах и индексировались с помощью специальных средств. Постреляционные СУБД привнесли во все эти области возможности, которые дают возможность манипулировать данными с помощью стандартного языка, интегрирующего запросы к обычным и специфичным данным.

Принципиальное отличие объектно-ориентированной модели представления данных от реляционной заключается в способе описания абстракций реального мира. Если в реляционной СУБД объекты разделяются на составные

элементы, которые размещаются в отдельных таблицах, то в ООСУБД модель данных более близка сущностям реального мира. Сложно структурированные объекты можно сохранять и использовать непосредственно, не раскладывая их по таблицам. Типы данных определяются разработчиком и не ограничены набором predetermined типов.

При чтении объекта из реляционной базы он собирается из отдельных элементов и только затем пригоден для использования. В объектно-ориентированных СУБД все иначе. Данные объекта, а также его методы помещаются в хранилище как единое целое. Таким образом, объектные языки получили средство долговременного хранения данных. На ранних стадиях проектирования системы, основанном на таком подходе, выбор ООСУБД позволит в полной мере реализовать упомянутые выше преимущества – возможность объектного моделирования предметной области и анализа отображения ее сущностей в проектируемые объекты и классы. Однако на последующих стадиях проектирования основная нагрузка приходится на программную реализацию функциональных элементов системы.

В целом объектно-ориентированные базы данных обычно рекомендованы для тех случаев, когда требуется высокопроизводительная обработка данных, имеющих сложную, но **малоизменяемую структуру**. Это позволяет работать с объектами баз данных так же, как с объектами в программировании на объектно-ориентированном языке. ООСУБД расширяют возможности традиционных баз данных, используя средства долговременного хранения данных, управление параллелизмом, восстановление данных, ассоциированные запросы и другие инструменты, связанные с распределенной обработкой данных. Все шире СУБД этого типа применяются крупными биржами, банками, страховыми компаниями, а также в сфере телекоммуникаций, проектирования и производства.

Многомерные модели данных, появившиеся в связи с необходимостью обработки большого количества накапливаемой информации, направлены на оперативный анализ хранимых данных. За счет встроенных механизмов по комплексной обработке данных, они позволяют в наглядной форме осуществлять просмотр выборочных сведений в различных информационных срезах, выполнять автоматическое получение агрегированных данных, аналитических

операций свертки, детализации, сравнения различных параметров во времени и т.п.

Конечный результат такого анализа – это попытка объективировать неявные знания в большом массиве данных, а также осуществить прогнозирование возможных будущих состояний системы. Это делает многомерные СУБД решением с очевидными преимуществами в области подготовки данных для всех видов бизнес-отчетности, предполагающих представление данных в различных разрезах и разных уровнях иерархии – например, отчетов по продажам, различных форм бюджетов и т.д.

Явные ограничения. Для уменьшения потенциальных противоречий хранимых данных, исключения дублирующейся и зависимой информации реляционная модель подразумевает нормализацию исходных данных. На каждом этапе преобразований определены четкие рамки вводимых ограничений. Приведение информации к виду третьей нормальной формы обеспечивает целостность данных. Однако способы достижения нормальных форм не регламентированы, оставаясь вне рамок реляционной модели.

Объектно-ориентированная модель реализует механизм инкапсуляции объектов и их методов. Данные защищены от внешнего воздействия, но при этом встроенных механизмов поддержания согласованной и непротиворечивой информации не предусмотрено. Пользователь может сам реализовать их как дополнительные расширения методов классов на объектно-ориентированном языке программирования.

Ограничения многомерной модели связаны с совместным описанием отдельных граней гиперкуба по различным измерениям. Если реализуется гиперкубическая схема организации, то требуются их обязательная одинаковая размерность. В любом случае все ячейки гиперкуба должны иметь значения (хотя бы и неопределенные) – зачастую это приводит к хранению больших объемов разреженной информации, предполагающей использование методов оптимизации и сжатия данных.

Неявные ограничения. В реляционных СУБД проявляются в наличии достаточно жестких ограничений накладываемых реализацией языка SQL, ярким примером которых является предположение о том, что данные в реляционной базе неупорядочены (или более точно, упорядочены случайным образом). При этом их упорядочивание требует дополнитель-

ных затрат времени на сортировку при каждом обращении к базе данных. Кроме того, в качестве основного механизма, обеспечивающего быстрый поиск и выборку отдельных строк в таблице (или в связанных через внешние ключи таблицах), обычно используются различные модификации индексов, основанных на B-деревьях. Такое решение оказывается эффективным только при обработке небольших групп записей и высокой интенсивности модификации данных в базах данных.

В постреляционных системах считается, что данные остаются неизменными в течение длительного периода времени. И здесь более эффективным оказывается хранение данных в форме частично денормализованных таблиц, в которых для увеличения производительности могут храниться не только детализированные, но и предварительно вычисленные агрегированные значения. А для навигации и выборки могут использоваться специализированные, основанные на предположении о малой изменчивости и малой подвижности данных, методы адресации и индексации связей между атрибутами из разных таблиц (операции соединения).

Ориентация на какой-либо объектно-ориентированный язык программирования приводит к тому, что ООСУБД, как правило, сильно зависит от его возможностей, а любые запросы к базе данных должны быть реализованы в его конструкциях. Кроме того, изменение схемы класса объектов обычно означает, что должны быть изменены и другие классы приложения, которые взаимодействуют с экземплярами данного класса. Это ведет к необходимости модификации всей системы.

В многомерных СУБД наряду с единичными данными, извлекаемыми из оперативных систем, используются и суммарные показатели (агрегированные параметры), такие, как суммы объемов продаж по месяцам, по категориям товаров и пр. Агрегаты хранятся в явном виде с единственной целью – ускорить выполнение запросов. Ведь, с одной стороны, в хранилище накапливается, как правило, очень большой объем данных, а с другой – аналитиков в большинстве случаев интересуют не детальные, а обобщенные показатели. Однако за скорость обработки запросов к суммарным данным приходится платить увеличением объема данных и времени на их загрузку. Плюс к этому, малейшее изменение структуры

базы данных требует перестройки всего пространства гиперкубов, что существенно сужает рамки использования модели.

9. Предпосылки и обоснование модели представления данных на гипердоменах

Таким образом, анализ наиболее известных из существующих моделей данных показал, что их использование для создания прикладных автоматизированных систем в рамках методологии автоматизации интеллектуального труда достаточно затруднительно в силу перечисленных проблем и недостатков этих моделей. Необходимо разработка принципиально новой модели данных для представления регулярных многослойно-многомерных конструкций в даталогическом моделировании.

