ГОСТ 34.320—96

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

Информационные технологии
Система стандартов по базам данных

КОНЦЕПЦИИ И ТЕРМИНОЛОГИЯ ДЛЯ КОНЦЕПТУАЛЬНОЙ СХЕМЫ И ИНФОРМАЦИОННОЙ БАЗЫ

Издание официальное

БЗ 3—2001

МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ
ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
Минск

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Институтом программных систем НАН Украины

ВНЕСЕН Государственным комитетом Украины по стандартизации, метрологии и сертификации

2 ПРИНЯТ Межгосударственным Советом по стандартизации, метрологии и сертификации (протокол № 10 от 3 октября 1996 г.)

За принятие проголосовали:

Наименование государства	Наименование национального органа по стандартизации
Азербайджанская Республика	Азгосстандарт
Республика Армения	Армгосстандарт
Республика Беларусь	Госстандарт Республики Беларусь
Республика Казахстан	Госстандарт Республики Казахстан
Республика Кыргызстан	Кыргызстандарт
Республика Молдова	Молдовастандарт
Российская Федерация	Госстандарт России
Республика Таджикистан	Таджикстандарт
Туркменистан	Главгосинспекция «Туркменстандартлары»
Республика Узбекистан	Узгосстандарт
Украина	Госстандарт Украины

3 Настоящий стандарт соответствует международному стандарту ISO/TR 9007:1987 «Concepts and terminology for the conceptual schema and the information base»

4 Постановлением Государственного комитета Российской Федерации по стандартизации и метрологии от 22 февраля 2001 г. № 87-ст межгосударственный стандарт ГОСТ 34.320—96 введен в действие непосредственно в качестве государственного стандарта Российской Федерации с 1 июля 2001 г.

5 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 2001

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания на территории Российской Федерации без разрешения Госстандарта России

Содержание

1 Область применения 1
2 Проблематика концептуальной схемы и информационной базы 1
2.1 Проблемная область 1
2.2 Описание проблемной области 2
2.3 Статические и динамические аспекты концептуальной схемы и информационной базы . . 2
2.4 Взаимодействие между реальным миром и информационной системой 2
2.5 Роль пользователей и информационных процессоров 3
2.6 Общие принципы описания проблемной области 3
2.7 Общие принципы определения содержимого концептуальной схемы 3
2.8 Роли концептуальной схемы 4
2.9 Требования к средству поддержки концептуальной схемы 4
3 Основные понятия для концептуальной схемы и информационной базы 4
3.1 Основные понятия и определения для действий над концептуальной схемой и информационной базой 6
3.2 Поведение информационного процессора 8
3.3 Заполнение концептуальной схемы. Минимальная концептуальная схема 9
3.4 Правила поведения для среды 9
3.5 Статические и динамические правила и ограничения 9
3.6 Формулирование правил и ограничений 10
3.7 Координация допустимых действий 11
4 Понятия и принципы реализации 13
4.1 Принципы заполнения концептуальной схемы 13
4.2 Принципы описания проблемной области 13
4.3 Абстрактный синтаксис для концептуальной схемы и информационной базы 14
4.4 Семантика концептуальной схемы и информационной базы 14
4.5 Принципы составления концептуальной схемы 14
4.6 Трехуровневая архитектура 15
4.7 Модель системы словарей информационных ресурсов 16
4.8 Соответствие трехуровневой архитектуры информационных систем и эталонной модели архитектуры открытых систем 16
5 Подходы к моделированию 17
Приложение А Алфавитный указатель терминов и определений 18
Приложение Б Пример проблемной области 20
Приложение В Система синтаксических обозначений Паскаля 22
Приложение Г Подходы сущность—атрибут—связь 23
Приложение Д Подходы на основе бинарных отношений 30
Приложение Е Подходы на основе интерпретируемой логики предикатов 35

М Е Ж Г О С У Д А Р С Т В Е Н Н Ы Й С Т А Н Д А Р Т

Информационные технологии
Система стандартов по базам данных

КОНЦЕПЦИИ И ТЕРМИНОЛОГИЯ ДЛЯ КОНЦЕПТУАЛЬНОЙ СХЕМЫ И ИНФОРМАЦИОННОЙ БАЗЫ

Information technology.
Database standards system. Concepts and terminology for the conceptual schema and the information base

Дата введения 2001—07—01

1 Область применения

Настоящий стандарт устанавливает основные понятия и термины концептуальных схем и информационных баз, охватывающие разработку, описание и применение концептуальных схем и информационных баз, манипулирования информацией, а также описание и реализацию информационного процесса.

Стандарт определяет роль концептуальной схемы. Положения, изложенные в стандарте, носят рекомендательный характер и могут использоваться для оценки систем управления базами данных (СУБД).

Стандарт не описывает конкретные методы применения средств поддержки концептуальных схем.

Описанные в стандарте языки концептуальных схем не следует рассматривать как стандартные.

2 Проблематика концептуальной схемы и информационной базы

Концептуальная схема описывает содержимое базы данных, которое включает перечень действий, допустимых над этими данными.

Концептуальное представление концентрирует внимание на смысле информации. Концептуальная схема описывает это представление. Внешние представления сосредоточены на том, как представлена информация для пользователя. Внутреннее представление сосредоточено на внутреннем физическом представлении данных в компьютерной системе и описывается во внутренней схеме.

Такая архитектура принимается за основу в стандарте. Она должна играть ключевую роль в системном анализе и проектировании баз данных. Концептуальная схема должна быть как моделью предметной области, полученной в результате системного анализа, так и центральным узлом между представлениями пользователя и проектом базы данных на физическом уровне.

Необходимо проводить четкое разделение между смыслом информации, внешним представлением данных и внутренней физической структурой хранения данных.

2.1 Проблемная область

Основной характеристикой баз данных является совместное использование общих данных многими пользователями системы. Совместно используя общие данные, пользователи устанавливают диалог друг с другом через систему. Должно существовать какое-то общее понимание информации, представленной данными. Общее понимание должно относиться к чему-либо внешнему по отношению к пользователям. Оно должно быть зафиксировано. Для этого необходима общая предварительно определенная грамматика.

Объекты и события, к которым относится общее понимание информации, являются проблемной областью. Проблемные области могут быть как вещественными, так и абстрактными.

В стандарте к проблемным областям применяется неформальный подход.

Издание официальное

Типичная проблемная область состоит из реальных и абстрактных объектов, которые являются сущностями. Она состоит из классов сущностей. Эта классификация основывается на сходстве и учитывает характеристики, общие для нескольких сущностей. Выбор характеристик для группировки сущностей в классы произволен; выбор осуществляется прагматически, в зависимости от целей проблемной области.

Общие свойства сущностей неформально характеризуются как «классификации», «правила», «законы» или «ограничения», касающиеся состояния и поведения сущностей в проблемной области.

То, что считается частью проблемной области, зависит от времени. Это относится к объектам, событиям, классификациям, правилам, законам и т. д.

2.2 Описание проблемной области

В соответствии с принятой терминологией информация о проблемной области «описывает» или «моделирует» эту область.

Информация, содержащаяся в системе обработки данных, дает описание проблемной области. Конкретное физическое представление этой информации является базой данных.

Концептуальная схема указывает на сущности, которые могут существовать в проблемной области, т. е. на сущности, которые существуют, существовали или могли когда-либо существовать. Она также указывает на факты и события, которые являются возможными или обязательными для этих сущностей. Вся другая информация о сущностях и их фактическом состоянии в определенный момент или период времени относится к информационной базе.

Каждое описание обязательно имеет форму представления. Для того чтобы описание можно было передавать, необходима интерпретация представления, в первую очередь смысл описания. Термин «информация» следует использовать, если есть необходимость в интерпретации, а термин «данные» — если необходимо сконцентрировать внимание на формах представления информации.

Концептуальную схему и информационную базу следует рассматривать как находящиеся на концептуальном уровне и обеспечивающие концептуальное представление информации о проблемной области.

Базу данных или ее части с точки зрения пользователя системы (строки данных) следует рассматривать как находящиеся на внешнем уровне, представляя внешнее представление информации о проблемной области. Формы внутреннего хранения в компьютере относятся к внутреннему уровню, который является внутренним представлением информации проблемной области. Для внешнего и внутреннего представлений формы представления имеют первостепенное значение. Интерпретация этих форм является интерпретацией, которая задана в концептуальном представлении.

2.3 Статические и динамические аспекты концептуальной схемы и информационной базы

Набор понятий для концептуальной схемы должен учитывать статические и динамические аспекты. Концептуальная схема может изменяться для правильного представления изменений в выбранной части реального или предполагаемого мира. Динамические аспекты рассматриваются в описаниях манипуляций, необходимых для ознакомления с содержимым концептуальной схемы и информационной базы.

Время появления изменений в проблемной области и время внесения соответствующих изменений в концептуальную схему и информационную базу могут быть не связаны между собой: изменения в проблемной области могут быть зарегистрированы ретроспективно, даже в другой последовательности. В других случаях временны̂е масштабы так тесно связаны друг с другом, что концептуальная схема и информационная база при необходимости становятся частью проблемной области; в этих случаях особенно описание такого взаимодействия должно также быть частью динамических аспектов.

Между статическими и динамическими аспектами не определена четкая граница; она может изменяться в различных подходах, а в некоторых случаях — отсутствовать.

2.4 Взаимодействие между реальным миром и информационной системой

Информационный процессор — это процессор, который осуществляет изменения в концептуальной схеме и информационной базе. Информационная система — это система, состоящая из концептуальной схемы, информационной базы и информационного процессора.

Информационный процессор производит изменения в информационной базе или концептуальной схеме только при получении сообщения. Сообщение содержит информацию и/или выражает команды. Сообщения исходят из части реального мира, которую называют средой, которая может быть или не быть внешней по отношению к проблемной области. При получении сообщения, содержащего команду, информационный процессор может выдавать информацию, имеющуюся в концептуальной схеме и информационной базе.

Термин «информационная система» необходимо использовать пользователям информационной системы вне среды.

2.5 Роль пользователей и информационных процессоров

Пользователями информационной системы могут быть машины или другие системы, а также люди. Пользователь — это некто или нечто, посылающий команды и сообщения в информационную систему и получающий сообщения от информационной системы. Он является частью среды. Некоторые пользователи могут иметь полномочия устанавливать правила поведения или ограничения для информационной системы.

Информационный процессор передает сообщения между средой и информационной базой или концептуальной схемой. Он не имеет своей инициативы, а может только в точности следовать правилам.

Информационный процессор является компьютерной системой или ее частью. Человек тоже может играть роль информационного процессора при условии, что он не нарушает установленных правил и не действует по своей инициативе. С другой стороны, компьютерные системы могут действовать как пользователи информационной системы. Примером может служить сеть информационных систем, обменивающихся сообщениями друг с другом. Если каждая из них имеет набор правил, независимый от других, тогда каждая выполняет роль пользователя других информационных систем. Поэтому роль определяет, должно ли нечто рассматриваться как пользователь или как информационный процессор.

Определение пользователей и информационных процессоров в терминах ролей подразумевает, что среда и информационная система необязательно должны быть разделены. Если в информационной базе содержится информация о пользователях информационной системы, то среда и проблемная область не будут разделены. Но даже если они не разделены, их всегда можно различить.

2.6 Общие принципы описания проблемной области

Концептуальная схема и информационная база должны описывать концептуальное представление. Это подразумевает, что концептуальная схема определяется в терминах и конструкциях, относящихся к объектам в самой проблемной области и выражающих состояние этих объектов.

Конструкции, используемые в концептуальной схеме и информационной базе, должны опираться на формальную логику в качестве теоретического обоснования. Можно ограничиться простейшими конструкциями, выражающими фундаментальные понятия. Но всегда можно определить над этими основами более сложные конструкции («макроконструкции»), которые могут быть более удобными или эффективными для описания различных аспектов проблемной области.

Выбор конкретных макроконструкций основывается на таких практических соображениях, как легкость понимания и использования. Этот выбор зависит от области применения информационной системы, для которой предусматривается концептуальная схема и информационная база.

Необходимо тщательно провести различие между сущностями и их описаниями. В этих описаниях сущности идентифицируются именами, которые ссылаются на сущности. Сюда входят синонимы — различные имена, относящиеся к одной и той же сущности, и омонимы — идентичные имена, относящиеся к различным сущностям.

Концептуальная схема описывает статические и динамические аспекты и зависимости проблемной области. Она определяет описания, которые могут присутствовать в концептуальной схеме и информационной базе, и манипуляции, допустимые над описаниями. Для этого необходимы конструкции как для самих описаний, так и для манипулирования ими в информационной системе.

2.7 Общие принципы определения содержимого концептуальной схемы

Системный аналитик должен учитывать ряд факторов при определении содержания концептуальной схемы. Среди них могут быть:

- описания классов (типов) сущностей проблемной области, а не отдельных экземпляров;
- описания понятий, менее подверженных изменениям;
- включение правил или ограничений, имеющих широкое воздействие на поведение проблемной области (и поэтому на поведение концептуальной схемы и информационной базы).

В любом случае должны соблюдаться общие принципы концептуальной схемы:

П р и н ц и п 100 %

Принцип, согласно которому все общие аспекты, т. е. все правила, законы и т. д., проблемной области должны быть описаны в концептуальной схеме, причем информационная система не может нести ответственность за несоблюдение правил и законов, описанных не в концептуальной схеме.

Принцип концептуализации

Принцип, согласно которому концептуальная схема должна включать статические и динамические аспекты проблемной области только концептуального уровня, не касаясь внешних и внутренних аспектов представления и организации данных (физической организации данных и доступа к ним, аспектов представления, касающихся отдельных пользователей).

2.8 Роли концептуальной схемы

Основные роли концептуальной схемы:

- обеспечить общую основу для понимания общего поведения проблемной области;
- определить допустимую эволюцию и манипулирование информацией о проблемной области;
- обеспечить основу для интерпретации внешних и внутренних синтаксических форм, представляющих информацию о проблемной области;
- обеспечить основу для преобразований внешних и внутренних схем.

2.9 Требования к средству поддержки концептуальной схемы

Средство поддержки концептуальной схемы должно:

- обеспечивать основные понятия, пригодные для адекватного описания как статических, так и динамических аспектов проблемной области, и тем самым описания в терминах концептуальной схемы и информационной базы;
- иметь язык, на котором можно описать концептуальную схему таким образом, чтобы она была понятна пользователям;
- обеспечить язык описания концептуальной схемы, который мог бы интерпретироваться компьютером;
- иметь удобные средства внесения изменений в концептуальную схему для отражения изменений в общих классификациях, правилах, законах и т. д. проблемной области и для предсказания непосредственных последствий таких изменений.

Представление информации, необходимой для различных пользователей, не должно противоречить утверждениям в концептуальной схеме. Если такие внешние схемы подлежат изменению, средство поддержки должно быть таким, чтобы изменение не влияло на концептуальную схему. Концептуальная схема должна быть инвариантной по отношению к изменениям во внутреннем (физическом) представлении данных в компьютере.

3 Основные понятия для концептуальной схемы и информационной базы

Раздел содержит пояснения основных понятий для концептуальной схемы и информационной базы. Определения приведены в указателе терминов в приложении А.

сущность: Любой конкретный или абстрактный объект, включая связи между объектами.

высказывание: Возможное состояние сущностей, по поводу которого можно утверждать или отрицать, что такое состояние имеет место.

Высказывание может относиться к одной сущности, нескольким отдельным сущностям, к группе сущностей и т. д.

лингвистический объект: Грамматически допустимая языковая конструкция.

Лингвистические объекты сами могут считаться сущностями.

Обмениваться информацией о сущностях можно только посредством описаний высказываний.

предложение: Лингвистический объект, представляющий определенное высказывание.

Предложения состоят из термов и предикатов.

терм: Лингвистический объект, обозначающий сущность.

предикат: Лингвистический объект, аналогичный глаголу, сообщающий что-либо о сущностях, обозначенных термами.

Например, предложение «Автомобиль РСХХ999 является моделью Мустанг» является примером высказывания. В этом предложении глагол «является» представляет собой предикат.

Термы «автомобиль РСХХ999» и «модель Мустанг» относятся к сущностям.

лексический объект (имя): Простой лингвистический объект, используемый только для ссылки на сущность.

Как правило, лексический объект состоит из одного или нескольких существительных.

Особый вид связи между «основными» сущностями и лексическими объектами, относящимися к ним, называется соглашением об именовании. Если такое соглашение между сущностью и лексическим

объектом установлено корректно, всегда возможно, хотя бы в принципе, выявить причинную цепочку по поводу использования этого лексического объекта.

Часть работы проектировщика информационной системы состоит в том, чтобы гарантировать, что на все рассматриваемые сущности можно каким-то образом сослаться. По этой причине проектировщику информационной системы требуется дополнительное описание (в концептуальной схеме и информационной базе) согласованных способов обращения к сущностям.

синонимы: Различные термы, обозначающие одну и ту же сущность.

омонимы: Идентичные термы, обозначающие различные сущности.

Омонимы могут привести к некоторой неоднозначности. В некоторых информационных системах они запрещены. Это запрещение не является фундаментальным требованием.

пространство сущностей: Совокупность сущностей, воспринимаемых как единое целое.

В одно и то же или в различное время можно различать много разных пространств сущностей. Кроме того, сущность может принадлежать многим пространствам сущностей.

пространство высказываний: Совокупность высказываний, каждое из которых истинно для данного пространства сущностей. Совокупность предложений, выражающих высказывания пространства высказываний, содержит информацию о соответствующем пространстве сущностей.

проблемная область: Все сущности, являющиеся объектом интереса, которые были, есть или когда-либо могут быть.

Проблемная область является областью возможных сущностей. Проблемная область ограничивается возможными сущностями, представляющими для нас интерес. В примере из приложения Б проблемная область Органа Регистрации состоит из всех автомобилей, изготовителей, моделей изготовителей, гаражей, людей и т. д., которые существовали, существуют или когда-либо будут существовать и которые интересуют Орган Регистрации.

Все высказывания, которые могут выполняться в одном или нескольких пространствах сущностей, вместе составляющих проблемную сферу, формируют область возможных высказываний. Проектировщика информационной системы должны интересовать те высказывания, которые выполняются во всех возможных пространствах сущностей.

необходимое высказывание

Высказывание, которое должно содержаться во всех возможных пространствах высказываний и является истинным во всех пространствах сущностей.

Необходимые высказывания определяют состояния, обязательные для всех сущностей во всех пространствах сущностей. Эти высказывания представляют абстракцию всех пространств сущностей, обобщающую то, что у них общее.

Необходимые высказывания определяют, какие сущности могут встретиться в любом пространстве сущностей (возможные сущности). В соответствующих случаях они определяют также, какие сущности должны появляться в каждом пространстве сущностей (необходимые сущности).

Классификации, правила, законы и т. д. проблемной области, описанные в разделе 2, образуют необходимые высказывания.

класс [сущностей]: Сущности проблемной области, для которых данное высказывание является истинным.

Каждый класс сущностей полностью определяется его возможными элементами. Любая сущность может быть элементом многих классов, поэтому классы могут пересекаться.

Сами классы являются сущностями, и им могут быть даны имена.

тип [сущности]: Высказывание, устанавливающее, что сущность принадлежит к определенному классу сущностей.

Подразумевается, что такой класс сущностей существует.

На тип можно ссылаться посредством имени типа. Для обозначения имени типа используется имя (существительное) в единственном числе, а для обозначения имени класса — имя (существительное) во множественном числе.

