



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Департамент программной инженерии и искусственного интеллекта

Титовченко Екатерина Дмитриевна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗРАБОТКА РЕДАКТОРА ИЕРАРХИИ И ПРАВИЛ ДЛЯ ВЫБОРА МЕТОДОВ
МОДЕЛИРОВАНИЯ АРХИТЕКТУРНОЙ АКУСТИКИ**

по направлению подготовки 09.04.04 – Программная инженерия,
магистерская программа «Разработка программно-информационных систем»

Владивосток
2024

Аннотация

В данной работе рассматриваются проблемы выбора методов моделирования архитектурной акустики. В первую очередь обозначена проблема отсутствия специализированного инструмента, способного помочь специалистам-акустикам сделать выбор в сторону той или иной акустической теории /метода. В данной работе предложен способ решения проблемы – использование редактора онтологий, предназначенного для использования специалистом предметной области, не имеющих опыта работы с программным обеспечением. Работа посвящена разработке программной системы, реализующей управление методами моделирования архитектурной акустики, обеспечивающей упрощение процесса выбора подходящего метода моделирования, основываясь на специфических характеристиках здания и требованиях к акустическому пространству.

Оглавление

Введение.....	5
1 Существующие методы моделирования в архитектурной акустике. Обзор литературы	8
1.1 Теории описания акустических процессов.....	8
1.2 Методы акустических теорий в архитектурной акустике.....	12
1.3 Анализ существующих методов управления знаниями в области архитектурной акустики.....	15
1.3.1 Онтологический подход	15
1.3.2 Описание принципов работы редакторов онтологий.....	19
1.4 Обзор существующих редакторов онтологий	21
Выводы по первой главе.....	28
2 Построение модели предметной области «Архитектурная акустика»	31
2.1 Определение объектов предметной области.....	31
2.2 Задачи предметной области	34
2.3 Модель онтологии предметной области.....	35
2.4 ER-диаграмма предметной области	40
2.5 Математическая модель представления онтологии	41
Выводы по второй главе.....	44
3 Проектирование редактора онтологий в области архитектурной акустики	45
3.1 Выбор и обоснование модели жизненного цикла редактора онтологий в области архитектурной акустики	45
3.2 Выбор и обоснование подхода к разработке редактора онтологий.....	45
3.3 Спецификация требований к редактору онтологий.....	46
3.3.1 Общее описание требований к редактору онтологий	46
3.3.2 Функциональные требования к редактору онтологий	47
3.3.3 Ограничения дизайна и реализации редактора онтологий.....	48
3.3.4 Классы и характеристики пользователей редактора онтологий	49
3.3.5 Функции редактора онтологий	53
3.4 Архитектурно-контекстная диаграмма редактора онтологий.....	55

3.5 Модульная структура и схема потоков данных.....	57
3.6 Разработка базы данных онтологии предметной области	61
3.7 Модель базы данных для хранения иерархии классов	61
Выводы по третьей главе.....	63
Список использованных источников	64

Введение

Архитектурная акустика, как область науки, занимается управлением звука в помещениях с целью создания качественной акустической среды. От того, насколько корректно звукоинженер рассчитает параметры помещения, будет зависеть итоговый звуковой образ и, как следствие, какую эмоциональную картину получит слушатель во время прослушивания звуковых программ [1]. Обеспечение удовлетворительного восприятия звука напрямую влияет на эмоциональное состояние людей, их способность к концентрации.

Звук является аспектом, определяющим характер помещения, и нет сомнений в том, что акустика влияет на людей и их комфорт, а также определяет атмосферу вокруг всех нас. Качественное акустическое окружение способно на порядок улучшить человеческую жизнь, и, если говорить об интерьере помещения, помнить об этом аспекте необходимо с самого начала работы над ним [2].

Проблемы в области архитектурной акустики могут возникнуть при проектировании абсолютно любых типов зданий, такие как школы, библиотеки, рестораны, концертные залы и живые помещения. Они могут проявляться в плохой звукоизоляции, нежелательном эхо и недостаточной ясности речи, что, безусловно, влияет на комфорт и качество предоставляемых услуг.

Также необходимо подчеркнуть, что особенностями задачи вычислительной акустики являются особенности источников звука, наличие, как и объективных показателей акустики помещения, таких как реверберация, так и субъективных, опирающихся на человеческое восприятие. Звук также можно определить множеством моделей, описывающих звук с различной степенью достоверности [3].

На этапе проектирования помещений одной из основных и приоритетных задач является обеспечение акустического комфорта. Несмотря

на то, что архитектурная акустика является древнейшей дисциплиной, в этой области до сих пор не решена проблема отсутствия одной единственной теории, способной полно описать акустические процессы. На данный момент в силу слабой степени формализации существует три теории описания этих процессов, но специалисты предметной области чаще всего подбирают ее «на глаз», основываясь на полученные эмпирическим путем знания и ощущения.

В целях поддержки принятия решения создаются онтологии – концептуальные модели знаний предметной области, состоящие из понятий и отношений, позволяющих построить формальное описание объектов и процессов [4]. Но если создаются такие узконаправленные базы знаний, то и необходимы инструменты для редактирования этих знаний. В случае области архитектурной акустики инструмент должен быть доступен и понятен специалисту этой предметной области, как главному источнику знаний.

Актуальность темы. Таким образом, на современном этапе развития области архитектурной акустики для успешной работы специалисту данной предметной области необходим инструмент для управления знаниями, т. е. специализированный редактор онтологий.

Практическая ценность заключается в доступности разрабатываемого редактора онтологии для специалистов, ничего не смыслящих в программном обеспечении.

Объектом исследования магистерской диссертации в данном случае является редактор иерархии и правил (редактор онтологии), а предметом – архитектурная акустика.

Целью магистерской диссертации является создание редактора иерархии и правил, который поможет специалистам в области акустики и архитектуры выбирать наиболее подходящие методы моделирования на основе специфических характеристик здания.

Для достижения этой цели необходимо решить следующие задачи:

1. Провести анализ предметной области «Архитектурная акустика»;

2. Построить модель предметной области «Архитектурная акустика»;

3. Разработать проект системы;

4. Реализовать рабочий прототип системы.

Задачи определяют структуру магистерской диссертации:

В первой главе рассматриваются существующие акустические теории, проблема их выбора в зависимости от акустических характеристик помещения, разнообразность методов этих теорий. Проанализированы существующие инструменты управления знаниями, т.е. редакторы онтологий, а также описан принцип их работы. Выявлены недостатки существующих решений по отношению к предметной области «Архитектурная акустика».

Во второй главе строится модель предметной области архитектурной акустики с использованием онтологического подхода: выделяются сущности и информационные объекты, каждый из которых описывается на языке прикладной логики, так же описываются связи между объектами.

В третьей главе проектируется будущая система в соответствии с методологией программной инженерии, рассматриваются заинтересованные лица, определяются функциональные требования, строится логическая модель данных, выбирается средство для реализации.

В четвертой главе разрабатывается рабочий прототип системы, приводятся примеры работы системы.

1 Существующие методы моделирования в архитектурной акустике. Обзор литературы

1.1 Теории описания акустических процессов

Архитектурная акустика — прикладная наука, изучающая законы распространения звуковых волн в помещениях, отражение и поглощение звука поверхностями, влияние отраженных волн на слышимость речи и музыки. В предмет изучения архитектурной акустики также входят методы управления структурой звукового поля и акустическими характеристиками интерьеров [5].

Цель этой науки — создание приемов проектирования залов с заранее предусмотренными хорошими условиями слышимости. К списку практических задач относятся:

- обеспечение комфортных акустических условий в помещениях еще на стадии их проектирования;
- оценка акустических условий в готовых помещениях различного назначения;
- исправление присутствующих акустических дефектов.

Истоки же архитектурной акустики восходят к глубокой древности, она тесно связана с развитием как общества, так и культуры. Уже в древнеримских и древнегреческих театрах можно было наблюдать первые зачатки акустических знаний, применяемых в целях максимально эффективного распространения звука в помещениях, рассчитанных на большое количество слушателей. Осовремененная архитектурная акустика зародилась благодаря работе У. Сэбина, что исследовал процесс затухания звука в закрытых пространствах. Именно он определил, что ключевым показателем «слышимости» является скорость затухания [5].

В настоящее время первоочередными задачами акустического проектирования являются исследование акустических характеристик помещений и процессов отражения и поглощения звука поверхностями,

влияние отраженных волн на слышимость речи и музыки, методы управления шумовыми характеристиками источников, структурой поля и др.

Тем не менее, все еще не существует одной единственной теории, способной объяснить все происходящие акустические процессы и позволившие бы решать все задачи оптимизации для помещений различного назначения с единой позиции. Это связано с особым характером этих задач проблемами акустики залов большой вместительности.

Согласно общепринятой архитектурно-акустической теории, одни и те же акустические процессы в помещениях описываются тремя теориями: волновой, статистической и геометрической [6]. Все они имеют значительную взаимосвязь, дополняя друг друга, и как показывает опыт, только одним методом этих теорий не удаётся решить конкретную задачу.

В статистической теории акустические процессы в помещении рассматриваются как постепенный спад энергии многократно отраженных преградами помещения волн. Этот спад происходит после прекращения действия источника звука. Идеализируя, считают этот процесс непрерывным. Предпосылкой к рассмотрению такого процесса в линейном и полулогарифмическом масштабе является выполнение нескольких условий: все направления движения волн равновероятны, а плотность звуковой энергии в каждой точке пространства помещения одинакова.

Реверберация — это процесс постепенного уменьшения интенсивности звука при его многократных отражениях [7]. Она существенно влияет на качество звучания: чрезмерная реверберация приводит к тому, что новые слоги звучат быстрее, проявляясь на фоне предыдущих затухающих слогов. Соответственно при этом ухудшается разборчивость речи, что имеет значение, к примеру, при прослушивании музыки: затянутый «отзвук» искажает эстетичность восприятия тем больше, чем быстрее звуки «набегают» друг на друга. Но также при очень коротком отзвуке или его отсутствии, как бывает при исполнении музыки вне помещения, утрачивается слитность звучания.

В контексте статистической теории реверберация рассматривается больше как процесс затухания, и эта теория подчеркивает, что качество акустически помещения зависит не столько от материалов, сколько от способности помещения поглощать и рассеивать звуковые волны [8].

Статическая теория занимается второй частью отзвука с повышающейся плотностью импульсов во времени и их затухающей энергией. Прямой звук и начальные отражения статистической теорией не рассматриваются, так же соотношение между энергией прямого и диффузного звука, направленных потоков энергии и фокусировки звука.

Но также она имеет ряд уязвимых мест: не объясняет акустические процессы на очень низких и очень высоких звуковых частотах, в очень больших помещениях, в которых возможно преобладание какого-то линейного размера и с неравномерным распределением звукопоглощающего материала.

В основу же волной теории акустических процессов лег взгляд о том, что сутью реверберации являются не многократные отражения, а постепенно затухающие колебания объемного резонатора, что не зависят от внешних параметров. Она представляет более строгую и точную методику моделирования и распространения звуковых волн.

Из волновой теории вытекает, что помещения простой правильной геометрической формы менее удовлетворяют условию диффузности поля, чем помещения сложной геометрической формы с непараллельными стенами, косо поставленными плоскостями или выпуклыми поверхностями. Линейные размеры этих поверхностей должны быть соизмеримы с длиной волны или быть больше ее. Все сводится к численному интегрированию, поэтому любые численные методы – это методы волновой теории [6].

Волновая теория основывается на том, что при включённом источнике звука звуковые волны распространяются в различных направлениях: осевом, касательном и наклонном, отражаются от ограничивающих помещение плоскостей и, складываясь с прямыми волнами, создают стоячие волны.

Спектра сложного звукового (музыкального) сигнала может содержать частоты, которые отсутствуют или которых мало в спектре собственных колебаний воздуха, находящегося в помещении. Что вызовет ответное (в резонанс) колебание воздушной среды на частотах, совпадающих с частотами источника. При получении добавочной энергии затухание собственных колебаний будет продолжаться дольше, соответственно, изменится период реверберации, а значит и акустические параметры помещения.

Волновая теория особенно полезна в ситуациях, где необходимо уделить внимание деталям, поскольку позволяет точно определить параметры звукового поля, такие как распространение интенсивности звука, характеристики стоячих волн и уровни реверберации.

В разнице в подходах к объяснению акустических процессов в помещениях статистическая и волновая теории дополняют друг друга и потому в известной мере работают на практику совместно. Использование волновой теории способно дать наиболее точные результаты, но это очень дорогой и трудоемкий в вычислительном отношении процесс.

Геометрическая же (лучевая) теория акустических процессов основана на законах геометрической оптики. Движение звуковых волн, исходя из этой теории, подобно движению световых лучей. В соответствии с законами геометрической оптики при отражении от зеркальных поверхностей угол отражения равен углу падения, и падающий и отраженный лучи лежат в одной плоскости. Характер отражения зависит от формы отражающей поверхности [9].

Минимальные размеры помещения для воспроизведения музыки (высота и ширина) не связаны с его вместимостью, а определяются необходимой структурой начальных отражений. Даже если помещение предназначено для исполнения музыки в отсутствии слушателей (студия звукозаписи, звукового вещания, ателье записи музыки, зал прослушивания киностудии), его размеры должны определяться только качеством звучания

музыки. Пренебрегать размером – ухудшать качество звука в значительной степени.

Геометрическая теория учитывает влияние формы помещения на распространение звукового поля, поскольку учитывает степень диффузии звука и наличие стоячих волн: помещения с длинными и узкими формами могут способствовать возникновению стоячих волн, что приведет к искажению звука [10].

Важно знать сильные и слабые стороны акустических теорий, с пониманием и осторожностью применять расчетные формулы, а также сознательно этим пользоваться.

1.2 Методы акустических теорий в архитектурной акустике

В настоящее время специалистами предметной области для решения задачи расчета акустики помещений наиболее часто используется статистическая и геометрическая теории, поскольку волновая теория за счет включенных в нее дифференциальных уравнений расчета звукового поля является достаточно сложным и энергоемким процессом для вычислений.

