

O. I. Davydov, A. K. Platonov, Database for semantic model of the operating environment of mobile service robot, *Keldysh Institute preprints*, 2017, 007

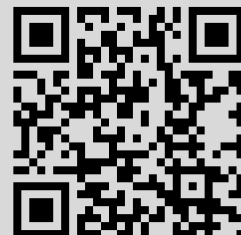
DOI: 10.20948/prepr-2017-7

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use  
<http://www.mathnet.ru/eng/agreement>

Download details:

IP: 79.139.215.77

December 17, 2023, 00:47:21





ISSN 2071-2898 (Print)  
ISSN 2071-2901 (Online)

Давыдов О.И., Платонов А.К.

База данных для  
семантической модели  
операционной среды  
мобильного сервисного  
робота

**Рекомендуемая форма библиографической ссылки:** Давыдов О.И., Платонов А.К. База данных для семантической модели операционной среды мобильного сервисного робота // Препринты ИПМ им. М.В.Келдыша. 2017. № 7. 24 с. doi:[10.20948/prepr-2017-7](https://doi.org/10.20948/prepr-2017-7)  
URL: <http://library.keldysh.ru/preprint.asp?id=2017-7>

**О р д е н а   Л е н и н а**  
**ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ**  
**имени М.В.Келдыша**  
**Р о с с и й с к о й   а к а д е м и и   н а у к**

**О.И. Давыдов, А.К. Платонов**

**База данных для семантической модели  
операционной среды  
мобильного сервисного робота**

**Москва — 2017**

**Давыдов О.И., Платонов А.К.**

**База данных для семантической модели операционной среды  
мобильного сервисного робота**

В работе описывается база данных для семантической модели операционной среды сервисного мобильного робота РБ-2. В соответствии с архитектурой семантической модели операционной среды в разделах базы данных содержится информация обо всех элементах модели, включая Термы, Предикаты, Высказывания, Правила и Сценарии. В данной работе приводятся форматы элементов семантической модели, приводятся их примеры, описывается структура базы данных, в том числе состав таблиц, отдельных полей в таблицах, связей между таблицами.

**Ключевые слова:** мобильный робот, семантическая модель, операционная среда, база данных, таблица базы данных, поля таблицы.

**Oleg Izmailovich Davydov, Alexandr Konstantinovich Platonov**

**Database for semantic model of the operating environment of mobile  
service robot**

The paper describes the database for the semantic model of the operating environment of the service mobile robot RB-2. In accordance with the architecture of the semantic model of the operating environment the database partitions contain information about model elements, including Terms, Predicates, Propositions, Rules, and Scripts. This paper represents formats of the elements of the semantic model, provides examples, describes the database structure, including composition of tables, individual fields in the tables and relationships between tables.

**Keywords:** mobile robot, semantic model, operating environment, database, database table, table fields.

Работа выполнена при финансовой поддержке РФФИ (грант № 15-08-06431) и РНФ (грант № 16-19-10705).

**Оглавление**

Введение .....	3
1. База данных в системе управления робота .....	4
2. Термы .....	6
3. Предикаты.....	8
4. Высказывания.....	11
5. Правила .....	13
5.1. Правило типа Логика.....	15
5.2. Правило типа Действие.....	16
5.3. Правило типа Событие.....	17
6. План, Сценарий .....	18
7. Заключение .....	23
Список литературы .....	24

## Введение

Проблемой создания семантических моделей операционной среды робота занимаются научные и исследовательские центры во всем мире. В этом направлении ведется разработка специализированных языков для описания среды, алгоритмов построения семантических моделей на основании данных сенсоров, создаются специализированные базы семантической информации, [1], [2], [3]. Однако в настоящее время, несмотря на значительные усилия, предпринимаемые мировыми производителями, на рынке практически нет роботов, которые использовали бы семантическую информацию для построения целенаправленного поведения.

Предлагаемая нами работа проводится в рамках проекта, цель которого – создание недорогой, но эффективной системы управления робота с элементами искусственного интеллекта. В рамках этого проекта разрабатывается семантическая модель операционной среды, которая предназначена для управления поведением сервисного мобильного робота РБ-2 в помещениях, в динамически меняющейся среде обитания человека, где робот должен полуавтономно выполнять различные команды человека и перемещаться внутри этих помещений.

Робот РБ-2 представляет собой подвижную платформу, на которой установлен манипулятор с пятью степенями свободы. Подвижная платформа робота оборудована дифференциальным приводом с двумя передними ведущими колесами и двумя задними поддерживающими омни-колесами. Робот оборудован контактными датчиками препятствий, инфракрасными и ультразвуковыми датчиками малых расстояний. На корпусе робота установлен лазерный сканер, который является основным источником данных для управления движением робота. Программное обеспечение системы управления установлено на бортовом компьютере с двухъядерным процессором Intel, тактовой частотой 1.8 ГГц, работающим под операционной системой Windows 7 Embedded.

Робот РБ-2 должен управляться в командно-супервизорном режиме, а в ряде случаев действовать автономно. При таком управлении человек-оператор робота формулирует текущее задание, которое робот выполняет в автономном режиме. Человек контролирует действия робота и корректирует задание. Такой способ управления должен иметь большую степень независимости от человека, обладать хорошими способностями «понимания» окружающей обстановки и иметь возможность естественного взаимодействия с людьми. Только иногда, когда это просто необходимо, оператор берет на себя прямое управление механизмами робота.

Для того чтобы иметь возможность управлять поведением робота, его система управления должна обладать адекватной моделью операционной среды [4]. Система управления робота РБ-2 состоит из двух уровней.

Верхний уровень, формирующий поведение робота, использует семантическую модель операционной среды. Он генерирует последовательность инструкций, которые передаются на исполнение нижнему уровню системы [5]. Нижний уровень, используя метрико-топологическую модель среды, на основе этих инструкций формирует последовательность конкретных команд для исполнительных механизмов робота [6].

Семантическая модель, используемая верхним уровнем системы, описывает среду в виде набора предикатных выражений, в виде логических отношений между явлениями, объектами и свойствами в этой операционной среде. В сервисном роботе такая модель важна как для обеспечения диалога с человеком-оператором робота, так и для выполнения его заданий. Для общения с человеком робот должен иметь описание семантики предметной области, уметь интерпретировать словарную и грамматическую структуру задания, которое формулируются оператором. Для выполнения этого задания робот должен обладать возможностью формировать последовательность целенаправленных действий, которые с большой долей вероятности приведут его к цели.

В данной работе описывается только та часть семантической модели, которая позволяет роботу планировать действия и прогнозировать развития ситуации, а также исполнять построенные планы. Модель создана пока только для решения задач управления движением подвижной платформы робота. Модель для управления манипулятором робота находится в стадии разработки. Задачи общения робота с человеком, задачи обработки сенсорной информации, а также задачи занесения в базу данных модели операционной среды в данной работе не рассматриваются.

## 1. Базы данных в системе управления робота

Архитектура системы управления робота РБ-2 соответствует архитектуре, приведенной в работе [7]. Она воспроизведена на рисунке 1. Из всех элементов этой архитектуры в управлении движением робота принимают участие **Монитор**, **Планировщик** и **Агент навигации и построения движения платформы робота**. Семантическая модель операционной среды хранится в **Базе данных**.