Теоретическим основанием для формирования новой модели представления данных является формальное описание взаимоувязанных статических составляющих инфологических моделей объектного и конкретного уровней представления предметных задач в рамках методологии автоматизации интеллектуального труда.

Статические составляющие инфологических моделей обладают следующими отличительными особенностями [24]:

- наличие трех типов структурных единиц (СЕ), позволяющих описывать конструкции разной сложности – информационный модуль (СЕ-«ИМ»), информационная сущность (СЕ-«ИС»), информационный атрибут (СЕ-«ИА»);
- наличие уровней абстрагирования, в соответствии с которыми структурные единицы различаются именами (ИСЕ) на объектном уровне и состояниями (ССЕ) на конкретном уровне;
- наличие различных типов и видов связей, позволяющих формировать многообразие синтаксических основных и производных конструкций разных уровней сложности (бинарных связей – «состав», «упорядочение», «компоновка» и тернарных связей на структурных единицах, схем структурных единиц и их бинарных связей);
- производные конструкции – схемы структурных единиц – обеспечивают увязку структурных единиц всех типов в единое целое, как для имен, так и для состояний.

Закон цикличности вербальных знаковых представлений, сформулированный в рамках

методологии автоматизации интеллектуального труда, определяет связность производных конструкций и, по сути, отражает формирование взаимосвязей синтаксических конструкций разных уровней сложности. Т.е. область допустимых значений структурной единицы типа «информационный атрибут» одного уровня сложности вербального описания может быть представлена (раскрыта) как структурная единица типа «информационный модуль» другого, менее сложного уровня вербального описания.

Идея формирования принципиально новой модели данных для представления регулярных многослойно-многомерных конструкций заключается в том, что такая сложная область допустимых значений может быть отображена в элемент данных под названием «гипердомен» [25].

Требования к модели представления данных на основе гипердоменов можно разделить на несколько групп. Во-первых: требования к полноте модели, во-вторых: требования, вытекающие из особенностей организации семантических и синтаксических моделей в рамках МАИТ.

Требования первой группы состоят в том, что в модели должны быть описаны и статические конструкции, и ограничения на них, и операции над конструкциями.

К требованиям второй группы можно отнести:

- наличие множества структурных единиц инфологической модели, обеспечивающих описание сложноорганизованных данных на различных уровнях сложности (абстракции) и на нескольких уровнях абстрагирования (общности),
- наличие формального описания связей структурных единиц для простых и сложных конструкций модели,
- наличие механизма и закономерностей порождения конструкций модели.

10. Обоснование элементов и конструкций модели представления данных на основе гипердоменов

Исходя из определения модели данных, статическая ее составляющая включает: элементы данных и множества их значений, структуры данных и множества их состояний, ограничения на элементы и структуры данных.

В новой модели данных элементом данных является гипердомен, определяемый с учетом

уровней абстрагирования, именем и совокупностью состояний.

В основе понятия гипердомена лежит понятие сложной конструкции, в которую могут быть отображены схемы структурных единиц как элементы производной статической составляющей инфологической модели в рамках МАИТ.

Эти особенности позволяют интерпретировать понятие «именованного гипердомена» как сложную конструкцию на множестве схем именovaných структурных единицах, удовлетворяющих определенным условиям на заданном уровне сложности (u) ((1) и Рис.1).

$$\overline{E2}_{us}(n) = \{\overline{e}_{usil}^2, \dots, \overline{e}_{usiv}^2, \dots, \overline{e}_{usid}^2\} \quad (1)$$

где $\overline{e}_{usil}^2$ - именованная подструктура или схема именovaných структурных единиц, а индексы u, i - отвечают за уровень синтаксической сложности, s - за имя гипердомена (или разнообразие на уровне), l - за имя подструктуры.

Любая схема именovaných структурных единиц $\overline{e}_{usil}^2$ определяется двумя структурными единицами типов «ИМ» - c_{ts}^p и «ИС» - c_{il}^q и, в свою очередь, определяет множество вариантов строения такой подструктуры из-за разнообразия подмножеств структурных единиц типа «ИА».

Для пояснения сравним эти конструкции с конструкциями реляционной модели данных (РМД) – доменом и отношением [6]. По определению отношение фиксируется как именованная структура (2) и как множество элементов (3).

$$R \subset D_1 \times D_2 \times \dots \times D_n \quad (2)$$

$$R = \{r_1, \dots, r_i, \dots, r_m\} \quad (3)$$

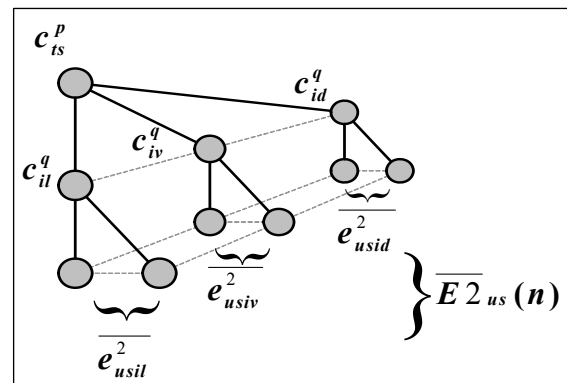


Рис.1. Структура именованного гипердомена $\overline{E2}_{us}(n)$

Графически формула (2) интерпретируется как «шапка» таблицы или соотношение имени таблицы с именами столбцов, а (3) – как соотношение имени таблицы с идентификаторами строк (состояниями) такой таблицы. При этом каждая строка обладает определенной структурой, включающей состояния всех столбцов (4).

$$r_i = (d_s^1, d_t^2, \dots, d_t^n) \quad (4)$$

Предположим, что база данных (**BD**) состоит только из одного отношения (**R**). Тогда графически и геометрически это можно представить, как показано на Рис.2 и Рис.3.

Таким образом, в рамках реляционной модели данных представление однородных состояний базы данных (**BD**) накладывается на одну ось (или одномерное пространство).

В случае, когда база данных (**BDI**) состоит из нескольких отношений-таблиц (**R** и **P**), графическая и геометрическая интерпретация будет иметь вид, приведенный на Рис.4 и Рис.5.

В этом случае состояния базы данных представляют собой набор кластеров, в каждом из которых состояния однородны. При этом в обоих случаях все состояния базы данных имеют определенную структуру, как в (4).

Таким образом, понятие схемы ИСЕ - $\overline{e_{usil}^2}$ будет эквивалентно понятию имени кластера базы данных – **BDI.R** или **BDI.P**, соответствующего второму случаю.