Волновые методы расчета отклика помещения, например **метод конечных элементов** [11], дают значительно более точные результаты, однако применять их на практике можно только для низких частот, так как с ростом частоты резко увеличивается время вычислений: показательно – от числа элементов, и экспоненциально – от глубины переотражении учитываемых компонент акустических волн. Метод позволяет получить точное решение уравнений акустики с учетом интерференции, дифракции и прочих волновых эффектов, в связи с чем в коммерческих программах для расчета акустических свойств помещений волновой подход применяется весьма ограниченно и с высокой степенью численной аппроксимации математических моделей поля.

Основными характеристиками, влияющими на выбор теории, необходимо считать [12]:

- характеристики помещения, такие как форма, размер, назначение и используемые в нем материалы;
- характеристики источника звука, такие как положение, характеристика направленности и амплитудно-частотная характеристика;
- анализируемые процессы.

К методам статистической теории относится метод У. Сэбина, основанный на модели идеального помещения. В нем звуковое поле рассчитывается на основе статистического рассмотрения процесса затухания звука. Предполагается, что в волновом движении нет преобладающих направлений потоком и симметрии в распределении амплитуд. Это позволяет считать, что звуковое поле изотропно, и средняя плотность звуковой энергии в любой точке помещения одинаково. Такое поле называется диффузным [6].

Его рассмотрение дает возможность пренебречь явлением наложения волн и применять при расчетах энергетическое суммирование.

В методах геометрической теории проводят аналогию с геометрической оптикой: считается, что звук распространяется прямолинейно и зеркально отражается от стен. Эти методы позволяют определить пути распространения звука от источника до слушающего, оценить влияние особенностей помещения на такие характеристики как реверберация, диффузия звука и разборчивость речи. Согласно классификации [12], методы геометрической теории можно поделить на группы по принципу их действия, т. е. на методы распространения энергии и методы отслеживания пути.

Основными же методами группы методов отслеживания пути считаются метод мнимых источников и метод лучевых траекторий (метод трассировки лучей).

Метод мнимых источников – метод, основанный на теории Эйринга, согласно которой звуковое поле, создаваемой точечным источником звука, можно представить как звуковое поле множества мнимых источников, возникающих в результате зеркального отражения звуковых пучков от границы помещения [9]. Этот метод позволяет заменить отраженный звук на

расчет прямого звука от мнимого источника, проанализировать поведение отраженных сигналов и оценить плотность звуковой энергии касательно точки наблюдения.

Он так же применим и при построении отражений от кривых поверхностей. Если требуется найти отражение от какой-либо точки кривой поверхности при заданном положении источника, то следует построить касательную плоскость к этой поверхности. Мнимым источником в это случае будет являться точка, симметричная источнику относительно касательной плоскости.

Следует также отметить, что поле мнимых источников обладает двумя важными свойствами. Во-первых, при внезапном включении источника звука мнимые же источники возникают последовательно, друг за другом, а после выключения – исчезают в такой же последовательности. Во-вторых, свойство, согласно которому акустическая мощность мнимого источника зависит от кратности отражения и коэффициента отражения относится к особенности поля мнимых источников [13].

Метод трассировки лучей – метод, заключающийся в отслеживании и построении траектории наборов звуковых лучей, испускаемых источником, при их последовательном переотражении о граничащих поверхностях до приемлемого затухания амплитуды волны.

Данный метод не накладывает ограничений на форму помещения, а применяемая модель отражения звука может быть основана на зеркальном или диффузном законах.

Остальные алгоритмы являются модификацией метода трассировки лучей: это методы трассировки частиц, пучка, конуса. Они отличаются лишь взглядом на то, что испускает источник. К примеру, термин «метод трассировки частиц» относится больше к подходу, используемому при визуализации данных, где траектории «частиц», обычно представляемых под собой данные, отображаются для изменения во времени. В контексте

архитектурной акустики этот метод используется для визуализации пути звуковых волн от источника, включая в себя отражения от поверхностей.

К методам геометрической акустики можно также отнести **метод распространения излучения** (Radiance Transfer) – метод, основанный на методах граничных элементов, но используемой величиной в данном случае будет являться энергия. Если элементы поверхности достаточно малы, можно предположить, что интенсивность звука не сильно меняется на одном элементе. Направленное пространство так же делится на части для каждого элемента. Фактически, это представляет собой энергию, покидающую участок в одном направлении, но отражающуюся от уже другого участка и в другом направлении [9].

Все эти методы представляют собой важную составляющую геометрической теории, позволяя специалистам применять тот или иной метод в зависимости от характеристик помещения и источника.

1.3 Анализ существующих методов управления знаниями в области архитектурной акустики

1.3.1 Онтологический подход

Для решения проблемы выбора теории и методов моделирования в предметной области архитектурной акустики необходимо создать систему, способную на основе входных данных о помещении сформировать поддержку решения в сторону той или иной акустической теории, а для этого наполнить ее знаниями и организовать их управление – систематизацию, то есть создать базу знаний, используя онтологический подход.

Онтологический подход является частным случаем типологического метода (метода классификации) – совокупности методологических процедур, ориентированных на понимание сложных явлений в структурной самодостаточности, в их становлении и обособлении [14].

Онтология представляет собой формальную модель понятийной структуры предметной области, включающее в себя словарь терминов и

отношений между ними, модель понятий и отношения между понятиями, а также представление факторов, влияющих на выбор той или иной акустической теории и запись их в виде аксиом [15].

Основное назначение онтологий – интеграция информации, т. е. объединение данных и субъекта, потребляющего эти данные. Онтологии связывают два важных аспекта: во-первых, они определяют формальную семантику информации, позволяя обработке этой информации компьютером, и, во-вторых, определяют семантику реального мира, позволяя на основе общей терминологии связывать информацию, представленную в виде, требуемом для компьютерной обработки, с информацией, представленной в удобной форме для восприятия человеком.

При онтологическом управлении знаниями основными элементами онтологии являются:

- классы (концепты) – общие категории, описывающие понятия предметной области и группу индивидуальных сущностей;
- атрибуты, описывающие свойства классов и их экземпляров;
- экземпляры – отдельные представители класса явлений, содержащие конкретные значения атрибутов;
- отношения – вид атрибута, определяющий зависимости между классами.

Основным отношением, учитываемым при построении онтологии, является отношение между концептами (класс-подкласс, гипоним-гипероним), на основе которого формируется таксономия классов.

Составляющие онтологии подчиняются своеобразной иерархии. На нижнем уровне находятся экземпляры, конкретные индивиды, выше идут понятия, то есть категории. На уровень выше располагаются отношения между этими понятиями, а обобщающей и связующей является ступень правил или аксиом. Слоты (параметры) описывают свойства классов и экземпляров.

Онтология определяет общий словарь для людей, которым нужно совместно использовать информацию в предметной области. Единого

универсального подхода к созданию онтологий, который бы однозначно привел к успешному результату, не существует [16]. Создание концептуальной онтологической модели – один из основных методов управления знаниями.

В формальном виде онтология представляется как тройка, что показано в формуле 1 [17]:

$$O = \langle X, R, F \rangle \quad (1)$$

где X – конечное множество концептов предметной области, которую представляет онтология;

R – конечное множество отношений между концептами.

F – конечное множество функций интерпретации, заданных на концептах или отношениях между ними.

Стоит отметить, что множества R и F в отдельных моделях онтологий могут быть пустыми множествами.

Онтология позволяет более эффективно управлять сложной и разнообразной информацией, позволяет приложениям распознавать семантические отличия, неизвестные компьютеру.

Для описания онтологий существуют различные языки представления онтологий (RDF, OWL, KIF, SCL и другие) и системы, однако, наиболее перспективным представляется визуальный подход, позволяющий специалистам непосредственно "рисовать" онтологии, что помогает наглядно сформулировать и объяснить природу и структуру явлений.

В графическом представлении онтология является ориентированным графом с вершинами. Вершины являются понятиями предметной области, а ребра указывают на непосредственную связь между ними. Начальная вершина представляет собой модель базового класса (концепта) предметной области, а связана отношениями «родительское-дочернее» с вершинами следующего

уровня. Эти вершины представляют понятия наиболее общих категорий предметной области.

Онтологии следует классифицировать по уровню и характеру постройки их формальных моделей, здесь они делятся на:

- онтологии верхнего уровня – онтологии, включающие универсальные концепты;

- онтологии предметной области – онтологии, содержащие концепты, характеризующие определенную предметную область:

- прикладные онтологии – онтологии, описывающие связи концептов в определенном приложении.

В рамках исследования наибольший интерес представляют онтологии предметной области. В качестве моделей, описывающих онтологию, выделяют два типа моделей: логическую и фреймовую модели [18-20].

Логическая модель использует логику предикатов первого порядка, формальные языки и системы логического вывода. В этой модели знания можно представить в виде утверждений, связанных логическими операторами, и не могут включать переменных различных значений [21].

Фреймовая модель использует описание онтологии в виде связанных иерархическими отношениями фреймов [22]. Фрейм представляет отдельный концепт, а слоты (другими словами, поля фреймов) содержат информацию об атрибутах. Они могут быть связаны друг с другом через отношения, такие как «являются частью», «имеет» для того, чтобы создавать сложные структуры знаний.

Если сравнивать эти две модели между собой по структуре, то можно вычленил, что фреймовая модель делает акцент на объектах и атрибутах, а логическая – на утверждениях и логических связях между ними.

По выразительности логическая модель имеет больший потенциал для формулирования сложных правил (исходя из использования систем логического вывода). По применению фреймовую модель стоит использовать при структурированных данных и объектно-ориентированном подходе, а

логическую – для систем искусственного интеллекта и задач, требующего формального анализа.

При создании онтологии области архитектурной акустики необходимо учитывать ряд факторов, влияющих на выбор группы методов [23]:

- характеристики поверхностей: сложность помещения, наличие рельефных поверхностей;
- характеристики аппаратного обеспечения: возможность параллельного выполнения алгоритма, объем оперативной памяти;
- потребности моделирования: необходимость отслеживания структуры ранних и поздних отражений, потребность в высокой точности результатов и т.п.

1.3.2 Описание принципов работы редакторов онтологий

Интерес к онтологическим технологиям и их построению значительно возрос в последнее время. Существует стандарт, регламентирующий процесс создания онтологий – стандарт онтологического исследования IDEF [24]. В рамках этого стандарта процесс проектирования онтологии выглядит следующим образом:

1. Изучение и систематизация начальных условий. На данном этапе построения необходимо определить основной контекст онтологии, распределить работу между членами проекта.
2. Сбор и накопление данных для построения онтологии.
3. Анализ и группировка собранных данных для облегчения согласования терминологии.
4. Начальное развитие онтологии. Во время этого шага на основе систематизированных данных о предметной области формируется предварительная онтология.
5. Уточнение и утверждение онтологии.

Для упрощения работы с онтологиями целесообразно пользоваться подходящими инструментами – программными средствами, специально

созданными для проектирования, редактирования и анализа онтологий, называемыми редакторами онтологий [25-28].

Основная их функция – поддержка процесса формализации знаний, представление онтологии как точного и полного описания предметной области, т. е. создание, редактирование, тиражирование и визуализация онтологии. Зачастую технология построения онтологии в рамках этих редакторов заключается в формировании на основе модели системы понятий предметной области, проектируемой экспертом, онтологии «сверху-вниз» [29].

Редакторы онтологий поддерживают такие операции, как ввод понятия и его состава, задание ограничений на этот состав, создание экземпляров, классов и т.д. Только после первоначального формирования базиса следует переназначение связей между экземплярами классов, изменение их состава и другие операции.

Важной характеристикой редактора онтологий является функциональность редактора. Базовый набор функций обеспечивает [30]:

- работу с одним и более проектами, экспорт проекта в нужном формате;
- открытие проекта, представление его структуры;
- редактирование и представление данных проекта;
- создание, редактирование, удаление понятий, отношений.

Уровень автоматизации работы сводится к контролю наличия противоречий в самой онтологии, и при использовании такого рода инструмента структура полностью определяется экспертом «вручную».

Очевидными преимуществами построенной на основе онтологии системы управления знаниями являются [31]:

- устранение дублирования;
- исключение противоречий и непоследовательности;
- избежание появления синонимичных терминов;
- сокращение недочетов;

- повышение степени упорядоченности и структурированности;
- поддержка принципов системности и комплексности;
- универсальность и простота доступа к нужной информации.

Редактор онтологии архитектурной акустики должен быть ориентирован на непрограммирующего специалиста предметной области, быть интуитивно ясен, понятен, иметь понятные алгоритмы действия и обладать наглядностью.

1.4 Обзор существующих редакторов онтологий

Инструменты для работы с онтологиями играют ключевую роль в поддержании и работе со сложными информационными системами, в частности с системами искусственного интеллекта, работе с машинным обучением и семантической обработкой данных.

Инструменты для построения онтологий можно классифицировать по предложенным критериям [32]:

1. По функциональности:

- онтологические редакторы, представляющие графический интерфейс, позволяющий работать с онтологиями без навыков написания кода;
- трансформеры, используемые просто для преобразования онтологий в разные форматы;
- визуализаторы, помогающие наглядно представить онтологию, что облегчает понимание и анализ.

2. По платформе:

- стандартные инструменты, работающие в различных операционных системах с возможностью интегрирования в другие среды разработки;
- онлайн-сервисы, предоставляющие возможность работать через web-интерфейс [33].

3. По уровню автоматизации [34]:

— автоматические генераторы, использующие алгоритмы автоматического создания структур онтологии из текстовых источников и данных;

— semi-automatic tools, комбинирующие как ручную, так и автоматическую работу с онтологиями.

В общем же случае редакторы построения онтологий можно разделить на два типа: разработанные для редактирования онтологий на определенном языке онтологий и интегрированные наращиваемые инструментальные сайты (Web-приложения, на основе форм HTML и/или Java-апплетов), большинство из которых не зависит от языка представления.

Поскольку в свободном доступе отсутствуют узконаправленные редакторы онтологии в области архитектурной акустики, стоит провести обзор существующих программных реализаций общего назначения, являющиеся наиболее известными. Их краткое описание и поддерживаемые языки представлены в таблице 1.

