**Введение**

В настоящее время в связи с развитием компьютерных технологий, появилась потребность в программных системах, автоматизирующих деятельность в таких сложных предметных областях, как химия.

Одним из классов программных систем являются системы, основанные на знаниях, отличительная особенность которых состоит в том, что знания, необходимые для выполнения профессиональной деятельности, отделены в этих системах от программ для решения прикладных задач.

В рамках диссертационной работы д.т.н., профессора кафедры ПО ЭВМ ДВФУ – Артемьевой Ирины Леонидовны «Многоуровневые модели сложно-структурированных предметных областей и их использования при разработке систем, основанных на знаниях» автором были разработаны теоретические положения и получено практическое решение проблемы создания расширяемых специализированных оболочек систем, основанных на знаниях, для сложно-структурированных предметных областей.

На настоящий момент имеется необходимость в создании системы, выполняющей те же функции, что и специализированная оболочка интеллектуальных систем для химии, но в виде веб-приложения.

**Цель дипломной работы**

Целью дипломной работы является разработка компонентов программной системы, которая позволяет создавать и редактировать метаонтологии и онтологии предметных областей в области химии, формировать и редактировать базу знаний для каждой созданной онтологии.

**Содержание Дипломной работы**

**\*\*\*\*\*\*\***

**Глава 1. Онтологии и программные системы, позволяющие их редактировать, обзор литературы**

В данной главе рассматриваются существующие онтологии и модели онтологий предметной области «Химия». Также в обзоре рассматриваются программные решения, дающие возможность описывать, редактировать, удалять и использовать онтологии предметных областей в различных целях.

**Глава 3. Техническая документация**

В данной главе представлены: требования к системе (функциональные требования, требования к входным данным, требования к выходным данным, требования к интерфейсу, требования к интерфейсу, требования к надежности, требования к среде), архитектурно-контекстная диаграмма, внешние спецификации (спецификация входных и выходных данных, сценарий диалога с пользователем, спецификация функций), архитектура программной системы (проектное решение), внутренние спецификации.

**3.1 Характеристики пользователей**

В данном разделе описываются характеристики всех групп пользователей программной системы.

**3.1.1 Характеристика эксперта**

* Знает русский язык.
* Понимает сообщения на русском языке.
* Умеет работать на компьютере в операционных системах семейства Windows, MacOS, Linux, Unix (умеет работать с контроллером типа “Мышь” и клавиатурой, включать и выключать компьютер, запускать приложения и завершать работу с ними).
* Умеет работать с хотя бы с одним из следующих интернет-браузеров Internet-Explorer, Mozilla Firefox, Google Chrome, Safari, Opera (запуск браузера, переход по заданному web-адресу).
* Является экспертом в том разделе химии, онтология которого доступна и для которой он может задать знания.

**3.1.2 Характеристика инженера знаний**

* Знает русский язык.
* Понимает сообщения на русском языке.
* Умеет работать на компьютере в операционных системах семейства Windows, MacOS, Linux, Unix (умеет работать с контроллером типа “Мышь” и клавиатурой, включать и выключать компьютер, запускать приложения и завершать работу с ними).
* Умеет работать с хотя бы с одним из следующих интернет-браузеров Internet-Explorer, Mozilla Firefox, Google Chrome, Safari, Opera (запуск браузера, переход по заданному web-адресу).
* Знает, что такое метаонтология предметных областей.
* Умеет задавайть метаонтологии в области химии (умеет выделить сущности метаонтологии, определить компоненты сущностей, задать общие свойства сущностей и их компонентов, собственные свойства сущностей, свойства указанных типов, совместные свойства сущностей, совместные свойства сущностей, свойства компонентов нескольких типов, общие свойства процесса и его компонентов, свойства компонентов сущночти процесса и общие свойства процесса).
* Умеет определять взаимосвязи между метаонтологиями и онтологиями предметных областей в области химии.
* Знает, что такое онтологии предметных областей.
* Умеет задавать онтологии в области химии на основе метаонтологий в этой области (задавать термины-функции, определять их аргументы, результаты и диапазоны значений результатов и аргументов).
* Умеет определять вспомогательные термины для онтологий (если это необходимо).
* Знает , что такое кортеж значений.
* Умеет задать кортеж значений.