Именованные гипердомены могут быть объединены во множества как на одном уровне сложности (5), так и на всех уровнях синтаксической сложности (6).

$$\overline{E2}_u(n) = \bigcup_s \overline{E2}_{us} \quad (5)$$

$$\overline{E2}(n) = \bigcup_u \overline{E2}_u \quad (6)$$

Для конкретного уровня понятие гипердомена раскрывается через его состояние, т.е. понятие «состояние гипердомена» рассматривается как сложная конструкция на множестве схем состояний структурных единицах, удовлетворяющих определенным условиям на заданном уровне сложности (**u**) ((7) и Рис.6).

$$\overline{E3}_{usr}(nm) = \{\overline{e_{usrila}^3}, \dots, \overline{e_{usrive}^3}, \dots, \overline{e_{usride}^3}\} \quad (7)$$

где $\overline{e_{usrila}^2}$ - состояние подструктуры или схема состояний структурных единиц, а индексы **u**, **I**

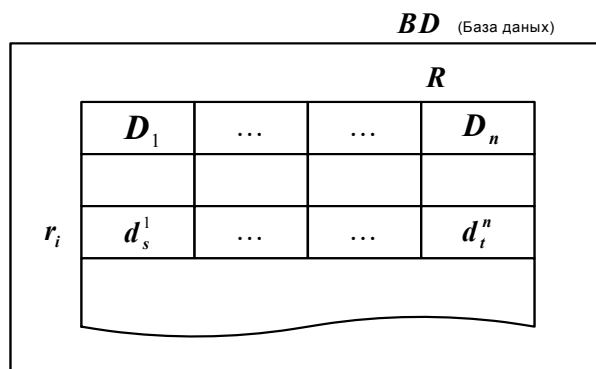


Рис.2. Графическое представление базы данных **BD**

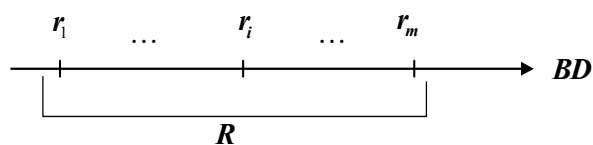


Рис.3. Геометрическое представление базы данных **BD**

– отвечают за уровень синтаксической сложности, **s** - за имя гипердомена (или разнообразие на уровне), **r** - за его состояние, **l** - за имя подструктуры, **a** - за состояние подструктуры.

Таким образом, понятие схемы состояний структурных единиц - $\overline{e_{usrilt}^2}$ будет эквивалентно отдельному значению кластера базы данных - r_i , а состояние гипердомена будет определяться сочетанием подмножества значений из кластеров базы данных.

Состояния гипердомена могут быть объединены в множества:

$$\overline{E3}_{us}(nm) = \bigcup_d \overline{E3}_{usd}(nm) \quad (8)$$

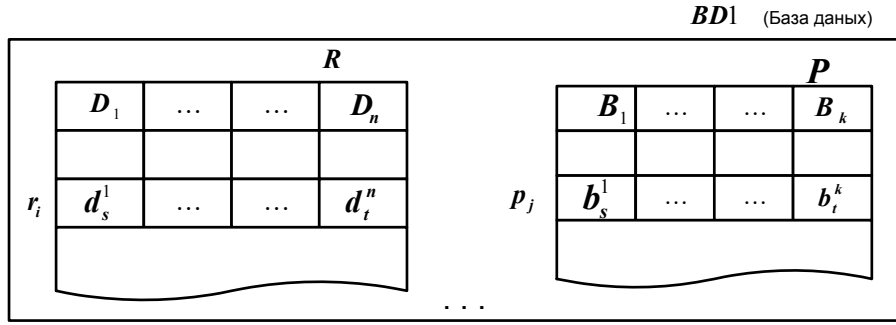
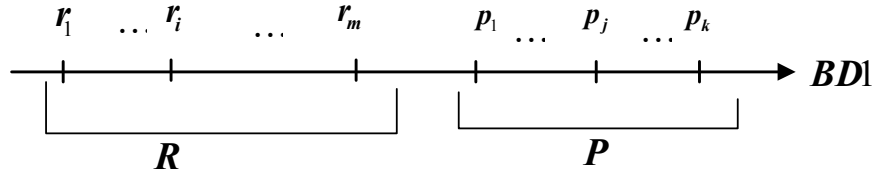
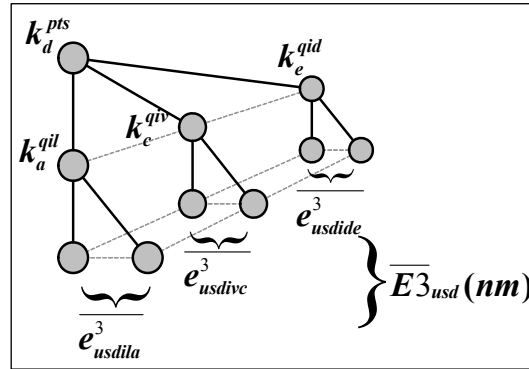
- множество состояний для одного именованного гипердомена;

$$\overline{E3}_u(nm) = \bigcup_s \overline{E3}_{us}(nm) \quad (9)$$

- множество состояний для всех именованных гипердоменов одного уровня синтаксической сложности;

$$\overline{E3}(nm) = \bigcup_u \overline{E3}_u(nm) \quad (10)$$

- множество состояний для всех именованных гипердоменов всех уровней синтаксической сложности.


 Рис.4. Графическое представление базы данных **BDI**

 Рис.5. Геометрическое представление базы данных **BDI**

 Рис.6. Структура состояния гипердомена $\overline{E3}_{usr}(nm)$

11. Формальное описание модели представления данных на основе гипердоменов

11.1. На объектном уровне

Сформулированный в рамках МАИТ закон цикличности знаковых вербальных конструкций обеспечивает правила сопряжения схем структурных единиц разных уровней сложности.

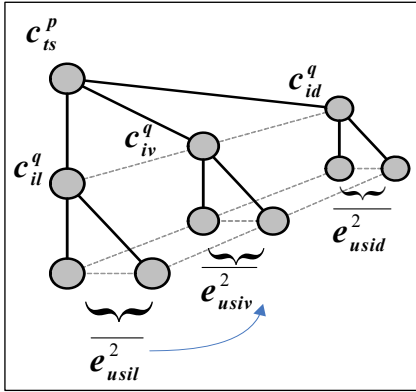
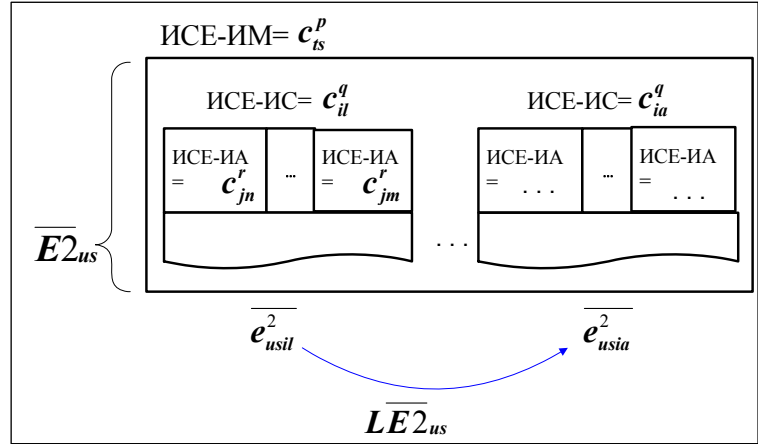
Многообразие бинарных связей схем именованных структурных единиц предопределяют многообразие связей именованных гипердоменов и их подструктур. Для всех конструкций на именованных гипердоменах выполнена геометрическая и графическая интерпретация.