Таблица 1. – Наиболее распространенные редакторы онтологий

Название	Краткое описание	Формализмы, языки, форматы
Ontolingua	Совместная разработка онтологий	OKBC, KIF
Protégé	Создание, просмотр онтологий	JDBC, UML, XML, XOL, SHOE, RDF/RDFS, DAML+OIL, OWL
OntoSarus	Web-браузер баз знаний на языке Loom	Loom
OntoEdit	Разработка и поддержка онтологий	F-Logic, RDFS, OIL, OXML

Продолжение *Таблицы 1.*

OliEd	Разработка онтологий, поддержка логического вывода	DAML+OIL
WebOnto	Многопользовательская разработка онтологий	OCML
WebODE	Создание онтологий с помощью методологии Methontology	F-Logic, LOOM, Ontolingua

1. Ontolingua [35] – первый инструмент инженерных технологий. Система, состоящая из сервера и языка представления знаний, была разработана в KSL (Knowledge Systems Laboratory) Стэнфордского университета. Редактор онтологий является самым важным приложением сервера Ontolingua и web-приложением. Она состоит из сервера и языка представления знаний.

Сервер Ontolingua также представляет архив онтологий различных предметных областей, что позволяет создать из них свою собственную онтологию. Он поддерживает возможность разработки онтологии несколькими пользователями, используя понятия пользователей и их групп. Система так же включает в себя графический браузер, позволяющий взглянуть на иерархию концептов вместе с экземплярами.

Сохраненные же на сервере онтологии преобразуются в различные форматы для использования приложениями, а также могут быть импортированы из ряда других языков.

2. Protégé [36] – свободно распространяемая Java-программа, предназначенная для создания, редактирования и просмотра онтологии прикладной области. Включает редактор онтологий, позволяющий разворачивать иерархию абстрактных и конкретных классов и слотов.

Структура онтологии сделана аналогично структуре каталога, и на основе созданной онтологии программа может генерировать формы получения знаний для введения экземпляров классов и подклассов. Так же данный инструмент редактирования оснащен графическим интерфейсом, снабжен примерами и разного рода правками, а также плагинами, что позволяет адаптировать его для редактирования моделей в разных форматах (к примеру, стандартный текстовый или в базе данных JDBC) [37].

3. OntoEdit [38] – редактор онтологий на базе Java, выполняющий их проверку, просмотр, кодирование и модификацию. Способен поддерживать языки FLogic, включая машину вывода, OIL, расширение RDFS и внутреннюю, основанную на XML, сериализацию модели онтологии используя OXML - язык представления знаний OntoEdit (OntoEdit's XML-based Ontology representation Language).

Данный инструмент имеет ряд объективных плюсов: удобство использования, разработка онтологии под руководством методологии и процесса логического вывода, а также обладает схожей с Protégé архитектурой и способен расширить структуру посредством плагинов.

Данный редактор онтологий имеет две версии: свободно распространяемая, но ограниченная концептами, отношениями и экземплярами, и лицензированная, имеющая большее количество возможностей.

На рисунке 1 представлена архитектура программы, подобная архитектуре Protégé.

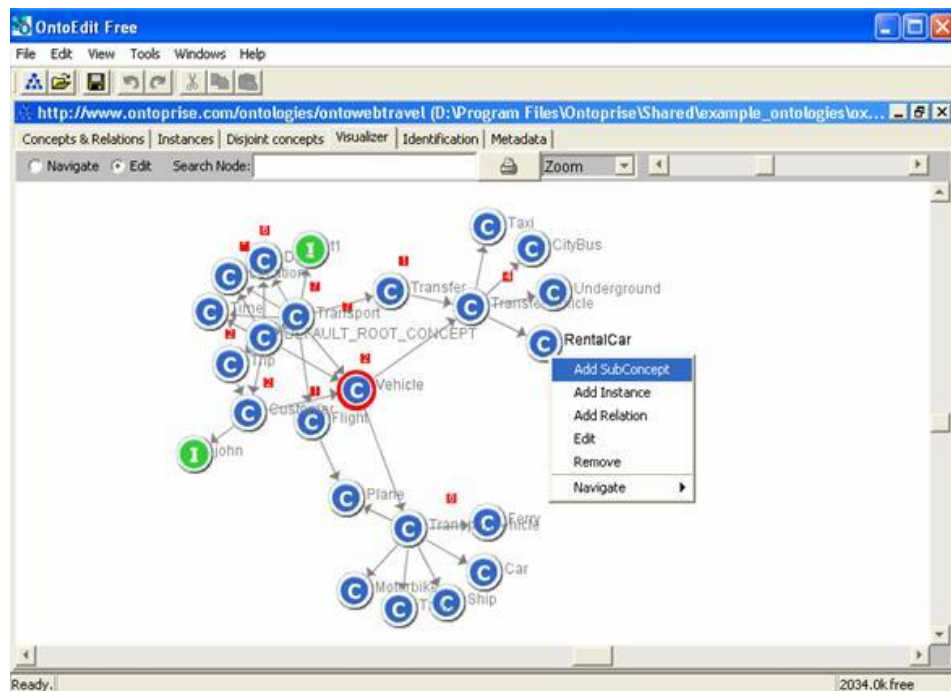


Рисунок 1 – Навигация по онтологии и ее редактирование с помощью OntoEdit

4. OilEd [39] – автономный графический редактор онтологий, основанный на языке OIL, который сочетает в себе фреймовую структуру и выразительность дескриптивной логики с сервисами рассуждения. Это позволило системе обеспечить интуитивный интерфейс пользователя и обнаружение противоречивых классов и скрытых отношений подклассов.

К недостаткам можно отнести отсутствие поддержки экземпляров, разработки онтологий большого масштаба, их интеграции. Но с помощью OilEd возможно просто строить онтологии, а после демонстрировать механизм для проверки онтологии на противоречивость.

5. WebOnto [40] – программа, разработанная для Tadzebao – инструмента исследования онтологий. Главные цели редактора – простота использования, представления средств для построения больших онтологий.

Для моделирования онтологий используется язык OCML [41]. Пользователь может создавать структуры, включая классы с множественным наследованием, где все слоты наследуется корректно.

Данный инструмент так же имеет ряд достоинств: сохранение структурных диаграмм, отдельный просмотр отношений классов, а также работа нескольких пользователей над онтологией при использовании диаграмм, функций передачи / приема и других.

6. OntoSarus – это web-браузер для баз знаний LOOM [42]. Он включает в себя два основных модуля: сервер онтологий web-браузера для редактирования и просмотра онтологий LOOM. OntoSarus предоставляет функции классификации концепций и сопоставления экземпляров. Веб-редактор и браузер генерируют HTML-страницы, включая изображения и другую документацию, из онтологий LOOM, хранящихся на сервере онтологии.

Однако стоит отметить, что данное средство предлагает лишь ограниченный набор инструментария, и для редактирования сложных онтологий может потребоваться более серьезный инструмент.

7. ODE [43] – конструктор онтологий, взаимодействующий с пользователями на концептуальном уровне (в отличие от OntoSaurus, общающейся на символьном). Программа обеспечивает пользователей набором таблиц для заполнения их концептами, атрибутами и отношениями между ними и автоматически генерирует для них код для таких программ, как LOOM, Ontolingua и Flogic. Дальнейшее свое развитие этот инструмент получил в WebODE [44], которая интегрирует все сервисы ODE в одну архитектуру.

Отличия инструментов для редактирования онтологий по основным категориям представлены в таблице 2.

Таблица 2. Сравнение редакторов онтологий

	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protege	WebODE	WebOnto
Доступность	Открытый код	Свободная лицензия	Свободный доступ	Открытый код, свободный доступ	Открытый код	Свободный доступ	Свободный отступ
Поддержка методологией	-	On-To-Knowledge	-	-	-	METHONTOLOGY	-
Расширяемость	-	Плагины	-	-	Плагины	Сервер приложения	-
Хранение онтологий	файлы	файлы	файлы	файлы	Файлы, СУБД	СУБД	СУБД
Язык ПО	Java	Java	Lisp	Lisp	Java	Java	Java + Lisp
Формализм	DL	Фреймы +FOL	Фреймы +FOL	DL	Фреймы +FOL	Фреймы +FOL	Фреймы +FOL
Основной язык представления знания	DAML+OIL	OXML	Ontolingua	LOOM	OKBC	-	OCML
Интерфейс пользователя	Локальное приложение	Локальное приложение	HTML	HTML	Локальное приложение	HTML и апплеты	Апплеты
Совместная разработка онтологии	нет	да	да	да	нет	да	ла
Проверка непротиворечивости	да	да	нет	да	да	да	да

Стоит отметить, что большинство рассмотренных инструментов разрабатываются университетскими группами, который представляют открытый код либо свободный доступ к функциям. Но наиболее перспективные из них, как OntoEdit Professional, все же передаются коммерческим компаниям.

Исходя из представленной выше таблицы, можно подвести, что такие инструменты, как OntoEdit, WebODE и KADS22 дают поддержку методологиям построения онтологий, что не мешает им использоваться в других методологиях или вообще без них. OntoEdit, Ontolingua, OntoSaurus, WebODE и WebOnto поддерживают совместную разработку онтологий, предоставляя отдельным пользователям или группам пользователей разрешение на доступ и написание различных наборов онтологий.

Более ранние инструменты Ontolingua, OntoSaurus и WebOnto имеют клиент-серверную архитектуру. Protégé, OntoEdit и OilEd имеют 3-уровневую архитектуру, где существует четкое разделение между хранением онтологий, модулями бизнес-логики логики приложений и приложениями интерфейса пользователя. Эти инструменты обладают большими возможностями по наращиванию (например, при помощи плагинов). Большинство инструментов хранит свои онтологии в текстовых файлах, что ограничивает размер онтологий. Только Protégé и WebODE могут хранить свои онтологии в базах данных и таким образом управлять большими онтологиями. Наконец, большинство инструментов реализовано на Java [45].

Разнообразие инструментов для редактирования онтологий делает их сравнение сложным. Выбор одного единственного программного продукта в таком случае должен быть обусловлен зависимостью от выполнения конкретных задач.

Выводы по первой главе

В первой главе были рассмотрены понятие «архитектурная акустика», цели этой науки, три акустические теории, методы этих теорий и проблема их

выбора. Также были проанализирован метод управления знаниями путем онтологического подхода, описаны его общие аспекты и сравнены существующие инструменты для работы с онтологиями – редакторы онтологий.

В настоящее время создано огромное количество инструментов редактирования онтологий, часть которых была представлена выше в качестве обзора. Они обладают своими характерными плюсами и минусами, но являются общими, не прикрепленными к какой-то конкретной предметной области.

В случае предметной области архитектурной акустики редактор онтологий как инструмент должен отвечать ряду как функциональных, так и нефункциональных требований:

1. Редактор должен обладать возможностью загрузки, сохранения и визуализации онтологии;
2. Редактор должен предоставлять специалисту предметной области дружелюбный и интуитивно понятный интерфейс, рассчитанный на непрограммирующего пользователя, позволяющий быстро редактировать и загружать новые данные в онтологию;
3. Редактор должен обладать гибким механизмом поиска и редактирования концептов онтологии.
4. Редактор должен обладать функцией формирования отчетов: распечатка описания баз знаний, правил, шаблонов факторов, а также описание результатов логического вывода.

Таким образом, редактор онтологий для области архитектурной акустики будет являться незаменимым инструментом для акустического проектирования таких объектов, как здания, концертные залы, кинотеатры, стадионы и другие. Он поможет экспертам и специалистам предметной области точно определять акустические параметры и находить наилучшие способы улучшения звукового комфорта в помещениях.

Кроме того, полученные онтологии можно будет использовать для обучения и подготовки квалифицированных специалистов, что поспособствует развитию этой области.

Из этого следует, что создание собственного инструмента для редактирования онтологии в области архитектурной акустики является актуальным.

2 Построение модели предметной области «Архитектурная акустика»

2.1 Определение объектов предметной области

Объектами профессиональной деятельности в области архитектурной акустики являются:

1. Поверхность – элемент, который может оказывать влияние на звуковое поле в пространстве. Поверхность имеет свойства, такие как положение, материал, кривизна, площадь;

Имеем конечное множество поверхностей с индивидуальным наименованиями. Названия поверхностей представляются скалярным значением. Каждая поверхность определяется следующими характеристиками:

а. положение – скалярное значение, характеризующее поверхность в пространстве;

б. материал – скалярное значение, которое характеризует, из чего именно сделана поверхность;

с. кривизна – вещественное число от 0 до 6,28319, характеризующее искривление поверхности;

д. площадь – вещественное число от 0 до 4294967295, которое характеризует размеры поверхности;

е. коэффициент поглощения – вещественное число от 0 до 100, которое характеризует величину коэффициента поглощения.

2. Звук – колебания акустических волн, которые передаются через среду и воспринимаются слуховой системой человека. Изучение характеристик звука позволяет проанализировать его взаимодействие с окружающей средой;

3. Источник – объект, создающий акустические колебания и излучающий звуковые волны;

Имеет множество источников с индивидуальными наименованиями. Наименования представляются скалярным значением. Каждое направление определяется следующими характеристиками:

- a. Положение – скалярное значение, обозначающие местоположение источника в пространстве;
 - b. Частота АЧХ – вещественное число от 3 до $3 \cdot 10^9$, характеризующее амплитудно-частотную характеристику;
 - c. Частота ХН - вещественное число от 3 до $3 \cdot 10^9$, характеризующее характеристику направленности.
 - d. Направление ХН – скалярное значение, обозначающее направление характеристики направленности;
 - e. Интенсивность ХН - вещественное число от 0 до 4294967295? , характеризующее величину характеристик направленности;
4. Среда распространения – окружающая среда, через которую звук распространяется;
 5. Помещение – объект в архитектурной акустике, включающий в себя различные математические и физические модели, а также зависящий от источника звука, характеристик поверхности и назначения;

Имеет множество помещений с индивидуальными наименованиями. Наименования представляются скалярным значением. Каждое направление определяется следующими характеристиками:

- a. Источник – скалярное значение, обозначающее множество наименований источников;
- b. Поверхность - скалярное значение, обозначающее множество наименований поверхностей;

Назначение - скалярное значение, обозначающее предназначение источника излучения.

6. Акустические теории – три теории: волновая, геометрическая и статистическая, применяемые для создания и оптимизации акустически комфортных и функциональных пространств. Характеризуется:

а. Волновая теория – фокусируется на анализе звуковых волн и их распространении в пространстве. Она учитывает такие параметры, как длина волны, скорость звука и взаимодействие звуковых волн с различными объектами и поверхностями.