Проектировщик информационной системы произвольно определяет, следует ли связывать понятие типа с определенным классом сущностей. Это зависит от того, что считается удобным или обычным для пользователей концептуальной схемы.

реализация (экземпляр) [типа сущности]: Отдельная сущность, для которой истинно некоторое высказывание по поводу ее типа.

При проектировании информационных систем понятия класса и типа используются, в частности, для установления совокупности необходимых высказываний. С некоторым классом или типом

может быть связана совокупность соответствующих необходимых высказываний, которые выполняются для возможных сущностей, являющихся элементами этого класса.

Высказывания, определяющие такие классы или типы, сами относятся к необходимым высказываниям.

концептуальная схема: Непротиворечивая совокупность предложений, выражающих необходимые высказывания, относящиеся к проблемной области.

Все возможные пространства сущностей, составляющие проблемную область, имеют общую концептуальную схему. Эта концептуальная схема определяет проблемную область, так как она несет информацию о том, какой именно должна быть совокупность всех возможных сущностей.

Какие высказывания являются необходимыми, а следовательно, какими будут границы концептуальной схемы, определяется произвольно и зависит от того, какой степени подробности добивается разработчик информационной системы. Эти границы со временем могут изменяться, требуя дополнительных изменений в уже сформулированной концептуальной схеме.

информационная база: Совокупность предложений, выражающих высказывания, отличные от необходимых высказываний, согласующиеся друг с другом и с концептуальной схемой, а также истинные в некотором пространстве сущностей.

Совокупность предложений, составляющих эту концептуальную схему и информационную базу, вместе описывают все высказывания, относящиеся к некоторому пространству сущностей, и тем самым дают описание пространства высказываний для этого пространства сущностей. Предполагается, что эти высказывания выполняются для этого пространства сущностей, воспринимаемого как «реальность». Совокупность предложений, составляющих концептуальную схему и информационную базу, должна обязательно быть непротиворечивой, если она претендует на то, чтобы быть истинным описанием этих высказываний.

Информационная база вместе с концептуальной схемой определяют некоторое пространство сущностей. Пространство сущностей состоит в точности из тех конкретных и абстрактных объектов, сущностей, на которые можно сослаться с помощью термов в предложениях, содержащихся в информационной базе и концептуальной схеме, вместе взятых.

Возможно описание одной проблемной области или одного определенного пространства сущностей в более чем одной концептуальной схеме и информационной базе.

Информационная база может предназначаться для того, чтобы выдавать информацию о сущностях, которые присутствуют в момент или в период времени, обычно называемый «сейчас». Тем не менее, «текущее» состояние информационной базы – актуальная информационная база — может относиться к «прошлому» или «будущему» пространства сущностей.

актуальная информационная база: Информационная база, существующая в указанный момент или период времени, который классифицируется как «сейчас», и отражающая дополнительные высказывания, отличные от необходимых.

актуальное пространство сущностей: Совокупность сущностей, которая описана в актуальной информационной базе и ее концептуальной схеме.

3.1 Основные понятия и определения для действий над концептуальной схемой и информационной базой

Информационная база и концептуальная схема со временем изменяются, отражая изменения в выбранной части мира, составляющей проблемную область, так как в информационной базе и концептуальной схеме должны быть только предложения, истинные в этой проблемной области.

Изменения в выбранной части мира могут быть, например, такими:

- сущности, появляющиеся или исчезающие в выбранной части;
- сущность, изменяющая свое состояние или связи с другими сущностями;
- классификация сущностей или какие-либо правила или ограничения, относящиеся к изменяющимся сущностям;
- исследуемая область изменяется в том смысле, что выбранная часть мира расширяется или сокращается.

Такие изменения могут повлечь за собой изменения и информационной базы, и концептуальной схемы. Влияние первых двух типов изменений может ограничиться только информационной базой, последние два типа вызовут также изменения и в концептуальной схеме.

Основным понятием манипулирования информацией в информационной базе или концептуальной схеме является элементарное действие. Определено три вида элементарных действий: вставка, удаление и выборка.

вставка: Добавление предложения к информационной базе или концептуальной схеме.

Другие предложения, не являющиеся логически выводимыми до вставки, могут стать таковыми и, таким образом, становятся выводимой частью информационной базы или концептуальной схемы.

Логически выводимое предложение само по себе фактически не вставляется.

удаление: Извлечение ранее вставленного предложения из информационной базы или концептуальной схемы.

Выводимое предложение, которое не может быть выводимо без исключенного предложения, больше не будет выводимым и поэтому не будет больше выводимой частью информационной базы или концептуальной схемы.

Выводимое предложение может быть одновременно вставлено явно. В этом случае исключение другого предложения, от которого существенно зависит выводимость данного предложения, не приводит к автоматическому удалению явно вставленного предложения. Исключение этого другого предложения только сделает невозможным выведение явно вставленного предложения.

выборка: Поиск и выдача предложения, которое есть в информационной базе или концептуальной схеме или выводится из предложений, имеющихся в них.

Выборка выводимого предложения из информационной базы или концептуальной схемы возможна только в случае, если информационная система знает, как вывести это предложение из других доступных или выводимых предложений в концептуальной схеме и информационной базе.

Допустимы комбинации элементарных действий, предназначенные для получения определенного результата.

действие: Одно или несколько элементарных действий, которые в качестве единого целого изменяют определенную совокупность предложений в информационной базе или концептуальной схеме и/или предоставляют о них информацию.

Типичным примером действия является замена определенного предложения другим, т. е. удаление, за которым следует вставка. Поскольку этот класс действий встречается часто, целесообразно определить его как вид действия.

модификация: Замена предложения в информационной базе или концептуальной схеме на другое.

Для исключения недопустимых действий и управления ими необходимо определить для действий правила и ограничения. Поэтому введем определение действий, рассматриваемых как атомарные единицы исполнения.

допустимое действие: Действие, удовлетворяющее установленным правилам или ограничениям и изменяющее непротиворечивую совокупность предложений в информационной базе или концептуальной схеме на другую непротиворечивую совокупность предложений и/или предоставляющее информацию о непротиворечивой совокупности предложений в информационной базе или концептуальной схеме.

Только окончательная совокупность предложений, полученная в результате допустимого действия, должна быть непротиворечивой совокупностью предложений. Если промежуточные совокупности предложений распознаваемы, они не обязательно непротиворечивы.

Определенные допустимые действия могут изменять предположительно непротиворечивую, но в действительности не «истинную» совокупность предложений, не соответствующую действительности, в непротиворечивую и истинную. Допустимые действия требуются для корректировки разрушенных информационных баз или концептуальных схем, какова бы ни была причина разрушения. Таким действиям разрешается игнорировать некоторые правила о допустимых или требуемых последовательностях состояний совокупностей предложений.

элементарное действие: Вызывается элементарной командой информационной системы.

элементарная команда: Приказ или сигнал для выполнения элементарного действия.

И действие, и допустимое действие вызываются командой информационной системы.

команда: Приказ или сигнал для выполнения действия. Если допустимость действия будет нарушена, в ответ на команду действие может быть проигнорировано.

Описание команд и действий должно быть дано на подходящем языке.

командное предложение: Лингвистический объект, являющийся командой.

Предъявление командного предложения информационной системе представляет собой команду.

Необходимы также средства для записи комбинаций элементарных действий и их идентификации как единого целого, определяющего неделимое действие или допустимое действие.

описание действия: Лингвистический объект, описывающий действие.

Синтаксическая и семантическая сложность командных предложений и описаний действий зависит от выбранного языка.

3.2 Поведение информационного процессора

Взаимодействие между средой и информационной системой осуществляется посредством сообщений.

сообщение: Совокупность, состоящая из одного или более предложений, возможно командных предложений, и использующаяся как элемент информационного обмена между средой и информационной системой.

Сообщения обрабатываются информационным процессором информационной системы.

информационный процессор: Процессор, который в ответ на команду выполняет действие над концептуальной схемой и/или информационной базой.

Информационный процессор распознает, относятся ли сообщения, полученные из среды, к данному языку. Сообщения, не относящиеся к данному языку, отбрасываются. Правильные сообщения могут выражать изменение в проблемной области или требовать извлечения одного или нескольких предложений, имеющихся в концептуальной схеме или информационной базе или выводимых из присутствующих в них предложений.

Сообщение, выражающее изменение в проблемной области, должно содержать командное предложение или сопровождаться им. Это предложение ссылается на описание действий для того, чтобы допустимое действие произвело соответствующее изменение в концептуальной схеме и информационной базе. Информационный процессор интерпретирует команду и изменяет информационную базу или концептуальную схему в соответствии с ограничениями и правилами поведения. Эти правила не только определяют, будет ли непротиворечивой полученная совокупность предложений в концептуальной схеме и информационной базе, но также решают, можно ли вообще изменять концептуальную схему и информационную базу в зависимости от того, какие предложения в них уже имеются.

Информационный процессор игнорирует команду, если изменение не может быть произведено в соответствии с ограничениями или правилами поведения. В результате этого информационная база и концептуальная схема остаются без изменения.

Если сообщение выражает команду для извлечения совокупности предложений, имеющихся в концептуальной схеме и информационной базе или выводимых из них, информационный процессор интерпретирует команду. Он выдает сообщение, содержащее соответствующую совокупность предложений согласно ограничениям или правилам поведения, которые определяют, когда и какая совокупность предложений, присутствующих в концептуальной схеме и информационной базе, должна быть результатом команды. Эти правила должны включать правила логического вывода, если применяются выводимые предложения.

Информационный процессор игнорирует команду, если при выборке необходимых предложений нарушаются ограничения или правила поведения.

Когда выдается команда, среде необходимо знать, принята ли команда или отвергнута. Поэтому информационный процессор должен выдавать среде сообщения по этому поводу.

Результат допустимого действия — это функция «управляющих» предложений, формулирующих правила и ограничения предложений, подлежащих изменению или выборке, и входного сообщения, включающего командное предложение и дополнительные предложения, если это необходимо.

Управляющие предложения находятся в концептуальной схеме, хотя предложения в информационной базе также могут играть управляющую роль. Выбираемые или изменяемые предложения находятся в информационной базе. Но эти предложения могут быть и в концептуальной схеме в тех случаях, когда сама концептуальная схема подлежит изменению или выборке. В последнем случае некоторые из управляющих предложений могут быть также найдены и в информационной базе.

Предложение, выражающее высказывание как таковое, имеет значение истинности. Значением истинности является:

- «истина», если оно явно объявлено истинным посредством сообщения, принятого от среды согласно правилам, известным информационному процессору. Оно также считается истинным, если предложение выводимо из таких явно сформулированных предложений в соответствии с правилами вывода, известными информационному процессору;
- «ложь», если его отрицание может быть подобным образом выведено из явно сформулированных предложений.

Значение истинности всех других предложений считается неопределенным.

Информационный процессор имеет двоякую задачу: принятие решения о допустимости изменения совокупности предложений и выведение «новых» предложений или генерация выходных сообщений.

3.3 Заполнение концептуальной схемы. Минимальная концептуальная схема

Первоначально информационному процессору известны только правила, встроенные в него. Информационный процессор обеспечивается механизмом интерпретации, который делает возможным распознавание и интерпретирование некоторого минимального языка. Этот язык позволяет среде:

- расширить язык для включения конструкций, подходящих для описания проблемной области;
- указывать команды;
- указывать полномочия;
- устанавливать новые правила поведения и логического вывода, относящиеся к концептуальной схеме и информационной базе;
- осуществлять описания действий для допустимых действий.

Далее информационный процессор будет снабжен набором алгоритмов, позволяющих ему выводить новые предложения из уже имеющихся.

Эти встроенные правила поведения составляют минимальную концептуальную схему информационной системы.

Имея эту встроенную минимальную концептуальную схему и первоначально пустую информационную базу, среда может использовать этот минимальный язык для систематического построения требуемой концептуальной схемы, а также информационной базы.

Должно существовать фундаментальное правило, касающееся вставки нового правила или ограничения, определяющее, что предпринять в том случае, когда эта вставка сделает противоречивыми определенные совокупности предложений, уже присутствующих в концептуальной схеме и информационной базе. Такое правило может игнорировать существующие предложения или предлагаемое новое правило. Оно не должно принимать и то, и другое.

Первоначально любое предложение, принятое посредством встроенных правил поведения, становится дополнительным правилом или ограничением и включается в концептуальную схему или в информационную базу. По мере добавления предложений число правил или ограничений, определенных в концептуальной схеме и информационной базе, увеличивается и все более ограничивает и управляет допустимыми действиями над концептуальной схемой и информационной базой.

3.4 Правила поведения для среды

Упомянутые выше правила поведения определяют поведение информационного процессора и то, что допустимо в информационной базе или концептуальной схеме.

Информационные системы могут выдавать среде сообщения, предназначенные для того, чтобы вызвать в ней изменения. Соответственно должны быть расширены правила поведения в рамках таких информационных систем. Однако эта ситуация не подразумевает, что информационная система управляет средой. Эти сообщения генерируются согласно правилам или ограничениям, установленным средой. Информационная система не может заставить среду подчиняться правилам, выраженным такими сообщениями.

Дополнительные правила состоят не только из допускающих и предписывающих правил информационного процессора, но также и из допускающих и предписывающих правил для среды. Допускающие правила для среды устанавливают критерии, необходимые информационному процессору для проверки фактической информации, чтобы система могла выдавать предупреждающие сообщения. Предписывающие правила для среды позволяют информационному процессору посылать необходимые запросы к среде.

Эти последние правила поведения для среды никогда не диктуют, что допустимо в информационной базе или концептуальной схеме, включая правила непротиворечивости для совокупностей предложений. Что касается информационной системы, они формулируют только то, что желательно. Как таковые они составляют отдельный класс правил или ограничений в информационной системе.

3.5 Статические и динамические правила и ограничения

Статическими являются аспекты системы, которые применимы для каждого из ее отдельных состояний. Статические правила или ограничения устанавливают зависимости между частями системы в любой момент времени. Динамическими являются аспекты, которые управляют развитием системы во времени. Рассматривать динамические аспекты системы означает рассматривать законы ее изменения. Динамические правила устанавливают зависимости между частями системы, развивающейся во времени.

Статические правила и ограничения связаны с непротиворечивостью и допустимостью совокупности предложений. Сфера влияния статических правил может быть ограничена единичными предложениями, или она может включать в себя несколько предложений в пределах одной совокупности предложений в одном состоянии информационной базы.

Динамические правила связаны с допустимыми переходами от одной совокупности предложений к другой, определяя возможные последовательности состояний информационной базы.Они являются правилами перехода. Правила перехода абстрагируются от причин изменений, а также от влияния, оказываемого этими изменениями на среду. Они указывают, какие состояния информационной базы могут следовать после других данных состояний информационной базы.

Соблюдение или нарушение правил перехода может контролироваться проверкой состояний. Статические правила могут рассматриваться как специальные случаи правил перехода. Хотя статические правила определяют, какие состояния допустимы, статическое правило можно интерпретировать как утверждение, что определенные состояния допустимы или запрещены независимо от того, каким было предыдущее состояние. То же самое можно выразить в виде правила перехода. Фразу «Независимо от того, каким было предыдущее состояние» можно понимать так: любое из возможных состояний информационной базы может выступать в качестве предыдущего состояния.

Допустимость состояния информационной базы может зависеть от любого из предыдущих состояний, а не только от самого последнего. История состояний может воздействовать на допустимость очередного состояния лишь в той степени, в какой она отражена в самом последнем состоянии. Поэтому достаточно учитывать только последнее состояние.

Такие правила являются статическими, так как они применяются к предложениям в одном состоянии информационной базы. Но их можно считать динамическими, так как они управляют процессом вывода.

Выводимые предложения не только могут порождаться в процессе вывода в информационной системе. Такие предложения могут быть вставлены явно. В таких случаях «правила вывода» гарантируют непротиворечивость полученной совокупности предложений, включая вставленные «выводимые» предложения.

3.6 Формулирование правил и ограничений

В ориентированных на состояния описаниях правила и ограничения формулируются как требования, предъявляемые к последовательным состояниям информационной базы. Тогда правило или ограничение является по сути описанием множества пар состояний информационной базы <СТАРОЕ, НОВОЕ>. Переход из состояния СТАРОЕ в состояние НОВОЕ допустим, независимо от того, каким образом он осуществляется, тогда и только тогда, когда пара <СТАРОЕ, НОВОЕ> присутствует в этом множестве.

Правило перехода различает допустимые и запрещенные пары состояний информационной базы. Оно может рассматриваться как двоичная функция, которая помечает каждую пару состояний информационной базы или «+» (допустимая), или «—» (запрещенная), что определяет дихотомию декартова произведения множества всех состояний информационной базы самой на себя. Такая дихотомия есть декомпозиция на два множества – $T+$ допустимых и $T-$ запрещенных изменений. $T+$ можно назвать положительным, а $T-$ отрицательным расширением правила. Для описания дихотомии можно использовать любое из двух множеств. Эффект правила перехода полностью определяется как его положительным, так и отрицательным расширением.

В ориентированных на действие описаниях допустимые изменения определяются последовательностями допустимых действий. Допустимость действий или последовательности действий может зависеть от текущего состояния. Поэтому правило или ограничение является множеством пар, каждая из которых состоит из компонента состояния информационной базы и компонента последовательности действий.

В ориентированных на действие описаниях правило определяет, что переход допустим, если, начиная от допустимого состояния СТАРОЕ, переход осуществляется посредством последовательности действий Q такой, что пара <СТАРОЕ, Q> присутствует в множестве, заданном правилом. Допустимость может прослеживаться назад, до начального состояния и всех действий, осуществленных над ним вплоть до текущего момента.

Для того чтобы получить конечные описания теоретически бесконечных последовательностей действий (на длину последовательностей действий ограничений нет), необходимо определить классы действий. Ориентированное на действие правило определено на классах последовательностей действий. В частности, последовательность действий может состоять только из одного элементарного действия.

Сложные правила образуются посредством композиций действий в допустимое действие (для данного отправного состояния).

Допустимое действие завершается успешно либо неуспешно как одно целое. Действия, из которых складывается допустимое действие, могут не быть допустимыми по отдельности.

Все установленные правила и ограничения вместе можно рассматривать как одно всеобщее правило, управляющее всей информационной базой. Требованием к языку является возможность составления сложных правил и ограничений из более простых. Разложение на более простые правила должно заканчиваться предопределенными примитивными правилами. Примитивное правило должно быть просто структурированным, легко обозримым множеством пар состояний информационной базы или состояния информационной базы и последовательности действий.

3.7 Координация допустимых действий

Все действия, выполняемые информационным процессором над информационной базой и концептуальной схемой, должны производиться в терминах допустимых действий. В силу этого любой процесс, выполняемый информационным процессором, будет состоять из одного или нескольких допустимых действий.

Концептуальная основа для координации допустимых действий основывается на следующих понятиях.

событие: Факт, состоящий в том, что нечто произошло или в проблемной области, или в среде, или в информационной системе.

команда: См. приложение А.

допустимое действие: См. приложение А.

условие команды: Предусловие выполнения допустимого действия.

Информационная система реагирует только вследствие события.

С точки зрения информационной системы не имеет значения, что считать событием: факт происшедшего или восприятие происшедшего. Но событие (стимул) должно выдавать информационному процессору уведомления о событии, на которое информационный процессор должен реагировать.

внешнее событие: Событие, происходящее в среде или проблемной области информационной системы.

Уведомление о таком событии — это сообщение для информационной системы, выражающее, по меньшей мере, команду.

внутреннее событие: Событие, которое наступает вследствие выполнения какого-либо допустимого действия в информационной системе.