Целенаправленное поведение – это способность робота планировать и выполнять последовательность действий для достижения поставленной цели, учитывая влияние всех внутренних и внешних факторов. Целенаправленное поведение робота реализуется системой управления в два этапа. На первом этапе осуществляется прогнозирование ситуации и формирование плана поведения робота. Эти задачи решаются Планировщиком. На втором этапе сформированный план выполняется Агентом навигации и построения

движения платформы робота. Монитор контролирует весь этот процесс и управляет им.

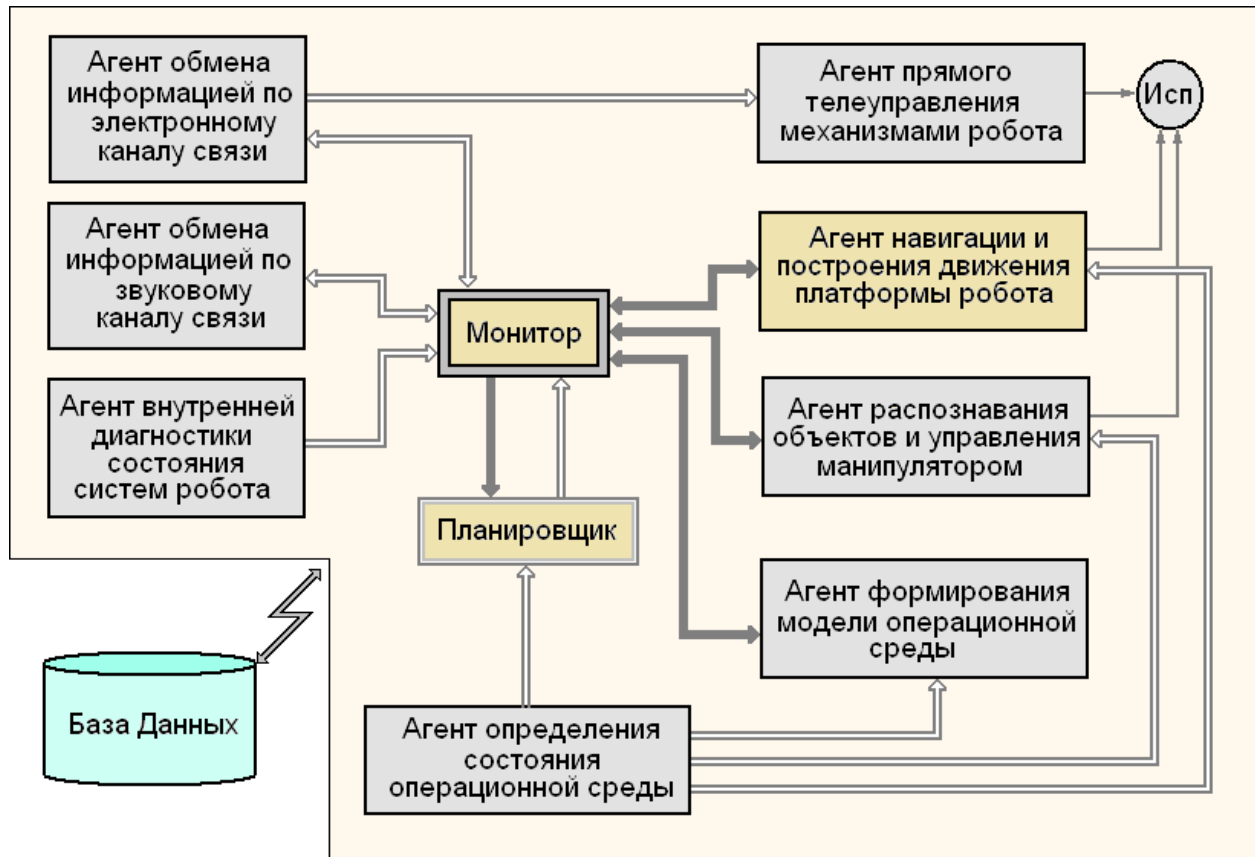


Рис. 1. Архитектура системы управления робота РБ-2

Монитор и Планировщик представляют собой верхний уровень системы управления, а Агент навигации и построения движения платформы – нижний. Семантическая модель среды используется Монитором и Планировщиком, которые формируют последовательность специальных инструкций для Агента навигации и построения движения платформы. На основании этих инструкций Агент управляет движением робота, контролирует состояние робота в операционной среде и передает на верхний уровень информацию об успешности выполнения инструкций.

Семантическая модель операционной среды хранится в базе данных системы управления робота РБ-2. В данной публикации описывается структура базы данных для семантической модели операционной среды робота РБ-2, которая основана на следующих элементах: **Термы, Предикаты, Высказывания, Правила и Сценарии**. Для описания модели используется версия языка STRIPS [8]. Обработки семантических выражений, логический вывод и планирование осуществляется Планировщиком, работа которого основана на классическом алгоритме Графплан [9].

Таблицы базы данных связаны между собой полями, названия которых содержат сочетание букв "ID". На рисунках связи между таблицами

обозначены стрелками между связанными полями. Направление стрелок обозначает объединение всех записей из таблицы, откуда стрелка исходит, и только тех записей из второй таблицы, в которых связанные поля совпадают.

Почти все таблицы имеют три поля – Deleted, RecDate и RecUser, которые используются как вспомогательные поля, служащие для документирования истории изменений, сделанных в базе данных.

Для отладки совместной работы базы данных с системой управления создана семантическая модель некой городской квартиры, которая является тестовой средой для робота РБ-2. Эта тестовая модель постоянно видоизменяется и пополняется. В отличие от структуры базы данных и концепции самой семантической модели число записей в таблицах базы данных постоянно меняется.

## 2. Термы

**Термы** являются основополагающими элементами семантической модели операционной среды РБ-2. К Термам относятся любые понятия, используемые в системе управления робота, в том числе объекты, события, свойства. Каждый Терм определяется своим названием и объемом. Название – это слово или словосочетание, уникальное для каждого Терма, объем – это некоторый набор свойств, который связывает данный Терм с другими Термами из базы данных, то есть с другими объектами, событиями или свойствами.

Примерами Термов, используемых роботом РБ-2, являются:

**Комната, Стол, Стол-Офисный, Шкаф-Платяной, Дверь, Стул.**

В рамках проекта РБ-2 все слова в названиях Термов, которые состоят из словосочетаний, связываются между собой дефисом(-).

Структура информации, связанной с понятием Терма в базе данных робота, приведена на рисунке 2.

Корневая таблица – Terms – содержит основные данные о Термах:

ID – идентификатор Терма;

Name – имя Терма;

Desc – словесное описание Терма объемом до 255 символов.

С корневой таблицей связана таблица TermTypes, в которой содержится классификация Термов. Здесь:

TermTypeID – идентификатор Терма;

TermID – идентификатор дочернего Терма.