б. Геометрическая теория – изучает влияние формы и размеров объектов на распространение звука. Она учитывает такие характеристики, как кривизна поверхности, площадь объекта и его расположение относительно источника звука и точки приема. Геометрическая теория позволяет определить, как звук отражается от поверхностей и как это влияет на звуковое поле в помещении.

с. Статистическая теория – использует статистические методы для анализа звукового поля и его диффузии в пространстве. Она учитывает такие параметры, как коэффициент поглощения материала, его текстуру и другие свойства, влияющие на способность материала поглощать звук. Статистическая теория позволяет предсказать поведение звукового поля в помещении и оптимизировать его для достижения желаемых акустических характеристик.

7. Методы – способы расчета, применяемые для анализа и предсказания акустических свойств объектов и среды распространения звука. Каждый метод можно классифицировать по ряду характеристик, и он может быть снабжен теми или иными свойствами:

а. Тип метода – скалярное значение, обозначающее категорию метода. Тип метода определяет основные принципы его применения и может влиять на точность и применимость метода.

б. Параметры моделирования – вещественные числа, отражающие параметры, используемые в методе для анализа акустических свойств, такие как длина волны, скорость звука, коэффициент поглощения материала и другие. Эти параметры могут быть адаптированы для различных условий и материалов.

с. Точность и разрешающая способность – вещественные числа, отражающие точность и разрешающую способность метода. Точность определяет, насколько близко результаты метода соответствуют реальным акустическим свойствам, а разрешающая способность указывает на детализацию анализа.

d. Время вычислений – вещественное число, отражающее время, необходимое для выполнения расчетов с использованием данного метода. Это может быть критическим фактором при выборе метода для реальных акустических задач.

е. Применимость – скалярное значение, обозначающее область применения метода, например, для внутренних помещений, подводных акустических каналов или для анализа звукового поля в открытых пространствах.

8. Потребности моделирования – характеристики, влияющие на выбор метода: скорость моделирования, учет волновых эффектов, набор интересующих частот и отслеживание отдельных отражений.

2.2 Задачи предметной области

Общая цель специалистов предметной области заключается в обеспечении комфортной и функциональной среды в соответствующих помещениях, создании эстетически приятного акустического окружения. Для достижения этой цели решаются следующие задачи:

1. Определение и описание звуковых характеристик помещений, чтобы определить, как звук взаимодействует с помещением и влияет на звуковое восприятие;

2. Идентификация проблем и разработка решений, включающие поиск отрицательно влияющих акустических характеристик помещения и нахождение способов решения данных проблем;

3. Оптимизация акустической среды, заключающаяся в моделировании и симуляции акустической среды для предсказания и оценки звуковых характеристик перед реализацией проектов помещений;

4. Консультирование и рекомендации по вопросам акустики помещений и выбору специального оборудования для конкретного объекта или проекта.

5. Анализ и оценка существующих объектов с целью выявления возможных проблем и предложение рекомендаций по улучшению.

2.3 Модель онтологии предметной области

Для стандартизации понятий и терминов и систематизации знаний следует принять определенные онтологические соглашения:

1. Онтология должна определять основные объекты и явления, с которыми работает архитектурная акустика, такие как звук, шум, поглощение звука, отражение звука, диффузия звука, реверберация и др. Каждый объект и явление должны быть явно определены и иметь соответствующие свойства и отношения;

2. Онтология должна предоставлять классификацию объектов архитектурной акустики, например, различные типы акустических материалов, конструкций и систем. Каждый класс должен иметь свои характеристики и отношения с другими классами;

3. Онтология должна определять основные свойства объектов архитектурной акустики, такие как коэффициент поглощения звука, коэффициент отражения звука и др. Каждое свойство должно иметь определенные значения и единицы измерения;

4. Онтология должна определять отношения между объектами архитектурной акустики, такие как влияние материала на поглощение звука, влияние формы помещения на рассеивание звука и др. Каждое отношение должно быть явно определено и иметь соответствующие параметры;

5. Онтология должна учитывать основные методы и технологии, используемые в архитектурной акустике;

6. Каждый метод и технология должны быть описаны с учетом их цели и принципов работы.

Необходимо определить термины, которые задают знания. Онтология разбита на модули, каждый из которых представляет информационный объект.

1) Модуль «Поверхность»

сорт поверхность: $\{ \} N \setminus \emptyset$

Термин «поверхность» обозначает непустое конечное множество наименований поверхностей;

сорт положение: $\{ \} N \setminus \emptyset$

Термин «положение» обозначает непустое конечное множество наименований положений;

сорт положение поверхности: поверхность \rightarrow положение

Термин «положение поверхности» - функция, сопоставляющая наименованию поверхности наименование ее положения;

сорт материал: $\{ \} N \setminus \emptyset$

Термин «материал» обозначает непустое конечное множество наименований материалов;

сорт коэффициент поглощения: материал $\rightarrow R(0, \infty)$

Термин «коэффициент поглощения» - функция, сопоставляющая наименованию материала положительное вещественное число;

сорт фактура: $\{ \} N \setminus \emptyset$

Термин «фактура» обозначает непустое конечное множество наименований фактур;

сорт фактура материала: материал \rightarrow фактура

Термин «фактура материала» - функция, сопоставляющая наименованию материала наименование фактуры;

сорт материал помещения: помещение \rightarrow материал

Термин «материал помещения» - функция, сопоставляющая наименованию помещения наименование материала;

сорт кривизна: помещение $\rightarrow \mathbb{R} (-\infty; +\infty)$

Термин «кривизна» - функция, сопоставляющая наименованию помещения диапазон вещественных чисел;

сорт площадь: помещение $\rightarrow \mathbb{R} (0; +\infty)$

Термин «площадь» - функция, сопоставляющая наименованию помещения множество положительных вещественных чисел;

2) Модуль «Направление»

сорт направление: $\{ \} N \setminus \emptyset$

Термин «направление» обозначает непустое конечное множество значений;

сорт азимут: направление $\rightarrow \mathbb{R} (-\infty; +\infty)$

Термин «азимут» - функция, сопоставляющая значению направления множество вещественных чисел;

сорт зенит: направление $\rightarrow \mathbb{R} (-\infty; +\infty)$

Термин «зенит» - функция, сопоставляющая значению направления множество вещественных чисел;

3) Модуль «Источник»

сорт источник: $\{ \} N \setminus \emptyset$

Термин «источник» обозначает непустое конечное множество названий источников.

сорт положение: $\{ \} N \setminus \emptyset$

Термин «положение» обозначает непустое конечное множество положений.

сорт положение источника: источник \rightarrow положение

Термин «положение источника» - функция, сопоставляющая наименованию источника наименование положения.

сорт АЧХ источника: источник \rightarrow АЧХ

Термин «АЧХ источника» - функция, сопоставляющая значению источника значение АЧХ.

сорт ХН источника: источник \rightarrow ХН

Термин «ХН источника» - функция, сопоставляющая значению источника значение ХН.

4) Модуль «АЧХ и ФН»

сорт АЧХ: источник $\rightarrow R(0; +\infty)$

Термин «АЧХ» - функция, сопоставляющая наименованию источника положительное множество вещественных чисел;

сорт амплитуда АЧХ; источник $\rightarrow R(0; +\infty)$

Термин «амплитуда» - функция, сопоставляющая множеству значений АЧХ множество вещественных чисел амплитуды.

сорт ХН: источник $\rightarrow R(0; +\infty)$

Термин «ХН» - функция, сопоставляющая наименованиям источника положительное множество вещественных чисел.

сорт направление ХН: источник \rightarrow направление

Термин «направление ХН» - функция, сопоставляющая значения ХН значениям направления;

сорт интенсивность ХН: источник $\rightarrow R(0; +\infty)$

Термин «интенсивность» - функция, сопоставляющая значениям ХН множество вещественных чисел.

сорт частота АЧХ: источник $\rightarrow R(0; +\infty)$

Термин «частота» - функция, сопоставляющая значениям АЧХ и ФН множество вещественных чисел.

сорт частота ХН: источник $\rightarrow R(0; +\infty)$

Термин «частота ХН» - функция, сопоставляющая значениям АЧХ и ФН множество вещественных чисел.

5) Модуль «Помещение»

сорт помещение: $\{ \} N \setminus \emptyset$

Термин «помещение» обозначает непустое конечное множество наименований.

сорт источники помещения: помещение $\rightarrow \{\}$ источник

Термин «источники помещения» - функция, сопоставляющая наименованию помещения значения источника;

сорт поверхности помещения: помещение \rightarrow поверхность

Термин «поверхности помещения» - функция, сопоставляющая названию помещения значения поверхности;

сорт назначение помещения: помещение $\rightarrow \{\}$ назначение

Термин «назначение помещения» - функция, сопоставляющая названию помещения его значение назначения;

б) Модуль «Методы»

сорт методы: $\{\} N \setminus \emptyset$

Термин «методы» обозначает непустое конечное множество наименований.

сорт метод трассировки частиц: $\rightarrow \{\}$ методы

Термин «метод трассировки частиц» - функция, сопоставляющая методам значение метода трассировки частиц.

сорт метод трассировки луча: $\rightarrow \{\}$ методы

Термин «метод трассировки луча» - функция, сопоставляющая методам значение метода трассировки луча.

сорт метод трассировки пучка: $\rightarrow \{\}$ методы

Термин «метод трассировки пучка» - функция, сопоставляющая методам значение метода трассировки пучка.

сорт методы трассировки конуса: $\rightarrow \{\}$ методы

Термин «методы трассировки конуса» - функция, сопоставляющая методам значение методов трассировки конуса.

сорт метод мнимых источников: $\rightarrow \{\}$ методы

Термин «метод мнимых источников» - функция, сопоставляющая методам значение метода мнимых источников.

сорт метод распространения излучения: $\rightarrow \{ \}$ методы

Термин «метод распространения излучения» - функция, сопоставляющая методам значение метода распространения излучения.

сорт метод эффективного излучения: $\rightarrow \{ \}$ методы

Термин «метод потока эффективного излучения» - функция, сопоставляющая методам значение метода потока эффективного излучения.

сорт методы распространения энергии: $\{ \}$ метод распространения излучения \times метод потока эффективного излучения

сорт методы отслеживания пути: $\{ \}$ метод мнимых источников \times метод трассировки луча \times метод трассировки конуса \times метод трассировки пучка \times метод трассировки частиц

методы геометрической теории: $\{ \}$ методы распространения энергии \times методы отслеживания пути

2.4 ER-диаграмма предметной области

ER-диаграмма предметной области, также известная как диаграмма отношений сущностей (Entity Relationship Diagram, ERD), является инструментом моделирования данных, используемым для визуализации структуры баз данных. Она позволяет представить сущности (объекты), атрибуты этих сущностей (характеристики объектов) и отношения между ними. ER-диаграммы применяются на различных этапах разработки информационных систем, начиная от определения требований и заканчивая проектированием и реализацией баз данных [46].

1. Сущности: Сущности представляют основные объекты или концепции в области архитектурной акустики. Например, «Материал», «Звуковое свойство», «Здание», «Помещение». Каждая сущность имеет свои атрибуты. Например, «Материал» может иметь атрибуты «Коэффициент поглощения», «Тип», «Фактура», «Название» и т.д.

2. Отношения: Отношения показывают, как сущности связаны друг с другом. Например, «Помещение» может иметь отношение «Содержит» с «Поверхность», которое описывает поверхности в используемом помещении.

Диаграмма представлена на рисунке 2.

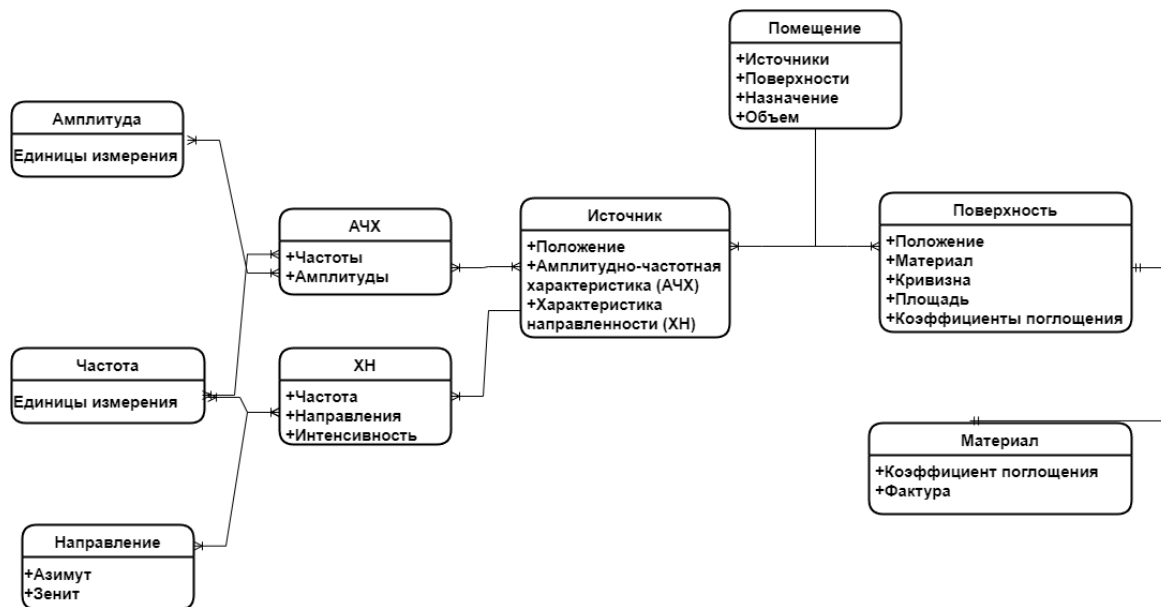


Рисунок 2 – ER-диаграмма предметной области

2.5 Математическая модель представления онтологии

Для программной реализации модели онтологии необходимо разработать структуры для представления концептов и экземпляров онтологии предметной области.

Концепт онтологии представлен следующей структурой:

$$c_i = \langle nc_i, dc_i, D_i, A_i \rangle,$$

Где nc_i – имя концепта; dc_i – описание концепта (содержит текстовый комментарий);

D_i – множество ссылок на дочерние концепты;

$A_i = \{ a_i \}$ – множество атрибутов концепта.