Между именованными гипердоменами могут быть установлены бинарные отношения, включающие совокупность следующих подмножеств:

$$De2(n) \subset \overline{E2}(n) \times \overline{E2}(n)$$

$$De2(n) = \{ \overline{LE2}_{us}(n), \overline{LE2}_{us}(n) \overline{E2}_{uq}(n), \overline{LE2}_{uq}(n) / \overline{E2}_{us}(n), \\ \overline{LE2}_{us}(n) \overline{E2}_{zn}(n), \overline{LE2}_{zn}(n) \overline{LE2}_{zm}(n) / \overline{E2}_{us}(n) \}$$

Подмножество $\overline{LE2}_{us}(n)$ определяет упорядоченность подструктур (il, iv) именованного гипердомена (s) уровня сложности (u) (Рис.7 и Рис.8):

Рис.7. Геометрическая интерпретация подмножества $\overline{LE2}_{us}(n)$ Рис.8. Графическая интерпретация подмножества $\overline{LE2}_{us}(n)$

$$\overline{LE2}_{us}(n) = \{(\overline{e_{usil}^2}, \overline{e_{usiv}^2}) \mid \overline{e_{usil}^2}, \overline{e_{usiv}^2} \in \overline{E2}_{us}(n)\}$$

Подмножество $\overline{LE2}_{us}(n)\overline{E2}_{uq}(n)$ определяет связность подструктур (il, ia) разных именованных гипердоменов (s, q) одного уровня сложности (u) :

$$\overline{LE2}_{us}(n)\overline{E2}_{uq}(n) = \{(\overline{e_{usil}^2}, \overline{e_{uqia}^2}) \mid \overline{e_{usil}^2} \in \overline{E2}_{us}(n), \overline{e_{uqia}^2} \in \overline{E2}_{uq}(n)\}$$

Подмножество $\overline{LE2}_{uq}(n) / \overline{E2}_{us}(n)$ определяет компоновку подструктур (ia, ic) одного именованного гипердомена (q) при условии связности этих подструктур с подструктурой (il) другого именованного гипердомена (s) одного уровня сложности (u) :

$$\begin{aligned} \overline{LE2}_{uq}(n) / \overline{E2}_{us}(n) &= \{(\overline{e_{uqia}^2}, \overline{e_{uqic}^2}) \mid \overline{e_{uqia}^2}, \overline{e_{uqic}^2} \in \overline{E2}_{uq}(n) \& \\ &(\exists \overline{e_{usil}^2} \in \overline{E2}_{us}(n) : (\overline{e_{usil}^2}, \overline{e_{uqia}^2}), (\overline{e_{usil}^2}, \overline{e_{uqic}^2}) \in \overline{LE2}_{us}(n)\overline{E2}_{uq}(n))\} \end{aligned}$$

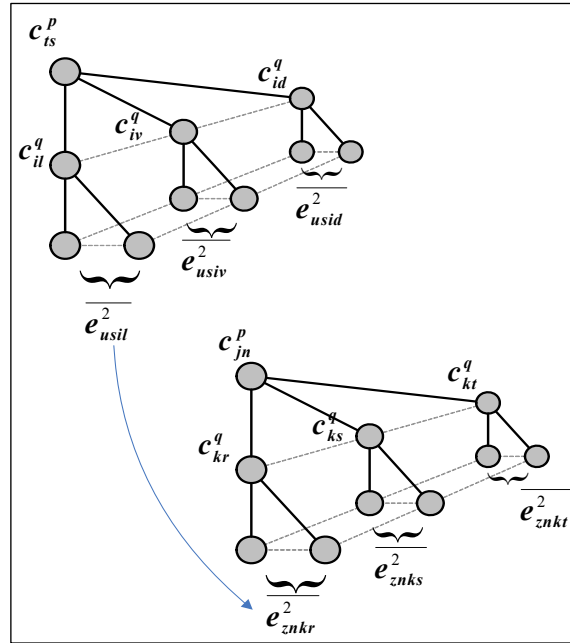
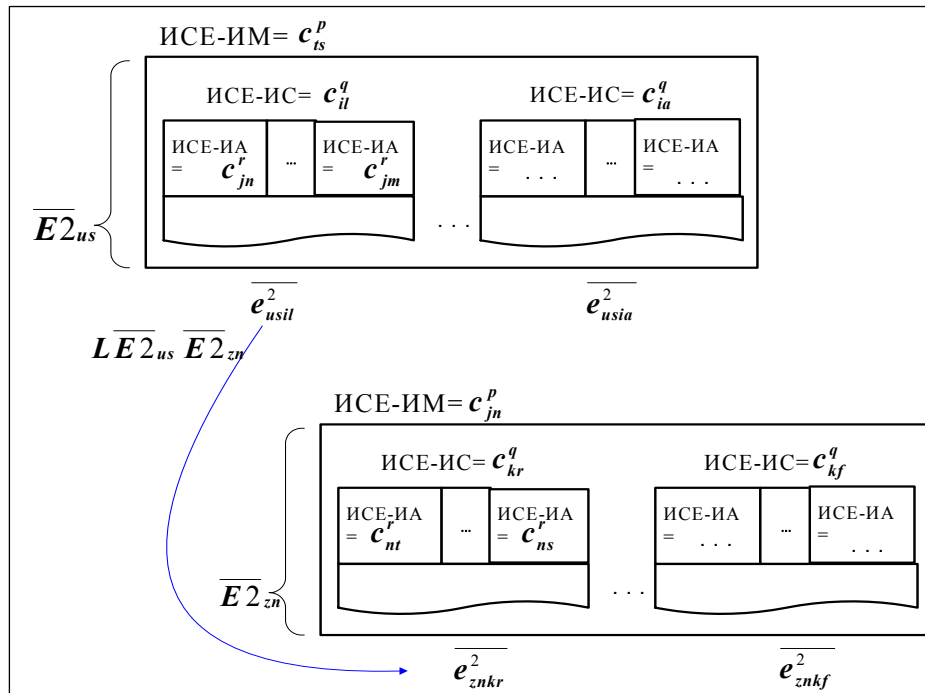
Подмножество $\overline{LE2}_{us}(n)\overline{E2}_{zn}(n)$ определяет связность подструктур (il, kr) разных именованных гипердоменов (s, n) разных уровней сложности (u, z) (Рис.9 и Рис.10):

$$\overline{LE2}_{us}(n)\overline{E2}_{zn}(n) = \{(\overline{e_{usil}^2}, \overline{e_{znkr}^2}) \mid \overline{e_{usil}^2} \in \overline{E2}_{us}(n), \overline{e_{znkr}^2} \in \overline{E2}_{zn}(n)\}$$