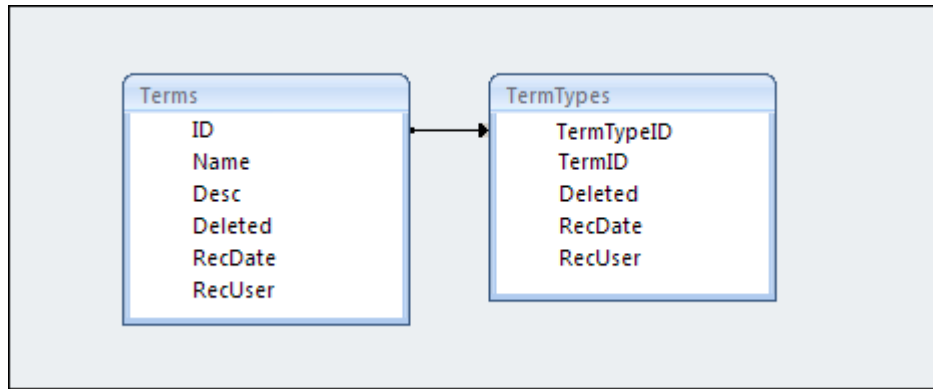


Рис. 2. Структура информации, формирующая понятие Терм

Классификация определяет структуру Термов в модели операционной среды. В качестве примера рассмотрим фрагмент классификации модели операционной среды робота РБ-2:

### **Шкаф-Платяной это Шкаф** **Шкаф это Мебель**

Здесь **Шкаф-Платяной** является подклассом, т.е. дочерним классом, а **Шкаф** для **Шкаф-Платяной** является суперклассом, т.е. родительским. В свою очередь **Шкаф** является подклассом для **Мебель**, следовательно, и **Шкаф-Платяной** является подклассом для **Мебель**. Для этой классификации в таблице TermTypes сформированы следующие три записи:

TermTypeID	TermID
<b>Мебель</b>	<b>Шкаф</b>
<b>Мебель</b>	<b>Шкаф-Платяной</b>
<b>Шкаф</b>	<b>Шкаф-Платяной</b>

В структуре модели среды робота РБ-2 заложена концепция сетевой классификации Термов, что позволяет определять для каждого Терма несколько суперклассов.

Структура описываемой модели среды предполагает наличие Термов-Констант, которые не являются суперклассами для других Термов. Термы-Константы используются для описания индивидуальных объектов или понятий, присутствующих в операционных средах сервисного робота, например: **Кресло-кабинетное**. Образом этого Терма является конкретное кресло в кабинете тестовой квартиры, которая используется как среда для отладки системы управления РБ-2. В таблице TermTypes Термы-Константы в поле TermTypeID никогда не появляются.

В отличие от Термов-Констант, Термы-Типы являются суперклассами для других Термов. Они обобщают в одном понятии группу однотипных объектов, свойств или явлений. Например: **Кресло-офисное** – это Терм-Тип,

к которому относится **Кресло-кабинетное**. Термы-Типы могут заноситься в таблицу TermTypes и в поле TermTypeID, и в поле TermID.

### 3. Предикаты

Термы не изолированы друг от друга. Они находятся между собой во множестве связей, отношений. Эти связи представлены в виде **Предикатов**, т.е. суждений, которые что-либо утверждают или отрицают об отношениях между Термами, являющихся параметрами Предиката. Каждый Предикат определяется своим названием – словосочетанием, обозначающим данный Предикат, и набором параметров – Термов, между которыми определяется связь. Универсальный формат всех Предикатов в базе данных робота:

#### **НАЗВАНИЕ(Терм1, Терм2,... ТермN)**

В рамках проекта РБ-2 названия Предикатов записываются прописными (заглавными) буквами. Все слова в названиях Предикатов, состоящих из словосочетаний, связываются между собой дефисом(-) или нижним подчеркиванием(\_) , которое используется только там, где по синтаксису русского языка должны стоять названия Термов.

В качестве примера, рассмотрим Предикат, используемый роботом РБ-2:

#### **НАХОДИТСЯ-В(Стол-письменный, Кабинет)**

Данный Предикат состоит из названия **НАХОДИТСЯ-В** и двух Термов – **Стол-письменный, Кабинет**.

Предикаты определяют отношения только между Термами-Типами и описывают только абстрактные операционные среды. Связи между Термами-Константами и/или Термами-Типами, описывающими реальные объекты, события, свойства операционной среды робота, определяются **Высказываниями**, которые строятся на основе Предикатов. Подробнее Высказывания будут обсуждаться в разделе 4.

Отметим, что совокупность всех Предикатов и Высказываний, в которых в качестве параметра участвует один и тот же Терм, определяет объем этого Терма. Для каждого Терма объем в базе данных робота должен быть уникальным.

На рисунке 3 приведена структура информации в базе данных робота, связанной с понятием Предиката. Корневая таблица Predicates:

ID – идентификатор Предиката;

Name – имя Предиката;

Desc – словесное описание Предиката объемом до 255 символов;

TypeID – идентификатор типа Предиката;

TermCount – количество Термов в Предикате.

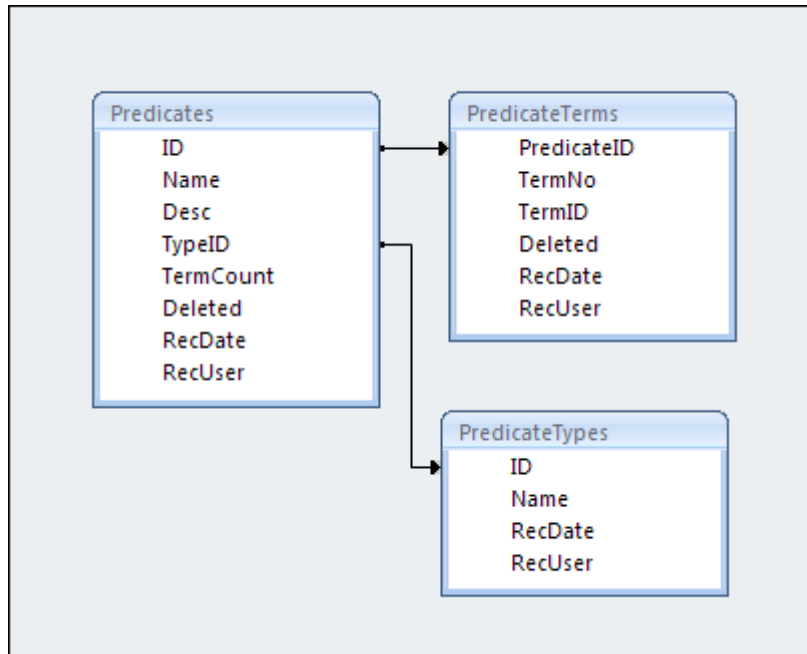


Рис. 3. Структура информации, формирующая понятие Предикат

С корневой таблицей связана таблица PredicateTerms, которая содержит списки Термов, относящихся к каждому Предикату. Здесь:

PredicateID – идентификатор Предиката;  
 TermNo – порядковый номер Терма в Предикате;  
 TermID – идентификатор Терма.

С корневой таблицей связана также сервисная таблица PredicateTypes, в которой приведен список типов Предикатов, идентификаторы типов и их наименования. В этой таблице в настоящее время содержатся пять типов Предикатов: **Свойство, Классификация, Состояние, Позиция, Принадлежность (Часть-целое)**.

Предикаты типа Свойство определяют отличительные признаки Термов. Все они имеют один из нескольких унифицированных форматов, например:

**\_ИМЕЕТ\_=\_ (Терм, Признак, Значение)**

Это означает, что **Терм ИМЕЕТ Признак =Значение**. Например:

**\_ИМЕЕТ\_=\_ (Стол, Цвет, Черный)**

Предусмотрены также и другие форматы Предикатов типа Свойство, например:

**\_ИМЕЕТ\_>\_ (Терм, Признак, Значение)**

**\_РАВЕН\_ (Терм, Значение)**

Отметим, что **Признак** и **Значение** в формате Предикатов типа Свойство – это Термы, причем **Признак** – это всегда Терм-Тип, а **Значение** – это или подкласс, или Терм-Константа, относящийся к типу **Признак**.