Совокупность множеств $\{ D_i \}$ является реализацией связей в онтологии, каждый концепт ссылается на множество других концептов, что позволяет обойти все древо концептов, начиная с корневого концепта всей онтологии.

Атрибуты концепта онтологии представлены следующей структурой:

$$a_i = \langle na_j, da_j, ta_j \rangle,$$

где na_j – имя атрибута; da_j – описание атрибута; ta_j – тип атрибута.

В качестве основных типов данных выбираются числовой, текстовый, логический, тип ссылки на объект:

$$T = \{Int, Num, Txt, Bool, ObjR\}, ta_j \in T,$$

Структура данных экземпляра концепта онтологии:

$$o_k = \langle no_k, to_k, V_k \rangle,$$

где no_k – имя экземпляра; to_k – тип экземпляра;

$V_k = \{V_k\}$ – множество значений атрибутов k-го экземпляра.

Процесс построения онтологии происходит за счет ввода новых атрибутов, моделей концептов и экземпляров в модель и дальнейшее ее редактирование.

1. Добавление нового атрибута в онтологию:

$$F_{add_attribute}: M = \{m_l, \dots, m_n\} \rightarrow M' = \{m_l, \dots, m_n, m_{n+1}\},$$

Где M – множество атрибутов концептов;

Тип атрибута может повторяться, но имя должно быть уникальным.

Добавленные атрибуты могут быть использованы при вводе в онтологию новых концептов.

2. Удаление атрибута из онтологии:

$$F_{remove_attribute}: M \rightarrow M' = M \setminus \{m_i\},$$

Где $|M| = n$, $|M'| = n-1$; m_i – удаляемый атрибут.

3. Редактирование атрибута:

При редактировании атрибута могут быть изменены его тип и описание, и эта операция выполняется только на этапе перед созданием экземпляра онтологии. Операция затрагивает модели концепта:

$$F_{change_attribute}: m_j = \langle nm_j, tm_j, dm_j \rangle \rightarrow m_j' = \langle nm_j, tm_j', dm_j' \rangle$$

4. Добавление нового концепта в онтологию:

$$F_{add_concept}: \left(\begin{array}{c} C = \{c_l, \dots, c_n\} \\ R \end{array} \right) \rightarrow \left(\begin{array}{c} C' = \{c_l, \dots, c_n, c_{n+1}\} \\ R' \end{array} \right)$$

5. Удаление концепта из онтологии:

$$F_{remove_concept}: \left(\begin{array}{c} C = \{c_l, \dots, c_n\} \\ R \end{array} \right) \rightarrow \left(\begin{array}{c} C' = \{c_l, \dots, c_n, c_{n-1}\} \\ R' \end{array} \right)$$

Из онтологии может быть удален концепт, не имеющий экземпляров, иначе будет нарушена целостность данных онтологии.

Удаление концепта должно сопровождаться переносом определений атрибутов, входящих в состав удаляемого концепта. Такие определения должны быть включены в состав прямых потомков удаляемого концепта.

6. Удаление атрибута онтологии из состава концепта:

$$F_{remove_attribute}: MD_i \rightarrow MD'_i = MD_i \setminus \{m_{rem}\},$$

Где m_{rem} – атрибут, удаляемый из состава концепта.

При наличии в онтологии созданных экземпляров концептов данная операция будет нарушать целостность данных онтологии и должна блокироваться, но в онтологиях, не использующих экземпляры концептов, она работает без ограничений.

7. Добавление атрибута в состав концепта:

В процессе редактирования можно добавить новые атрибуты. Операция будет затрагивать модели концептов, наследующих от редактируемого концепта.

$$F_{add_attribute}: MD_i \rightarrow MD'_i = MD_i \cup \{m_{new}\},$$

где m_{new} – атрибут, включаемый в состав концепта.

8. Добавление связи наследования:

Технология построения онтологии предполагает последовательное задание для нового концепта его родительских концептов, когда состав концепта формируется на основе отношений наследования.

$$F_{add_parent} = R \rightarrow R' = R \cup \{(c_{par}, c_i)\},$$

где c_i – редактируемый концепт;

c_{par} – добавляемый родительский концепт.

9. Удаление связи наследования:

$$F_{remove_parent}: R \rightarrow R' = R / \{(c_{par}, c_i)\},$$

где c_i – редактируемый концепт;

c_{par} – добавляемый родительский концепт.

При удалении связи с родительским концептом из состава редактируемого концепта исключаются атрибуты, унаследованные только от того концепта, связь с которым удаляется.

10. Создание экземпляра концепта:

При создании экземпляра концепта атрибутам должны быть присвоены отдельные значения, но также могут использоваться значения атрибутов по умолчанию и пустые значения.

$$F_{add_object} = O \rightarrow O' = O \cup \{o_j\}$$

Выводы по второй главе

3 Проектирование редактора онтологий в области архитектурной акустики

3.1 Выбор и обоснование модели жизненного цикла редактора онтологий в области архитектурной акустики

Модель жизненного цикла, которая наиболее подходит для редактора онтологий в области архитектурной акустики, — это модель V или модель проверки и валидации. Эта модель подразумевает, что продукт (в данном случае, редактор онтологий) должен быть тщательно проверен и валидирован перед его выпуском, чтобы убедиться, что он соответствует требованиям и ожиданиям пользователей. Это особенно важно для редактора онтологий, поскольку он будет использоваться для создания и редактирования онтологий, которые отражают сложные концепции и отношения в области архитектурной акустики [48].

Эта модель подходит для разработки также по ряду причин:

1. Модель V обеспечивает поддержку в планировании и реализации проекта. В ходе проекта ставятся следующие задачи: минимизация рисков, повышение и гарантии качества, уменьшение общей стоимости проекта и повышение качества коммуникации между участниками проекта.
2. В модели V деятельность разбивается на отдельные шаги, каждый из которых включает в себя необходимые для него действия, инструкции к ним, рекомендации и подробное объяснение деятельности.

3.2 Выбор и обоснование подхода к разработке редактора онтологий

Для разработки редактора онтологий, объектно-ориентированный подход более уместный, чем структурный.

Объектно-ориентированный подход позволяет эффективно моделировать сложные структуры данных и отношения между ними, что является ключевой особенностью работы с онтологиями. Он позволяет представить понятия предметной области в виде классов и объектов, что

делает его особенно полезным при работе с онтологиями, где понятия и отношения между ними часто сложны и многоуровневые [49].

Можно также привести несколько причин:

1. Объектно-ориентированный подход предоставляет гибкость и мощность, необходимые для работы со сложными системами, такими как редактор онтологий. Он поддерживает принципы инкапсуляции, наследования и полиморфизма, которые помогают организовать код, повторно использовать код и легко расширять систему в будущем;

2. Объектно-ориентированный подход поддерживает изменения в системе. Это значит, что если в процессе разработки появятся новые требования или потребуется внести изменения в существующий код, это будет относительно просто выполнить с помощью объектно-ориентированного подхода;

3. Объектно-ориентированный подход хорошо поддерживает семантический веб, который является ключевым для работы с онтологиями. С помощью объектно-ориентированного подхода можно легко создавать, прототипировать и комбинировать модели предметной области с другими компонентами приложения, такими как пользовательский интерфейс [50];

3.3 Спецификация требований к редактору онтологий

3.3.1 Общее описание требований к редактору онтологий

Программа для редактирования онтологий в области архитектурной акустики представляет собой инструмент для создания и управления онтологиями, которые отражают специфические концепции и отношения в этой области. Онтологии используются для формализации знаний в определенной области, предоставляя структуру и терминологию, которую можно использовать для представления этих знаний в компьютерных системах.

В контексте архитектурной акустики онтология включает классы, такие как "Источник", "Помещение", "Поверхность", и так далее. Отношения между этими классами могут быть выражены через свойства.

Редактор онтологий предоставляет интерфейс для создания и редактирования этих классов и отношений. Это включает возможности для добавления новых классов или свойств, изменения существующих, а также удаления нежелательных элементов.

3.3.2 Функциональные требования к редактору онтологий

Редактор онтологий в области архитектурной акустики должен решать следующие задачи:

1. Создание и редактирование онтологий: Редактор должен предоставлять инструменты для создания и редактирования онтологий, включая работу с концептами, атрибутами, экземплярами.
2. Поддержка форматов онтологий: Редактор должен поддерживать стандартные форматы онтологий, такие как OWL (Web Ontology Language) или RDF (Resource Description Framework), а также их импорт и экспорт.
3. Визуализация онтологий: Редактор должен предоставлять возможность визуализации онтологий, что помогает лучше понять структуру и связи между концепциями;
4. Интеграция с другими системами: Редактор должен быть способен интегрироваться с другими системами и платформами, например, с программным обеспечением для моделирования звука или анализа акустических характеристик зданий;
5. Экспорт и импорт онтологий: Редактор должен предоставлять возможности экспорта и импорта онтологий в различных форматах;
6. Пользовательские роли и права доступа: Редактор должен предоставлять возможность управления доступом к онтологиям и редактированию их содержимого различными пользователями;

7. Аудит и отслеживание изменений: Редактор должен предоставлять возможности аудита и отслеживания изменений в онтологиях, что помогает контролировать версию и обеспечивает совместимость;

8. Поддержка стандартов и рекомендаций: Редактор должен соответствовать всем актуальным стандартам и рекомендациям в области архитектурной акустики и онтологий.

9. Редактор онтологий должен быть предназначен для специалиста предметной области, имеющего навыки работы на компьютере или для обучающегося в данной предметной области;

10. Процедура запуска редактора онтологий должна быть как можно более простой для пользователя;

11. Редактор онтологий должен быть переносимым без изменений из одной среды в другую в рамках Windows 7, Windows 8, Windows 10, Windows 11;

12. Прототип редактора онтологий должен позволять пользователю заходить в программу, добавлять и сохранять внесенные в нее данные;

13. Редактор онтологий должен быть надежным и стабильным, обеспечивая точность и целостность данных. Это может включать в себя требования к обработке ошибок, восстановлению после сбоев и безопасности данных;

14. Редактор онтологий должен предоставлять интуитивно понятный и удобный интерфейс для пользователей. Это может включать в себя требования к дизайну, навигации, взаимодействию с пользователем и т.д.

3.3.3 Ограничения дизайна и реализации редактора онтологий

Поскольку для разработки редактора онтологий был выбран язык Python, необходимо обозначить ряд ограничений, которые следует учесть:

1. Несмотря на то, что Python имеет множество полезных библиотек для работы с онтологиями, таких как `rdflib` и `owlready2`, они могут не

поддерживать все функции, доступные в других языках программирования или форматах онтологий.

2. Python использует исключения для обработки ошибок, что может сделать обработку ошибок в редакторе онтологий более сложной, особенно при работе с внешними библиотеками или API;

3. Python поддерживает множество операционных систем, но некоторые библиотеки могут работать только на определенных платформах. Кроме того, некоторые функции Python могут не поддерживаться в некоторых версиях Python, что может ограничить совместимость редактора онтологий с различными средами;

4. Python может использовать больше памяти по сравнению с некоторыми другими языками программирования, что может быть проблемой при работе с большими онтологиями.

3.3.4 Классы и характеристики пользователей редактора онтологий

Основные участники, которые взаимодействуют с программным обеспечением:

— Пользователь (инженер-акустик) – внешнее лицо, которое использует редактор онтологий для создания, редактирования и просмотра онтологий;

— Администратор – внешний исполнитель, который наполняет онтологическую базу знаний и сопровождает систему.

Диаграмма использования программы пользователем инженером-акустиком представлена на рисунке 3.

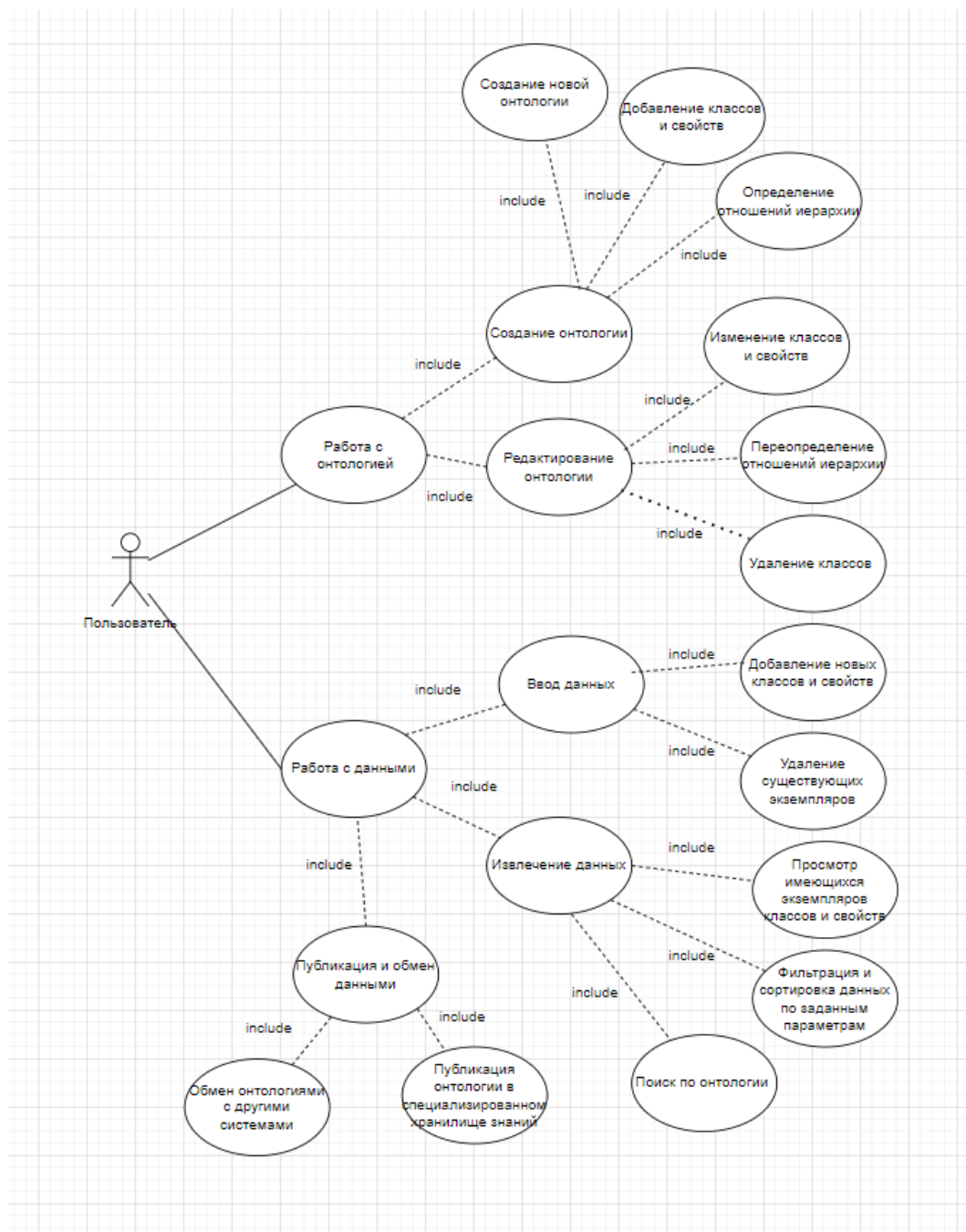


Рисунок 3 – Диаграмма использования редактора онтологий пользователем

Функциональные возможности пользователя:

1. Работа с онтологией

Пользователь может перемещаться рабочему пространству редактора онтологий, создавать, редактировать и удалять онтологии.