В зависимости от результата такого допустимого действия внутренние события нужны только в том случае, если уведомление о завершении приведет к последующей реакции информационной системы или к отправке сообщения в среду.

События имеют определенные характеристики, которые важны для информационной системы. Они могут относиться к определенным типам (классам) событий. С понятием типа события тесно связано понятие реализации события.

Важной характеристикой типа события (или уведомления о нем) является тип (типы) команды, связанный с этим типом события, который определяет допустимое действие или допустимые действия, ожидаемые в качестве реакции информационной системы.

К другим важным характеристикам типов событий относятся:

- ожидаемое число реализаций событий;
- частота возникновения событий.

Большинство событий нуждаются также в механизме передачи «параметров», чтобы настроить допустимое действие на то, что конкретно должно быть сделано. В этих случаях сообщение, уведомляющее о событии, содержит (или должно сопровождаться) одно или более предложений, являющихся входными предложениями для допустимого действия.

Допустимые действия, выполняемые информационным процессором, могут осуществлять вставку, выборку, удаление или модификацию предложения в концептуальной схеме и информационной базе. Однако вводить сами входные предложения в концептуальную схему или информационную базу необязательно, если входные предложения только передают параметры для допустимого действия. Иногда входные предложения вводятся, в других случаях они приводят к вставке других выведенных предложений. Входное предложение, которое фактически не вставляется, не воспроизводимо после окончания допустимого действия.

Предложения, связанные с допустимым действием, определяют фактический результат. Вместе они определяют фактическое допустимое действие, т. е. реализацию типа допустимого действия.

Допустимое действие запускается событием или, более точно, командой, представленной в виде сообщения, извещающего о событии. Тип события определяет, какой тип допустимого действия запускается. В этот момент допустимое действие будет активным. Активное допустимое действие будет непрерывным до его окончания. Это окончание может вызывать внутреннее событие в зависимости от результата допустимого действия.

В любое время в информационной системе могут быть активны несколько допустимых действий. Допустимые действия могут принадлежать к одному и тому же или к различным типам допустимых действий. Два допустимых действия, активных в одно и то же время, считаются работающими совершенно независимо друг от друга.

Между самим событием и уведомлением о нем в информационный процессор или между уведомлением и реакцией информационного процуессора может возникать определенная задержка. Поэтому возможно, что последовательность, в которой информационная система уведомляется о событиях, отличается от последовательности самих событий, происходящих в проблемной области. Если эта последовательность событий во времени существенна, например из-за обязательно предшествующих состояний, которые должны быть зарегистрированы в концептуальной схеме и информационной базе, соответствующие допустимые действия должны быть координированными, включая, в случае необходимости, синхронизацию допустимых действий.

Условие команды для некоторого допустимого действия состоит из событий (внешних и внутренних), которые должны произойти для запуска допустимого действия, а также из правила или множества правил, устанавливающих, каким образом события формируют необходимое условие для запуска допустимого действия. Условие команды может быть расширено и сделано более точным при помощи правил, учитывающих предложения, извещающие о событиях, или уже присутствующих в концептуальной схеме и информационной базе.

Если в условие команды для допустимого действия включено несколько событий, тогда последнее по времени событие, независимо от того, каким оно есть, выполняет условие команды и запускает допустимое действие. «Поступление» других событий, за исключением последнего, переводит условие команды в «состояние ожидания».

Условие команды может иметь ограниченное состояние ожидания, т. е. могут быть ограничения на продолжительность состояния ожидания. Если в этот ограниченный период последнее необходимое событие не происходит, допустимое действие не будет выполняться, несмотря на события, которые уже прошли. Тем не менее новая серия событий может подготовить новое состояние ожидания для возможного допустимого действия. Если не определено ограниченное состояние ожидания, состояние ожидания может продолжаться как угодно долго.

Одна реализация событий может участвовать в запуске только одного допустимого действия. Если условие команды удовлетворено и запуск допустимого действия состоялся, то «поступление» новых событий может подготовить условие команды только для нового запуска допустимого действия.

В контексте условий команд необходимо специальное замечание по поводу событий. Само событие (реализация) несет информацию, так как оно удостоверяет, что нечто произошло. Этот факт устанавливается не в общем смысле, а конкретно. Поэтому событие удостоверяет не то, что имел место какой-то факт определенного типа, но что имел место конкретный факт (реализация) с конкретным результатом.

Вследствие этого внутренние события контролируются предписывающими правилами для данного события. Тип допустимого действия может завершиться несколькими различными внутренними событиями, в зависимости от результата допустимого действия. Такие различные события могут быть взаимоисключающими, но это не обязательно.

Во время выполнения допустимого действия (а также до выполнения самого допустимого действия) информационный процессор использует такие правила и ограничения:

- распознает уведомление о событии и получение выходного предложения согласно правилам распознавания;
- вычисляет условие команда и запускает фактическое допустимое действие или помещает условие команды в состояние ожидания;
- после запуска допустимого действия выполняет само допустимое действие в соответствии с правилами, сформулированными в описании допустимого действия, манипулирующего концептуальной схемой и информационной базой;
- контролирует манипулирование концептуальной схемой и информационной базой согласно соответствующим статическим и динамическим правилам и ограничениям для концептуальной схемы и информационной базы и соответствующим правилам полномочий;

- возбуждает одно или более соответствующих внутренних событий согласно предписывающим правилам для внутренних событий;

- генерирует выходные сообщения в соответствии с предписывающими правилами для выходных сообщений.

4 Понятия и принципы реализации

4.1 Принципы заполнения концептуальной схемы

Для проектирования концептуальной схемы проектировщик информационной системы должен начинать изучение с выбранной части мира, составляющей проблемную область, и строить в уме ее абстрактную модель.

Такая абстрактная модель должна быть достаточно общей для того, чтобы охватить многие (предпочтительно все) реализации некоторого типа явлений. Эта модель представляет собой высказывания, обязательно выполняющиеся во всех возможных пространствах сущностей.

Следует придерживаться разделения предложений на такие, которые выражают общие законы и правила, которым должны подчиняться возможные сущности в проблемной области, и такие, которые устанавливают факты о конкретных сущностях и проблемной области в соответствии с законами и правилами, описываемыми в концептуальной схеме. К предложениям первой группы относятся:

- описание классов (типов) сущностей в проблемной области;

- описание понятий, менее подверженных изменениям;

- правила или ограничения, имеющие широкое воздействие на поведение проблемной области (и поэтому также на поведение концептуальной схемы и информационной базы).

Необходимо придерживаться таких принципов концептуальной схемы: принцип 100 % и принцип концептуализации.

Некоторые соображения и замечания могут расширять эти два принципа.

Язык концептуальной схемы должен позволять описывать любое множество общих правил проблемной области в концептуальной схеме. Для этого некоторые правила можно описывать процедурно.

Если возникает необходимость расширять или модифицировать общие правила проблемной области, то желательно все их поместить в концептуальную схему.

Принцип концептуализации требует, чтобы концептуальная схема включала в себя только концептуалыные аспекты проблемной области. Не должны быть частью концептуальной схемы аспекты и конструкции, относящиеся к другим компонентам или факторам информационной системы: аспекты представления данных с точки зрения пользователей, аспекты физической организации данных и эффективности реализации, организованные аспекты информационной системы и т. д.

4.2 Принципы описания проблемной области

Теоретической основой описания проблемной области является интерпретируемая формальная система логики. Разработка такой формальной системы логики для описания проблемной области требует:

- определения алфавита, в котором можно однозначно и автоматически распознать, входит ли данный символ в алфавит. Эти условия естественным образом удовлетворяются при использовании обычных устройств символьного ввода информации в ЭВМ;

- обеспечения конечного набора эффективных правил (алгоритмов), определяющего, какие строки символов являются правилами построения и, в частности, какие правильно построенные строки следует рассматривать как предложения;

- обеспечения конечного набора эффективных правил, определяющих, является ли данное предложение аксиомой;

- обеспечения конечного набора эффективных правил, определяющих, может ли данное предложение быть непосредственно выводимым из данного множества предложений;

- обеспечения конечного набора правил интерпретации, приписывающих каждому предложению значение таким образом, что предложение однозначно интерпретируется как истинное или ложное утверждение о проблемной области.

В данном контексте аксиома определяется следующим образом.

аксиома: Любое предложение с точным содержанием, утверждаемое в качестве такового авторитетным источником.

Аксиомы и правила вывода выбираются так, чтобы:

- каждая аксиома интерпретировалась как истинное утверждение о проблемной области;
- каждое предложение, непосредственно выводимое из множества предложений, интерпретированных как истинные утверждения о проблемной области, само интерпретировалось как истинное утверждение о проблемной области.

Проектировщик информационной системы должен иметь возможность выразить концептуальную схему в терминах, соответствующих рассматриваемой проблемной области и восприятию ее пользователем. В частности, не должно налагаться никаких ограничений на сущности, наличие которых допускается в проблемной области, или на свойства, которыми они могут обладать. Проектировщик информационной системы ограничен только требованием придерживаться основных принципов логики, встроенных в информационный процессор.

Концептуальная схема должна быть простой в использовании и понимании для разнообразных пользователей. В связи с этим необходим механизм добавления лингвистических конструкций произвольной сложности («макроконструкций») в формальную систему. Кроме того, в самой формальной системе должен существовать механизм определения новых конструкций в уже существующих терминах.

4.3 Абстрактный синтаксис для концептуальной схемы и информационной базы

Для точного определения проблемной области требуется формальный язык.

В стандарте синтаксис формального языка определяется посредством введения грамматических понятий без какого-либо указания, как эти понятия могут быть выражены на каком-либо определенном языке. Понятиями являются терм, предложение, функтор.

функтор: Лингвистический объект, который обозначает функцию от других лингвистических объектов, имеющую в качестве аргументов (вход) список лингвистических объектов и выдающую в качестве значения (выход) единственный уникально определенный лингвистический объект.

Термы и предложения могут включать в себя переменные.

переменная: Терм, который обозначает неконкретизированную сущность в проблемной области.

4.4 Семантика концептуальной схемы и информационной базы

Чтобы приписать смысл (семантику) различным выражениям в языке, необходимо начать с множества неопределенных понятий — примитивов. Другие понятия тогда получают смысл, выводимый из неформальных примитивных понятий с помощью формальных определений. Соответствующие аспекты смысла каждого примитивного понятия формально вводятся посредством задания аксиом, которые считаются истинными. Правила вывода должны сохранять истинность.

Предложение выражает истинное высказывание, если оно утверждает, что состояние проблемной области — такое-то, и оно действительно является таковым.

Интерпретация предложений зависит от значения термов и предикатов, составляющих это предложение. Интерпретация термов соотносит им конкретные сущности проблемной области.

Интерпретировать предикат в предложении означает возможность определить для любого списка сущностей истинность высказывания, выраженного этим предложением, для данного списка сущностей.

4.5 Принципы составления концептуальной схемы

При рассмотрении принципов составления концептуальной схемы и информационной базы следует решить три воспроса:

- что входит в концептуальную схему?;
- каковы возможности информационного процессора?;
- как информационный процессор работает с концептуальной схемой?

Информационный процессор имеет встроенную минимальную концептуальную схему, которая не может быть изменена без изменения самого информационного процессора. Любая подходящая совокупность предложений, согласующихся с этой минимальной концептуальной схемой, может быть вставлена при помощи механизмов информационного процессора для того, чтобы расширить эту минимальную концептуальную схему для формирования актуальной концептуальной схемы. Этот процесс можно применять рекурсивно.

Информационный процессор должен будет в каждый момент времени «знать», к какой части фактической совокупности предложений, составляющих концептуальную схему и информационную базу, он должен обращаться как к «концептуальной схеме» для того, чтобы управлять манипулированием другими предложениями. Важно решить в процессе проектирования и моделирования, какие предложения будут в минимальной концептуальной схеме, какие — вне ее.

Информационная база -- это непротиворечивая совокупность предложений (т. е. непротиворечивая внутри себя и непротиворечивая по отношению к концептуальной схеме), которая выражает дополнительные высказывания для соответствующего пространства сущностей, описываемого в информационной системе.

При наличии концептуальной схемы и, возможно, совокупности существенных или необходимых предложений остальная часть информационной базы является необязательной при единственном условии, что все вместе составляют непротиворечивую совокупность. В любой данный момент состояние концептуальной схемы и информационной базы — это точно определенная совокупность предложений.

4.6 Трехуровневая архитектура

Необходимо снабдить любую реализацию информационной системы по меньшей мере двумя типами интерфейса. Первый тип интерфейса, по отношению к пользователям информационной системы, обрабатывает внешние формы представления, удобные для конкретного пользователя.

Эти внешние интерфейсы описываются во внешних схемах.

внешняя схема: Определение форм внешнего представления для возможных совокупностей предложений в пределах представления конкретного пользователя, а также аспектов манипулирования этими формами.

Внешний интерфейс — это фактически интерфейс между пользователем в среде и информационной системой.

Второй тип интерфейса — это интерфейс между пользователем и информационной системой. Он связан со следующими аспектами:

- формы внутреннего (физического) представления информации;
- эффективность работы программ и механизмы эффективного доступа к хранимым данным;
- управление параллельным использованием, восстановлением после сбоев и т. д.

Эти интерфейсы определяются во внутренних схемах.

внутренняя схема: Определение форм внутреннего представления в компьютере совокупностей предложений в концептуальной схеме и информационной базе, а также аспектов манипулирования этими формами.

Внутренний интерфейс — это интерфейс между информационной системой и средствами управления данными в компьютере.

В информационную систему поступают сообщения, содержащие информацию, которая добавляется в информационную базу. Информационный процессор, управляемый правилами, описанными в концептуальной схеме и, возможно, другими предложениями, уже присутствующими в информационной базе, или вставит эту новую информацию, или проигнорирует сообщение, выдав соответствующее сообщение, извещающее о результате. Другие действия будут выполняться подобным же образом.

Прикладной процесс, обрабатывающий сообщения пользователя, имеет дело с определенным внешним представлением данных (строками символов), составляющих его внешнюю базу данных, содержащую соответствующую информацию.

Это конкретное внешнее представление описано во внешней схеме, соответствующей данному прикладному процессу. Такая внешняя база данных является виртуальной, отображаемой на информационную базу (или соответствующую ее часть). Это подразумевает, что информационная система должна обрабатывать:

- «интеграцию» действий различных пользователей;
- отображение их внешних представлений в общее (концептуальное) представление, известное информационной системе.

Задачей информационного процессора остается осуществление правил, определенных в концептуальной схеме, и контроль за вставкой. Дополнительные задачи выполняются внешним процессором. Поэтому внешние схемы содержат также правила отображения.

Сама информационная база по существу является виртуальной. Информация представлена в вычислительной системе во внутренних формах физических данных (записи, сегменты, поля и т. д.) во внутренней базе данных. Эти формы описаны во внутренней схеме. Имеется ввиду дополнительный процесс отображения, выполняемый внутренним процессором. Правила отображения для него также описаны во внутренней схеме.

К самой внутренней базе данных доступ осуществляется с помощью средства управления внешней памятью компьютерной системы.

Согласно трехсхемной архитектуре и внутренняя, и внешняя схемы, и процессоры могут иметь множество уровней. Точно также внутренняя база данных может быть реализована как семейство

внутренних баз данных, каждая из которых «хранит» часть информационной базы. Такие базы данных пересекаются.

Представления пользователей можно объединить в концептуальную подсхему. Концептуальная схема в информационной системе представляет собой «объединение» этих различных концептуальных подсхем. На концептуальном уровне формы представления несущественны.

На внешнем уровне определяются формы представления, удобные для пользователя. Они описываются во внешних схемах. Каждое представление пользователя (концептуальная подсхема) отображается в одну или более внешних схем, определяющих соответствующие формы представления, каждая из которых описывает внешнюю базу данных, существующую в пределах представления этого пользователя, хотя и в виртуальной форме.

В случае, когда внешнее представление является объединением нескольких представлений, полученная внешняя схема будет охватывать несколько отдельных внешних схем и описывать общую базу данных во внешней, но объединенной форме.Функция разложения внешней схемы на подмножества поддерживается и управляется внешним процессором.

Внешние базы данных отображаются в физические базы данных. Несколько внешних баз данных могут отображаться в одну физическую базу данных; одна внешняя база данных может отображаться в несколько физических баз данных; возможно любое другое сочетание. Возможны и распределенные сети.

Физическая база данных определяется во внутренней схеме. Преобразование из внешней во внутреннюю форму главным образом осуществляется внутренним процессором. В распределенных базах данных взаимосвязь между внешней и внутренней базами данных может быть описана в схеме распределения, которая может быть объединенной частью внешней схемы (внешних схем), взаимодействующей с внутренней схемой (внутренними схемами).

Задача обработки правил, выполняемая информационным процессором, может быть реализована в виде набора процедур. Они не обязательно выполняются одним специальным (информационным) процессором, отличным от внешнего и внутреннего процессоров. В частности, в системах распределенных баз данных эти процедуры могут распределяться по соответствующим внешним и внутренним процессорам.

Концептуальная схема рассматривается как описание необходимых высказываний для проблемной области и поэтому определяет, что описывается в информационной базе, а не как это описывается. Концептуальная схема управляет семантическим значением всех представлений – то есть определяет набор проверяющих, генерирующих и выводящих процедур, определенных на концептуальном уровне информационной системы. Но они не дают описания промежуточного состояния в процессе преобразования из внешней формы во внутреннюю.

Внешний процессор устанавливает связь непосредственно с пользователями и координирует их потоки информации.

Внутренние схемы описывают внутреннее физическое представление информации. Отображение между внешними и внутренними формами выполняется, главным образом, внутренним(ми) процессором(ами). Поэтому внешний(ние) процессор(ры) устанавливает(ют) связь с внутренним(ними) процессором(ами). Отображение внешних схем во внутренние схемы должно сохранять смысл информации в соответствии с концептуальной схемой.

4.7 Модель системы словарей информационных ресурсов

Система словарей информационных ресурсов для информационной системы сама является информационной системой. Она описывает проблемную область, состоящую из целевой информационной системы.

Концептуальная схема системы словарей информационных ресурсов должна описывать проблемную область, содержащую все возможные целевые информационные системы, которые могут документироваться с использованием системы словарей информационных ресурсов. Соответственно, концептуальная схема системы словарей информационных ресурсов должна обеспечивать описание того, что может быть описано в информационной базе системы словарей информационных ресурсов, а также допустимых действий над информационной базой системы словарей информационных ресурсов.

Информационная база системы словарей информационных ресурсов должна описывать все целевые информационные системы, для которых используется система словарей информационных ресурсов, в том числе концептуальные, внешние и внутренние схемы, различные процессоры, программы, пользователей и т. д.

4.8 Соответствие трехуровневой архитектуры информационных систем и эталонной модели архитектуры открытых систем

Концептуальный и внешний уровни трехуровневой архитектуры информационных систем соответствуют функциям прикладного уровня и уровня представления.

Внутренний уровень имеет дело с внутренним представлением данных, включая внутреннее манипулирование данными и практическое физическое хранение данных на запоминающих устройствах. Область деятельности здесь сравнима по характеру с уровнями сеанса, транспортным, сетевым, уровнем данных и физическим уровнем.

5 Подходы к моделированию

Следует использовать такие критерии подходов к моделированию:

- сопоставление формы и содержания;
- статические аспекты в сравнении с динамическими аспектами;
- возможность различать лексические и нелексические сущности;
- выразительная мощность (степень полноты, с которой данный метод может формально выразить все нужные аспекты и ограничения проблемной области в концептуальной схеме);
- гранулирование информации;
- разделение различных видов высказываний.