Для целостного описания каждого Терма необходимо иметь полный набор его свойств. Однако хранение в базе данных всех возможных свойств не реализуемо, поскольку их количество и сложность могут быть неисчерпаемыми. На практике в базе данных робота РБ-2 содержатся лишь те Предикаты типа Свойство, которые используются в задачах, решаемых роботом.

Большинство предикатов типа Классификация имеют следующий формат:

### **ЭТО (Терм-А, Терм-В)**

Суть данного предиката состоит в том, что **Терм-А** для **Терм-В** является подклассом, т.е. дочерним классом, а **Терм-В** для **Терм-А** является суперклассом, т.е. родительским.

Таблица TermTypes, описанная в предыдущем разделе, формируется автоматически в процессе занесения/удаления записей типа Классификация в таблице Predicates.

Предикаты типа Состояние определяют состояние Термов, которые по своей природе могут иметь несколько состояний. Большинство предикатов данного типа имеют следующее название и формат:

### **НАХОДИТСЯ-В-СОСТОЯНИИ (Терм-А, Терм-В)**

В качестве примера приведем Предикаты:

**НАХОДИТСЯ-В-СОСТОЯНИИ (Дверь-Х, Открыта)**  
**НАХОДИТСЯ-В-СОСТОЯНИИ (Дверь-Х, Закрыта)**

Предикаты типа Позиция определяют положение в пространстве, чаще всего относительно других объектов среды:

**НАХОДИТСЯ-ОКОЛО (Кресло-офисное, Стол-письменный)**  
**НАХОДИТСЯ-В (Стол-письменный, Кабинет)**

Пример Предиката типа Принадлежность:

**ЯВЛЯЕТСЯ-ЧАСТЬЮ (Спинка-стула, Стул)**

Как правило, Предикаты такого типа определяют состав сложных объектов событий или свойств.

## 4. Высказывания

Как было отмечено в предыдущем разделе, связи между конкретными объектами реальной среды робота описываются в виде **Высказываний**. Высказывания определяют отношения между любой комбинацией Термов-Типов и Термов-Констант. По сути Предикаты являются шаблонами для Высказываний, поэтому по формату они идентичны друг другу. Однако, в отличие от Предикатов, каждому Высказыванию приписывается целое число в диапазоне от 0 до 100, называемое достоверностью, которое отражает степень доверия к его истинности.

Высказывания формально можно разделить на Факты и Предположения, которые отличаются друг от друга по степени их достоверности. Фактом является отношение между Термами, истинность которого никогда не подвергается сомнению, т.е. достоверность равна 100. Достоверность неложных Предположений всегда меньше 100, но больше 0. Если достоверность равна 0, то Предположение является ложным.

Робот, функционирующий в операционной среде, должен иметь стационарную модель этой среды, которая описывает ее постоянную, неменяющуюся часть. Например, двигаясь в помещении, робот должен обладать информацией о числе комнат, о наличии дверей, о мебели, которая постоянно и стационарно находится в помещении и т.д. Эти данные в виде набора Фактов и являются стационарной моделью среды. Например:

НАХОДИТСЯ-В\_(Стол-письменный, Кабинет)  
ИМЕЕТ =\_(Кабинет, Длина, 4-метра)

Если Фактом является априори достоверное отношение между Термами, то достоверность Предположения не очевидна, и в каждом конкретном случае требует оценки. Предположениями описывается непостоянная, меняющаяся часть операционной среды робота, например:

НАХОДИТСЯ-ОКОЛО\_(Кресло-офисное, Стол-письменный)

Априорных данных об истинности или ложности данного Факта нет и быть не может. Можно только оценить степень достоверности того, что данный факт является истинным. В каждом конкретном случае это зависит от некоторого набора внешних условий, причем одно и то же Высказывание в зависимости от условий может быть и Фактом, и Предположением.

В базе данных семантической модели среды робота РБ-2 все Высказывания хранятся в таблице Props, которая представлена на рисунке 4. Таблица Props по составу полей почти полностью повторяет таблицу PredicateTerms, приведенную на рисунке 3.

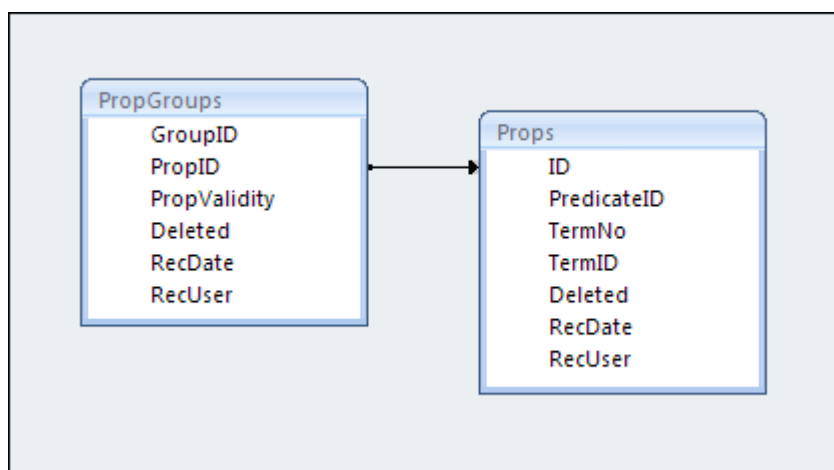


Рис. 4. Структура информации, формирующая понятие Высказывание

В таблице Props:

ID – идентификатор Высказывания;

PredicateID – идентификатор Предиката-шаблона;

TermNo – порядковый номер Терма в Высказывании;

TermID – идентификатор Терма.

Отметим, что в таблице Props в поле TermID заносится или идентификатор соответствующего термина из таблицы PredicateTerms, или идентификатор его любого дочернего термина.

Дополнительно для удобства агрегирования элементов введено понятие **Группы**. Обычно в Группы объединяются Высказывания, которые относятся к одному и тому же набору внешних условий. Одно и то же Высказывание может быть членом нескольких Групп. Высказывания объединяются в группы вручную, по неформальным признакам, позволяющим облегчить разработку и сопровождение базы данных.

Таблица PropGroups, связанная с таблицей Props, содержит идентификаторы Групп и списки Высказываний, относящихся к этим Группам. Поле PropValidity в таблице PropGroups определяет достоверность каждого Высказывания в зависимости от Группы. Если значение этого поля равно 100, то данное Высказывание является абсолютно достоверным Фактом. Если значение этого поля равно 0, то данное Высказывание ложно. Все остальные Высказывания, у которых значения находятся в интервале от 0 до 100, являются Предположениями, имеющими соответствующую достоверность в данной Группе.

Отметим, что записи Высказывания с одним и тем же PropID могут присутствовать в таблице в нескольких экземплярах, но они должны относиться к разным Группам. Это означает, что в каждой группе Высказывание имеет единственное значение достоверности в этой Группе.