2. Создание онтологии

Пользователь может создать новую онтологию посредством ввода ее названия и описания.

3. Добавление классов / свойств

Пользователь может добавить новый класс в онтологию, указав его название и описание, добавить суперклассы для нового класса, а также атрибуты или свойства класса.

4. Определение отношений иерархии

Пользователь может добавить отношения между классами и задать для них ограничения.

5. Изменение классов и свойств

Пользователь может редактировать существующие классы и свойства в онтологии, изменяя их атрибуты и характеристики. Класс или свойство онтологии обновляются согласно внесенным изменениям.

6. Переопределение отношений иерархии

Пользователь может переопределять иерархические отношения между классами в онтологии. Иерархические отношения обновляются согласно внесенным изменениям.

7. Удаление классов / свойств / иерархии

Пользователь может удалять классы, свойства и связи между классами.

8. Добавление новых классов и свойств

Пользователь может добавлять новые классы и свойства в онтологию.

9. Удаление существующих экземпляров

Пользователь может удалить существующие экземпляры онтологии.

10. Просмотр имеющихся классов и свойств

Пользователь может посмотреть имеющиеся в системе классы и свойства.

11. Фильтрация и сортировка данных

Пользователь может выбрать данные, указать критерии и отсортировать данные в соответствии с ними.

12. Поиск по онтологии

Пользователь может осуществить поиск по выбранному критерию в онтологии.

13. Публикация онтологии в базе знаний

Пользователь может выполнить публикацию онтологии в базе знаний.

14. Обмен онтологиями с другими системами

Пользователь может выбрать онтологию и передать ее экспортированную в иной формат другой системе.

Диаграмма использования программы администратором представлена на рисунке 4.

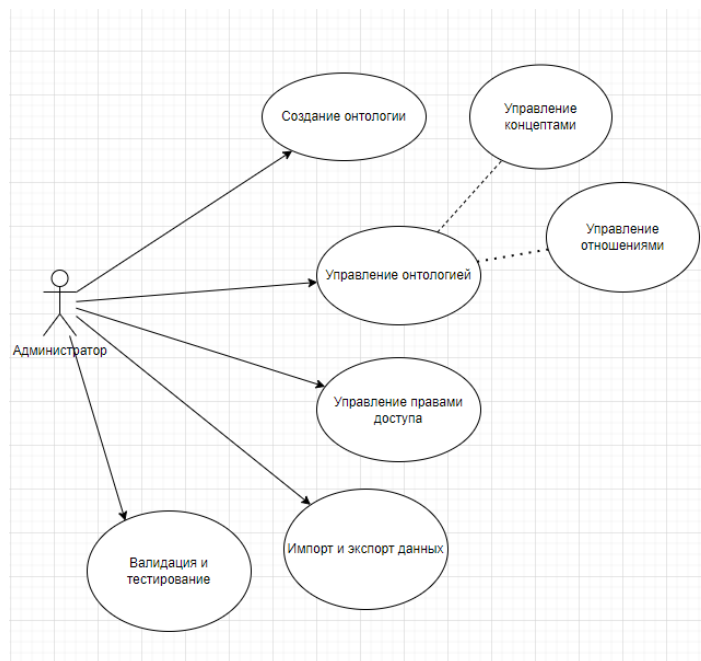


Рисунок 4 – Диаграмма вариантов использования редактора онтологии администратором

Функциональные возможности администратора:

1. Создание онтологии

Администратор может создать новую онтологию, определяя ее имя, цель, область применения и основные концепты.

2. Управление онтологией

Администратор может управлять концептами, отношениями онтологии и доступом к ней.

3. Управление концептами

Администратор может добавлять, редактировать или удалять концепты в онтологии. Концепт — это основная единица знания, которая обозначает сущность или идею.

4. Управление отношениями

Администратор может определять отношения между концептами в онтологии, такие как "является частью", "имеет отношение к" и т.д.

5. Управление правами доступа

Администратор может определять права доступа для других пользователей к онтологии, чтобы регулировать их возможности редактирования или чтения.

6. Импорт и экспорт данных

Администратор может импортировать и экспортировать данные в разные форматы, чтобы обмениваться или использовать онтологию с другими системами

7. Валидация и тестирование

Администратор может проводить проверку онтологии на соответствие определенным правилам или проводить тестирование для обнаружения ошибок или проблем в онтологии.

3.3.5 Функции редактора онтологий

В таблице 3 показаны требования к функционалу редактора онтологий.

Таблица 3 – Требования к функционалу системы

Название:	Создать новую онтологию
Действующие лица:	Пользователь
Описание:	Пользователь входит в меню создания онтологии и, выбирая или создавая нужные классы, сущности и атрибуты, создает онтологию.
Предусловия:	Пользователь зашел в программу как «Пользователь»
Постусловия:	Созданная онтология предметной области

Продолжение Таблицы 3

Основной поток:	Создать новую онтологию 1. Пользователь входит в меню создания онтологии. 2. Выбирает / добавляет классы. 3. Выбирает / добавляет сущности. 4. Выбирает / добавляет атрибуты. 5. Выбирает / добавляет связи. 6. Система проверяет корректность введенных / выбранных данных. 7. Система сохраняет в базе онтологию.
Альтернативные потоки:	1. Пользователь создает для проекта геометрическую модель (ответвление шага 3) 1. Переход к варианту использования «Создать геометрическую модель».
Исключения:	Создание некорректной онтологии (отсутствуют данные) Система извещает, что задана некорректная онтология а. Переход к шагу 4 основного потока.
Включает:	нет
Приоритет:	высокий
Особые требования:	нет
Допущения:	нет
Замечания:	нет
Название:	Загрузить существующую онтологию
Действующие лица:	Пользователь
Описание:	Пользователь входит в меню загрузки онтологии, выбирает нужную и выгружает её.
Предусловия:	Пользователь зашел в программу как «Пользователь».
Постусловия:	Загруженная онтология предметной области.
Основной поток:	Загрузить онтологию предметной области 1. Пользователь входит в меню загрузки онтологии. 2. Выбирает необходимую онтологию. 3. Система загружает онтологию. 4. Система отображает выбранную онтологию.
Альтернативные потоки:	нет
Исключения:	Система извещает, что выбрана некорректная онтология а. Пользователь загружает другую онтологию. б. Переход к шагу 3 основного потока.
Включает:	нет
Приоритет:	средний
Особые требования:	нет
Допущения:	нет
Замечания:	нет
Название:	Добавить атрибут
Действующие лица:	Пользователь
Описание:	Пользователь входит в меню создания атрибута и добавляет новый атрибут.
Предусловия:	Пользователь зашел в программу как «Пользователь»

Окончание Таблицы 3

Постусловия:	Добавленный атрибут
Основной поток:	Добавить атрибут. 1. Пользователь входит в меню создания атрибута. 2. Выбирает сущность, в которую будет входить атрибут. 3. С помощью инструментов системы вводит название атрибута и его связь с сущностью. 4. Система сохраняет атрибут в базу данных.
Альтернативные потоки:	нет
Исключения:	нет
Включает:	нет
Приоритет:	средний
Особые требования:	нет
Допущения:	нет
Замечания:	нет
Название:	Наполнение онтологической базы
Действующие лица:	Администратор
Описание:	Администратор добавляет новые данные в базу данных
Предусловия:	Администратор открыл программу как «Администратор», база данных пуста
Постусловия:	База данных, наполненная классами, сущностями и атрибутами предметной области архитектурной акустики
Основной поток:	Наполнить онтологическую базу 1. Администратор заходит в меню «База данных» 2. Администратор добавляет недостающие данные о методах, сущностях, атрибутах 3. Система обрабатывает полученные данные 4. Система сохраняет полученные данные в базе данных
Альтернативные потоки:	нет
Исключения:	нет
Включает:	нет
Приоритет:	высокий
Особые требования:	нет
Допущения:	нет
Замечания:	нет

3.4 Архитектурно-контекстная диаграмма редактора онтологий

Архитектурно-контекстная диаграмма представляет собой высокоуровневое представление компонентов и связей в системе.

На рисунке 3 представлена архитектурно-контекстная диаграмма нулевого уровня разрабатываемого редактора онтологий. Выделено два вида

пользователей: администратор и инженер-акустик. Их функциональные возможности представлены на рисунке 5.

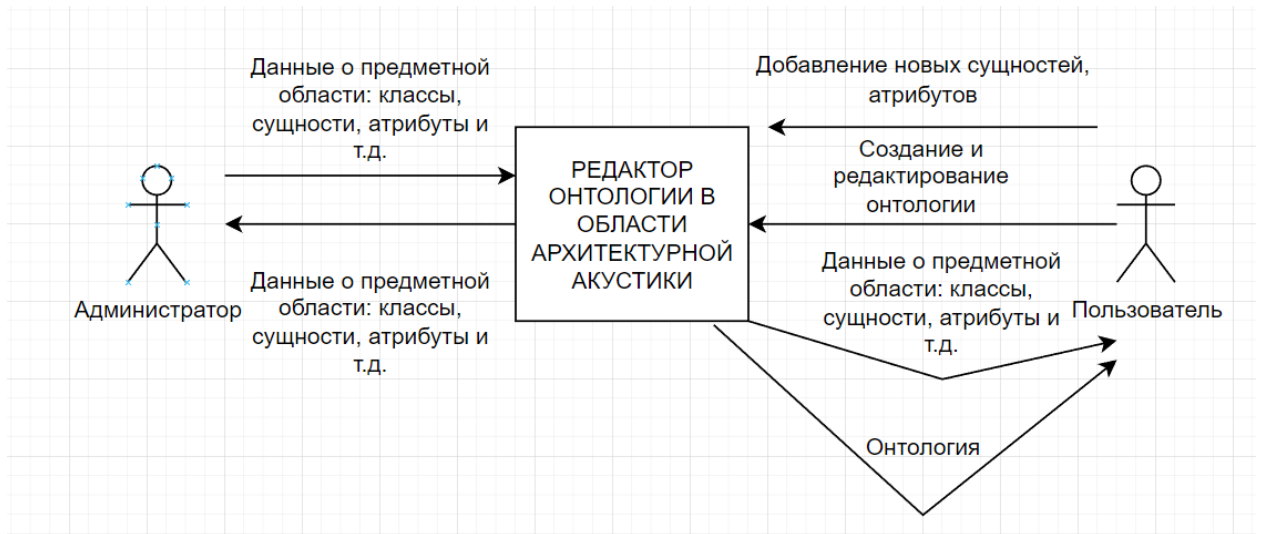


Рисунок 5 - Архитектурно-контекстная диаграмма 0 уровня

На рисунке 6 представлена архитектурно-контекстная диаграмм первого уровня. Она позволяет углубиться в устройство системы. На рисунке показано, что программа состоит из двух подсистем «Серверная часть» и «Клиентская часть», связанных между собой.

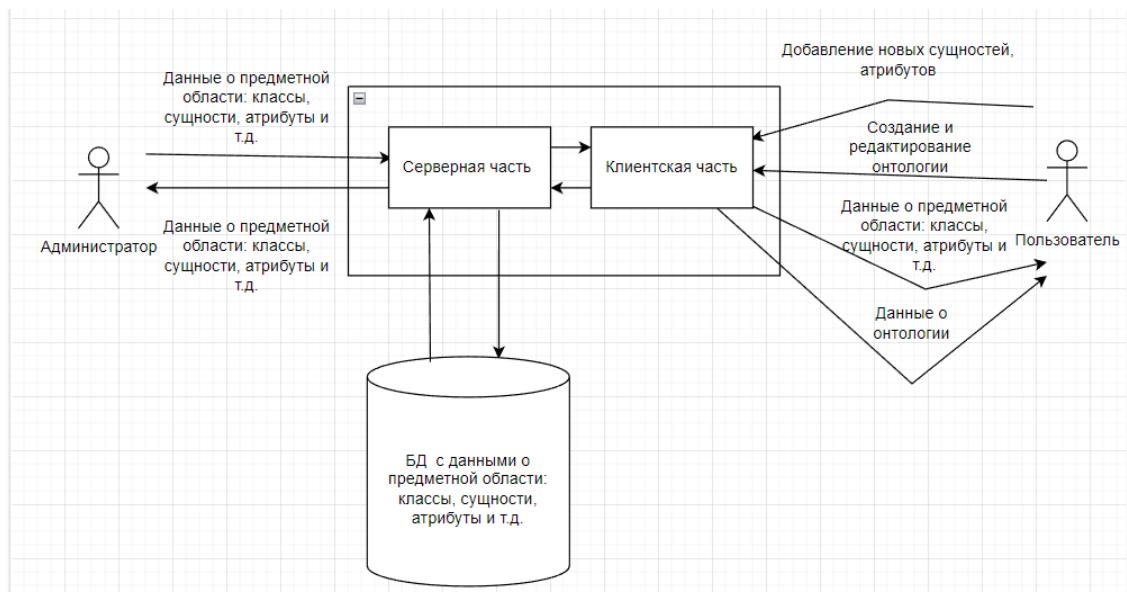


Рисунок 6 - Архитектурно-контекстная диаграмма 1 уровня

На рисунке 7 показана архитектурно-контекстная диаграмма серверной части.