**3.2 Требования к программной системе**

**3.2.1 Требования к редактору метаонтологий**

**3.2.1.1 Функциональные требования**

Редактор метаонтологии должен:

3.2.1.1.1 обеспечить поэтапное создание метаонтологии;

3.2.1.1.2 позволять задавать сущности метаонтологии и их типы в процессе создания;

3.2.1.1.3 позволять использовать сущности других метаонтологий, изменять его в процессе создания и запоминать эту связь;

3.2.1.1.4 позволять редактировать список сущностей в процессе добавления метаонтологии и учитывать все измененния на всех этапах создания метаонтологии;

3.2.1.1.5 позволять изменять название метаонтологии в процессе ее создания;

3.2.1.1.6 позволять задавать компоненты сущности, редактировать их в процессе создания метаонтологии и учитывать все изменения на каждом из этапов создания;

3.2.1.1.7 позволять удалять общие свойства сущностей и изменять их формулировки в процессе создания;

3.2.1.1.8 позволять удалять собственные свойства сущностей и изменять их формулировки в процессе создания;

3.2.1.1.9 позволять удалять свойства указанных типов и изменять их формулировки в процессе создания;

3.2.1.1.10 позволять задавать совместные свойства сущностей и удалять их в процессе создания;

3.2.1.1.11 позволять задавать свойства компонентов нескольких типов и удалять их в процессе создания;

3.2.1.1.12 позволять определять уровень рассмотрения химического процесса (сущности процесса и их типы, состав компонентов процесса) и изменять его в процессе создания;

3.2.1.1.13 позволять определять общие свойства процесса и его компонент, удалять их и изменять формулировки в процессе создания;

3.2.1.1.14 позволять определятья общие свойства компонентов сущностей процесса и изменять их в процессе создания;

3.2.1.1.15 позволять определятья общие свойства процесса и его схему, а также удалять их в процессе создания;

3.2.1.1.16 позволять редактировать созданные метаонтологии;

3.2.1.1.17 позволять изменять название метаонтологии и учитывать это в созданных онтологиях и метаонтологиях при редактировании метаонтологии;

3.2.1.1.18 позволять изменять список используемых метаонтологий и учитывать это во всех свойствах редактируемой метаонтологии, во всех свойствах метаонтологий и онтологий, которые использовали редактируемую метаонтологию при их создании при редактировании метаонтологии;

3.2.1.1.19 позволять изменять список сущностей метаонтологии и учитывать эти изменения во всех свойствах редактируемой метаонтологии, во всех свойствах метаонтологий и онтологий, которые использовалиредактируемую метаонтологию при их создании при редактировании метаонтологии;

3.2.1.1.20 позволять изменять компоненты сущностей метаонтологии и учитывать эти изменения во всех свойствах редактируемой метаонтологии, во всех свойствах метаонтологиий и онтологий, которые использовали редактируемую метаонтологию при их создании при редактировании метаонтологии;

3.2.1.1.21 позволять удалять общие свойства сущностей и их компонентов и изменять их формулировки при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.22 позволять удалять собственные свойства сущностей и изменять их формулировки при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, сощданных на основе редактируемой метаонтолоогии;

3.2.1.1.23 позволять удалять свойства указанных типов и изменять их формулировки при редактировании метаонтологии, а также учитывать внесенные измененния в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.24 позволять изменять список совместных свойств сущностей при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.25 позволять изменять список свойств компонентов нескольких типов при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.26 позволять изменять уровень рассмотрения химического процесса (сущности процесса и их типы, состав компонентов процесса) при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.27 позволять изменять формулировки свойств компонентов сущностей процесса при редактировании метаонтологии, а также учитывать внесенные изменения в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.28 позволять изменять список общих свойств процесса при редактировании метаонтологии, а также учитывать внесенные измененния в онтологиях, созданных на основе редактируемой метаонтологии;

3.2.1.1.29 позволять удалять метаонтологии и автоматически удалять созданные на их основе онтологии;

3.2.1.1.30 ввод всех данных должен осуществляться с помощью мыши или и клавиатуры;

3.2.1.1.31 осуществлять контроль ввода входных данных и выдавать диагностические сообщение в случае их неверного ввода, позволяя их отредактировать;

3.2.1.1.32 отображать введенные входные данные на дисплее;

3.2.1.1.33 предупреждать пользователя об опасных действиях.

**3.2.1.2 Требования к входным данным**

3.2.1.2.1 Название метаонтологии – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов.