## 5. Правила

**Правила** также являются элементами семантической модели среды робота РБ-2. Формат любого Правила представляется в следующем виде:

**Тип:**

**Название**

**(Параметры:**

**Список-параметров)**

**(Предусловие:**

**Список-Условий)**

**(Эффект+:**

**Список-Последствий+)**

**(Эффект-:**

**Список-Последствий-)**

**Параметры** Правила – это список Термов-Типов. В процессе применения Правила параметры могут быть заменены на любые дочерние Термы или на Термы-Константы, относящиеся к данному типу. Все параметры имеют имена-идентификаторы, у которых стоит в качестве первого символа стоит вопросительный знак. Если производится замена параметра на дочерний Терм или на Терм-Константу, то эта замена должна быть выполнена идентично для всех Термов с одинаковыми именами-идентификаторами, представленными в Правиле. Эту замену будем называть **означиванием**.

Означивание выполняется Планировщиком для того, чтобы сформировать в разделах **Предусловие**, **Эффект+** и **Эффект-** Высказывания, описывающие операционную среду робота и обстоятельства, в которых используется данное Правило.

**Предусловие** Правила – это список Предикатов, которые после их означивания становятся Высказываниями и которые должны быть истинными, чтобы Правило было применимо.

**Эффект+** содержит список Предикатов, которые после их означивания становятся Высказываниями и про которые заранее известно, что они будут истинными в результате применения Правила. Соответственно, в подразделе **Эффект-** содержится список Предикатов, которые после означивания становятся Высказываниями и про которые заранее известно, что они станут ложными в результате применения Правила.

Предположения в разделе **Предусловие** отличаются по степени своей достоверности. На стадии планирования и/или прогнозирования Правило считается применимым, если степень достоверности всех Высказываний в разделе **Предусловие** в данных конкретных обстоятельствах не ниже, чем некоторое пороговое значение. Пороговая величина задается вручную, на основании опыта использования данного Правила. Достоверность всех



Высказываний в разделе **Эффект+** и **Эффект-** после применения Правила устанавливается равной минимальному значению достоверности всех Высказываний в разделе **Предусловие**.

На рисунке 5 приведена структура таблиц в базе данных робота, связанных с понятием Правила.

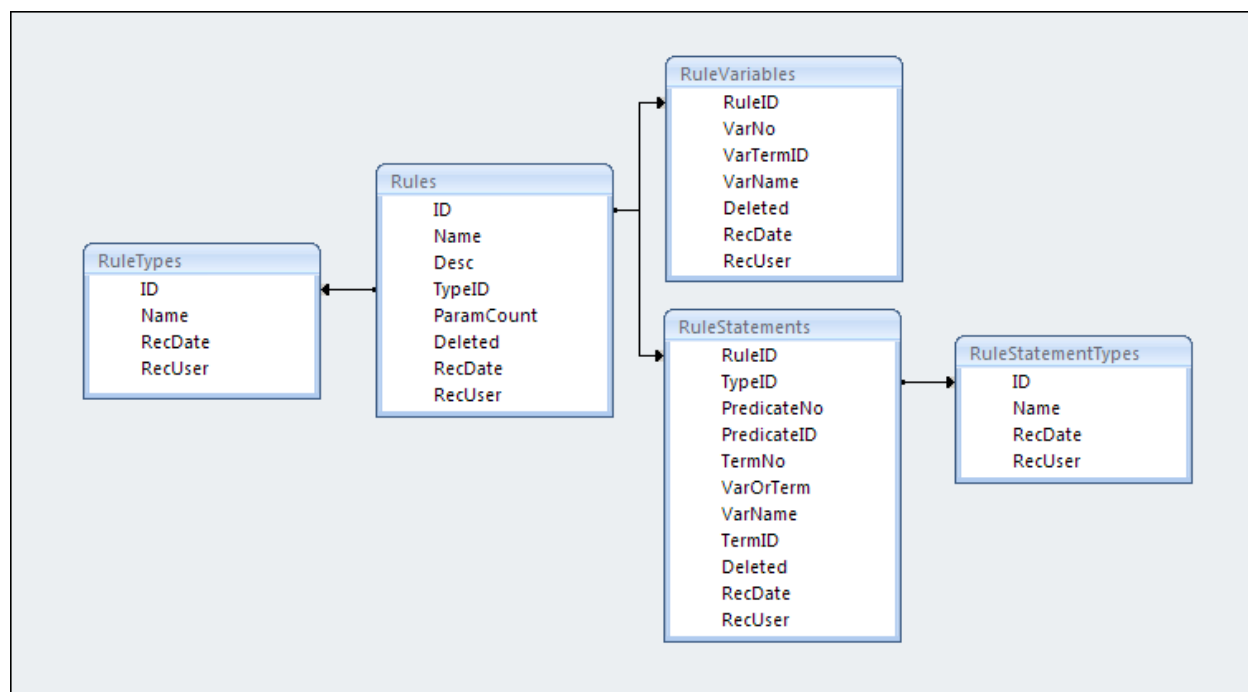


Рис. 5. Структура информации, формирующая понятие Правило

В этой структуре корневой таблицей является Rules, в которой:

- ID – идентификатор Правила;
- Name – имя Правила;
- Desc – словесное описание Правила;
- TypeID – идентификатор типа Правила;
- ParamCount – количество параметров в Правиле.

С корневой таблицей Rules связаны две таблицы, определяющие структуру каждого Правила. Таблица RuleVariables содержит список параметров Правила. Здесь:

- RuleID – идентификатор Правила;
- VarNo – порядковый номер параметра в Правиле;
- VarTermID – идентификатор Терма-Типа параметра в Правиле;
- VarName – имя-идентификатор параметра в Правиле.

В таблице RuleStatements содержится список всех разделов Правила, Условий и Последствий применения данного Правила. Здесь:

- RuleID – идентификатор Правила;
- TypeID – идентификатор раздела Правила;



PredicateNo – порядковый номер Предиката в Правиле;  
 PredicateID – идентификатор Предиката;  
 TermNo – порядковый номер Терма в Предикате;  
 VarOrTerm – признак, допускает ли Терм в Предикате замену;  
 VarName – имя-идентификатор параметра в Правиле;  
 TermID – идентификатор Терма в Предикате.

С таблицей RuleStatements связана сервисная таблица RuleStatementTypes, в которой приведен список разделов Правила, их идентификаторы разделов и их наименования. В этой таблице содержатся только три типа разделов: Предусловие, Эффект+ и Эффект-.

С корневой таблицей Rules связана сервисная таблица RuleTypes, в которой приведен список типов Правил, идентификаторы типов и их наименования. В этой таблице содержатся только три типа Правил, которые используются системой управления робота РБ-2 в настоящее время: **Логика**, **Действие** и **Событие**.