Серверная часть состоит из интерфейса, базы данных и обработчика данных/запросов. Интерфейс отображает доступные для администратора функции: добавление и удаление данных об онтологиях, классах, сущностях и атрибутах.

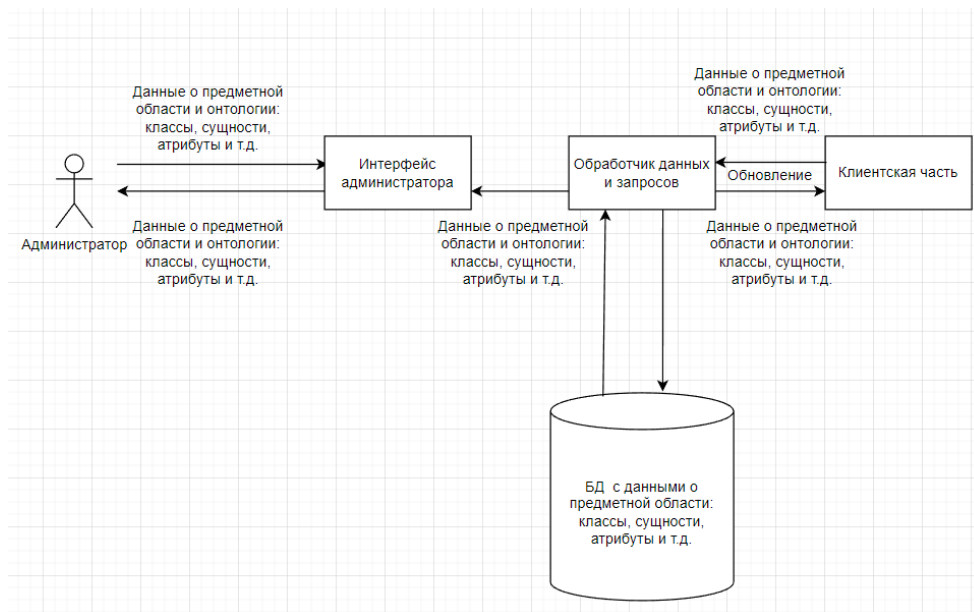


Рисунок 7 – АКД серверной части

3.5 Модульная структура и схема потоков данных

В соответствии с задачами, которые должен решать проектируемый редактор онтологии, разработана модульная структура и схема потоков данных, представленная на рисунке 8.

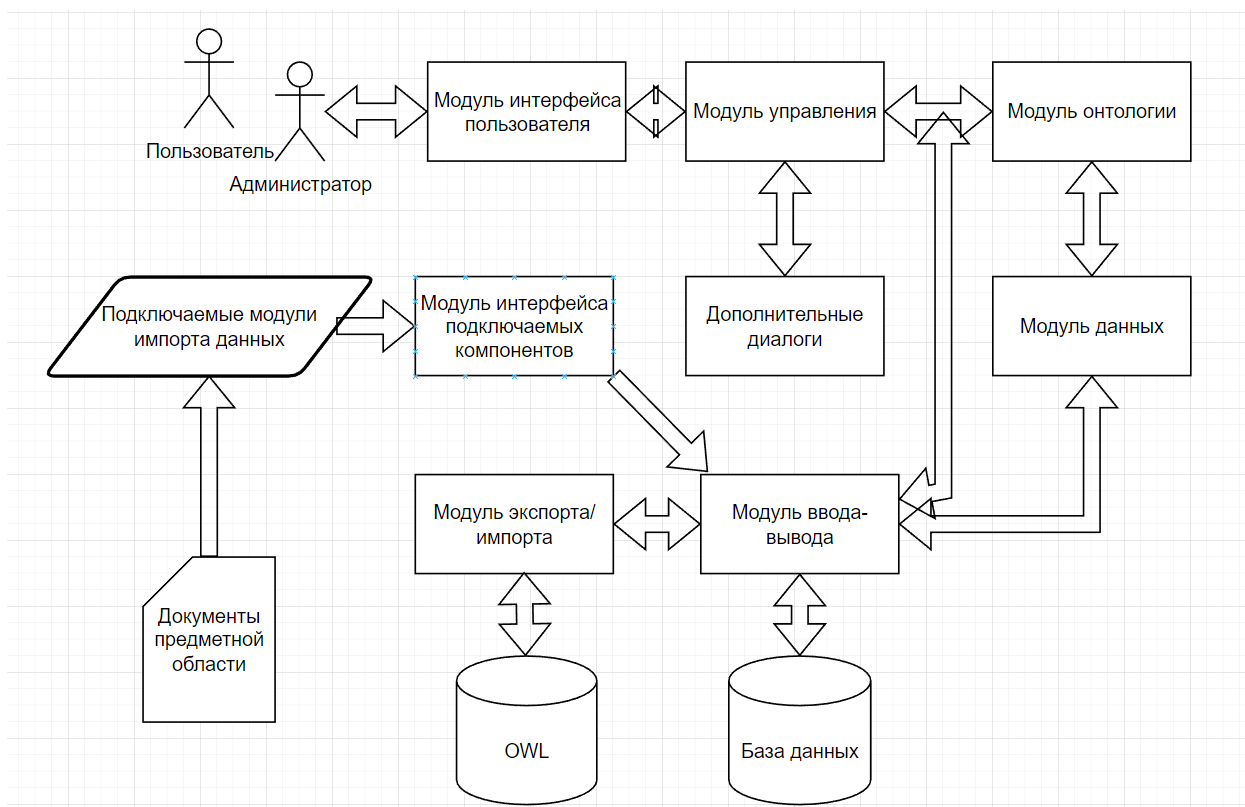


Рисунок 8 – Схема потоков данных

В архитектуру программного продукта входят следующие модули:

1. Модуль онтологии.

Модуль реализует интерфейс программного управления онтологией предметной области и содержит реализацию базовых операций, выполняемых с концептами и экземплярами онтологий.

2. Модуль данных.

Модуль обеспечивает размещение данных онтологии в удобном для программного доступа представлении в оперативной памяти компьютера.

3. Модуль управления.

Модуль содержит реализацию основных алгоритмов управления онтологией.

4. Модуль ввода-вывода.

Модуль обеспечивает загрузку моделей онтологии из файлов базы данных и сохранение в файлах созданной или измененной онтологии.

5. Модуль экспорта/импорта.

Модуль обеспечивает сохранение модели онтологии в широко используемом формате OWL (Ontology Web Language), что позволяет работать с созданной онтологией другим программным продуктам. Также с использованием этого модуля реализуется загрузка онтологии, подготовленных в других редакторах и сохраненных в формате OWL

6. Модуль интерфейса пользователя.

Модуль предоставляет пользователю возможности управления программой посредством меню и кнопок, размещенных в панели программы, отображает информацию о структуре онтологии в панели программы и обеспечивает пользователям возможности просмотра и редактирования онтологии.

7. Модуль дополнительных диалогов.

Модуль реализует вспомогательные диалоговые окна, которые отображаются для выполнения ряда операций с онтологией.

8. Модуль интерфейса подключаемых компонентов.

Модуль реализует программный интерфейс, обеспечивающий импорт в текущую онтологию данных об экземплярах и сущностях предметной области из внешних подключаемых программных компонентов. Этот компонент обеспечивает программе открытую архитектуру и позволяет реализовать автоматический режим построения онтологии в случае использования дополнительных модулей, формирующих модели экземпляров и сущностей.

Блок-схема обобщенного алгоритма редактора онтологий изображена на рисунке 9.

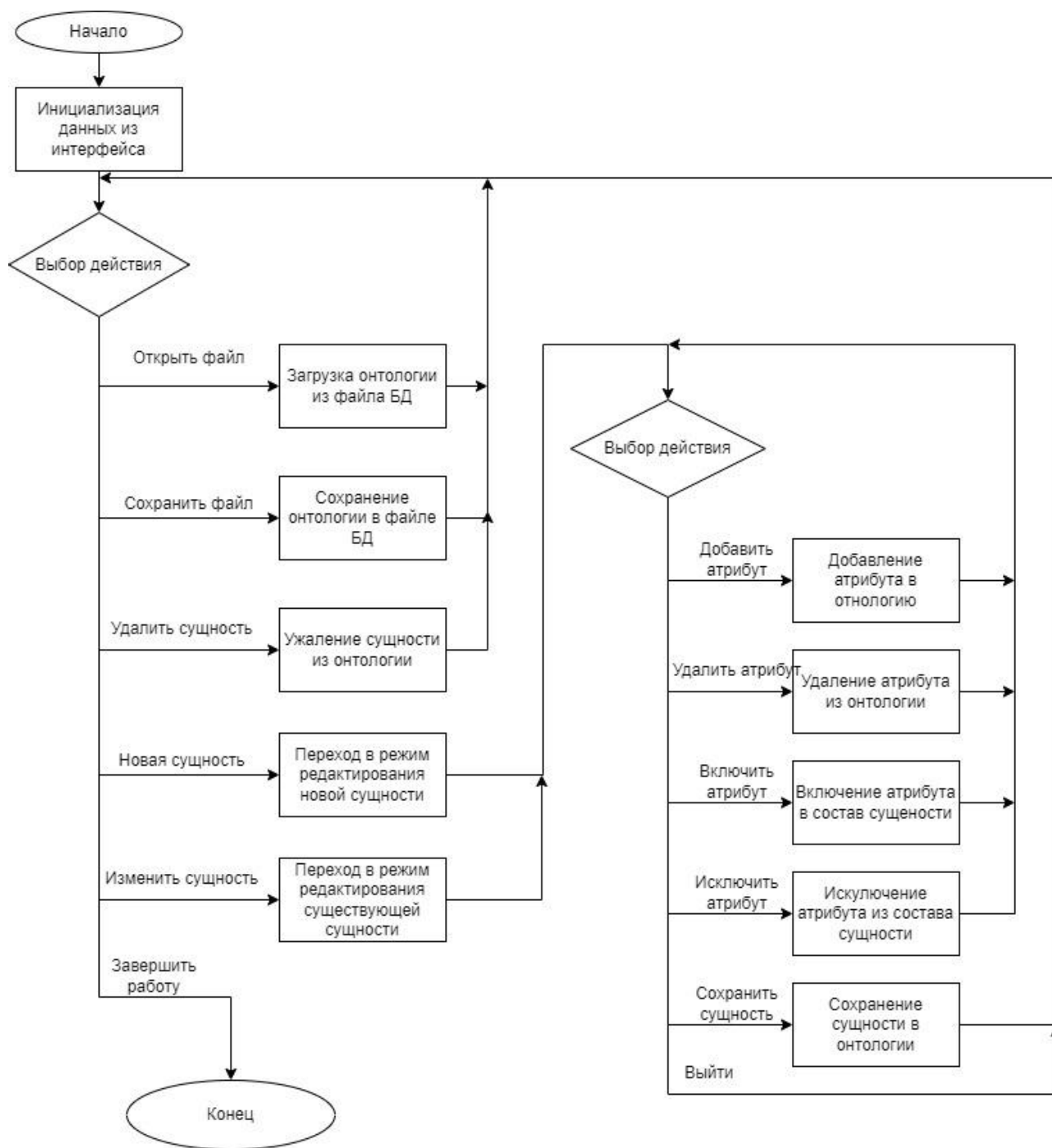


Рисунок 9 – Блок-схема обобщенного алгоритма редактора онтологии

Реализация обобщенного алгоритма является задачей модуля управления и модуля интерфейса пользователя. Остальные модули предназначены для выполнения вспомогательных операций и дополнительных функций редактора онтологии

3.6 Разработка базы данных онтологии предметной области

При разработке системы редактора онтологий необходимо решить вопрос о выборе формы хранения данных онтологии предметной области. В распространенных редакторах онтологий сохранение данных выполняется, как правило, в формате текстовых файлов XML формата.

На основе XML созданы используемые для описания онтологий языки RDFS и OWL. Экспорт и хранение данных в основных редакторах онтологий выполняется на основе перечисленных форматов. Этот подход приемлем при создании учебных и демонстративных прототипов онтологий, но при использовании онтологии в качестве компонента других систем такой способ существенно ограничивает скорость доступа и приводит к избыточности объема данных.

Оптимальным вариантом для решения таких задач будет являться использование реляционных СУБД в качестве хранилища онтологической информации. Основой построения структуры реляционной базы данных будет являться концептуальная модель предметной области.

3.7 Модель базы данных для хранения иерархии классов

На основе решения хранения в базе данных информации об иерархии и атрибутом составе сущностей может быть реализован следующий набор таблиц:

- Таблица «Сущности», содержащая идентификаторы, имена и описания сущностей, включенных в онтологию;
- Таблица «Отношение сущностей», фиксирующая отношения наследования, связывающие концепты онтологии;
- Таблица «Тип атрибута», содержащая идентификаторы, имена и описания типов атрибутов;
- Таблица «Атрибуты», содержащая идентификаторы, имена, типы и описания каждого атрибута;

— Таблица «Поля сущностей», содержащая состав атрибутов каждой из сущностей.

В рассматриваемом решении добавление новых сущностей и классов в онтологию реализуется путем модификации таблицы «Сущности», связи фиксируются в «Отношения сущностей», а состав атрибутов каждой сущности сохранится в таблице «Поля сущностей».

Таблица «Тип атрибута» формируется на этапе создания проекта или заполняется пользователем «Администратор», так как она содержит базовые типы данных, которые используются в предметной области «Архитектурная акустика». При включении пользователем в онтологию сущностей, содержащих новые атрибуты, таблица «Атрибуты» также добавляется новыми атрибутами. Таблица «Поля сущностей» содержит записи лишь о тех атрибутах, которые определены при вводе данной сущности. Наследуемые сущностью поля представлены в таблице «Поля сущностей» как поля родительских сущностей.

В качестве типа данных для первичных ключей таблиц онтологии необходимо выбрать тип индивидуального идентификатора, остальные поля рассматриваемых таблиц будут иметь текстовый тип.

Базовый набор атрибутов онтологии должен включать в себя:

- текстовый тип;
- целочисленный тип;
- вещественный тип;
- логический тип;

Примерная схема базы данных представлена на рисунке 10.