Подмножество $\overline{LE2}_{zn}(n)\overline{LE2}_{zm}(n) / \overline{E2}_{us}(n)$ определяет компоновку подструктур (kr, kp) разных именованных гипердоменов (n, m) одного уровня сложности (z) при условии связности этих подструктур с подструктурой (il) другого именованного гипердомена (s) другого уровня сложности (u) :

$$\begin{aligned} \overline{LE2}_{zn}(n)\overline{LE2}_{zm}(n) / \overline{E2}_{us}(n) &= \{(\overline{e_{znkr}^2}, \overline{e_{zmkp}^2}) \mid \\ &\overline{e_{znkr}^2} \in \overline{E2}_{zn}(n), \overline{e_{zmkp}^2} \in \overline{E2}_{zm}(n) \& \\ &(\exists \overline{e_{usil}^2} \in \overline{E2}_{us}(n) : (\overline{e_{usil}^2}, \overline{e_{znkr}^2}) \in \overline{LE2}_{us}(n)\overline{E2}_{zn}(n), \\ &(\overline{e_{usil}^2}, \overline{e_{zmkp}^2}) \in \overline{LE2}_{us}(n)\overline{E2}_{zm}(n))\} \end{aligned}$$

Таким образом, определено и формально представлено все многообразие бинарных связей именованных гипердоменов и их подструктур, что позволяет формировать многослойно-многомерные конструкции для размещения данных в вычислительной среде.


 Рис.9. Геометрическая интерпретация подмножества $\overline{LE2}_{us}(n)\overline{E2}_{zn}(n)$

 Рис.10. Графическая интерпретация подмножества $\overline{LE2}_{us}(n)\overline{E2}_{zn}(n)$

11.2. На конкретном уровне

Статические составляющие инфологических моделей конкретного уровня представления предметных задач определяют структурные характеристики состояний модели представления данных на основе гипердоменов, обладают аналогичными отличительными особенностями и дополнительно введены производные конструкции – схемы состояний структурных единиц, которые обеспечивают увязку состояний структурных единиц всех типов в единое целое.

Многообразие бинарных связей схем состояний структурных единиц предопределяют многообразие связей состояний гипердоменов и их подструктур.

Между состояниями гипердоменов и их подструктур могут быть установлены бинарные отношения, включающие совокупность следующих подмножеств:

$$De3(nm) \subset \overline{E}3(nm) \times \overline{E}3(nm),$$

$$De3(nm) = \{ \overline{LE}3_{usd}(nm),$$

$$\overline{LE}3_{usd}(nm)\overline{E}3_{usf}(nm), \overline{LE}3_{usf}(nm) / \overline{E}3_{usd}(nm),$$

$$\overline{LE}3_{usd}(nm)\overline{E}3_{uqf}(nm), \overline{LE}3_{uqf}(nm)\overline{LE}3_{uqg}(nm) / \overline{E}3_{usd}(nm),$$

$$\overline{LE}3_{usd}(nm)\overline{E}3_{znp}(nm), \overline{LE}3_{znp}(nm)\overline{E}3_{zmq}(nm) / \overline{E}3_{usd}(nm) \}$$

Подмножество $\overline{LE}3_{usd}(nm)$ определяет упорядоченность состояний подструктур (*ila*, *ivc*) в одном состоянии (*d*) гипердомена (*s*) на уровне сложности (*u*) (Рис.11).

$$\overline{LE}3_{usd}(nm) = \{ (\overline{e^3_{usdila}}, \overline{e^3_{usdivc}}) \mid \overline{e^3_{usdila}}, \overline{e^3_{usdivc}} \subset \overline{E}3_{usd}(nm) \}$$

Подмножество $\overline{LE}3_{usf}(nm) / \overline{E}3_{usd}(nm)$ определяет компоновку состояний подструктур (*ilj*, *ivk*) в одном состоянии (*f*) гипердомена (*s*) при условии их связности с состоянием подструктуры (*ila*) другого состояния (*d*) гипердомена (*s*) на уровне сложности (*u*):

$$\overline{LE}3_{usf}(nm) / \overline{E}3_{usd}(nm) = \{ (\overline{e^3_{usfilj}}, \overline{e^3_{usfivk}}) \mid \overline{e^3_{usfilj}}, \overline{e^3_{usfivk}} \subset \overline{E}3_{usf}(nm) \&$$

$$(\exists \overline{e^3_{usdila}} \in \overline{E}3_{usd}(nm) :$$

$$(\overline{e^3_{usdila}}, \overline{e^3_{usfilj}}), (\overline{e^3_{usdila}}, \overline{e^3_{usfivk}}) \in \overline{LE}3_{usd}(nm)\overline{E}3_{usf}(nm) \}) \}$$

Подмножество $\overline{LE}3_{usd}(nm)\overline{E}3_{uqf}(nm)$ определяет связность состояний подструктур (*ila*, *iax*) в разных состояниях (*d,f*) разных гипердоменов (*s,q*) на уровне сложности (*u*):

$$\overline{LE}3_{usd}(nm)\overline{E}3_{uqf}(nm) = \{ (\overline{e^3_{usdila}}, \overline{e^3_{uqfiar}}) \mid \overline{e^3_{usdila}} \subset \overline{E}3_{usd}(nm),$$

$$\overline{e^3_{uqfiar}} \subset \overline{E}3_{uqf}(nm) \}$$

Подмножество $\overline{LE}3_{uqf}(nm)\overline{LE}3_{uqg}(nm) / \overline{E}3_{usd}(nm)$ определяет компоновку состояний подструктур (*ial*, *iax*) в разных состояниях (*f,g*) гипердомена (*q*) при условии их связности с состоянием подструктуры (*ila*) другого состояния (*d*) гипердомена (*s*) на уровне сложности (*u*):

$$\overline{LE}3_{uqf}(nm)\overline{E}3_{uqg}(nm) / \overline{E}3_{usd}(nm) = \{ (\overline{e^3_{uqfiar}}, \overline{e^3_{uqgial}}) \mid$$

$$\overline{e^3_{uqfiar}} \subset \overline{E}3_{uqf}(nm), \overline{e^3_{uqgial}} \subset \overline{E}3_{uqg}(nm) \&$$

$$(\exists \overline{e^3_{usdila}} \in \overline{E}3_{usd}(nm) :$$

$$(\overline{e^3_{usdila}}, \overline{e^3_{uqfiar}}) \in \overline{LE}3_{usd}(nm)\overline{E}3_{uqf}(nm))$$

$$(\overline{e^3_{usdila}}, \overline{e^3_{uqgial}}) \in \overline{LE}3_{usd}(nm)\overline{E}3_{uqg}(nm) \}) \}$$