### 5.1. Правило типа Логика

Правило типа Логика – это Правило логического вывода, которое позволяет формировать новые Высказывания, основываясь на исходном наборе Высказываний. Правила типа Логика в системе управления робота РБ-2 используются для пополнения Высказываний в модели операционной среды робота. В качестве примера рассмотрим Правило типа Логика, которое реально используется в семантической модели операционной среды робота РБ-2:

**Логика:**

**Наследование-типа-терма**

**(Параметры:**

**Терм ?x, Терм ?y, Терм ?z)**

**(Предусловие:**

**\_ЭТО\_ (Терм ?x, Терм ?y)**

**\_ЭТО\_ (Терм ?y, Терм ?z))**

**(Эффект+:**

**\_ЭТО\_ (Терм ?x, Терм ?z))**

Это Правило определяет тот факт, что если **?x** имеет тип **?y**, а **?y** имеет тип **?z**, то **?x** относится также к типу **?z**. Например, рассмотрим Факты:

**\_ЭТО\_ (Шкаф-Платяной, Шкаф)**

**\_ЭТО\_ (Шкаф, Мебель)**

На основании представленного выше Правила типа Логика можно сделать заключение:

**\_ЭТО\_ (Шкаф-Платяной, Мебель)**

По определению все дочерние Термы, в том числе и Термы-Константы, обладают свойствами своих Термов-Типов. В семантической модели операционной среды робота РБ-2 это сформулировано в виде следующего Правила:

**Логика:**

**Наследование-свойств**

**(Параметры:**

**Терм ?x, Терм ?t, Терм ?s, Терм ?z)**

**(Предусловие:**

**\_ЭТО\_ (Терм ?x, Терм ?t)**

**\_ЭТО\_ (Терм ?z, Терм ?s)**

**\_ИМЕЕТ=\_ (Терм ?t, Терм ?s, Терм ?z))**

**(Эффект+:**

**\_ИМЕЕТ=\_ (Терм ?x, Терм ?s, Терм ?z))**

В соответствии с этим правилом можно утверждать, что **Шкаф-Платяной** обладает свойствами Терма-Типа **Шкаф** и Терма-Типа **Мебель**.

## 5.2. Правило типа Действие

Правило типа Действие определяет действие робота, которое может быть им выполнено при наличии определенных исходных условий, и указывает на последствия, возникающие после выполнения этого действия. Правила типа Действие используются при планировании и прогнозировании поведения робота. Конкретное действие определяется названием соответствующего Правила. Система этих Правил сформирована таким образом, что действия, заложенные в их названии, могут быть реально исполнены роботом, поэтому можно утверждать, что эти Правила являются также моделью функциональных возможностей самого робота.

В качестве примера рассмотрим Правило типа Действие, которое используется в семантической модели операционной среды робота РБ-2:

**Действие:**

**Движение-в\_от-двери\_до-позиции\_объекта\_**

**(Параметры:**

**Помещение ?x, Дверь ?y, Позиция ?z, Объект ?r)**

**(Предусловие:**

**\_НАХОДИТСЯ-В\_ (Объект ?r, Помещение ?x)**

**\_НАХОДИТСЯ-У\_ (Объект ?r, Дверь ?y)**

**\_НАХОДИТСЯ-В\_ (Позиция ?z, Помещение ?x)**

**\_НАХОДИТСЯ-В\_ (Дверь ?y, Помещение ?x))**

**(Эффект+:**

**\_НАХОДИТСЯ-НА\_ (Объект ?r, Позиция ?z))**

**(Эффект-:**

**\_НАХОДИТСЯ-У\_ (Объект ?r, Дверь ?y)**

Это правило определяет условия, при выполнении которых объект может переместиться от двери до некоторой целевой позиции в помещении. Если в ходе движения робота это правило будет применено и действие будет исполнено роботом, то в результате робот окажется не у двери помещения, а в целевой позиции.

Разновидностью Правила типа Действие являются Правила, у которых отсутствует список Высказываний в разделах **Эффект+** и **Эффект-**. Такие Правила определяют действия робота, которое должны быть выполнены при наличии некоторых исходных условий без учета последствий, возникающих после выполнения таких действий. Правила такого типа используются в ситуациях, когда планировать действия робота не имеет смысла, когда заранее известно как должен действовать робот независимо от последствий этих действий. Эти Правила моделирует рефлекторную реакцию робота и используются системой робота РБ-2 только для экстремальных ситуаций, например:

**Действие:**

**Остановка-из-за-близости-препятствия**

**(Параметры:**

**Препятствие ?х, Мин-Дист)**

**(Предусловие:**

**ПЕРЕД-РБ-НАХОДИТСЯ\_(Препятствие ?х)**

**ДИСТАНЦИЯ-ОТ-РБ-ДО\_<\_(Препятствие ?х, Мин-Дист))**

**(Эффект+: )**

**(Эффект-: )**

Данное Правило используется Планировщиком для формирования условий, при которых робот должен однозначно останавливаться, несмотря на все возможные последствия этой остановки. В этом правиле два параметра: **Препятствие** и **Мин-Дист**. Параметр **Мин-Дист** не допускает замены, так как он является Термом-Константой. В базе данных робота РБ-2 для этого Терма определен Факт типа Свойство:

**\_РАВЕН\_(Мин-Дист, 40)**

Параметр **Препятствие** допускает замену на любой дочерний Терм, в число которых входит все, что находится в помещении и может являться препятствием движению робота, в том числе: **Человек**, **Мебель**, **Стена**.

### 5.3. Правило типа Событие

Правило типа Событие определяет последствия какого-то внешнего события, которое может возникнуть при наличии определенных исходных условий и оказывает влияние и на самого робота и/или на его операционную среду. Эти события не зависят от действий робота и являются или следствием воздействия людей или свойствами самой операционной среды.

Человек обычно знает о происходящих и о возможных событиях, об их истоках и последствиях и использует эти знания в своей повседневной деятельности. Робот должен также обладать моделями внешних событий, чтобы адекватно реагировать на них.

Конкретное событие определяется названием соответствующего Правила. В качестве примера рассмотрим Правило типа Событие, которое используется в семантической модели операционной среды робота РБ-2:

**Событие:**

**Стул\_отодвинут-человеком\_от-стола\_**

**(Параметры:**

**Стул ?x, Человек ?y, Стол ?z)**

**(Предусловие:**

**\_НАХОДИТСЯ-У\_(Стул ?x, Стол ?z)**

**\_НАХОДИТСЯ-У\_(Человек ?y, Стул ?x)**

**(Эффект+: )**

**(Эффект-:**

**\_НАХОДИТСЯ-У\_(Стул ?x, Стол ?z))**

Выполнение события, описанное данным Правилем, не зависит от робота, а зависит от человека, находящегося у стула. Предугадать заранее, состоится это событие или нет, невозможно. Можно только оценить вероятность того, что данное событие произойдет, причем эта оценка должна основываться на статистической информации. Оценка вероятности событий производится вручную в процессе занесения Правил в базу данных семантической модели робота.

Правила типа **Событие** используются при планировании поведения робота. В ходе выполнения построенного плана робот должен проверять, состоялось ли предполагаемое событие или нет, и в зависимости от результата этой проверки следовать по одной из альтернативных ветвей плана.

## 6. План, Сценарий

Используя Термы, Факты, Предположения и Правила, Планировщик создает последовательность инструкций, которые должны выполняться всеми Агентами системы управления робота РБ-2. Эта последовательность инструкций называется **Планом Поведения** робота. Из этого плана Монитор выделяет куски и передает их для исполнения Агенту навигации и построения движения платформы робота. Каждый из таких кусков является **Планом Движения** робота. Так как наша семантическая модель создается пока только для решения задач управления движением платформы робота, мы в дальнейшем под термином **План** будем понимать только План Движения.