Рекомендуется использовать следующие методы моделирования в области информационных систем и баз данных:

- абстрактные типы данных;
- модели, основанные на бинарных отношениях;
- концептуальные графы;
- модели предложений с глубокой структурой;
- модели сущность — связь;
- модели, ориентированные на функции или на действия;
- модели, основанные на *n*-арных отношениях;
- сетевые модели;
- модели объект — роль;
- модели взаимодействия процессов;
- реляционные модели;
- семантические сети;
- теоретико-множественные модели.

Каждый из этих методов является особым взглядом на проблемы концептуальных схем и информационных баз.

Основными подходами к моделированию информационных систем и баз данных являются:

- подходы сущность — атрибут — связь;
- подходы на основе бинарных и элементарных *n*-арных отношений;
- подходы на основе интерпретируемой логики предикатов.

Подходы сущность — атрибут — связь основываются на следующих понятиях:

- сущности;
- связи между сущностями;
- атрибуты-ассоциации между значениями и сущностями или между значениями и связями;
- значения.

В этих подходах используются также понятия тип и реализация, применяемые к каждому из этих примитивных понятий.

Основой этих подходов является моделирование данных.

Подходы сущность — атрибут — связь следует характеризовать как ориентированные на определение статических аспектов. Они представлены в общих чертах в приложении Г.

Подходы, основанные на бинарных и элементарных *n*-арных отношениях, различают сущности и имена сущностей, но не различают атрибуты и связи. Подходы базируются на трех основных понятиях:

- сущности;
- имена сущностей;
- бинарные отношения.

В этих подходах используются также понятия тип и реализация, применяемые к каждому из примитивных понятий.

Подходы, основанные на бинарных и элементарных *n*-арных отношениях, базируются на моделировании проблемной области прямолинейным образом и по отдельным элементам, используя предложения, выражающие простые элементарные высказывания, не вводя специфической классификации элементов.

Общие принципы подходов на основе бинарных и элементарных *n*-арных отношений представлены в приложении Д.

Подходы на основе интерпретируемой логики предикатов воспринимают проблемную область как состоящую исключительно из сущностей, для которых выполняются определенные высказывания. Концептуальная схема и информационная база образуют описание, состоящее только из множества предложений, закодированных на некотором формальном языке, основанном на формальной логике. Такие предложения состоят из:

- термов и переменных;
- предикатов;
- логических связей;
- кванторов.

Термы и переменные относятся к сущностям в проблемной области, а предложения выражают высказывания об этих сущностях.

Суть подходов — формирование интерпретируемой, аксиоматизированной, дедуктивной, формальной системы логики, описывающей проблемную область и не налагающей какого-либо ограничения со стороны метода моделирования на саму проблемную область.

Обзор этих подходов на основе интерпретируемой логики предикатов более подробно представлен в приложении Е.

ПРИЛОЖЕНИЕ А
(справочное)

Алфавитный указатель терминов и определений

аксиома (AXIOM): Любое предложение с точным содержанием, утверждаемое в качестве такового авторитетным источником

актуальная информационная база (ACTUAL INFORMATION BASE): Информационная база, существующая в указанный момент или период времени, который квалифицируется как «сейчас», и отражающая дополнительные высказывания, отличные от необходимых

актуальное пространство сущностей (ACTUAL ENTITY WORLD): Совокупность сущностей, которая описана в актуальной информационной базе и ее концептуальной схеме

база данных (DATA BASE): Совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь

внешнее событие (EXTERNAL EVENT): Событие, происходящее в среде или проблемной области информационной системы

внешний уровень (EXTERNAL LEVEL): Все аспекты представления информации для пользователя во внешних интерфейсах информационной системы

внешняя схема (EXTERNAL SCHEMA): Определение форм внешнего представления для возможных совокупностей предложений в пределах представления конкретного пользователя, а также аспектов манипулирования этими формами

внутреннее событие (INTERNAL EVENT): Событие, которое наступает вследствие выполнения какого-либо допустимого действия в информационной системе

внутренний уровень (INTERNAL LEVEL): Все аспекты незаметного для пользователя представления информации в компьютерной физической реализации информационной системы

внутренняя схема (INTERNAL SCHEMA): Определение форм внутреннего представления в компьютере совокупностей предложений в концептуальной схеме и информационной базе, а также аспектов манипулирования этими формами

вставка (INSERTION): Добавление предложения к информационной базе или концептуальной схеме

выборка (RETRIEVAL): Поиск и выдача предложения, которое есть в информационной базе или концептуальной схеме или выводится из предложений, имеющихся в них

высказывание (PROPOSITION): Возможное состояние сущностей, по поводу которого можно утверждать или отрицать, что такое состояние имеет место

данные (DATA): Информация, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами

действие (ACTION): Одно или несколько элементарных действий, которые в качестве единого целого изменяют определенную совокупность предложений в информационной базе или концептуальной схеме и/или представляют о них информацию

допустимое действие (PERMISSIBLE ACTION): Действие, удовлетворяющее установленным правилам или ограничениям и изменяющее непротиворечивую совокупность предложений в информационной базе или концептуальной схеме на другую непротиворечивую совокупность предложений и/или предоставляющее информацию о непротиворечивой совокупности предложений в информационной базе или концептуальной схеме

информационная база (INFORMATION BASE): Совокупность предложений, выражающих высказывания, отличные от необходимых высказываний, согласующиеся друг с другом и концептуальной схемой, а также истинные в определенном пространстве сущностей

информационная система (INFORMATION SYSTEM): Концептуальная схема, информационная база и информационный процессор, составляющие вместе формальную систему для хранения и манипулирования информацией

информационный процессор (INFORMATION PROCESSOR): Процессор, который в ответ на команду выполняет действие над концептуальной схемой и/или информационной базой

информация (INFORMATION): Любой вид знаний о предметах, фактах, понятиях и т. д. проблемной области, которыми обмениваются пользователи информационной системы

класс [сущностей] (CLASS [of entities]): Сущности проблемной области, для которых данное высказывание является истинным

команда (COMMAND): Приказ или сигнал для выполнения действия

командное предложение (COMMAND STATEMENT): Лингвистический объект, являющийся командой

концептуальная подсхема (CONCEPTUAL SUBSCHEMA): Часть концептуальной схемы информационной системы совместного использования, ограниченная представлением отдельного пользователя

концептуальная схема (CONCEPTUAL SCHEMA): Непротиворечивая совокупность предложений, выражающих необходимые высказывания, относящиеся к проблемной области

концептуальный уровень (CONCEPTUAL LEVEL): Все аспекты, относящиеся к интерпретации информации и манипулированию информацией о проблемной области или пространстве сущностей в информационной системе

лексический объект (LEXICAL OBJECT) **имя** (NAME): Простой лингвистический объект, используемый только для ссылки на сущность

лингвистический объект (LINGUISTIC OBJECT): Грамматически допустимая языковая конструкция

модификация (MODIFICATION): Замена предложения в информационной базе или концептуальной схеме на другое

необходимое высказывание (NECESSARY PROPOSITION): Высказывание, которое должно содержаться во всех возможных пространствах высказываний и является истинным во всех пространствах сущностей

омонимы (HOMONYMS): Идентичные термы, обозначающие различные сущности

описание действия (ACTION DESCRIPTION): Лингвистический объект, описывающий действие

переменная (VARIABLE): Терм, который обозначает неконкретизированную сущность в проблемной области

пользователь (USER): Некто (или нечто), выдающий команды и сообщения информационной системе и получающий сообщения от информационной системы

предикат (PREDICATE): Лингвистический объект, аналогичный глаголу, сообщающий что-либо о сущностях, обозначенных термами

предложение (SENTENCE): Лингвистический объект, представляющий определенное высказывание

принцип концептуализации (CONCEPTUALIZATION PRINCIPLE): Принцип, согласно которому концептуальная схема должна включать статические и динамические аспекты проблемной области только концептуального уровня, не касаясь внешних и внутренних аспектов представления и организации данных (физической организации данных и доступа к ним, аспектов представления, касающихся отдельных пользователей)

принцип 100 % (100 PERCENT PRINCIPLE): Принцип, согласно которому все общие аспекты, т. е. все правила, законы и т. д. проблемной области должны быть описаны в концептуальной схеме, причем информационная система не может нести ответственность за несоблюдение правил и законов, описанных не в концептуальной схеме

проблемная область (UNIVERSE OF DISCOURCE): Все сущности, являющиеся объектом интереса, которые были, есть или когда-либо могут быть

пространство высказываний (PROPOSITION WORLD): Совокупность высказываний, каждое из которых истинно для данного пространства сущностей

пространство сущностей (ENTITY WORLD): Совокупность сущностей, воспринимаемых как единое целое

реализация [типа сущности] (OCCURENCE [of an entity-type]), **экземпляр** [типа сущности] (INSTANCE) [of an entity-type]): Отдельная сущность, для которой истинно некоторое высказывание по поводу ее типа

синонимы (SYNONYMS): Различные термины, обозначающие одну и ту же сущность

система словаря информационных ресурсов (INFORMATION RESOURCE DICTIONARY SYSTEM): Информационная система, содержащая информацию об одной или нескольких прикладных системах

событие (EVENT): Факт, состоящий в том, что нечто произошло или в проблемной области, или в среде, или в информационной системе

сообщение (MESSAGE): Совокупность, состоящая из одного или более предложений, возможно командных предложений, и использующаяся как элемент информационного обмена между средой и информационной системой

среда (ENVIRONMENT): Часть реального мира с пользователями, которые обмениваются сообщениями с информационной системой

сущность (ENTITY): Любой конкретный или абстрактный объект, включая связи между объектами

схема базы данных (DATABASE SCHEMA): Формальное описание данных в соответствии с конкретной схемой данных

схема данных (DATA SCHEMA): Логическое представление организации данных

терм (TERM): Лингвистический объект, обозначающий сущность

тип [сущности] (TYPE [of an entity]): Высказывание, устанавливающее, что сущность принадлежит к определенному классу сущностей

удаление (DELETION): Извлечение ранее вставленного предложения из информационной базы или концептуальной схемы

условие команды (COMMAND CONDITION): Предусловие выполнения допустимого действия

функтор (FUNCTOR): Лингвистический объект, который обозначает функцию от других лингвистических объектов, имеющую в качестве аргументов (вход) список лингвистических объектов и выдающую в качестве значения (выход) единственный уникально определенный лингвистический объект

элементарная команда (ELEMENTARY COMMAND): Приказ или сигнал для выполнения элементарного действия

элементарное действие (ELEMENTARY ACTION): Вставка, удаление или поиск предложения

язык концептуальной схемы (CONCEPTUAL SCHEMA LANGUAGE): Формальный язык для описания концептуальной схемы, ее составных частей и действий над ними

ПРИЛОЖЕНИЕ Б
(справочное)

Пример проблемной области

В разделе Б.1 дано словесное описание классификаций и правил для нашего примера проблемной области. Это описание можно рассматривать как неформальную концептуальную схему.

В разделе Б.2 приведены примеры сущностей и событий в проблемной области. Их можно рассматривать как неформальную и неполную информационную базу, описывающую рассматриваемое пространство сущностей.

Б.1 Правила и другие требования для проблемной области

Описываемая проблемная область связана с регистрацией автомобилей и рассматривается с точки зрения Органа Регистрации. Орган Регистрации существует для того, чтобы:

— знать, кто является или являлся зарегистрированным владельцем автомобиля в любой момент времени от изготовления до уничтожения автомобиля;

— управлять определенными законами, например связанными с потреблением топлива автомобилями и с передачей права владения ими.

Изготовители автомобилей

Существует некоторое число изготовителей, имеющих уникальные имена. Изготовители могут начать функционирование с разрешения Органа Регистрации (которое нельзя отменить). Одновременно могут работать не более пяти изготовителей. Изготовитель может прекратить работу при условии, что он не владеет ни одним автомобилем. В этом случае разрешение на функционирование теряет силу.

Автомобили

Автомобиль является определенной моделью; изготовитель присваивает ему серийный номер, уникальный для машин, сделанных этим изготовителем. Изготовитель регистрируется как владелец автомобиля немедленно. В этот момент присваивается регистрационный номер, уникальный для всех автомобилей и на все времена. Регистрируется также год выпуска. Только в январе автомобиль может быть объявлен произведенным в предыдущем году. В конечном счете автомобиль уничтожается и регистрируется дата его уничтожения. История автомобиля должна храниться до конца второго календарного года после его уничтожения.

Модели автомобилей

Модель автомобиля имеет одно уникальное имя. Автомобили одной модели производятся только одним изготовителем. Новые модели могут вводиться без ограничений. Считается, что все автомобили одной модели расходуют то же количество топлива.

Расход топлива

Расход топлива — это некоторое число литров углеводородного топлива на 100 км пути, которое находится в пределах 4—25 л. Усредненный расход топлива зарегистрированных автомобилей, произведенных одной фирмой-изготовителем в определенном году, не должен превышать максимального значения, которое устанавливается для всех изготовителей и может меняться от года к году. В конце января каждого года Орган Регистрации посылает соответствующее сообщение каждой фирме-изготовителю, которая не выполнила этого требования.

Гаражи

Существует некоторое число гаражей, каждый с уникальным именем. Могут открываться новые гаражи. Гаражи могут владеть автомобилями, но в любое время автомобили, которыми они владеют, должны поступать от не более чем трех изготовителей (какие три — не имеет значения, и они со временем могут меняться). Гараж не может прекратить торговлю, пока у него есть автомобили.

Лица

Существует ряд лиц, которые могут владеть одним или несколькими автомобилями. Каждый человек имеет уникальное имя. Интерес представляют только те лица, которые владеют или когда-то владели автомобилем, до сих пор известным Органу Регистрации.

Владение автомобилем

В любое время автомобилем может владеть или его фирма-изготовитель, или торгующий гараж, или некоторое лицо, или группа лиц. Если автомобилем владеет группа лиц, каждое из них считается владельцем.

Передача права владения

Право владения автомобилем передается при регистрации фактической передачи. Фирма-изготовитель может передавать свое право только гаражам и не может выступать в качестве получателя такого права. Гараж может передавать свое право только лицам. После уничтожения автомобиля он не может более никому передаваться. Однако могут регистрироваться более ранние передачи.

Кроме перечисленных, никаких других правил нет. В данном примере неизбежно делается некоторое число упрощающих допущений, например причины, цены и обстоятельства передачи права владения не рассматриваются.

Б.2 Некоторые факты и события в пространстве сущностей

Ниже описаны некоторые объекты и события в предлагаемом пространстве сущностей.

Форд, Дженерал Моторз, Рено, Фольксваген и Джовет являются фирмами — изготовителями автомобилей, имеющими разрешение выпускать машины. Модели Форда — Мустанг и Гранада. Дженерал Моторз среди прочих производит модель Импала.

История автомобиля модели Мустанг фирмы Форд, серийный номер PCXX999 такова: он изготовлен в 1975 г. и представлен на регистрацию 21 января 1975 г. Он получил регистрационный номер GMF117. 29 января 1975 г. он был направлен в гараж Смита, который продал его 15 марта 1975 г. г-ну Джонсону. Г-н Бейкер купил автомобиль у г-на Джонсона 24 мая 1978 г. Машина была уничтожена 13 января 1980 г.

Автомобиль модели Импала фирмы Дженерал Моторз, серийный номер QGTM783F, был зарегистрирован под номером ABC 653 9 апреля 1978 г. и направлен в гараж Джоунс Бразерс. Этот автомобиль был куплен г-ном Джонсоном 26 мая 1978 г. Автомобиль был уничтожен 14 августа 1979 г.

Дженерал Моторз выпустила автомобиль модели Импала, серийный номер QAVP864, в 1977 г. Он был зарегистрирован 21 января 1978 г., получив регистрационный номер PQR456. 14 февраля 1978 г. он был продан в гараж RN CARS, который уже владел другими автомобилями фирм Дженерал Моторз, Рено и Фольксваген. Г-н и г-жа Дж. Соуп купили этот автомобиль 31 марта 1978 г., но не смогли продать свой Датсун как часть этой сделки.

В 1978 г. новый изготовитель PSC (PRETTY SMALL CAPS) запросил разрешение на производство, но получил отказ. После разрешения фирмы Джовет 1979 г. запрос был повторен, и на этот раз PSC получил разрешение функционировать с 1 января 1980 г.

Первой выпущенной моделью была Гесмайзер. Первая партия этой модели с серийными номерами GAM1001, GAM1002 и GAM1003 была зарегистрирована 4 января 1980 г. Они получили регистрационные номера XYZ101, XYZ102, XYZ103 соответственно.

Автомобиль XYZ101 был передан в гараж South Station 25 января 1980 г., но был случайно уничтожен в тот же день. Последние два автомобиля были переданы в гараж North Station 20 января 1980 г. Обе машины были проданы г.г. Гедель, Эшер и Бах 26 января 1980 г. Г-н Бах погиб в катастрофе на автомобиле XYZ103 2 марта 1980 г. Автомобиль был зарегистрирован как уничтоженный 5 марта 1980 г. 5 марта 1980 г. г-да Гедель и Эшер были зарегистрированы как владельцы оставшегося автомобиля XYZ102. Они продали этот автомобиль в гараж Смита 15 марта 1980 г., а у него купили Мустанг, серийный номер PCXXX010, который был зарегистрирован 5 января 1980 г. как выпущенный в 1979 г. Регистрационный номер этого автомобиля был XYZ109. XYZ102 был уничтожен путем разборки, т. к. в гараже Смита был дефицит запчастей. Поэтому в конце 1982 г. Орган Регистрации может удалить все сведения об автомобилях XYZ101—XYZ103. 1 декабря 1980 г. фирма PSC дала объявление о свертывании деятельности.

В 1979 г. средний расход топлива был установлен в размере 12 л на 100 км. В 1980 г. этот показатель был 10/100 км, который сохранился и на 1981 г.

ПРИЛОЖЕНИЕ В
(справочное)

Система синтаксических обозначений Паскаля

В приложении принята система обозначений для описания синтаксиса ПАСКАЛЯ, предложенная комитетами ИСО/ТК 97/ПК 5 для языков, используемых при иллюстрации примера концептуальной схемы в различных подходах.

Эту систему обозначений для описания синтаксиса можно использовать и для самоописания. Последовательность, с которой записываются метаутверждения, несущественна. Единственное требование к тому, чтобы описание было полным: необходимо определение метаутверждения для каждой метапеременной, появляющейся в метавыражении.

синтаксис	= метаутверждение {метаутверждение}.
метаутверждение	= интервал метапеременная интервал "=" метавыражение интервал ".".
метапеременная	= буква {буква | цифра | "-"}.
метавыражение	= последовательность {или последовательность}.
последовательность	= интервал элемент {запятая элемент}.
элемент	= метапеременная | итерация | опция | гнездо | терминальный-символ.
итерация	= "{" метавыражение интервал "}".
опция	= "[" метавыражение интервал "]".
гнездо	= "(" метавыражение интервал ")".
терминальный-символ	= описание-строки | терминальный-знак.
описание-строки	= ограничитель-строки строка ограничитель-строки.
ограничитель-строки	= ' ".
строка	= {терминальный-знак | любой-знак}.
терминальный-знак	= ' ' (любой-знак | ' ").
пробел	= " ".
интервал	= {пробел}.
или	= интервал "|".
запятая	= интервал "," интервал].
буква	обозначает любую букву алфавита, более подробно здесь не определяется.
цифра	обозначает любую цифру от 0 до 9, более подробно здесь не определяется.
любой-знак	обозначает любой символ, кроме " (двойная кавычка) в наборе символов, более подробно здесь не определяется.