Подмножество $\overline{LE}3_{usd}(nm)\overline{E}3_{znp}(nm)$ определяет связность состояний подструктур (*ila*, *krx*) в разных состояниях (*d,p*) разных гипердоменов (*s,n*) на разных уровнях сложности (*u,z*) (Рис.12):

$$\overline{LE}3_{usd}(nm)\overline{E}3_{znp}(nm) = \{ (\overline{e^3_{usdila}}, \overline{e^3_{znpkrx}}) \mid \overline{e^3_{usdila}} \subset \overline{E}3_{usd}(nm),$$

$$\overline{e^3_{znpkrx}} \subset \overline{E}3_{znp}(nm) \}$$

Подмножество $\overline{LE3}_{znp}(nm)\overline{E3}_{zmq}(nm)/\overline{E3}_{usd}(nm)$ определяет компоновку состояний подструктур (k_{rx}, k_{pl}) в разных состояниях (p, q) разных гипердоменов (n, m) уровня сложности (z) при условии их связности с состоянием подструктуры (ila) другого состояния (d) гипердомена (s) на другом уровне сложности (u):

$$\begin{aligned} \overline{LE3}_{znp}(nm)\overline{E3}_{zmq}(nm)/\overline{E3}_{usd}(nm) &= \{(\overline{e_{znpkrx}^3}, \overline{e_{zmqkpl}^3}) \mid \\ &\overline{e_{znpkrx}^3} \subset \overline{E3}_{znp}(nm), \overline{e_{zmqkpl}^3} \subset \overline{E3}_{zmq}(nm) \& \\ &(\exists \overline{e_{usdila}^3} \in \overline{E3}_{usd}(nm) : \\ &(\overline{e_{usdila}^3}, \overline{e_{znpkrx}^3}) \in \overline{LE3}_{usd}(nm)\overline{E3}_{znp}(nm), \\ &(\overline{e_{usdila}^3}, \overline{e_{zmqkpl}^3}) \in \overline{LE3}_{usd}(nm)\overline{E3}_{zmq}(nm))\} \end{aligned}$$

Таким образом, определено все многообразие бинарных связей состояний гипердоменов и состояний их подструктур, что позволяет заполнять многослойно-многомерные конструкции данными в вычислительной среде.

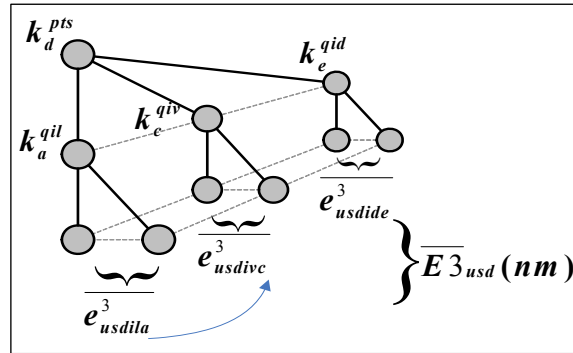


Рис.11. Геометрическая интерпретация подмножества $\overline{LE3}_{usd}(nm)$

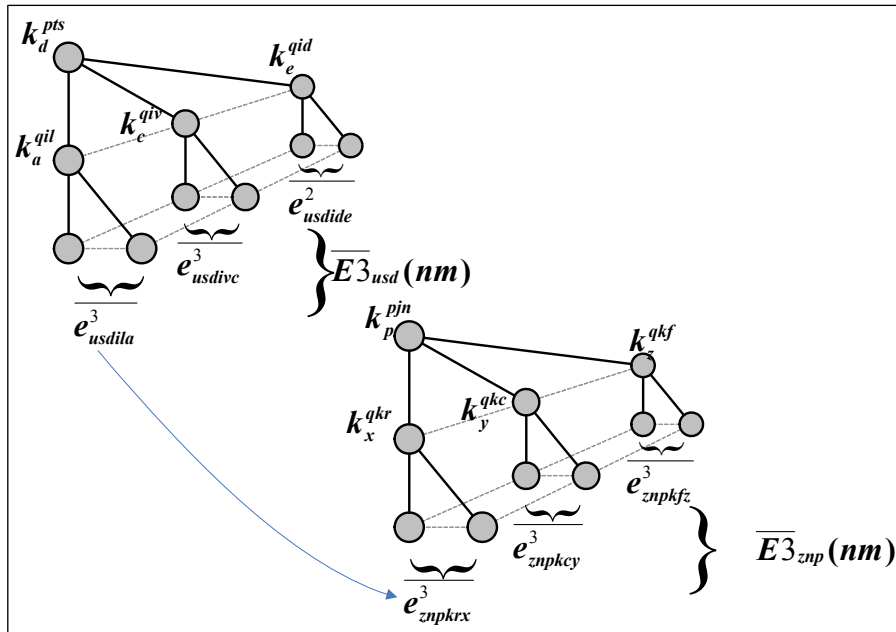


Рис.12. Геометрическая интерпретация подмножества $\overline{LE3}_{usd}(nm)\overline{E3}_{znp}(nm)$

12. Взаимосвязи конструкций объектного и конкретного уровней для модели данных на гипердоменах

Взаимосвязи статических конструкций объектного и конкретного уровня для модели представления данных на основе гипердоменов формально можно описать:

1) любой именованный гипердомен может быть представлен множеством своих состояний, а любое состояние гипердомена определяется совокупностью состояний его подструктур:

$$\overline{E2}_{ip}(n) \rightarrow \overline{E3}_{ip1}(nm), \dots, \overline{E3}_{ipg}(nm), \dots, \overline{E3}_{ipf}(nm),$$

$$\overline{E3}_{ipg}(nm) = \{\overline{e^3_{ipgmu}}\}, \overline{E3}_{ipg}(nm) \subset \overline{E3}(nm);$$

2) бинарные связи именованных подструктур именованных гипердоменов определяют бинарные связи состояний подструктур состояний гипердоменов:

$$De2(n) \rightarrow De3(nm):$$

Упорядоченность подструктур (*il, iv*) именованного гипердомена (*us*):