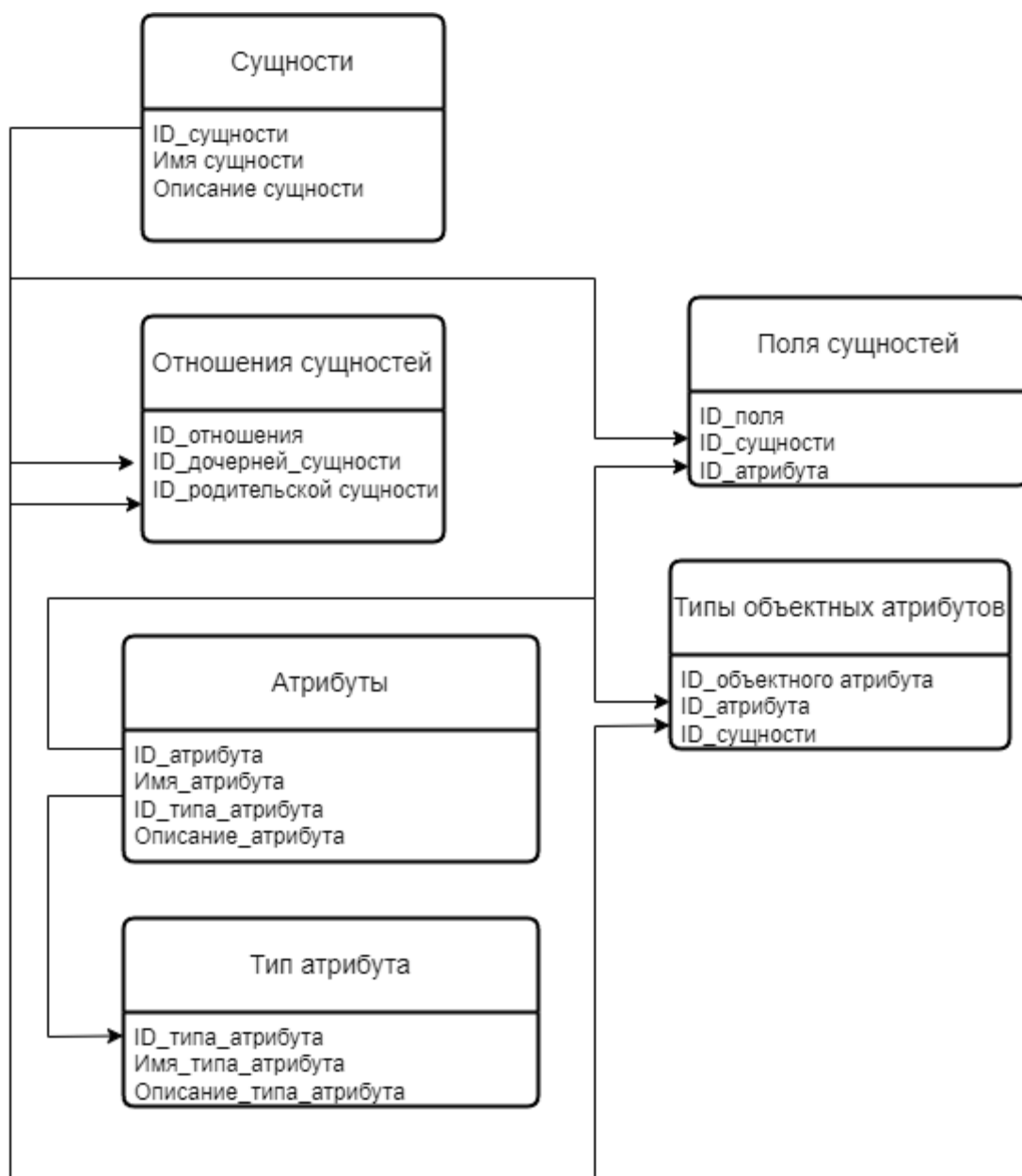


Рисунок 10 – Схема базы данных

Выводы по третьей главе

В результате выполнения главы определена архитектура системы редактора онтологий в области архитектурной акустики, определены функциональные требования, их спецификация, описаны ограничения реализации, а также классы и характеристики пользователей.

Также была представлена архитектурно-контекстная диаграмма редактора, модульная структура и часть модели базы данных онтологии области архитектурной акустики.

Список использованных источников

1. Акустическое оформление и звукоизоляция помещения [Электронный ресурс] // AlbusPro. – Режим доступа: <https://albus-pro.com/архитектурная-акустика/>
2. Архитектурная акустика [Электронный ресурс] // RKFN Sound. – Режим доступа: <https://rkfnsound.ru/about-us/news/2020/architectural-acoustics/>
3. Чусова А. Е., Артемьева И.Л. Автоматизированная система выбора оптимальных методов решения акустических задач на базе онтологии // Тематический выпуск по материалам Всероссийской научной конференции «Научный сервис в сети Интернет». – Том 26. - №6 – 2023г.
4. Симонова Е.В. Моделирование информационных систем: Учеб. Пособие - / Самара: Изд-во Самарского университета, 2022. – 8 с.
5. Архитектурная акустика [Электронный ресурс] // Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Архитектурная_акустика
6. Ефимов А.П. Три взгляда на акустику помещений. В журн. Install-pro. №№ 3-7, 2000 г.
7. Реверберация [Электронный ресурс] // Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Реверберация>.
8. Шишелова Т.И., Шепотько Н.А. СТРОИТЕЛЬНАЯ АКУСТИКА // Успехи современного естествознания. – 2009. – № 8. – С. 13-14;
9. Lauri Savioja, U. Peter Svensson. Overview of geometrical room acoustic techniques // The Journal of the Acoustical Society of America 138, 708 (2015).
10. Романюк Максим Алексеевич, Старченко Ирина Борисовна Теоретические исследования звукового поля в помещении и разработка интерактивной системы аудиоуправления // ИВД. 2011. №1. URL: <https://cyberleninka.ru/article/n/teoreticheskie-issledovaniya-zvukovogo-polya-v-pomeschenii-i-razrabotka-interaktivnoy-sistemy-audioupravleniya>

11. Метод конечно элементов [Электронный ресурс] // Википедия . – Режим доступа: https://ru.wikipedia.org/wiki/метод_конечных_элементов
12. Чусова А.Е. ОНТОЛОГИЯ МЕТОДОВ МОДЕЛИРОВАНИЯ В ОБЛАСТИ АРХИТЕКТУРНОЙ АКУСТИКИ // Прикаспийский журнал: управление и высокие технологии. 2023. №1 (61). URL: <https://cyberleninka.ru/article/n/ontologiya-metodov-modelirovaniya-v-oblasti-arhitekturnoy-akustiki> (дата обращения: 14.06.2024).
13. Вахитов Шакир Яшэрович, Давыдов Денис Андреевич, Алешкин Василий Михайлович Модификация метода мнимых источников с целью моделирования реального процесса первых отражений в помещении // Academia. Архитектура и строительство. 2017. №3. URL: <https://cyberleninka.ru/article/n/modifikatsiya-metoda-mnimyh-istochnikov-s-tselyu-modelirovaniya-realnogo-protssessa-pervyh-otrazheniy-v-pomeschenii> (дата обращения: 14.06.2024).
14. Основные понятия онтологии [Электронный ресурс] // Научно электронная библиотека. – Режим доступа: <https://monographies.ru/en/book/section?id=11245>
15. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. – М., БИНОМ, Интернет-университет информационных технологий-ИНТУИТ.ру, 2009. – 173с.
16. Оскерко В.С. Моделирование базы знаний на основе онтологии / Оскерко В.С., Пунчик З.В // Библиотека БГУИР. – 2020. – с 138-139.
17. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: учебное пособие для вузов. – СПб.: Питер, 2000. – 382с.
18. Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Ч. 1. Существующие подходы к определению понятия «онтология» //НТИ. Сер. 2. – 2001. - №2. – с. 20-27.

19. Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Ч. 2. Компоненты модели //НТИ. Сер. 2. – 2001. - №3. – с. 19-29.
20. Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Ч. 3. Сравнение разных классов моделей онтологий //НТИ. Сер. 2. – 2001. - №4. – с. 10-15.
21. Лутошкина, Н.В. Модели знаний и онтологии: учеб. Пособие /Н.В. Лутошкина ; СибГУ им. М.Ф. Решетнева. – Красноярск, 2021. – 80 с.
22. Минский М. Фреймы для представления знаний. – М.: Энергия, 1979. – 152с.
23. Инструментальные средства проектирования онтологий [Электронный ресурс] // Интуит: Национальный открытый университет. – Режим доступа: <https://intuit.ru/studies/courses/1078/270/lecture/6857>
24. IDEF5 Method Report. – Armstrong Laboratory AL/HRGA Wright-Patterson Air Force Base, Ohio 45433. – 1994, - 187p.
25. Ермалаев К. А. Использование онтологии для управления знаниями предприятия: Исслед. по информ., 2007, выпуск 12. – с. 65.
26. Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий // Электронные библиотеки – Москва: Институт развития информационного общества, т.7 вып.4, 2004. – Электронный журнал, посвященный созданию и использования электронных библиотек.
27. Gruinger M., Obstr L. An Ontology Framework // Ontology Summit – NIST, Gaithersburg, MD April 22-23, 2007.
28. Слета В.Д., Сергеев А.С. – Построение и эволюционное развитие онтологий // Известия ПГПУ им. В.Г. Белинского. 2010 №18 (22). С. 196-200.
29. Natalya F. Noy, Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. // Stanford Medical Informatics Techincak Report SMI-2001-0880. March 2001.

30. Farquhar A., Fikes R., Rice J. The Ontolingua server: A tool for collaborative ontology construction // International Journal of Human-Computer Studies, 46(6), pages 707–728, 1997.
31. Рубашин В. Ш. Онтологии – концептуальные границы, проблемы и решения. Точка зрения разработчика // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2007». М.: Издательский центр РГГУ, 2007. С. 481-485.
32. Буданова, Назигуль. Анализ инструментов для построения онтологии информационно-справочной системы гостиниц и хостелов / Назигуль Буданова. — Текст : непосредственный // Молодой ученый. — 2018. — № 17 (203). — С. 17-19. — URL: <https://moluch.ru/archive/203/49675/> (дата обращения: 18.06.2024).
33. Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологии // Сборник института программных систем НАН Украины. — 2004. — Том 7. — Выпуск 4. — URL: <https://rdl-journal.ru/article/download/254/253>
34. Noy N., Musen M. SMART: Automated Support for Ontology Merging and Alignment // Stanford Medical Informatics, Stanford Univ., 1999.
35. Farguhar A., Fikes R., Rice J. The Ontolingua server: A tool for collaborative ontology construction // International Journal of Human-Computer Studies, 46(6), pages 707-728, 1997.
36. Musen, M. Domain Ontologies in Software Engineering: Use of Prot?g? with the EON Architecture // Methods of Inform. in Medicine, pages 540-550, 1998.
37. Рубашкин В.Ш., Лахути Д.Г. Онтология: от натурфилософии к научному мировоззрению и инженерии знаний // Вопросы философии №1, 2005. С. 64-81.
38. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.

39. Bechhofer S., Horrocks I., Goble C., Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web // Joint German/Austrian conf. on Artificial Intelligence (KI'01). Lecture Notes in Artificial Intelligence LNAI 2174, Springer-Verlag, Berlin, pages.396-408, 2001.
40. Domingue J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web // Proc. of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, 1998.
41. Motta E. Reusable Components for Knowledge Modelling // Ph.D. Thesis. The Open University, 1997.
42. MacGregor R. Inside the LOOM classifier // SIGART bulletin, Vol.3, No.2, pages 70-76, 1991.
43. Fernandez M, Gomez-Perez A., Pazos J. A Building a Chemical Ontology Using Methontology and the Ontology Design Environment // IEEE Intelligent Systems, Jan./Feb. pages 37-46, 1999.
44. Kifer M., Lausen G., Wu J. Logical Foundations of Object-Oriented and Frame-Based Languages // Journal of the ACM, 1995
45. Батищев С. В., Искварина Т. В., Скобелев П. О. Методы и средства построения онтологий для интеллектуализации сети Интернет // Известия Самарского научного центра РАН. 2002. №1. URL: <https://cyberleninka.ru/article/n/metody-i-sredstva-postroeniya-ontologiy-dlya-intellektualizatsii-seti-internet> (дата обращения: 18.06.2024).
46. Бардина Наталья Владимировна Диаграмма классов и модель «Сущность–связь» как логические модели информационной системы // Модели, системы, сети в экономике, технике, природе и обществе. 2012. №2 (3). URL: <https://cyberleninka.ru/article/n/diagramma-klassov-i-model-suschnost-svyaz-kak-logicheskie-modeli-informatsionnoy-sistemy> (дата обращения: 19.06.2024).
47. Антонов И.В. Актуальные проблемы машинной обработки текстов на естественном языке // Фундаментальные исследования и инновации в технических университетах: Материалы XI Всероссийской конференции по

проблемами науки и высшей школы. – СПб: Изд-во Политехнического ун-та, 2007. – с. 161-162.

48. Шульга Татьяна Эриковна, Храмов Дмитрий Эдуардович ОНТОЛОГИЯ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2023. №2. URL: <https://cyberleninka.ru/article/n/ontologiya-zhiznennogo-tsikla-razrabotki-programmnogo-obespecheniya> (дата обращения: 23.01.2024);

49. Чусова А.Е. ОНТОЛОГИЯ МЕТОДОВ МОДЕЛИРОВАНИЯ В ОБЛАСТИ АРХИТЕКТУРНОЙ АКУСТИКИ // Прикаспийский журнал: управление и высокие технологии. 2023. №1 (61). URL: <https://cyberleninka.ru/article/n/ontologiya-metodov-modelirovaniya-v-oblasti-arhitekturnoy-akustiki> (дата обращения: 23.01.2024);

50. Видия А.В., Дородных Н.О., Юрин А.Ю. ПОДХОД К СОЗДАНИЮ ОНТОЛОГИЙ НА ОСНОВЕ ЭЛЕКТРОННЫХ ТАБЛИЦ С ПРОИЗВОЛЬНОЙ СТРУКТУРОЙ // Онтология проектирования. 2021. №2 (40). URL: <https://cyberleninka.ru/article/n/podhod-k-sozdaniyu-ontologiy-na-osnove-elektronnyh-tablits-s-proizvolnoy-strukturoy> (дата обращения: 23.01.2024).