Формат Плана имеет следующий вид:

**План:**

**Идентификатор-Плана**

**(Предусловие:**

**Список-Фактов)**

**(Список-Шагов)**

Под названием **Список-Шагов** подразумевается перенумерованная последовательность Шагов Плана, каждый из которых имеет следующий формат:

**(Номер-Шага:**

**Идентификатора-Действия**

**Список-Фактов-Шага)**

В свою очередь **Идентификатор-Действия** имеет формат:

**Название-действия(Терм1, Терм2,... ТермN)**

Здесь **Название-действия** совпадает с Названием Правила типа Действие, которое выбрано Планировщиком для данного шага.

Все Факты в Списках Фактов и в разделе **Предусловие** и в составе каждого Шага проверяются на истинность Агентом навигации и построения движения платформы перед переходом к выполнению следующего Шага. Последний Список Фактов проверяется после выполнения действия, предусмотренного последним Шагом Плана. Если хотя бы один Факт из этих Списков оказывается ложным, то такая ситуация определяется как не предусмотренная Планом. При возникновении непредусмотренной ситуации Агент навигации и построения движения платформы останавливает движения робота, прерывает выполнения Плана и передает Монитору информации о том, какой именно Факт оказался ложным. Если в процессе выполнения Плана все Факты из всех списков оказываются истинными, то выполнение плана считается успешным и соответствующая информация передается Монитору.

В Списках Фактов могут содержаться Факты, истинность которых должна проверяться не только перед, но и в ходе выполнения следующего Шага. Такие Факты названы Контрольными. Они используются для контроля возникновения Событий, в результате которых План должен быть изменен. Контрольные Факты помечаются Планировщиком специальными символами.

План может не содержать **Список-Шагов**. В этом случае в разделе **Предусловие** не должны присутствовать Контрольные Факты. В любом Шаге Плана может отсутствовать **Список-Фактов-Шага**.

Любой Шаг Плана может представлять собой также План. В этом случае вместо **Идентификатора-Действия** Шага ставится **Идентификатор-Плана**.

В качестве примера рассмотрим План, в котором робот, двигаясь в операционной среде, должен выполнить три Шага. На первом Шаге он должен, двигаясь из исходной **Позиция-С**, приблизиться к **Дверь-Х**, ведущей из **Комната-А** в **Комната-В**. На втором Шаге он должен пройти через дверь из **Комната-А** в **Комната-В**. На третьем шаге он должен перейти к выполнению плана движения в **Комната-В** до **Объект-Z**, который, возможно, также состоит из нескольких шагов:

**План:**

**Движение-из-Комната-А-в-Комната-В**

**(Предусловие:**

**РБ-НАХОДИТСЯ-НА\_(Позиция-С)**

**РБ-НАХОДИТСЯ-В\_(Комната-А)**

**\_НАХОДИТСЯ-В\_(Объект-Z, Комната-В)**

**\_НАХОДИТСЯ-В-СОСТОЯНИИ\_(Дверь-Х, Открыта)\*)**

**(1:**

**Движение-РБ-в\_от\_до\_(Комната-А, Позиция-С, Дверь-Х)**

**РБ-НАХОДИТСЯ-У\_(Дверь-Х)**

**РБ-НАХОДИТСЯ-В\_(Комната-А)**

**\_НАХОДИТСЯ-В-СОСТОЯНИИ\_(Дверь-Х, Открыта)\*)**

**(2:**

**Движение-РБ-из\_в\_через\_(Комната-А, Комната-В, Дверь-Х)**

**РБ-2-НАХОДИТСЯ-У\_(Дверь-Х)**

**РБ-2-НАХОДИТСЯ-В\_(Комната-В))**

**(3:**

**План: Движение-в-Комната-В-до-Объект-Z)**

В этом примере второй Шаг можно выполнить, если **Дверь-Х** открыта. Да и первый Шаг имеет смысл, только если **Дверь-Х** открыта. Поэтому одним из исходных условий данного Плана является истинность Факта о том, что **Дверь-Х** открыта. Этот Факт необходимо контролировать постоянно в течение времени выполнения Шагов 1 и 2, то есть он является Контрольным и помечен звездочкой. После того как робот окажется в **Комната-В**, проверять данный Факт на истинность нет смысла.

При построении этого Плана Планировщик использовал следующее Правило типа Событие:

**Событие:**

**Дверь\_закрылась**

**(Параметры:**

**Дверь-Х,)**

**(Предусловие:**

**\_НАХОДИТСЯ-В-СОСТОЯНИИ\_(Дверь-Х, Открыта))**



(Эффект+:

НАХОДИТСЯ-В-СОСТОЯНИИ\_(Дверь-Х, Закрота))

(Эффект-:

НАХОДИТСЯ-В-СОСТОЯНИИ\_(Дверь-Х, Открота))

Вероятность осуществления этого События в любой момент времени выше 0, но ниже 100. Планировщиком был сформирован План, в котором проверка истинности Факта об открытой двери поставлена под контроль в течение всего времени движения робота до двери и через дверь.

Сервисный робот РБ-2 предназначен для выполнения широкого круга заданий. Однако очевидно, что некоторые из этих заданий однотипны и даже одинаковы. Например, робот вынужденно совершает некоторые типовые действия для обеспечения своей деятельности.

Сразу после включения робот должен обязательно протестировать все свои системы. Если что-то не работает или работает неправильно, робот должен сгенерировать соответствующее сообщение и выдать его оператору. Если все системы работают штатно, то далее робот должен определить свое местоположение. Если ему это не удастся, например, он оказался в среде, которую он не может идентифицировать, ему следует сделать соответствующий запрос оператору и перейти в режим ожидания ответа. Если же он определил свое местоположение или ему сообщил эту информацию человек-оператор, то робот должен перейти в режим ожидания задания от оператора.

Описанная последовательность представляет собой План Поведения, который должен выполняться каждый раз, когда включается питание робота. Существует также и типовой План Поведения при выключении робота.

Стандартные План Поведения и План Движения робота РБ-2 называются **Сценарием Поведения** и **Сценарием Движения** соответственно. Сервисный робот, выполняющий функции экскурсовода в музее, должен действовать по достаточно жесткому плану, называемому планом экскурсии. Такой робот регулярно выполняет практически одно и то же задание и может действовать по одному и тому же Сценарию. Робот, помогающий одиноким малоподвижным людям, чаще всего выполняет однотипные задания, например, выключить освещение в удаленной комнате или принести человеку небольшой предмет, и т.д. Можно каждый раз заново строить План Поведения и План Движения робота, но разумнее заранее предусмотреть стандартные Сценарии.

Формат Сценария Движения полностью идентичен формату Плана Движения. Если Планы создаются Планировщиком каждый раз, когда они необходимы и после выполнения забываются, то Сценарии являются частью семантической модели операционной среды робота РБ-2 и хранятся в базе данных. Монитор в соответствии с заданием человека-оператора ищет в базе

данных подходящий Сценарий и, если не находит, то инициирует Планировщик на формирование Плана.

На рисунке 6 приведена группа таблиц базы данных робота РБ-2, связанная с понятием Сценария.