Семантика в целом должна быть ясной из приведенного выше описания. Могут быть полезными следующие примечания:

Метасимвол	Смысл
=	определено как
|	альтернатива
.	конец определения
опция: [*x*]	0 или 1 экземпляр *x*
итерация: {*x*}	0 или более экземпляров *x*
гнездо: (*x* | *y* | *z*)	любой из *x* или *y*, или *z*
терминальный-символ: «*xyz*»	строка *xyz*
терминальный-знак: "*x*	символ *x*
' '	символ '
ограничитель-строки: ' "	символ "

Кавычка (') должна интерпретироваться следующим образом: «Не копировать этот символ, а взять следующий символ как он есть».

Необходим специальный терминальный-символ, чтобы определять:

- символ " в строке (' "),
- символ ' в строке (' '),

так как пустая строка определяется как " ".

Установлены «приоритеты операций»:

- любой тип скобок (опция, итерация, гнездо);
- последовательность;
- выбор.

ПРИЛОЖЕНИЕ Г
(справочное)

Подходы сущность-атрибут-связь

Основная цель данного приложения — изложить понятия, имеющиеся в большинстве подходов сущность-атрибут-связь.

Учитывая цели концептуальной схемы, можно указать следующие роли этих подходов:

а) обеспечить общую основу для понимания поведения проблемной области.

Основные понятия — сущность, атрибут и связь — кажутся простыми для понимания и обучения, хотя часто трудно сделать верный выбор на основе анализа части мира, составляющей проблемную область. Особое внимание обращается на структуру связей между сущностями;

б) определить допустимые изменения и манипулирование информацией о проблемной области.

В подходах сущность-атрибут-связь использование основных понятий создает некоторую жесткость по сравнению с другими классами подходов. Например, различие между атрибутом и связью приводит к возникновению некоторых трудностей при модификации концептуальной схемы. Поскольку в большинстве подходов сущность-атрибут-связь не сформулированы динамические ограничения, управление допустимым манипулированием довольно ограничено;

в) обеспечить основу для интерпретации внешних и внутренних синтаксических форм, которые представляют информацию о проблемной области;

г) обеспечить основу для взаимного отображения внешних и внутренних схем.

В подходах сущность-атрибут-связь роль концептуальной схемы чаще понимается как результат определенной стадии интерактивного процесса проектирования. На последующих стадиях эта концептуальная схема преобразуется вручную во внешние и внутренние модели данных.

Г.1 Фундаментальные понятия

В подходах сущность-атрибут-связь считается, что проблемная область состоит из базовых понятий сущностей, которые могут иметь атрибуты и связи между сущностями. В некоторых подходах связи также могут иметь атрибуты.

Сущность — это любой конкретный или абстрактный объект в проблемной области.

Атрибут — это наблюдаемое свойство сущности или наблюдаемое свойство ассоциации между сущностями проблемной области.

Атрибут имеет значение.

Связь — это наблюдаемая ассоциация между сущностями проблемной области.

Атрибуты и связи являются высказываниями о сущностях. Атрибуты — это одноместные или двухместные высказывания о единичной сущности или связи, часто ассоциирующие определенное значение с этой сущностью или связью. Связи — это высказывания о двух или более сущностях (*N*-местные высказывания).

Объекты, имеющие общие свойства, относятся к одному типу. Тип — это классификация подобных объектов. Три основных абстракции подходов сущность-атрибут-связь — тип-сущности, тип-атрибута и тип-связи.

В подходах сущность-атрибут-связь совместно с понятием типа часто используют понятия «реализация» и «совокупность». Реализация определенного типа определяется как уникальный отдельный объект, принадлежащий к этому типу. Совокупность типа — это некоторая совокупность реализации этого типа, которая время от времени может меняться. Обычно, это совокупность всех сущностей данного типа, которые встречаются в соответствующем пространстве сущностей. Тип-сущности — это классификация сущностей, каждая из которых имеет подобные атрибуты, связанные с ней. Каждая реализация типа-сущности должна быть уникальна и поэтому отличима от всех других реализаций этого типа-сущности. В подходах сущность-атрибут-связь сущности различаются посредством одного или более атрибутов, называемых идентификатором. Тип-атрибута — это классификация подобных атрибутов всех реализаций сущности, относящихся к одному типу-сущности, или всех реализаций связи соответствующего типа. Индивидуальная реализация типа-атрибута, ассоциируемая с индивидуальной реализацией типа-сущности или типа-связи, рассматривается как значение атрибута. В данном разделе используется термин «значение-атрибута». Пара <тип-атрибута, значение-атрибута> «идентифицирует» атрибут-сущности. Говорят также, что значения атрибутов, входящих в идентификатор, идентифицируют сущность.

Тип-связи в подходах сущность-атрибут-связь — это связь, определенная на одном или более типах-сущности.

Понятие связи между сущностями — основополагающее для подходов сущность-атрибут-связь. Однако не следует предполагать, что тип-сущности может участвовать только в одном типе-связи. Можно определить любое число типов-связи, в которых принимает участие тип-сущности. Не следует также предполагать, что только один тип-связи может определяться на совокупности типов-сущности. Может быть определено любое число типов-связи, включающих в себя одну и ту же совокупность типов-сущности. Там, где определяются два

или более типа-связи на одной и той же совокупности типов-сущности, необходимо различать типы-связи, приписывая им одно или несколько имен.

Некоторые подходы сущность-атрибут-связь также различают типы-значений. Тип-значения — это классификация совокупности значений-атрибута, которые могут образовывать пары с определенным типом-атрибута, и, таким образом, будут атрибутами для сущностей определенного типа-сущности или для связей определенного типа-связи. Тип-значения всегда тесно связан с одним или более типами-атрибута. Другой термин для типа-значения, используемый некоторыми авторами, — домен.

В некоторых подходах сущность-атрибут-связь типы-сущности не могут пересекаться, т. е. определенная сущность может принадлежать только к одному типу-сущности. В других случаях типы-сущности могут пересекаться, т. е. определенная сущность может относиться к нескольким типам-сущности. Это замечание применимо также к типам-атрибута, типам-связи и типам-значения.

Между вариантами подходов сущность-атрибут-связь существуют различия, особенно по отношению к характеристикам связей.

В подходах сущность-атрибут-связь связи могут классифицироваться по трем критериям:

- размерность типа-связи;
- функциональность типа-связи;
- полные и частичные связи.

Размерность типа-связи — это число реализаций сущности в одной реализации типа-связи.

Тип-связи размерности *n* называют *n*-арным типом-связи.

Некоторые подходы сущность-атрибут-связь ограничиваются бинарными типами-связи. Более того, некоторые из них не допускают бинарные типы-связи на одном типе-сущности.

Функциональность типа-связи иллюстрируется по отношению к двум различным типам-сущности. Возможными видами функциональности для бинарного типа-связи, определенного на типах-сущности *A* и *B*, являются:

- один-к-одному (1-1), когда одна реализация *A* может быть связана только с одной реализацией *B*, и каждая реализация *B* может быть связана только с одной реализацией *A*;
- один-ко-многим (1-*n*), когда одна реализация *A* может иметь одну или более связей, каждую с одной реализацией *B*, но каждая реализация *B* может иметь только одну связь с одной реализацией *A*;
- многие-ко-многим (*m*-*n*), когда одна реализация *A* может иметь одну или более связей, каждая с одной реализацией *B*, и каждая реализация *B* может иметь несколько связей, каждая с одной реализацией *A*.

Соответствующие типы функциональности можно определить также и для *n*-арных типов-связи.

На практике эту характеристику рассматривают вместе с характеристикой полной или частичной связи.

Полная связь, определенная на типах-сущностях *A* и *B*, требует, чтобы каждая реализация *A* и *B* принимали участие в какой-либо связи этого типа-связи.

Частичная связь, определенная на типах-сущности *A* и *B*, требует, чтобы некоторые, но не обязательно все, реализации *A* и некоторые, но не обязательно все, реализации *B* участвовали в какой-либо связи этого типа-связи.

Возможны четыре случая, полностью независимые от функциональности связи:

A	*B*	
все	все	(полная)
все	некоторые	(частичная)
некоторые	все	(частичная)
некоторые	некоторые	(частичная)

Комбинация функциональности и полноты/частичности устанавливает мощность связи.

Мощность связи — это число, указывающее, сколько раз реализация типа-сущность участвует в связях определенного типа-связи.

Понятие функциональной зависимости применяется только к множеству реализаций-связи, но можно определить функциональную зависимость между подмножеством совокупностей типов-сущности и другим типом-сущности.

Некоторые подходы сущность-атрибут-связь, которые допускают только бинарные типы-связи, ограничиваются типами-связи «один-ко-многим», включая «один-к-одному» и «нуль или один-ко-многим». В других допускаются также бинарные типы-связи «многие-ко-многим».

В общем случае, не только сущности, но также и связи могут иметь атрибуты. Но некоторые подходы сущность-атрибут-связь (обычно те, которые допускают только бинарные связи) не позволяют связям иметь атрибуты.

Далее в обсуждении подходов сущность-атрибут-связь выбраны следующие характеристики связей:

- допускаются *n*-арные типы-связи;
- допускаются типы-связи «многие-ко-многим»;
- допускаются атрибуты связей.

Г.2 Грамматика и семантика

Необходимы лингвистические конструкции для именования и описания сущностей, атрибутов, типов-сущности и т. д. Все подходы сущность-атрибут-связь имеют такие конструкции — явно определяемые или неявно предполагаемые.

Основными лингвистическими конструкциями являются:

- имена-типа-сущности — лексические объекты, которые ссылаются на типы-сущности;
- имена-типа-атрибута — лексические объекты, которые ссылаются на типы-атрибута;
- имена-типа-связи — лексические объекты, которые ссылаются на типы-связи.

Для подходов, различающих также типы-значения или домены, дополнительно необходимы имена-типа-значения или имена-доменов — лексические объекты, ссылающиеся на типы-значения или домены.

Для имен-типа будут использоваться прописные буквы.

В формальных языках, использующихся в подходах сущность-атрибут-связь для описания проблемной области, необходимы более сложные конструкции для описания различных типов. Для этих конструкций будет использоваться общий термин «описание». Описание — это граф. рисунок или языковая конструкция, которая описывает тип, а также перечисляет его характеристики. Некоторые авторы используют слово «тип» для понятия «описание-типа».

В подходах сущность-атрибут-связь, рассматриваемых в данном приложении, выделим следующие описания:

- описания-типа-сущности, перечисляющие типы-атрибута и другие характеристики (такие, как идентификатор) для типов-сущности;
- описания-типа-атрибута, перечисляющие характеристики типов-атрибута;
- описания-типа-связи, перечисляющие типы-сущности, на которых определен тип-связи, и дополнительные характеристики, такие, как категория типа-связи, мощность, типы-атрибута при их наличии.

В вариантах, в которых предусмотрены типы-значения или домены, они тоже должны быть описаны посредством описаний-типов-значения или описаний-доменов, перечисляющих характеристики типов-значения.

Ниже приводится синтаксис языка, используемого в этом приложении, для описания концептуальной схемы для примера проблемной области из приложения Б. Но, как уже было замечено, его не следует интерпретировать как предложение для стандартизации этого языка в подходах сущность-атрибут-связь.

```
концептуальная-схема        = "CONCEPTUAL SCHEMA"
                                  имя-концептуальной-схемы
                              описание-типа-сущности
                                  {описание-типа-сущности}
                              описание-типа-связи
                                  {описание-типа-связи}.
описание-типа-сущности      = "ENTITY-TYPE" имя-типа-сущности
                              "IDENTIFIER" идентификатор
                              "DESCRIPTION" описание-типа-атрибута
                                  {описание-типа-атрибута}.
описание-типа-связи         = "RELATIONSHIP-TYPE" имя-типа-связи
                              "DIMENSION" целое-без-знака
                              "COLLECTION" имя-типа-сущности
                                  {имя-типа-сущности}
                              "CARDINALITY"
                                  имя-типа-сущности minc "," maxc
                                  {имя-типа-сущности minc "," maxc}
                              {"FUNCTIONAL DEPENDENCY"
                                  имя-типа-сущности {имя-типа-сущности}
                                  "ON имя-типа-сущности}
                              ["IDENTIFIER" идентификатор]
                              ["DESCRIPTION"
                              описание-типа-атрибута
                                  {описание-типа-атрибута}].
описание-типа-атрибута      = имя-типа-атрибута.
minc                        = целое-без-знака | буква.
maxc                        = целое-без-знака | буква.
имя-концептуальной-схемы    = идентифицирующее-имя.
имя-типа-сущности           = идентифицирующее-имя.
идентификатор               = имя-типа-атрибута {"," имя-типа-атрибута}.
имя-типа-атрибута           = идентифицирующее-имя.
имя-типа-связи              = идентифицирующее-имя.
идентифицирующее-имя        = буква {буква | цифра | дефис}.
```

Буква, цифра, дефис и целое-без-знака далее не определяются.

Идентифицирующие-имена должны быть уникальными в пределах концептуальной схемы.

Г.3 Моделирование

Методы моделирования на основе подходов сущность-атрибут-связь. Эти методы можно поделить на две категории.

В методах непосредственного моделирования проблемной области сущности и связи идентифицируются в высказываниях проблемной области.

Моделирование в два этапа состоит из:

- идентификации атрибутов, т. е. любых значений, несущих информацию;
- построения типов-сущности и типов-связи посредством анализа типов-атрибута.

Г.3.1 Некоторые прагматические правила моделирования

В подходах сущность-атрибут-связь связи определяются между сущностями, а не между атрибутами, следовательно, важен выбор сущностей. Предложены некоторые прагматические правила, позволяющие решать такие проблемы:

а) если тип-сущности имеет только один соответствующий тип-атрибута, его, возможно, следует классифицировать как тип-атрибута какого-либо другого типа-сущности;

б) если атрибуты нескольких сущностей ссылаются на одну и ту же сущность, эта сущность должна классифицироваться как полноправный тип-сущности.

Объяснение этих прагматических правил показывает, что выбор должен производиться посредством изучения проблемной области, а не только ее описания. Решение о том, что является типом-атрибута, а что типом-сущности, не может быть принято априори. Подход сущность-атрибут-связь — итеративный, и то, что представляется атрибутом на ранней стадии процесса моделирования, может оказаться сущностью и наоборот.

Связи — это не сущности. Однако можно построить сущность, которая является трансформацией связи.

Для этого может существовать несколько причин. Одна причина — дать понятию, первоначально представленному типом-связи, особый идентификатор, а не использовать идентификаторы совокупности сущностей, другая — обеспечить наиболее удобное, по мнению проектировщика, представление проблемной области.

Г.3.2 Формальные правила моделирования

Верификация

Верификация позволяет гарантировать, что каждая реализация типа-сущности или типа-связи имеет только одно значение любого типа-атрибута.

Нормализация

Нормализация в подходе сущность-атрибут-связь гарантирует, что ни один атрибут связи не может быть верифицированным по отношению к подмножеству идентификатора связи.

Декомпозиция

Декомпозиция в подходах сущность-атрибут-связь позволяет без потери семантики разбить тип-связи размерности *n* на несколько типов-связи с меньшими размерностями при условии сохранения функциональных зависимостей, определенных для типа-связи. Необходимо убедиться, что в общей части используются одни и те же реализации.

Г.4 Пример описания

Описание концептуальной схемы на языке, определенном в разделе Г.2:

```
CONCEPTUAL SCHEMA       регистрация-автомобилей
ENTITY-TYPE             изготовитель
IDENTIFIER              ид-изготовителя
DESCRIPTION             ид-изготовителя
                        функционирует

ENTITY-TYPE             модель-автомобиля
IDENTIFIER              ид-модели
DESCRIPTION             ид-модели
                        описание-расхода-топлива

ENTITY-TYPE             автомобиль
IDENTIFIER              регистрационный-номер
DESCRIPTION             регистрационный-номер
                        серийный-номер
                        дата-уничтожения

ENTITY-TYPE             расход-топлива
IDENTIFIER              ид-года
DESCRIPTION             ид-года
                        max-расходы
```

ENTITY-TYPE	гараж
IDENTIFIER	ид-гаража
DESCRIPTION	ид-гаража торгует
ENTITY-TYPE	человек
IDENTIFIER	ид-человека
DESCRIPTION	ид-человека
ENTITY-TYPE	передача
IDENTIFIER	передаваемый-автомобиль дата-передачи, последовательный-номер
DESCRIPTION	передаваемый-автомобиль дата-передачи, последовательный-номер
RELATIONSHIP-TYPE	производится-кем
DIMENSION	2
COLLECTION	изготовитель модель-автомобиля
CARDINALITY	изготовитель 0, *n* модель-автомобиля 1, 1
RELATIONSHIP-TYPE	производится-кем
DIMENSION	2
COLLECTION	изготовитель автомобиль
CARDINALITY	изготовитель 0, *n* автомобиль 1, 1
RELATIONSHIP-TYPE	является-моделью
DIMENSION	2
COLLECTION	модель-автомобиля автомобиль
CARDINALITY	изготовитель 0, *n* автомобиль 1, 1
RELATIONSHIP-TYPE	год-выпуска
DIMENSION	2
COLLECTION	расход-топлива автомобиль
CARDINALITY	расход-топлива 0, *n* автомобиль 1, 1
RELATIONSHIP-TYPE	передача-иг
DIMENSION	4
COLLECTION	автомобиль передача изготовитель гараж
CARDINALITY	автомобиль 0, *n* передача 1, 1 изготовитель 0, *n* гараж 0, *n*
RELATIONSHIP-TYPE	передача-гч
DIMENSION	4
COLLECTION	автомобиль передача гараж человек
CARDINALITY	автомобиль 0, *n* передача 1, *n* гараж 0, *n* человек 0, *n*

RELATIONSHIP-TYPE	передача-чг
DIMENSION	4
COLLECTION	автомобиль
	передача
	человек
	гараж
CARDINALITY	автомобиль 0, *n*
	передача 1, *n*
	человек 0, *n*
	гараж 0, *n*

RELATIONSHIP-TYPE	передача-чч
DIMENSION	4
COLLECTION	автомобиль
	передача
	человек
CARDINALITY	автомобиль 0, *n*
	передача 1, *n*
	человек 0, *n*
	человек 0, *n*

Г.5 Контрольный список для концептуальной схемы

Дальнейший анализ иллюстрирует, собраны ли в концептуальной схеме все необходимые высказывания о проблемной области. Знак Е подразумевает, что в модели данное утверждение описано, * отсылает к примечаниям в конце раздела.