- определяет упорядоченность состояний подструктур (*ila, ivc*) в одном состоянии (*d*) гипердомена (*s*) на уровне сложности (*u*):

$$\overline{LE2}_{us}(n) = \{\{\overline{e^2_{usil}}, \overline{e^2_{usiv}}\}\}, \overline{LE2}_{us}(n) \subset De2(n)$$

$$(\overline{e^2_{usil}}, \overline{e^2_{usiv}}) \rightarrow \{\{\overline{e^3_{usdila}}, \overline{e^3_{usdivc}}\}\} \subset \overline{LE3}_{usd}(nm),$$

$$\overline{LE3}_{usd}(nm) \subset De3(nm);$$

- определяет связность состояний подструктур (*ila, ilj*) в разных состояниях (*d, f*) одного гипердомена (*s*) на уровне сложности (*u*):

$$\overline{LE2}_{us}(n) = \{\{\overline{e^2_{usil}}, \overline{e^2_{usiv}}\}\}, \overline{LE2}_{us}(n) \subset De2(n)$$

$$(\overline{e^2_{usil}}, \overline{e^2_{usiv}}) \rightarrow \{\{\overline{e^3_{usdila}}, \overline{e^3_{usfilj}}\}\} \subset \overline{LE3}_{usd}(nm) \overline{E3}_{usf}(nm),$$

$$\overline{LE3}_{usd}(nm) \overline{E3}_{usf}(nm) \subset De3(nm);$$

- определяет компоновку состояний подструктур (*ilj, ivk*) в одном состоянии (*f*) гипердомена (*s*) при условии их связности с состоянием подструктуры (*ila*) другого состояния (*d*) гипердомена (*s*) на уровне сложности (*u*):

$$\overline{LE2}_{us}(n) = \{\{\overline{e^2_{usil}}, \overline{e^2_{usiv}}\}\}, \overline{LE2}_{us}(n) \subset De2(n)$$

$$(\overline{e^2_{usiv}}, \overline{e^2_{usiv}}) \& (\overline{e^2_{usil}}, \overline{e^2_{usiv}}) \rightarrow$$

$$\{\{\overline{e^3_{usfilj}}, \overline{e^3_{usfivk}}\}\} \subset \overline{LE3}_{usf}(nm) / \overline{E3}_{usd}(nm),$$

$$\overline{LE3}_{usf}(nm) / \overline{E3}_{usd}(nm) \subset De3(nm);$$

Связность подструктур (*il, ia*) разных именованных гипердоменов (*s, q*) одного уровня сложности (*u*) определяет связность состояний подструктур (*ila, iax*) в разных состояниях (*d, f*) разных гипердоменов (*s, q*) на уровне сложности (*u*):

$$\overline{LE2}_{us}(n) \overline{E2}_{uq}(n) = \{\{\overline{e^2_{usil}}, \overline{e^2_{uqia}}\}\}, \overline{LE2}_{us}(n) \overline{E2}_{uq}(n) \subset De2(n)$$

$$(\overline{e^2_{usil}}, \overline{e^2_{uqia}}) \rightarrow \{\{\overline{e^3_{usdila}}, \overline{e^3_{uqfiat}}\}\} \subset \overline{LE3}_{usd}(nm) \overline{E3}_{uqf}(nm),$$

$$\overline{LE3}_{usd}(nm) \overline{E3}_{uqf}(nm) \subset De3(nm)$$

Компоновка подструктур (*ia, ic*) одного именованного гипердомена (*uq*) при условии связности этих подструктур с подструктурой (*il*) другого именованного гипердомена (*us*) одного уровня сложности (*u*) определяет компоновку состояний подструктур (*ial, iax*) в разных состояниях (*f, g*) гипердомена (*q*) при условии их связности с состоянием подструктуры (*ila*) другого состояния (*d*) гипердомена (*s*) на том же уровне сложности (*u*):

$$\begin{aligned} \overline{LE2}_{uq}(n) / \overline{E2}_{us}(n) &= \{\overline{e^2_{uqia}}, \overline{e^2_{uqic}}\}, \overline{LE2}_{uq}(n) / \overline{E2}_{us}(n) \subset De2(n) \\ (\overline{e^2_{uqia}}, \overline{e^2_{uqic}}) &\rightarrow \\ \{\overline{e^3_{uqfia}}, \overline{e^3_{uqgic}}\} &\subset \overline{LE3}_{uqf}(nm) \overline{E3}_{uqg}(nm) / \overline{E3}_{usd}(nm), \\ \overline{LE3}_{uqf}(nm) \overline{E3}_{uqg}(nm) / \overline{E3}_{usd}(nm) &\subset De3(nm); \end{aligned}$$

Связность подструктур (*il*, *kr*) разных именованных гипердоменов (*s*, *n*) разных уровней сложности (*u*, *z*) определяет связность состояний подструктур (*ila*, *krx*) в разных состояниях (*d*, *p*) разных гипердоменов (*s*, *n*) так же на разных уровнях сложности (*u*, *z*):

$$\begin{aligned} \overline{LE2}_{us}(n) \overline{E2}_{zn}(n) &= \{\overline{e^2_{usil}}, \overline{e^2_{znkr}}\}, \overline{LE2}_{us}(n) \overline{E2}_{zn}(n) \subset De2(n) \\ (\overline{e^2_{usil}}, \overline{e^2_{znkr}}) &\rightarrow \{\overline{e^3_{usdila}}, \overline{e^3_{znprkr}}\} \subset \overline{LE3}_{usd}(nm) \overline{E3}_{znpr}(nm), \\ \overline{LE3}_{usd}(nm) \overline{E3}_{znpr}(nm) &\subset De3(nm); \end{aligned}$$

Компоновка подструктур (*kr*, *kp*) разных именованных гипердоменов (*n*, *m*) одного уровня сложности (*z*) при условии связности этих подструктур с подструктурой (*il*) другого именованного гипердомена (*s*) другого уровня сложности (*u*) определяет компоновку состояний подструктур (*krx*, *kpl*) в разных состояниях (*p*, *q*) разных гипердоменов (*n*, *m*) уровня сложности (*z*) при условии их связности с состоянием подструктуры (*ila*) другого состояния (*d*) гипердомена (*s*) на другом уровне сложности (*u*):

$$\begin{aligned} \overline{LE2}_{zn}(n) \overline{E2}_{zm}(n) / \overline{E2}_{us}(n) &= \{\overline{e^2_{znkr}}, \overline{e^2_{zmkp}}\}, \\ \overline{LE2}_{zn}(n) \overline{E2}_{zm}(n) / \overline{E2}_{us}(n) &\subset De2(n) \\ (\overline{e^2_{znkr}}, \overline{e^2_{zmkp}}) &\rightarrow \\ \{\overline{e^3_{znprkr}}, \overline{e^3_{zmqkpl}}\} &\subset \overline{LE3}_{znpr}(nm) \overline{E3}_{zmq}(nm) / \overline{E3}_{usd}(nm) \end{aligned}$$

Таким образом, определены все взаимосвязи между именованными подструктурами именованных гипердоменов и состояниями подструктур состояний гипердоменов.