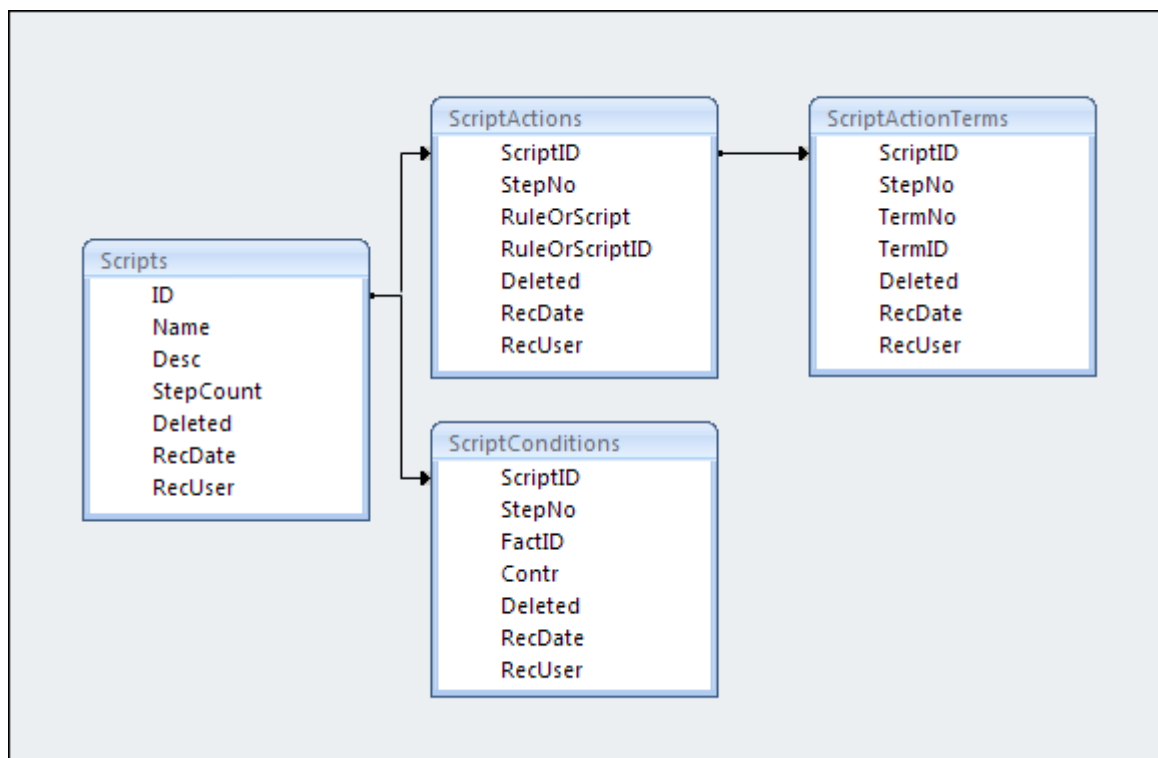


Рис. 6. Структура информации, формирующая понятие Сценария

В этой структуре корневой является таблица Scripts, в которой:

- ID – идентификатор Сценария;
- Name – имя Сценария;
- Desc – словесное описание Сценария;
- StepCount – количество шагов в Сценарии.

С таблицей Scripts связаны две таблицы, определяющие шаги каждого Сценария: ScriptActions и ScriptConditions.

В таблице ScriptActions задается действие, которое должно выполняться на данном шаге:

- ScriptID – идентификатор Сценария;
- StepNo – номер шага Сценария, нумерация начинается с шага 1;
- RuleOrScript – признак, определяющий тип действия на данном шаге, Правило или Сценарий;

RuleOrScriptID – идентификатор Правила или Сценария, которые должно выполняться на данном шаге. С таблицей ScriptActions связана таблица ScriptActionTerms, в которой содержится список Термов для действия, если оно определено Правилем. Если же действие определено



Сценарием, то в таблице ScriptActionTerms записи для такого шага отсутствуют. В таблице ScriptActionTerms:

ScriptID – идентификатор Сценария;  
 StepNo – номер шага Сценария, нумерация начинается с шага 1;  
 TermNo – порядковый номер Терма в параметрах Правила;  
 TermID – идентификатор Терма в параметрах Правила.

Таблица ScriptConditions содержит перечень Фактов, которые следует проверять на истинность по завершении действия, предусмотренного шагом. Здесь:

ScriptID – идентификатор Сценария;  
 StepNo – номер шага Сценария, нумерация начинается с шага 0;  
 FactID – идентификатор проверяемого Факта;  
 Contr – признак, является ли Факт Контрольным.

В этой таблице шаг с номером 0 используется для определения Предусловия выполнения Сценария.

## 8. Заключение

Описанная база данных для семантической модели операционной среды робота РБ-2 находится в завершающей стадии отладки. Разработаны алгоритмы Планировщика и Монитора системы управления робота, непосредственно использующие семантическую модель среды. Для занесения информации в базу данных модели, которое выполняется человеком вручную, создано программное обеспечение, работающее совместно с программами и базой данных метрико-топологической модели среды [4], [5].

Тестирование модели и программ показало, что процесс формирования модели достаточно трудоемкий. Он близок процессу программирования, связан с большим количеством ошибок, вносимых поправок и изменений. Отладка модели с использованием Планировщика в связке с Монитором и Агентом навигации и построения движения платформы робота требует много времени. Так как алгоритм Планировщика является наиболее сложным, для ускорения тестирования модели создано специальное программное обеспечение, в котором полностью используется Планировщик, а Монитор и Агент навигации и построения движения заменены имитаторами.

Текущая стадия формирования и тестирования семантической модели в совокупности с метрико-топологической моделью среды подтверждает возможность эффективного ее использования для решения задач навигации и построения движения робота, обладающего элементами искусственного интеллекта. На этой стадии робот РБ-2 при включении в новом месте квартиры способен определять свое положение, формировать план движения к поставленной цели, используя заложенные в базе данных Сценарии,

выполнять этот план, объезжая препятствия по ходу движения и определяя состояние дверей.

## Список литературы

1. Tenorth M., Kunze L., Jain D., Beetz M. KNOWROB-MAP– Knowledge-Linked Semantic Object Maps. //In 10th IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA, December 6-8 2010, pp. 430–435.
2. Galindo C., Saffiotti A., Coradeschi S., Buschka P., Fernandez-Madrigal J., Gonzalez J. Multi-hierarchical semantic maps for mobile robotics. //In Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2005, pp. 3492–3497.
3. Mozos O., Mizutani H., Kurazume R., Hasegawa T. Categorization of indoor places using the kinect sensor. // Sensors 2012, 12(5), pp. 6695–6711.
4. Давыдов О.И., Платонов А.К. Организация и структура комплексной карты операционной среды сервисного робота. // Препринты ИПМ им. М.В.Келдыша. 2015. № 72. 28 с.
5. Давыдов О.И., Платонов А.К. База данных для модели операционной среды сервисного робота. // Препринты ИПМ им. М.В.Келдыша. 2016. № 132. 20 с.
6. Давыдов О.И., Платонов А.К. Алгоритм управления дифференциальным приводом мобильного робота РБ-2. // Препринты ИПМ им. М.В.Келдыша. 2015. № 25. 17 с.
7. Давыдов О.И., Пряничников В.Е. Архитектура системы управления мобильного сервисного робота. // Информационно-измерительные и управляющие системы, № 7, т.13 2015, стр.41-50.
8. Lifschitz V. On the Semantics of STRIPS. // Reasoning about Actions and Plans. Morgan Kaufmann Publishers Inc., 1987, pp. 1–9.
9. Рассел С., Норвиг П. Искусственный интеллект: Современный подход. // М.: Издательский дом «Вильямс», 2006, с. 541-545.