НАЛИЧИЕ	НЕОБХОДИМЫЕ ВЫСКАЗЫВАНИЯ
Е	1. Проблемная область связана с регистрацией автомобилей и ограничена сферой интересов Органа Регистрации.
Е	2. Каждый изготовитель автомобилей имеет уникальное имя.
*	3. Новые изготовители автомобилей могут начать производство при условии, что они имеют разрешение Органа Регистрации.
*	4. Орган Регистрации не может отменить разрешение.
*	5. Одновременно могут работать не более пяти автономных изготовителей.
*	6. Изготовитель может прекратить работу при условии, что он больше не владеет автомобилями.
Е	7. Каждый изготовитель автомобилей производит автомобили нескольких моделей.
Е	8. Автомобиль относится к определенной модели.
*	9. Изготовитель автомобилей присваивает серийный номер каждому выпускаемому автомобилю.
*	10. Этот серийный номер является уникальным для всех автомобилей одного изготовителя.
*	11. Вновь произведенный автомобиль регистрируется Органом Регистрации, как только это окажется возможным.
(Е)	12. В этот момент автомобиль регистрируется как принадлежащий изготовителю. Поэтому первым владельцем автомобиля будет его изготовитель.
*	13. Только Орган Регистрации может назначить регистрационный номер каждому регистрируемому автомобилю.
(Е)	14. Этот регистрационный номер уникален для всех автомобилей в течение всего времени.
Е	15. Автомобиль имеет год выпуска.
*	16. Автомобиль может регистрироваться как произведенный в предыдущем году только в течение января.
Е	17. Автомобили могут быть уничтожены, после чего записывается дата уничтожения.
*	18. История автомобиля должна храниться до конца второго календарного года с момента его уничтожения. После этого она удаляется.
(Е)	19. Имя модели автомобиля уникально для моделей автомобилей в течение всего времени.
*	20. Любая определенная модель автомобиля производится только одним изготовителем.
*	21. Время от времени вводятся новые модели.
Е	22. Все автомобили одной и той же модели потребляют одинаковое количество топлива.
(Е)	23. Расход топлива должен быть известен Органу Регистрации.
*	24. Расход топлива может находиться в пределах 4—25 л на 100 км.
*	25. Расход топлива, усредненный по всем автомобилям, произведенным соответствующим изготовителем в определенный год, не должен превышать максимального значения, одинакового для всех изготовителей.
Е	26. Максимальный расход топлива может меняться от года к году.

*	27. В конце января изготовителю, не выполнившему требование по расходу топлива в предыдущем году, посылается уведомление.
E	28. Каждый гараж имеет уникальное имя.
*	29. Могут открываться новые гаражи.
E	30. Гаражи могут владеть автомобилями.
*	31. В любое время гараж может владеть автомобилями, произведенными не более чем тремя изготовителями (какими именно, не имеет значения, и они могут меняться со временем).
*	32. Существующий гараж может быть закрыт при условии, что в нем нет зарегистрированных за ним автомобилей.
E	33. Определенное лицо может иметь один или более автомобилей, зарегистрированных как принадлежащие ему.
E	34. Два или более человека могут иметь в одновременном владении один или несколько автомобилей.
E	35. Люди имеют уникальные имена.
*	36. Люди известны Органу Регистрации только в том случае, если они имеют или имели один или более автомобилей, которые известны Органу Регистрации.
*	37. В любой момент времени автомобилем владеет его изготовитель или гараж, или некое лицо, или группа лиц, но не вместе две или более из этих категорий.
E	38. Передача права на владение регистрируется, включая дату передачи предыдущего владельца (владельцев) и нового владельца (владельцев).
*	39. Передача права на владение может регистрироваться после уничтожения автомобиля.
*	40. Но передача права на владение может регистрироваться после уничтожения автомобиля при условии, что передача права на владение произошла до уничтожения автомобиля.
E	41. Каждый изготовитель распределяет новые автомобили нескольким независимым гаражам, каждый из которых может получить автомобили от нескольких изготовителей.
(E)	42. Поэтому гараж всегда будет вторым владельцем автомобиля.
E	43. Изготовители не распределяют автомобили другим изготовителям или непосредственно людям.
E	44. Каждый гараж может продавать (т. е. производить передачу зарегистрированного права владения) новые или бывшие в употреблении автомобили и может покупать (т. е. производить передачу зарегистрированного права владения) автомобили у людей.
E	45. Гаражам не разрешается продавать автомобили другим гаражам.
E	46. Гаражам не разрешается продавать автомобили изготовителям.
E	47. Люди могут продавать автомобили друг другу или покупать их друг у друга.

Замечания по подходам сущность—атрибут—связь

* Помета в круглых скобках означает, что концептуальная схема обеспечивает описание утверждения в информационной базе, но это утверждение не может быть выражено в виде правила. Ограничения уникальности в пунктах 14 и 19 поддерживаются, но не на все время. Так, отсутствует правило, которое запрещало бы иметь двум разным автомобилям один и тот же регистрационный номер, но в различные моменты времени.

* Правила установления полномочий не включены (например, пункты 3, 4, 9, 11 и 13).

* Правила проверки достоверности, хотя они и являются статическими, не включены (например, пункты 5, 10, 24, 25, 31, 36 и 39).

* Взаимное исключение связей не включено (например, пункт 37).

* Не включены динамические правила или ограничения, поэтому пункты 6, 20 и 32 не применимы. Заметим, что в пункте 20 не имеется в виду статическое ограничение «любая определенная модель автомобиля производится одним изготовителем в один момент времени».

* Предписывающие правила взаимодействия не являются частью концептуальной схемы (например, пункты 16, 18, 21, 29, 40).

Г.6 Отображение концептуальной схемы сущность-атрибут-связь в схему сетевой базы данных и схему реляционной базы данных

Процедуру преобразования концептуальной схемы сущность-атрибут-связь сначала в схему сетевой базы данных и затем в схему реляционной базы данных можно в общих чертах охарактеризовать следующим образом:

а) поместить типы-атрибута одного типа-сущности в тип-записи;

б) взять типы-связи «один-ко-многим» и преобразовать их в типы-набора;

в) преобразовать типы-связи «многое-ко-многим», n-арные ($n>2$) типы-связи и типы-связи, имеющие типы-атрибуты, в типы-записи и преобразовать соединительные линии в типы-набора.

Чтобы получить реляционную схему базы данных, следует отобразить типы-записи в отношениях. При этом наборы следует представить «ключами-владельцев», которые ссылаются на «кортежи-владельцев» (например, производится-кем в модель-автомобиля).

ПРИЛОЖЕНИЕ Д
(справочное)

Подходы на основе бинарных отношений

Д.1 Суть подходов

Общее во всех подходах к бинарной модели — это представление «элементов» информации посредством реализаций бинарных ассоциаций, т. е. предложений, в которых участвуют только два терма.

Общая особенность подходов бинарных отношений — это использование графической системы записей для иллюстрации частей концептуальной схемы и информационной базы.

Информационная база — это описание части проблемной области. Проблемная область в каждый момент времени представляет собой множество реализаций бинарных ассоциаций и реализаций сущностей. Состав и семантика проблемной области должны моделироваться способом, согласующимся с существующими высказываниями об этих реализациях.

Процесс моделирования проходит в две фазы: с одной стороны, имеется проблемная область, с другой стороны — информационная база и концептуальная схема.

Разработчик модели или аналитик выполняет три задачи: он дает имя или устанавливает соглашения об именовании для сущности, которую он наблюдает в проблемной области; он квалифицирует ее (что также является своего рода именованием) и готовит концептуальную схему, которая будет описывать эту квалификацию, а также позволит заполнить ее именем или ссылкой на наблюдаемую сущность.

В подходах на основе бинарных отношений отправным моментом является то, что основной функцией концептуальной схемы является введение классификации и порядка в описание проблемной области, а не просто исчерпывающее описание на уровне реализаций. Моделирование проблемной области на уровне реализаций может считаться достаточным для некоторых других подходов, так как оно может, в принципе, достичь любой требуемой степени полноты. Но этому методу моделирования не достает многих прагматических и практически полезных свойств, которые, например, являются следствием понятия типа.

Подходы на основе бинарных отношений, как они здесь представлены, полностью реализуют различие между лексическими и нелексическими объектами, как способ избежать смешения объектов и их имен, что часто имеет место при описании информации и при обменах информацией.

Эти подходы сами по себе не рассматривают язык, на котором описывается проблемная область, но подчеркивают его существование и необходимость в нем. Предполагается как постулат, что существует семантическая эквивалентность между:

-высказываниями о проблемной области;

- предложениями, описывающими их в концептуальной схеме и информационной базе (не обязательно в соответствии один к одному).

Реализуя цели концептуальной схемы, можно сделать к ним следующие замечания.

Подходы на основе бинарных отношений учитывают использование формального языка для полного описания концептуальной схемы, а также графического формализма или системы обозначений, охватывающей основные аспекты концептуальной схемы. Использование основных понятий, таких как сущность, имя-сущности, бинарное отношение и ограничение, дают возможность полного описания как статических, так и динамических правил и ограничений концептуальной схемы и информационной базы. Фундаментальное различие между объектами и их именами намного облегчает понимание и обучение. Это замечание следует учитывать при обеспечении общей основы для понимания общего поведения проблемной области.

В подходах на основе бинарных отношений основные понятия являются атомарными. Вследствие этого любое данное ограничение формулируется явно. Поэтому изменения в проблемной области можно отразить посредством добавления или удаления (но не изменения) типов бинарных отношений, сущностей или имен сущностей, или посредством изменения ограничений. Это приводит к большей стабильности по сравнению с подходами, где некоторые ограничения «упаковываются» в «базовые» понятия. Это замечание следует учитывать при манипулировании информацией о проблемной области и определении допустимых изменений в ней.

Концептуальная схема должна содержать возможно более глубокую структуру описания проблемной области с помощью семантически несводимых элементов. Различие между объектами и их именами, приводимое в подходах на основе бинарных отношений, обеспечивает глубокий структурный аспект, тогда как бинарные отношения и аппарат подтипов обеспечивают семантическую несводимость. Это замечание следует учитывать при обеспечении основы для интерпретации внешних и внутренних синтаксических форм, представляющих информацию о проблемной области.

В подходах на основе бинарных отношений максимальная степень стабильности достигается при использовании семантически несводимых конструкций и конструкций, свободных от дополнительных ограничений. Это имеет место в подходах на основе бинарных отношений. Это замечание следует учитывать при обеспечении основы для взаимного отображения внешних и внутренних схем.

Д.2 Фундаментальные понятия

Признавая различия между лексическими и нелексическими объектами и их разное назначение, необходимо ввести понятие типа. Действительно, один или несколько лексических объектов определяют один или несколько объектов, соответствующих сущностям в проблемной области.

Формально имеем:

а) лингвистические объекты:

1) лексические, соответствующие отдельным «произносимым» сущностям,

2) нелексические, соответствующие «непроизносимым» сущностям. Нелексические объекты определяют сущности посредством лексических объектов;

б) реализация таких бинарных отношений между двумя лингвистическими объектами:

1) идея — между двумя нелексическими объектами,

2) мостик — между лексическим и нелексическим объектами,

3) фраза — между двумя лексическими объектами;

в) типы лингвистических объектов:

1) лексические (LOT),

2) нелексические (NOLOT).

Такая классификация на типы — существенная часть процесса моделирования. Она зависит обычно от синтаксиса и семантики перевода высказываний о проблемной области на язык информационной системы. Она зависит также от понимания разработчиками модели слабых мест, абстракций, интерпретаций и соглашений об именовании, появляющихся в процессе этого перевода.

Этим типам лингвистических объектов соответствуют типы идей, мостиков и фраз;

г) типы бинарных отношений:

1) идея — передающие «реальную» информацию между нелексическими объектами,

2) мостик — связывающие именующие лексические объекты с именуемыми нелексическими объектами,

3) фраз — обозначающие связи только между лексическими объектами.

При наличии типов лингвистических объектов целесообразно классифицировать их далее на подтипы, основываясь на различиях между разделяемыми и неразделяемыми типами бинарных отношений на типах лингвистических объектов с одной стороны и на чисто «концептуальной» классификации (будущего) множества реализаций этих типов лингвистических объектов с другой. Между лексическим и нелексическим типами объектов отношение подтипа по очевидным причинам не разрешается. Подтипы могут быть пересекающимися;

д) подтипы:

1) нелексические — которые порождают подмножества нелексических типов объектов,

2) лексические — которые порождают подмножества лексических типов объектов.

Принимая во внимание, что некоторые высказывания проблемной области или, более точно, некоторые необходимые высказывания несут информацию о самих высказываниях и поэтому переводятся в предикаты о лингвистических объектах и экземпляры бинарных отношений, мы должны выразить их в статических и динамических ограничениях (на уровне типов). Так как существование этих высказываний, возможно, повлияло на форму проблемной области, такие ограничения также будут воздействовать на процесс моделирования;

е) ограничения — статические и динамические.

Все операторы, с точки зрения концептуальной схемы, действуют на уровне «реализации» или «экземпляра». Аксиомой, вводимой для того, чтобы эти операторы имели смысл, является следующее утверждение:

— в каждом состоянии информационной базы каждый экземпляр лингвистического объекта должен быть реализацией какого-либо типа лингвистического объекта, каждый экземпляр бинарного отношения является реализацией некоторого типа бинарного отношения, и для каждого лингвистического объекта существует способ уникальной ссылки на него посредством одного или нескольких лексических объектов.

Операторами, определенными для этого примера подходов на основе бинарных отношений, являются:

ж) операторы:

1) list (вывести) (...экземпляр бинарного отношения),

2) add (добавить) (...экземпляр бинарного отношения),

3) delete (удалить) (...экземпляр бинарного отношения),

4) qualify (квалифицировать) (...лингвистический объект, как относящийся к некоторому подтипу),

5) unqualify (деквалифицировать) (...лингвистический объект из подтипа),

6) equate (приравнять) (...идентифицировать, что два различных лингвистических объекта означают или ссылаются на один и тот же объект),

7) together do (совместно выполнить) (...выполнять последовательность операторов, но принимать решение о ее допустимости на основе всей последовательности в целом).

Семантику этих элементарных операторов можно непосредственно интерпретировать в терминах «выборка», «вставка», «удаление» и «допустимое действие». Те же или подобные им операторы существуют в

большинстве подходов. Но операторы qualify, unqualify и equate типичны именно для нашего примера подходов на основе бинарных отношений. Операторы qualify и unqualify появляются в связи с понятием подтипов.

Оператор qualify принимает ссылку на лингвистический объект и преобразует ее в ссылку на лингвистический объект в одном из подтипов типа лингвистического объекта, содержащего данный лингвистический объект.

Оператор unqualify делает противоположное: он исключает возможность использования ссылки для лингвистических объектов соответствующего подтипа.

Оператор «equate» — особый, так как он является следствием реализации различия между именами и объектами.

Оператор equate устанавливает, что два различных терма ссылаются на одну и ту же сущность в проблемной области и требуют «нелексической» идентификации, сохраняя при этом все существующие соглашения о ее именовании.

Д.3 Грамматика и семантика

Д.3.1 Язык и его связь с проблемной областью

Существуют некоторые общие фундаментальные понятия, характеризующие грамматику любого языка. К ним относятся:

ТЕРМ

Лингвистический объект, ссылающийся на сущность.

ПРЕДЛОЖЕНИЕ

Лингвистический объект, представляющий определенное высказывание.

ФУНКТОР

Лингвистический объект, который определяет функцию от других лингвистических объектов, имеющую в качестве аргументов (вход) список лингвистических объектов и выдающую в качестве значения (выход) единственный уникально определенный лингвистический объект.

Функторы, используемые в подходах на основе бинарных отношений, — это такие, как предикат, оператор, квантор и т. д.

Можно дать следующую интерпретацию этих понятий на языке, используемом для определения концептуальной схемы в подходах на основе бинарных отношений.

ТЕРМЫ

ЭЛЕМЕНТАРНЫЕ ТЕРМЫ — имена типов лингвистических объектов, имена типов бинарных отношений, имена ролей, индивидуальные имена.

ОПИСАТЕЛЬНЫЕ ТЕРМЫ — ссылки, связывающие один или более лексических объектов (элементарных термов) с нелексическим объектом.

ПРЕДЛОЖЕНИЯ

ЭЛЕМЕНТАРНЫЕ ПРЕДЛОЖЕНИЯ — объявления типов лингвистических объектов, типов бинарных отношений.

ФУНКТОРНЫЕ ПРЕДЛОЖЕНИЯ — предложения, полученные посредством применения функторов (главным образом, предикатов) к ТЕРМАМ и ЭЛЕМЕНТАРНЫМ ПРЕДЛОЖЕНИЯМ и используемые для определения ограничений.

Элементарные предложения (объявления) строятся, главным образом, при помощи примитивного предиката «является» в некоторой эквивалентной форме (а именно — «добавить» экземпляр «мостика»). Этот предикат связывает терм на уровне концептуальной схемы с термом на метаконцептуальном уровне, содержащем такие имена, как «ТИП ОБЪЕКТА» и «ТИП ИДЕИ». Другими примитивными предикатами в модели являются «является-подмножеством», «имеет-роли» и «связывает-с», которые используются в объявлении ролей для типов идеи, мостика и фразы.

На практике можно еще найти полезные примитивные (или полупримитивные) предопределенные предикаты, описывающие логическое представление для типов лексических объектов, но здесь они не рассматриваются, чтобы не отвлекаться от существенных вопросов.

Д.3.2 Формальный синтаксис

(R1) концептуальная схема = «begin»
«CONCEPTUAL-SCHEMA called» имя-схемы ";"
объявление-nolot {объявление-nolot}
объявление-lot {объявление-lot}
{объявление-подтипа}
{объявление-идеи}
объявление-мостика {объявление-мостика}
{объявление-фразы}
{объявление-ограничения}
«end».

(R2) объявление-nolot = «NOLOT called»
(имя-nolot | список-имен-nolot).

(R3) объявление-lot = «LOT called» (имя-lot | список-имен-lot).

```
(R4) объявление-подтипа      = подтип-nolot | подтип-lot.
(R5) подтип-nolot            = «NOLOT called»
                                   (имя-nolot-1 | список-имен-nolot-1)
                               «is subtype-of NOLOT called»
                                   имя-nolot-2 ";".
(R6) подтип-lot              = «LOT called»
                                   (имя-lot-1 | список-имен-lot-1)
                               «is subtype-of LOT called» имя-lot-2 ";".
(R7) объявление-идеи         = «IDEA (with-first ROLE (called»
                                           имя-роли-1
                                       «and on NOLOT called»
                                           имя-nolot-2 ")"
                                   «and with-second ROLE (called»
                                           имя-роли-2
                                       «and on NOLOT called»
                                           имя-nolot-2"))"
                               «in called» имя-идеи ";".
(R8) объявление-мостика      = «BRIDGE (with-first ROLE (called»
                                           имя-роли-1
                                       «and on NOLOT called»
                                           имя-nolot-2")"
                               «and with-second ROLE (called»
                                           имя-роли-2
                                       «and on LOT called»
                                           имя-lot-2"))"
                               «is called» имя-мостика " ; ".
(R9) объявление-фразы        = «PHRASE (with-first ROLE (called»
                                           имя-роли-1
                                       «and on LOT called»
                                           имя-lot-2 " ) "
                               «and with-second ROLE (called»
                                           имя-роли-2
                                       «and on LOT called»
                                           имя-lot-2"))"
                               «is called» имя-фразы " ; ".
(R10) объявление-ограничения = «CONSTRAINT called» имя-ограничения
                               «is declared as» предложение " ; ".
(R11) список-имен-nolot      = "{" имя-nolot {имя-nolot} "}" " ; ".
(R12) список-имен-lot        = "{" имя-lot {имя-lot} "}" " ; ".
```

Примечание — Список-имен-nolot-1 и список-имен-lot-1 имеют тот же синтаксис.

```
(R13) имя-схемы              = идентифицирующее-имя.
(R14) имя-nolot              = идентифицирующее-имя.
(R15) имя-lot                = идентифицирующее-имя.
(R16) имя-роли               = идентифицирующее-имя.
```

Примечание — Имя-nolot-1 и т. д. имеют тот же синтаксис.

```
(R17) имя-идеи               = идентифицирующее-имя.
(R18) имя-мостика            = идентифицирующее-имя.
(R19) имя-фразы              = идентифицирующее-имя.
(R20) идентифицирующее-имя   = " ' ' " буква {буква | символ | дефис}" ' ' ".
```

Примечание — Синтаксис предложений в определении ограничений, а также синтаксис букв, цифр и т. п. здесь не уточняется.