Заключение

1. В рамках методологии автоматизации интеллектуального труда остро встает проблема отображения многоуровневых конструкций в существующие СУБД, что требует новых подходов к организации информации и данных в компьютере на более высоком уровне, чем это было до сих пор.

2. Наиболее распространенные модели представления данных, используемые в современных СУБД, не обеспечивают возможности их применения в методологии автоматизации интеллектуального труда.

3. Для того чтобы компьютеры могли обрабатывать информацию на более высоком уровне, необходимо предложить модель описания данных, увязывающую семантику предметной области с ее формализованным представлением в вычислительной среде.

4. Новая модель данных на гипердоменах основана на теоретическом аппарате инфологического моделирования в рамках методологии автоматизации интеллектуального труда.

5. Формально определены элементы новой модели данных – гипердомены, как сложноорганизованных конструкций, которые на объектном уровне представляются именами, на конкретном уровне – своими состояниями.

6. Формально описаны связи, как между именованными конструкциями, так и между состояниями этих конструкций.

7. Формально описаны взаимосвязи именованными структурами и структурами состояний.

8. Для реализации модели данных на гипердоменах требуются дальнейшие теоретические исследования и практические разработки.

Литература

1. Соломенцев Ю.М., Волкова Г.Д. Проблемы развития конструкторско-технологической информатики / Материалы Третьего Международного конгресса "Конструкторско-технологическая информатика" КТИ-96, - М., МГТУ "СТАНКИН", 1996г., с. 128-129.
2. Волкова Г.Д. Об одном подходе к разработке методологии автоматизации интеллектуального труда / Материалы науч.-метод. конференции "Проблемы интеграции образования и науки"-М.:ВНИИТЭМР, 1990г., с.171-172.
3. Волкова Г.Д. Развитие методологии автоматизации интеллектуального труда как теоретической основы создания прикладных автоматизированных систем. Информационные технологии и автоматизированные системы. М.: №1. 2006. с.105-117.
4. Ульман Дж. Основы системы баз данных / -М.: Финансы и статистика, 1983, 334с.
5. Майер Д. Теория реляционных баз данных. /М.: Мир, 1987, 385с.
6. Цаленко М.Ш. Моделирование семантики в базах данных. - М.: Наука. Гл. ред. физ.-мат. лит., 1989. - 288с. - (Проблемы искусственного интеллекта).
7. Калиниченко Л.А., Рывкин В.М. Машины баз данных и знаний. М.: Наука. Гл. ред. физ.-мат. лит., 1990. - 226 с
8. Чен П.П.Ш. Модель "сущность-связь" - шаг к единому представлению данных. /СУБД,1993,№3,с.137-158.
9. Коголовский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2002.-800с.
10. Ирингер М., Кирстен В. Постреляционная СУБД Cach 5
11. Грачева Т.В. ADABAS- основа технологии SOFTWARE AG. /М.: Открытые системы, СУБД, №2, 1995г.
12. Ким Е.К., Шибаев И.Г., Бычков В.А. Проектирование трех мерных баз данных в СУБД uniVerse /М.: Открытые системы, СУБД, №3, 1996г.
13. Цаленко М.Ш. Моделирование семантики в базах данных. - М.: Наука. Гл. ред. физ.-мат. лит., 1989. - 288с. - (Проблемы искусственного интеллекта).
14. Буч Г. Объектно-ориентированное проектирование (с примерами применения) / Киев-Москва, Диалектика, 1992г., 519с.
15. Коголовский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2002.-800с.
16. Лисков Б., Гатэг Дж. Использование абстракций и спецификаций при разработке программ: Пер. с англ. - М.: Мир, 1989. - 424 с.
17. Кузнецов С.Д. Объектно-ориентированные базы данных - основные концепции, организация и управление: краткий обзор / www.citform.ru/database/articles/art_24.shtml
18. Иванов А.Г., Карпова А.В., Семик В.П., Филинов Ю.Е. Объектно-ориентированная среда программирования. Системы и средства информатики. Вып.2. М.: Наука, 1991.
19. Программирование. Специальный выпуск по объектно-ориентированному программированию. N 6, 1990.
20. Ким В. Технологии объектно-ориентированных баз данных//М.: Открытые системы, СУБД, №4, 1994г.
21. Сахаров А.А. Принципы проектирования и использования многомерных баз данных (на примере Oracle Express Server) // М.: Открытые системы, СУБД №3,1996г.
22. Заботнев М.С.. Многомерная модель представления данных по образовательной статистике // Телематика - 2003. Труды X Всероссийской научно-методической конференции. - Санкт-Петербург, 2003. - с. 245-246.
23. Григорьев О.Г., Волкова Г.Д., Шукин М.В. Анализ моделей представления данных в современных СУБД с учетом требований методологии автоматизации интеллектуального труда: Межотраслевая информационная служба: Науч.-метод.журнал - М.: ФГУП «ВИМИ» 2010, вып.3(154), стр.12-28.
24. Волкова Г.Д. Методология автоматизации интеллектуального труда. Часть 4.: Межотраслевая информационная служба: Науч.-метод.журнал - М.: ФГУП «ВИМИ» 2010, вып.1(150), стр.4-30.7
25. Волкова Г.Д., Григорьев О.Г., Новый подход к организации данных в вычислительной среде – модель данных на гипердоменах //Труды XVII международной научно-технической конференции «Информационные средства и технологии». В 3 томах. Т.1-М.: Издательский дом МЭИ, 2010, стр.97-105.

Волкова Галина Дмитриевна. Профессор кафедры «Информационные технологии и вычислительные системы» МГТУ «Станкин». Окончила Московский авиационный институт им. С.Орджоникидзе в 1976 году. Доктор технических наук. Автор 110 печатных работ. Область научных интересов: когнитивные технологии, автоматизация интеллектуального труда, концептуальное моделирование предметных задач, представление и обработка знаний.
E-mail: it_ktp@stankin.ru.

Григорьев Олег Георгиевич. Заместитель директора Института системного анализа РАН. Окончил Московский институт электронного машиностроения в 1980 году. Доктор технических наук. Автор 35 печатных работ.
E-mail: olegg@polikvart.ru.