3.2.1.2.2 Используемые метаонтологии – должны выбираться в списке доступных.

3.2.1.2.3 Название сущности – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов или выбираться из списка, содержащего список сущностей используемых в метаонтологии.

3.2.1.2.4 Тип сущности должен выбираться из списка, содержащего элементы {}N, {}R, {}I,{}L.

3.2.1.2.5 Компоненты сущностей – должны выбираться из списка доступных для каждой сущности.

3.2.1.2.6 Общее свойство сущности и ее компонента – должно формироваться автоматически. Если требуется его изменить, то название должно воодиться в текстовое поле, только на русском языке и не превышать 250 символовв длину.

3.2.1.2.7 Собственное свойство сущности – должно формироваться автоматически. Если тербуется его изменить, то название должно вводиться в текстовое поле, только на русском языке и не превышать 250 символов в длину.

3.2.1.2.8 Свойство указанного типа – должно формироваться автоматически.

Если требуется его изменить, то название должно вводиться в текстовое поле, только на русском языке и не превышать 250 символовв длину.

3.2.1.2.9 Совместное свойство сущностей – название свойства должно вводиться в текстовое поле, только на русском языке и не превышать 50 символов в длину. Компоненты свойства должны выбираться из списка доступных сущностей.

3.2.1.2.10 Свойство компонентов нескольких типов – название свойства должно вводиться в текстовое поле, только на русском языке и не превышать 50 символов в длину. Сущность для свойства должна выбираться из списка доступных сущностей. Комноненты сущности должны выбираться из списка допустимы компонентов сущностей.

3.2.1.2.11 Уровень рассмотрения химического процеса (сущности процесса и их типа) – должны выбираться в списке.

3.2.1.2.12 Общее свойство процесса и его компонент – должно формироваться автоматически. Если требуется его изменить, то название должно вводиться в текстовое поле, только на русском языке и не превышать 250 символов в длину.

3.2.1.2.13 Уровень рассмотрения химического процесса (состав компонентов процесса) – должны выбираться в списке.

3.2.1.2.14 Свойство компонента сущности процесса – должно формироваться автоматически. Если требуется его изменить, то название должно вводиться в текстовое поле, только на русском языке и не превышать 250 символов в длину.

3.2.1.2.15 Общее свойства процесса – Название свойства должно вводиться в текстовое поле, только на русском языке и не превышать 50 символов в длину. Сушность процесса должна выбираться из списка доступных. Компоненты сущности процесса должны выбираться из списков, причем для каждоого компонента должна быть возможность выбора доступных для него компонентов.

**3.2.1.3 Требования к выходным данным**

3.2.1.3.1 Каждая созданная метаонтология должна формироваться в базу данных с названием созданной метаонтологии в отдельной папке, предназначенных для хранения структур метаонтологий.

3.2.1.3.2 Список используемых метаонтологий должен записываться в отдельную таблицу бзы данных.

3.2.1.3.3 Список сущностей должен записываться в отдельную таблицу базы данных.

3.2.1.3.4 Список компонентов сущностей должен записываться в отдельную таблицу базы данных.

3.2.1.3.5 Список общих свойств сущностей и их компонентов должен записываться в отдельную таблицу базы данных.

3.2.1.3.6 Список собственных свойств сущностей должен записываться в отдельную таблицу базы данных.

3.2.1.3.7 Список свойств указанных типов должен записываться в отдельную таблицу базы данных.

3.2.1.3.8 Список совместных свойств сущностей должен записываться в отдельную таблицу базы данных.

3.2.1.3.9 Список свойств компонентов нескольких типов должен записываться в отдельную таблицу базы данных.

3.2.1.3.10 Список сущностей процесса и их типы должны записываться в отдельную таблицу базы данных.

3.2.1.3.11 Список компонентов процесса должен записываться в отдельную таблицу базы данных.

3.2.1.3.12 Список компонентов процесса должен записываться в отдельную таблицу базы данных.

3.2.1.3.13 Список свойств компонентов сущностей процесса должен записываться в отдельную таблицу базы данных.

3.2.1.3.14 Список общих свойств процесса должен записываться в отдельную таблицу базы данных.

3.2.1.3.15 При завершении создания метаонтологии должна формироваться база даных с названием этой метаонтологии в отдельной папке. База данных должна содержать аблицы с названиями сущностей метаонтологии имеющих тип отличный от {}L.