Определение концептуальной схемы можно «запомнить», если считать, что каждое «объявление» является предложением, которое нужно добавить командой add. «Удалить», «вывести» и т. д. имеют в данном контексте обычный смысл.

Пример:

```
add CONCEPTUAL-SCHEMA called «РЕГИСТРАЦИЯ-АВТОМОБИЛЕЙ»;
add NOLOT called {'МОДЕЛЬ-АВТОМОБИЛЯ' 'ДАТА' 'ПЕРЕДАЧА'
                 'ВЛАДЕЛЕЦ'};
add LOT called {'РЕГИСТРАЦИОННЫЙ-НОМЕР' 'СЕРИЙНЫЙ-НОМЕР'};
add NOLOT called 'ИЗГОТОВИТЕЛЬ' is subtype-of NOLOT called
                 'ВЛАДЕЛЕЦ';
```

Д.3.3 Семантика

Правило	Семантика	Описание
(R2)	(S1)	Экземпляры NOLOT с однозначно идентифицирующими именами «имя-nolot» добавляются к концептуальной схеме
(R3)	(S2)	То же, что и (S1) для экземпляров LOT с именами имя-lot LOT и NOLOT
	(S3)	— непересекающиеся подтипы типа метаобъектов «ТИП-ОБЪЕКТА», другими словами, «имя-lot» не должно уже существовать как «имя-nolot»
(R5)	(S4)	Упомянутые NOLOT должны уже существовать
	(S5)	Так как деление на подтипы — транзитивное свойство, не должно быть замкнутого «цикла» при определении подтипов
(R6)	(S6)	То же, что и (S4) с соответствующими изменениями
	(S7)	То же, что и (S5)
(R7)	(S8)	Добавляются экземпляры ролей с «именем-роли»; два «имени-роли» должны быть различными, если они определены «на» том же NOLOT
	(S9)	То же, что и (S4)
	(S10)	Добавляется экземпляр ИДЕИ с однозначно идентифицирующим именем «имя-идеи»
(R8)	(S11)	То же, что и (S8)
	(S12)	Должны уже существовать упомянутые LOT и NOLOT
	(S13)	То же, что и (S10) с соответствующими изменениями
	(S14)	Упомянутый LOT не должен появиться в другом объявлении МОСТИКА
(R9)	(S15)	То же, что и (S8) с соответствующими изменениями
	(S16)	То же, что и (S6)
	(S17)	То же, что и (S10) с соответствующими изменениями
(R10)	(S18)	Экземпляр ОГРАНИЧЕНИЯ добавляется с однозначно идентифицирующим именем «имя-ограничения»
	(S19)	«Предложение» должно синтаксически правильно описывать предикат (функтор), который должен сохраняться истинным всегда (для всех состояний информационной базы)
	(S20)	Никакие два ограничения не могут противоречить друг другу (это приводит к невозможности заполнения информационной базы)
	(S21)	Набор всех ограничений считается исчерпывающим списком всех существующих правил или ограничений для всех состояний информационной базы, т. е. экземпляр бинарного отношения вводится в информационную базу только при условии, если ни одно ограничение не нарушается
	(S22)	Множество ограничений подразумевает, что допустимы любые изменения в информационной базе, которые не нарушают объявленных ограничений
(R20)	(S23)	Идентифицирующие имена должны быть уникальными в пределах концептуальной схемы, за исключением имен-ролей, которые имеют более ограниченную область уникальности (S10, S11, S15)

Общие замечания по семантике:

- существует сходство между приведенным выше способом описания семантических свойств, связанных с синтаксическими единицами, и атрибутами Кнута;

- отдельно от этой «атрибутной» семантики существует семантика, присущая примитивным конструкциям, и способ ее использования в модели. Отчасти она уже более или менее неформально рассмотрена в предыдущих разделах и дальше будет только упоминаться. К ней относится понятие имени, например, «имя-notol» или «имя-роли». Например, «имя» — терм, ссылающийся на отдельную сущность в проблемной области и использующийся для обозначения единичной сущности. Он является лексическим по существу, т. е. существует соответствие один к одному с какой-либо строкой данного алфавита. Тем не менее следует подчеркнуть, что подходы на основе бинарных отношений полностью поддерживают как синонимы, так и омонимы;

- терминальные элементы, такие как NOTOL или IDEA (которые называют «ключевыми словами»), также являются экземплярами имен (или примитивных понятий), но на метаконцептуальном уровне. Во избежание путаницы в синтаксисе и соответствующем языке эти имена не берутся в кавычки;

- описание семантики (S1—S23) можно рассматривать как множество статических и динамических ограничений, действующих на этом метаконцептуальном уровне.

Ничто не препятствует рассматривать в этих подходах сами концептуальные схемы или любой другой «абстрактный» язык как проблемную область. Этим путем можно ввести несколько уровней описания. Но на каждом уровне описания и в каждом состоянии концептуальной схемы и информационной базы концептуальная схема и та часть проблемной области, которую она описывает, различимы средой.

Причина очевидна, так как концептуальная схема должна быть известна для того, чтобы служить единственным «посредником», под контролем которого возможно манипулирование и доступ к информационной базе.

Д.4 Моделирование

Не существует единственного алгоритма разработки концептуальной схемы.

Один из способов проектирования концептуальной схемы в подходах на основе бинарных отношений — взять в качестве отправного пункта неформальное восприятие проблемной области. Нужно выбрать необходимые высказывания о проблемной области и идентифицировать свободные переменные. Интуитивно это термы (имена типов), ссылающиеся на классы в проблемной области, которые можно заменить определенными экземплярами имен в экземплярах предложения о сущностях, принадлежащих к выбранным классам.

Можно использовать строки, представляющие свободные переменные в качестве возможных имен для типов в концептуальной схеме. Нужно идентифицировать предложения, которые сообщают нечто о других предложениях. Некоторые из них дадут в результате дополнительные типы лингвистических объектов и типы бинарных отношений, а другие будут иметь своим результатом ограничения. Некоторые, возможно, приведут и к тому, и к другому. Первые обычно встречаются там, где дается дополнительная информация об определенном типе предложения, которая может быть добавлена командой add, а другие ограничивают совокупность существующих экземпляров предложений.

После этого можно использовать реализации предложений, выражающие дополнительные высказывания о пространстве сущностей, для контроля экземпляров, выведенных из концептуальной схемы. Если встречаются противоречия, то концептуальную схему нужно модифицировать, чтобы исключить эти противоречия, предполагая, как всегда, что все высказывания выполняются для данной проблемной области.

Возможно, нужно расширить множество реализаций предложений о проблемной области. Это расширение должно генерироваться пользователем совместно с разработчиком информационной системы. Такой этап необходим в тех случаях, когда концептуальная схема и информационная база оказываются недостаточно полным представлением проблемной области.

Необходимо повторять описанные этапы, пока не будет достигнут приемлемый уровень качества.

ПРИЛОЖЕНИЕ Е
(справочное)

Подходы на основе интерпретируемой логики предикатов

Использование интерпретируемой логики предикатов для описания концептуальных схем основывается на представлении о концептуальной схеме и информационной базе как совокупности абстрактных элементов, называемых предложениями. На практике предложения представляются строками символов. Символы кодируются соответствующими наборами состояний физических элементов носителей памяти информационной системы.

Руководящий принцип реализации рассматриваемых подходов, связанных с применением интерпретируемой дедуктивной формальной логической системы, состоит в том, чтобы к информационному процессору, внешнему по отношению к концептуальной схеме и информационной базе, предъявлялись минимальные требования. На практике многие конструкции могут быть встроены в реальный информационный процессор в целях обеспечения эффективности и сокращения затрат.

Сформулированный принцип можно осуществлять по-разному. Для того чтобы обеспечить основной выбор из этих различных возможностей, необходим второй руководящий принцип. Он выводится из требования простоты концептуальной схемы для понимания и применения различными пользователями.

Необходим механизм добавления конструкций произвольной сложности к формальной системе, позволяющей пользователям взаимодействовать с ней на любом требуемом уровне общности понятий. Для соблюдения этого принципа без нарушения первого формальная система должна включать механизм, обеспечивающий возможность определения новых конструкций через уже существующие.

Е.1 Фундаментальные понятия

При моделировании проблемной области существующие в ней предметы — только сущности. Относящиеся к сущности ситуации, которые могут иметь или не иметь место в проблемной области — только высказывания.

Системы и правила дедукции выбираются таким образом, чтобы каждая аксиома интерпретировалась как истинное утверждение о проблемной области, а каждое предложение, непосредственно выводимое из набора предложений, интерпретируемых как истинное утверждение о проблемной области, само интерпретировалось как истинное утверждение о проблемной области.

Правило дедукции — единственное, что требуется в подходах на основе интерпретируемой логики предикатов, сохраняет истинность. Однако аксиомы, выбираемые разработчиком концептуальной схемы, в значительной степени зависят от проблемной области; гарантия корректности возлагается на разработчика.

Используемый формальный язык описывается лишь в той степени, которая необходима для определения основных категорий грамматических конструкций и обеспечения интерпретируемости универсальной машиной Тьюринга.

Конкретный синтаксис не представлен (за исключением случаев, когда необходимо детализировать пример), чтобы любой формальный язык, наследующий рассматриваемый абстрактный синтаксис, мог использоваться для выполнения описываемой программы.

Е.2 Грамматика и семантика

Для придания значения (семантики) различным выражениям в языке необходимо начать с (желательно малого) набора неопределенных понятий, известных как примитивы. Тогда другие понятия формально определяются через неформально понимаемые примитивы и получают значения. Суть каждого примитивного понятия формально вводится путем установления аксиом, предполагаемых истинными. То же самое имеет место в элементарной геометрии, где понятия точки, линии и плоскости берутся как основные (неопределенные), а более сложные конструкции, такие как треугольники и окружности, определяются в терминах этих примитивов и аксиом (постулаты Эвклида), формально устанавливающих свойства точек, линий и плоскостей.

Е.2.1 Абстрактный синтаксис

Определенные общие концепции — это основные понятия (примитивы), характеризующие абстрактный синтаксис (грамматику) любого языка. К ним относятся:

ТЕРМ — лингвистический объект, обозначающий сущность;

ПРЕДЛОЖЕНИЕ — лингвистический объект, представляющий определенное высказывание;

ФУНКТОР — лингвистический объект, который обозначает функцию от других лингвистичееких объектов, имеющую в качестве аргументов (вход) список лингвистических объектов и выдающую в качестве значения (выход) единственный уникально определенный лингвистический объект.

Функтор есть средство, позволяющее конструировать лингвистические объекты произвольной сложности. Все, что представляется необходимым для концептуальной схемы и информационной базы — это функторы первого уровня, т. е. функторы, аргументами которых являются только термы и предложения (никогда — функторы) и значениями которых являются термы или предложения (никогда — функторы). Тем не менее, ничто из поясняемого ниже не препятствует введению функторов высшего уровня в тех случаях, когда это требуется для концептуальной схемы и информационной базы.

Функторы называются чистыми, если их списки аргументов всегда представляют собой или только предложения, или только термы. Функторы называются функторами подстановки, если существует шаблон, в который могут быть подставлены элементы списка аргументов. Функторы могут быть фиксированными или независящими от того, фиксирован список аргументов по длине или нет.

Элементарными функторами называются фиксированные функторы подстановки первого уровня. Существуют четыре вида чистых элементарных функторов подстановки. Это:

Аргументы	Значение
Термы	Терм
Термы	Предложение
Предложения	Предложение
Предложения	Терм

Очевидно, что каждый из характеризуемых выше чистых элементарных функторов применим к концептуальным схемам. Два других вида функторов, являющихся элементарными, но не чистыми, также важны для концептуальных схем: кванторы и абстракторы. Переменные являются видом терма

Аргументы	Значение
Переменная + предложение	Предложение
Переменная + предложение	Терм

Переменные — это аналоги местоимений естественного языка, они относятся к неопределенным сущностям проблемной области, точно так же как местоимения в естественном языке относятся к неопределенным предметам в предложениях с анафорой.

Сконструированные (составные) предложения и термы могут быть открытыми или замкнутыми в зависимости от того, имеют они свободные переменные или нет. Только замкнутые предложения могут быть однозначно интерпретированы, и по этой причине они будут единственным видом предложений, встречающихся в концептуальной схеме и информационной базе.

Данный раздел не дает подробного описания открытых и замкнутых предложений и способа связывания переменных. Достаточно отметить, что путем добавления нужных кванторов из любого предложения можно связать его свободные переменные.

Собственные имена всегда можно исключить и заменить предикатами, использующими абстрактор, называемый «дескриптором» и определяющий собственное имя в контексте.

Е.2.2 Конкретный синтаксис

В противоположность другим примерам, приведенным в стандарте, конкретный синтаксис любой версии подхода интерпретируемой логики предикатов не может быть полностью описан в системе обозначений, представленной в приложении В, т. к. язык в существенной мере не контекстно свободен. Здесь приводится краткое описание конкретного синтаксиса. Часть этого описания зависит от контекста ввиду контекстной чувствительности языка, но следует понимать, что полный синтаксис может быть описан совершенно формально и исчерпывающе на любом языке, приемлемом для описания концептуальных схем и основанном на подходе интерпретируемой логики предикатов. Рассматриваемый ниже язык именуется «*L*». Выбор синтаксических деталей здесь производится для ограничения набора символов, чтобы уделить основное внимание разъяснению сути вопроса.

Итак:

прописная буква = "A" | "B" | "C" | ... | "Z"
строчная буква = "a" | "b" | "c" | ... | "z"
цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".
прим = " ' ".

П р и м е ч а н и е — Это означает единичный начальный символ.

точка = "."
другой знак = "&" | "(" | ")" | "*" | "+" | "-" | "/" | ":" | ";" | "," | "<" | "=" | ">" | "[" | "\" | "]" | "{" | "}" | " ".
прописная-курсив = "*A*" | "*B*" | "*C*" | ... | "*Z*".
строчная-курсив = "*a*" | "*b*" | "*c*" | ... | "*z*".
знак = прописная-буква | строчная-буква | цифра | прим | точка | прописная-курсив | строчная-курсив | другой знак.
строка = знак {знак}.
переменная = (прописная-курсив {прим}) | (строчная-курсив {прим}).
составной-символ = прописная-буква {строчная-буква} {прим}.
числовой-символ = {цифра} [точка] {цифра}.

П р и м е ч а н и е — Строка не должна начинаться и/или заканчиваться ни цифрой, ни точкой.

символ = переменная | составной-символ | числовой-символ | другой знак.
выражение = символ {символ}.

Нетрудно убедиться, что любую строку, являющуюся выражением, можно разложить на символы единственным способом.

предикат-примитив = "Pr" {строчная-буква} {прим}.
прим-переменная-список = (прим переменная) | (прим прим-переменная-список переменная).
атомарное-предложение = "Pr" {строчная-буква} прим-переменная список.
предложение-примитив = атомарное-предложение | ("Для всех" переменная предложение-примитив) | ("Не" предложение-примитив) | ("Если" предложение-примитив "То" предложение-примитив).

Вхождение переменной в предложение-примитив следует понимать буквально. Вхождение — основной текстно-чувствительный аспект *L*. "*x*" не считается входящим в "*x*"'. Вхождение переменной называется связанным в предложении-примитиве, если существует предложение-примитив, содержащее это вхождение переменной, являющееся частью или всем рассматриваемым предложением-примитивом и начинающееся с выражения "Для всех", за которым следует эта переменная. Вхождение переменной свободно в предложении-примитиве, если оно не связано в этом предложении-примитиве. Переменная называется свободной переменной предложения-примитива, если она имеет свободное вхождение в предложение-примитив.

Предложение-примитив называется открытым, если оно имеет по крайней мере одну свободную переменную, в противном случае оно называется замкнутым.

Все, что может быть сформулировано как утверждение, в *L* может выражаться только замкнутыми предложениями-примитивами. Однако даже простейшие утверждения становятся пространственными, громоздкими и неудобными при чтении, если их выразить как предложения-примитивы. Поэтому вводится следующая дополнительная конструкция:

ОПРЕДЕЛЕНИЕ

Любое выражение является определением, если оно утверждается как таковое уполномоченным источником.

За исключением небольшого числа специальных случаев, необходимых для начального развертывания, которые здесь не рассматриваются, все определения будут замкнутыми предложениями в одной из двух следующих форм:

выражение "|ff" выражение;
выражение "=" выражение.

Как будет показано далее, символ "|ff" интерпретируется как связка, которая, будучи вставлена между двумя предложениями, означает их логическую эквивалентность. Последовательность определений в этой форме строится так, что правостороннее выражение (определяющее выражение) каждого определения в последовательности является или предложением-примитивом, или левосторонним выражением (определяемое выражение) какого-либо предыдущего определения в последовательности. Благодаря взаимозаменяемости логически эквивалентных выражений очень сложные утверждения можно представить в сжатой форме, логически эквивалентной предложению-примитиву.

Аналогичный процесс в определениях второй формы, т. е. символ "=", интерпретируется как предикат идентичности, и также обеспечивает взаимозаменяемость выражений. Конечно, в этой ситуации выражения по обе стороны знака идентичности являются термами, а не предложениями. Используя способ концептуального определения, можно определить термы везде, где может появиться переменная (также терм) в атомарном предложении. Все, что может утверждаться о сущности, обозначенной термом, может быть интерпретировано.

Введя концепцию последовательности определений, можно еще кое-что сказать о конкретном синтаксисе *L*. Следующие концепции вводятся здесь для иллюстрации того, что они являются фактически синтаксическими понятиями.

АКСИОМА

Любое замкнутое предложение, утверждаемое в качестве такового авторитетным источником.

ДОКАЗАТЕЛЬСТВО

Любая последовательность замкнутых предложений является доказательством, если для любого предложения *Q*: либо *Q* есть аксиома, либо до *Q* в последовательности есть предложения *P* и "Если *P* То *Q*".

ТЕОРЕМА

Любое предложение, для которого существует доказательство, содержащее это предложение.

Хотя не существует общего алгоритма нахождения доказательства, которое содержит данное предложение и, следовательно, разрешающей процедуры для определения того, является ли предложение-кандидат теоремой или нет (определение «теоремы» — синтаксическое), однако существует механизм определения, является ли данная последовательность доказательством и содержит ли она предложение-кандидат.

Е.2.3 Семантика

Семантика формального языка *L* устанавливается путем присваивания значений его примитивным конструкциям.

Интерпретация двух примитивных связок и примитивного квантора очевидна; она заимствуется из традиционной логики первого порядка.

Е.3 Моделирование

Е.3.1 Классификация аксиом

Из предложений *L* в качестве аксиом выбираются определенные замкнутые предложения. Один набор аксиом *A*1 составляет аксиомы логики и математики. Применение правил вывода позволяет получить дополнительный набор предложений, которые являются теоремами, выводимыми исключительно на основе аксиом *A*1. Хотя существует большое разнообразие различных наборов предложений, любой из которых может быть выбран в качестве *A*1, общая совокупность теорем будет, как правило, одним и тем же во всех случаях. Разумно предположить, что все концептуальные схемы и информационные базы будут включать в себя один и тот же набор *A*1.

Следует отметить, что большинство аксиом *A*1 являются схемами аксиом — утверждениями метаязыка о том, что каждое предложение заданной формы — аксиома. Хотя существует ограниченное число таких схем, множество аксиом в *A*1 фактически не ограничено.

Концептуальная схема не исчерпывает всех предложений. Если бы все возможные предложения были бы выводимыми в концептуальной схеме, это противоречило бы определению. Кроме того, ничто не осталось бы для самой информационной базы. В общем, должно существовать много различных допустимых совокупностей предложений, наборов предложений, несовместимых между собой или с концептуальной схемой. Различные допустимые совокупности предложений, конечно, могут не согласовываться друг с другом. Информационная база будет представлять собой одну из этих допустимых совокупностей предложений вместе с теоремами, выводимыми в данный момент из предложений информационной базы и концептуальной схемы.

Часто для определенных концептуальной схемы и информационной базы требуется, чтобы в информационной базе присутствовал некоторый набор предложений. Эти требования отличаются от аксиом или теорем, из них выводимых. Концептуальная схема может потребовать, чтобы, например, было доступно какое-либо параметрическое значение. Здесь концептуальная схема не определяет точное значение, а утверждает, что оно существует. Предложение, указывающее это значение, должно быть в информационной базе, а не в концептуальной схеме. Может быть указано любое из допустимых значений, но должно присутствовать одно из них.

При наличии концептуальной схемы и требуемых предложений остальная часть концептуальной схемы и информационной базы — вспомогательная. Необходимо лишь, чтобы все предложения вместе образовывали

допустимую совокупность. Конечно, в любой данный момент состояние концептуальной базы — это точно определенный набор предложений.

Должны существовать и другие наборы предложений, совместимые с состоянием концептуальной схемы и информационной базы и, следовательно, допустимые для дополнения.

Е.3.2 К о н с т р у к ц и и

Цель данного подраздела — выработать более удобные формы для примитивных понятий. Сначала рассматриваются конструкции, существенные для базовой логики. В этом начальном изложении буквы «*P*» и «*Q*» будут означать произвольные предложения, а буквы «*X*» и «*Y*» — произвольные термы.

Связки, используемые обычно в элементарной логике, можно легко определить через связки-примитивы:

«(*P* Или *Q*)» эквивалентно «(Если Не *P* То *Q*)»;

«(*P* & *Q*)» эквивалентно «Не (Не *P* Или Не *Q*)»;

«(*P* | ff *Q*)» эквивалентно «((Если *P* То *Q*) & (Если *Q* То *P*))»;

Целесообразно использовать дополнительный квантор:

«Для некоторого *XP*» эквивалентно «Не Для всех *X* Не *P*»

Приведенные четыре определения (фактически, определяющие схемы) вместе с шестью схемами аксиом, устанавливающими свойства предложений «Не *P*», «(Если *P* Тогда *Q*)», «Для всех *XP*» достаточны для всей необходимой логики, не зависимой от специфических предикатов.

Для представления математических формулировок требуется пять предикатов-примитивов: три одноместные и два двухместных. Предикаты-примитивы вводились синтаксически в такой форме:

«Pr» {строчная буква} {прим}

Pr устанавливает, что символ — это предикат, строка строчных букв служит просто для отличия одного предиката от другого, а число символов прим определяет, сколько переменных должно следовать за предикатом, чтобы получилось правильно построенное атомарное предложение. Используя определения, можно ввести более удобные обозначения:

(Нуль *X* | ff Pra'*X*)

(Индивид *X* | ff Prb'*X*)

(Класс *X* | ff Prc'*X*)

Интерпретация этих трех предикатов проста. «Нуль *X*» утверждает, что сущность, обозначенная символом *X*, — нулевая сущность, «Индивид *X*» утверждает, что сущность индивидная, а «Класс *X*» утверждает, что сущность, обозначенная символом *X*, — класс. В каждой проблемной области, в подходах интерпретируемой логики предикатов, все множество сущностей распадается на эти взаимоисключающие категории. Нулевая сущность — грубый эквивалент отсутствия чего-либо. Нулевая сущность существует, т. е. должна существовать теорема «Для некоторого *x* Нуль *x*». Это вид сущности, к которой приводятся невозможные предметы, например «Нуль квадратная-окружность».

Следующие бинарные предикаты фундаментальны:

((*X* = *Y*) | ff Pra' '*X Y*)

((*X* Принадлежит *Y*) | ff Prb' '*X Y*).

«*X* = *Y*» утверждает, что сущность, обозначенная символом *X*, идентична сущности, обозначенной *Y*, и «*X* принадлежит *Y*» утверждает, что сущность, обозначенная *X*, — это член класса, обозначенного символом *Y*. Для того чтобы высказывание «*X* принадлежит *Y*» было истинным, *Y* должен обозначать класс, но это высказывание значимо независимо от того, что обозначает *Y*. Такая ситуация характерна для подходов интерпретируемой логики предикатов и вынуждает использовать явные ограничения. Должна быть введена аксиома, подобная следующей:

«Для любого *X* Для любого *Y* (Если (*X* Принадлежит *Y*) То Класс *Y*)».

Достаточность этих предикатов-примитивов для логики и математики давно продемонстрирована. В обычных формулировках достаточно пяти схем аксиом и семнадцати явных аксиом.

Количество необходимых определений зависит от объема математики, необходимой для описания определенной рассматриваемой проблемной области. В последующем будут излагаться только понятия, относящиеся к примеру приложения Б, и то неформально, главным образом, чтобы ввести обозначения. Во-первых, вводится понятие определенных описаний:

«The *X P*» обозначает такую единственную сущность *X*, что высказывание, утверждаемое P, — истинно, если такая единственная сущность существует.

Описания — это термы или термы специфической формы, которые определяются в контексте всех возможных позиций в атомарных предложениях. Все другие сложные термы, в конечном счете, редуцируются к определенным описаниям. Таким образом, точное определение понятий «класс всех *x* таких, что P», обозначаемого «{*X**P*}» определяется следующим образом: ({*X**P*} = The Y (Класс Y& Для всех *X* (Если Для некоторого *Z* (*X* Принадлежит *Z*) То (*X* Принадлежит *Y*) | ff *P*))).

Это определение вместе с соответствующими аксиомами (не приведенными здесь) гарантирует, что *L* — экстенциональный, т. е. что два класса идентичны тогда и только тогда, когда они имеют одни и те же члены, и что известные парадоксы класса исключаются.

Классы могут задаваться посредством перечисления членов:
({*a*, *b*, *c*, ... *z*} = {*X* \ *X* = *a* Или *X* = *b* Или *X* = *c* Или ...
Или *X* = *z*}).
Упорядоченные пары могут быть определены следующим образом:
([*x*, *y*] = ({*x*}, {*x*, *y*}}).
Тогда отношения являются классами упорядоченных пар:
({*xy* \ *P*} = {*z*\ Для некоторого *x* Для некоторого *y*
(*z* = [*x*, *y*] & *P*)}).
То есть отношение *x* с *y* такое, что *P* является классом упорядоченных пар [*x*, *y*], таких, что *P* — истинно. Утверждать, что *A* находится в связи типа «члена» с *B* значит утверждать, что ([*A*, *B*] Принадлежит {*xy* \ *x* принадлежит *y*}).

Функции, используемые в строго математическом смысле в подходе интерпретируемой логики предикатов, задаются следующим образом:
(*Fx Y* = {*yx* \ *y* = *Y*}).
Функция от *x*, значение которой (для аргумента *x*) есть *Y*, является отношением *y* с *x* таким, что *y* = *Y*. Функции — это просто специальные отношения, где для каждого аргумента имеется единственное значение.

По разным причинам удобно использовать специфическое обозначение для результата применения функции. Вместо *F*(*x*), *L* использует:
(*F*:*x* = The *y* ([*y*, *x*] Принадлежит *F*)).
Если Удвоить = *Fx*, 2*x*, то Удвоить : 3=6.

Подобная конструкция допускает идентификацию класса всех сущностей, вступающих в отношение *R* с каким-либо членом определенного класса. Так:
(*R*: *x* = {*y* \ Для некоторого *z*
(*z* Принадлежит *x* & [*y*, *z*] Принадлежит *R*)}).

Удобно использовать также некоторые элементарные понятия теории множеств. Это такие обычные понятия:
Объединение:
(*x U y* = {*z* \ *z* Принадлежит *x* или *z* Принадлежит *y*}).
Пересечение:
(*x N y* = {*z* \ *z* Принадлежит *x* & *z* Принадлежит *y*}).
Дополнение:
(*C x* = {*y* \ Не (*y* Принадлежит *x*)}).
Подмножество:
(*x B y* Iff Для всех *z* (Если *z* Принадлежит *x* То *z* Принадлежит *y*)).
Оконечное объединение:
(*Un x* = {*y* \ Для некоторого *z* (*z* Принадлежит *x* & *y* Принадлежит *z*)}).
Обратное утверждение:
(*Cnv R* = {*xy* \ [*y*, *x*] Принадлежит *R*}).
Декартово произведение:
(*x X y* = {*zw**z* Принадлежит *x* & *w* Принадлежит *y*}).
Некоторые конкретные сущности определяются и обозначаются особо:
Нулевая сущность:
(*W* = The *x* Нуль *x*).
Пустое множество:
(*O* = {*x* \ Не *x* = *x*}).
Класс индивидов:
(*I* = {*x* \ Индивид *x*}).
Класс элементов (Универсальный класс):
(*E* = {*x* \ Для некоторого *y* (*x* Принадлежит *y*)})
Класс всех функций:
(*Fcn* = {*F* \ Для некоторого *x* Для некоторого *y*,
(*F B x X y*
& для всех *x*' Для всех *y*' Для всех *y*' '
(Если [*y*', *x*'] Принадлежит *F*
& [*y*'', *x*'] Принадлежит *F*
То *y*' = *y*' '))}).
Левый домен произвольного отношения:
(*Dl R* = {*x* \ Для некоторого *y* ([*x*, *y*] Принадлежит *R*)}).
Правый домен произвольного отношения:
(*Dr R* = {*y* \ Для некоторого *x* ([*x*, *y*] Принадлежит *R*)}).

Известно, что все математические конструкции могут быть определены в терминах множеств. Здесь это не делается. Однако предполагается, что это проделано таким образом, что все промежуточные определения являются аксиомами. Явно необходимы для примера лишь такие конструкции:

0, 1, 2, ...	Последовательность натуральных чисел (нуль есть пустое множество);
Nn	Класс всех натуральных чисел;
Nrp	Класс всех положительных действительных чисел;
< , >	Предикаты арифметического порядка;
+, —, *, /	Традиционные арифметические операторы;
K	Функция мощности, т. е. *K* : *x* — это кардинальное число — число элементов в классе, обозначенное *x*;
Среднее	Арифметическое среднее функции, то есть Среднее: *F* — арифметическое среднее значение функции, обозначенной *F*, при условии, что эти значения являются действительными числами.

Для того чтобы говорить об именах сущностей, необходимо ввести в качестве индивидов надписи — экземпляры типографских строк (а также их отображения в памяти компьютера) и классы надписей, рассматриваемых как эквивалентные (представляющие одни и те же абстрактные символы).

Для этого используется тот же метод, что и рассматриваемый здесь для концептуальной схемы и информационной базы. Начинаем, как обычно, с примитивных предикатов. Существуют три таких необходимых примитива: надписи, соединение и лексикографическая последовательность.

Надписи — это конечные линейные последовательные образы, составленные из более или менее связанных графических элементов (глифов), выбранных из алфавита знаков. Важно отметить, что каждый экземпляр физической реализации надписи отличается от каждой другой такой физической реализации. Четвертый и шестой глифы в слове «надписи» — различные надписи, хотя они являются членами одного и того же знака. С учетом этих различий приводятся следующие пояснения соответствующих примитивных предикатов.

Надпись

Prd'*x* утверждает, что сущность, обозначенная *x*, является надписью.

Соединение

Pra' ' '*xyz* утверждает, что сущность, обозначенная *z*, является надписью, полученной путем приписывания к сущности, обозначенной *x*, сущности, обозначенной *y*.

Лексикографическая последовательность

Prc' '*xy* утверждает, что сущность, обозначенная *y*, является глифом и элементом знака, который непосредственно лексикографически следует за знаком, элементом которого является глиф, обозначенный *x*.

Необходимо пояснить последний предикат. Он предполагает, что в алфавите имеется порядок. Поскольку единственный рассматриваемый здесь алфавит описан в Г.2.2 данного приложения, естественно предположить, что лексикографический порядок — это порядок, в котором представлены его знаки. Таким образом, если *x* обозначает «*a*», тогда Prc' '*xy* истинно тогда и только тогда, когда *y* означает «*b*».

Теперь глифы можно характеризовать как надписи, которые не могут быть результатом соединения: первый знак — как набор всех глифов, которым лексикографически не предшествует ни один глиф: второй знак — как набор всех глифов, лексикографически следующих за глифами в первом знаке, и т. д. Теперь может быть определена алфавитная функция, например A на подмножестве натуральных чисел от 0 до 127. Значение функции для каждого такого числа — это знак, кодируемый бинарным представлением числа, которое является аргументом. То есть, A:9 = «A».

Определим:

({*x* (Conc *y*) = The *z* Для всех *x*' Для всех *y*' Для всех *z*'
(Если *x*' Принадлежит *x*
& *y*' Принадлежит *y*
& Pra' ' '*x*'*y*'*z*'
То *z*' Принадлежит *z*)).

Теперь имеется механизм орфографического анализа в *L*. Так, слово «Cat» в *L* — это
((*A*:25 Conc *A*:11) Conc *A*:39).

Хотя это не очень легко запоминающееся обозначение, оно совершенно точно. Таким же образом могут быть введены любые другие имена. Очевидно также, как можно описать формальный синтаксис *L* на *L*.

Можно определить новый примитивный предикат обозначение:

Prd' '*xy* утверждает, что сущность, обозначенная *x*, обозначает сущность, обозначенную *y* (является именем для нее).

Это приводит к функции

(Обозначение = *Fx* {*y* \ Prd' '*yx*}).

Обозначение: *x* тогда является набором всех сущностей, взятых как имена для *x* в проблемной области. Естественно, необходимо обеспечить приемлемый набор аксиом, чтобы определить, какие понятия являются именами в определенной проблемной области и какие правила применяются для их назначения, но в любой момент, когда встречается выражение «(*N* Принадлежит Обозначение: *x*)», он обозначает, что то, что обозначено *N*, т. е. именем того, что обозначено *x*.

Понятие реализации чего-либо, что имеет место в воспринимаемом мире, — основная конструкция, допускающая рассуждение о физическом мире в концептуальных схемах. Реализации, рассматриваемые как индивиды, связываются с определенными точками времени и пространства. Реализации считаются мгновенными и точечными, отличимыми одна от другой и от всех других индивидов. Чтобы описать основные свойства реализаций, вводятся примитивные предикаты, а для того, чтобы установить семантику этих свойств, вводятся аксиомы, налагающие ограничения на эти примитивные предикаты. Первоначальные высказывания о реализациях относятся к их местоположению во времени и пространстве.

Но размещение во времени существенно. Единственный примитивный предикат, достаточный для введения времени, — это

((X Раньше чем Y) |ff Pre' 'XY).

Основное отношение можно теперь определить следующим образом:

(Раньше = {xy \ x Раньше чем y}).

Удобно также явно определить обратное отношение:

(Позже = Cnv Раньше).

Класс всех реализаций (в смысле физического универсума) определяется как класс всего, что участвует в отношении Раньше:

(Реализации = Dl Раньше U Dr Раньше).

Теперь можно ввести аксиомы, относящиеся к предикату «Раньше чем».

(Реализации B l).

Для всех x Для всех y

(Если x Раньше чем y Тогда Не y Раньше чем x).

Для всех x Для всех y Для всех z

(Если x Раньше чем y & y Раньше чем z То x Раньше чем z).

Для всех x Для всех y

(Если x Раньше чем y То для некоторого z

(x Раньше чем z & z Раньше чем y)).

Для всех x

(Если x Принадлежит Реализация То Для некоторого y (y Раньше чем x)).

Для всех x

(Если x Принадлежит Реализациям То Для некоторого y (x Раньше чем y)).

Значение этих аксиом вполне очевидно. Первая утверждает, что реализации — это индивиды, вторая — что отношение Раньше асимметрично, третья — что оно транзитивно, а четвертая — что оно плотно. Последние две устанавливают, что для любой реализации всегда есть, по крайней мере, одна — раньше и одна — позже.

Для моделей интерпретируемой логики предикатов характерно отсутствие фундаментального индивида, представляющего принципиальный интерес; интерес представляют различные множества индивидов. Следовательно:

(События = {x \ Не x = 0 & x В Реализациях}).

Таким образом, события — это непустые подмножества множества реализации.

Раньше и Позже — это соответствующие отношения на реализациях. Перед и После — это соответствующие отношения на событиях.

(Перед = {xy \ x Принадлежит События
& y Принадлежит События
& Для всех x' Для всех y'
(Если x' Принадлежит x
& y' Принадлежит y
То x' Раньше чем y')}).

(После = Cnv Перед).

Таким образом, одно событие происходит перед другим, если каждая реализация в первом осуществляется раньше, чем каждая реализация в другом.

Каждое множество реализаций, не содержащее членов более ранних, чем другие, есть момент. Оно может быть получено из функции, определенной на реализациях и ставящей в соответствие каждой реализации момент, которому она принадлежит:

(Момент = Fx Между Реализациями {y \ y
Принадлежит Реализациям & Не
([x, y] Принадлежит Перед U После)}).

(Моменты = Dl Момент).

При определении понятия «Момент» был введен новый технический прием. Выражение

(Fx Между Y Z = {yx \ x Принадлежит Y & y = Z)} является примером определения, введенного не для того, чтобы описать новые понятия, а лишь для более компактной записи выражения.

Два события одновременны, если моменты, порождаемые их реализациями, совпадают. Таким образом:

(Одновременно = {xy Между События \ Момент: x = Момент: y}).

С помощью нескольких примитивных предикатов, не описанных здесь, которые устанавливают определенный момент как исходный, другой момент как отстоящий от первого на единицу времени (например, секунду) и момент, определяющий полпути между любыми двумя другими моментами, можно ввести метрику для времени. Тогда каждому моменту времени можно поставить в соответствие определенное число — число секунд, на которое момент отстоит от исходного. Каждому числу в таком случае можно поставить в соответствие событие, являющееся объединением всех моментов, которым соответствует это число. Класс секунды становится таким образом классом событий. Аналогично можно ввести Дни, Месяцы, Годы и в необходимых случаях, такие классы, как Первый день месяца, Январь и т. д.

УДК 681.324:006.354 МКС 35.100.01 35.240 П85 ОКСТУ 4002

Ключевые слова: информация, данные, схема данных, база данных, схема базы данных, информационная база, информационная система, концептуальная схема, информационный процессор

Редактор *В. П. Огурцов*
Технический редактор *В. Н. Прусакова*
Корректор *Н. И. Гаврищук*
Компьютерная верстка *Т. Ф. Кузнецовой*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 13.06.2001. Подписано в печать 21.09.2001. Усл. печ. л. 5,11. Уч.-изд. л. 5,40. Тираж 500 экз. С 2077. Зак. 1447

ИПК Издательство стандартов, 107076, Москва, Колодезный пер., 14.
http://www.standards.ru e-mail. info@standards.ru
Набрано в Калужской типографии стандартов на ПЭВМ.
Калужская типография стандартов, 248021, Калуга, ул. Московская, 256.
ПЛР № 040138

gost.e-ecolog.ru