**3.2.2 Требования к редактору онтологий**

**3.2.2.1 Функциональные требования**

Редактор онтологий должен:

3.2.2.1.1 обеспечить поэтапное создание онтологии;

3.2.2.1.2 позволять создавать онтологии на основе метаонтологий;

3.2.2.1.3 позволять использовать онтологии, созданные на основе выбранной метаонтологии и сохранять эту связь;

3.2.2.1.4 позволять использовать термины-функции онтологий, которые выбраны качестве используемых;

3.2.2.1.5 позволять задавать термины-функции для создаваемой онтологии;

3.2.2.1.6 позволять создавать кортежи значений с именем и использовать в дальнейшем в качестве результатов терминов функций;

3.2.2.1.7 позволять задавать термины-функции, результатами которых является кортеж значений;

3.2.2.1.8 позволять создавать вспомогательные термины и использовать в дальнейшем в качестве результатов терминов функций;

3.2.2.1.9 позволять использовать кортежи значений и вспомогательные термины используемых онтологий;

3.2.2.1.10 позволять изменять список терминов-функций в процессе создания онтологии;

3.2.2.1.11 позволять изменять список используемых онтологий в процессе создания онтологии и учитывать это на всех этапах при создании онтологии;

3.2.2.1.12 позволять изменять метаонтологию, на которой основывается создаваемая онтология, и учитывать эти изменения на всех этапах создания онтологии;

3.2.2.1.13 формировать структуру базы знаний при завершении создания онтологии;

3.2.2.1.14 позволять изменять название онтологии в процессе создания;

3.2.2.1.15 позволять редактировать созданные онтологии;

3.2.2.1.16 позволять изменять название созданных онтологий и учитывать это во всех онтологиях, в которых она используется при редактировании онтологии;

3.2.2.1.17 позволять изменять список используемых онтологий и учитывать эти изменения во всех свойствах онтологий и онтологиях, использующих редактируемую онтологию при редактировании онтологии;

3.2.2.1.18 позволять изменять список функций-терминов и учитывать эти изменения во всех онтологиях, использующих редактируемую онтологию при редактировании онтологии;

3.2.2.1.19 позволять создавать новые кортежи с именем и вспомогательные термины и добавлять их в онтологии которые используют редактируемую онтологию при редактировании онтологии.

**3.2.2.2 Требования к входным данным**

3.2.2.2.1 Название онтологии – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов.

3.2.2.2.2 Используемые онтологии – должны выбираться в списке доступных.

3.2.2.2.3 Используемая метаонтология – должна выбираться из созданных метаонтологий.

3.2.2.2.4 Название термина-функции – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов в длину.

3.2.2.2.5 Результат функции - должен выбираться из списка содержащего элементы R, I, L, N, {}R, {}I, {}L, {}N, а также сущности используемой метаонтологии и сущности используемой используемой метаонтологии со значком «{}», обозначающим подмножество.

3.2.2.2.6 Минимальное значение для результата типа I – целые числа от -1 000 000 до 1 000 000. Должно вводиться в текстовое поле.

3.2.2.2.7 Максимальное значение для результата типа I – целые числа от -1 000 000 до 1 000 000. Должно вводиться в текстовое поле.

3.2.2.2.8 Минимальное значение для результата типа R – вещественные числа от -1 000 000 до 1 000 000. Должно вводиться в текстовое поле.

3.2.2.2.9 Максимальное значение для результата типа R – вещественные числа от -1 000 000 до 1 000 000. Должно вводиться в текстовое поле.

3.2.2.2.10 Метатермин – должен выбираться из списка доступных метатерминов.

3.2.2.2.11 Имя кортежа значений – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов.

3.2.2.2.12 Элемент кортежа – должен выбираться из списка доступных элементов.

3.2.2.2.13 Название вспомогательного термина-множества – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов.

3.2.2.2.14 Название элемента вспомогательного термина-множества – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов.

**3.2.2.3 Требования к выходным данным**

3.2.2.3.1 Каждая созданная онтология должна формироваться в базу данных с названием созданной онтологии в отдельной папке, предназначенной для хранения структур онтологий.

3.2.2.3.2 Название метаонтологии, должно записываться в отдельную таблицу базы данных.

3.2.2.3.3 Список использованных онтологий должен записываться в отдельную таблицу базы данных.

3.2.2.3.4 Список созданных терминов-функций должен записываться в отдельную таблицу базы данных.

3.2.2.3.5 Список кортежей должен записываться в отдельную таблицу базы данных.

3.2.2.3.6 Список вспомогательных терминов-множеств должен записываться в отдельную таблицу базы данных.

3.2.2.3.7 Для каждой онтологии должна формироваться база знаний, имеющая такое же название, как и созданная онтология. Каждая таблица этой базы должна называться именами терминов-функций онтологий.

**3.2.3 Требования к редактору знаний**

**3.2.3.1 Функциональные требования**

Редактор знаний должен:

3.2.3.1.1 позволять задавать возможные значения терминов для любой созданной метаонтологии;

3.2.3.1.2 позволять задавать значения аргументов и результатов терминов-функций онтологий, формирующих базу знаний для любой созданной онтологии;

3.2.3.1.3 позволять изменять значения аргументов и результатов терминов-функций онтологий, формирующих базу знаний для любой созданной онтологии;

3.2.3.1.4 позволять изменять возможные значения терминов для любой созданной метаонтологии.

**3.2.3.2 Требования к входным данным**

3.2.3.2.1 Онтология – должна выбираться из списка созданных онтологий.

3.2.3.2.2 Термин онтологии – должен выбираться в списке доступны терминов.

3.2.3.2.3 Значение термина типа I – от – 1 000 000 до 1 000 000. Должно вводиться в текстовом поле при заполнении возможных значений терминов метаонтологии. Должно выбираться из списка возможных значений при заполнении базы знаний онтологии.

3.2.3.2.4 Значение термина типа R – от – 1 000 000 до 1 000 000. Должно вводиться в текстовом поле при заполнении возможных значений терминов метаонтологии. Должно выбираться из списка возможных значений при заполнении базы знаний онтологии.

3.2.3.2.5 Значение термина типа N – должно вводиться в текстовое поле только на русском языке и длина не должна превышать 50 символов при заполнении возможных значений терминов метаонтологии. Должно выбираться из списка возможных значений при заполнении базы знаний онтологии.

3.2.3.2.6 Значение термина типа L – должно выбираться из двух возможных значений «Ложь» и «Истина».

3.2.3.2.7 Значение термина типа {}I – должно создаваться в виде списка значений типа I и каждое значение в этом списке должно выбираться из списка возможных значений.

3.2.3.2.8 Значение термина типа {}R – должно создаваться в виде списка значений типа R и каждое значение в этом списке должно выбираться из списка возможных значений.

3.2.3.2.9 Значение термина типа {}L – должно создаваться в виде списка значений типа L и каждое значение в этом списе должно быть либо «Ложь», либо «Истина».

3.2.3.2.10 Значение термина типа {}N – должно создаваться в виде списка значений типа N и каждое значение в этом списке должно выбираться из списка возможных значений.

**3.2.3.3 Требования к выходным данным**

3.2.3.3.1 Введенные возможные значения любого типа терминов метаонтологий должны сохраняться в базе данных с названием этой метаонтологии и таблице с названием термина, чьи возможные значения были сформированы.

3.2.3.3.2 Введенные значения термина онтологии должны сохраняться в базе данных с названием этой онтологии в таблице с названием термина, чьи значения были сформированы.

**3.2.4 Требования к интерфейсу системы**

3.2.4.1 Интерфейс системы должен обеспечивать удобный ввод данных.

3.2.4.2 Все вводимые сообщения должны быть написаны на русском языке.

3.2.4.3 Интерфейс системы должен быть интуитивно понятным. В любой момент работы с программой пользователю должно быть понятно, что делать дальше.

3.2.4.4 Все сообщения, выводимые пользователю должны быть граммотными.

3.2.4.5 Общение с пользователем должно осуществялться посредством диалоговых окон.

3.2.4.6 Интерфейс должен быть дружелюбным.

3.2.4.7 Все выводимые сообщения должны быть диагностическими, т.е. указывать пользователю на совершенную им ошибку.

3.2.4.8 Процесс создания метаонтологии и онтологии должен осуществляться в режиме мастера.

**3.2.5 Требования к архитектурной среде**

**3.2.6 Требования к надежности**

**3.3 Проект базы данных**

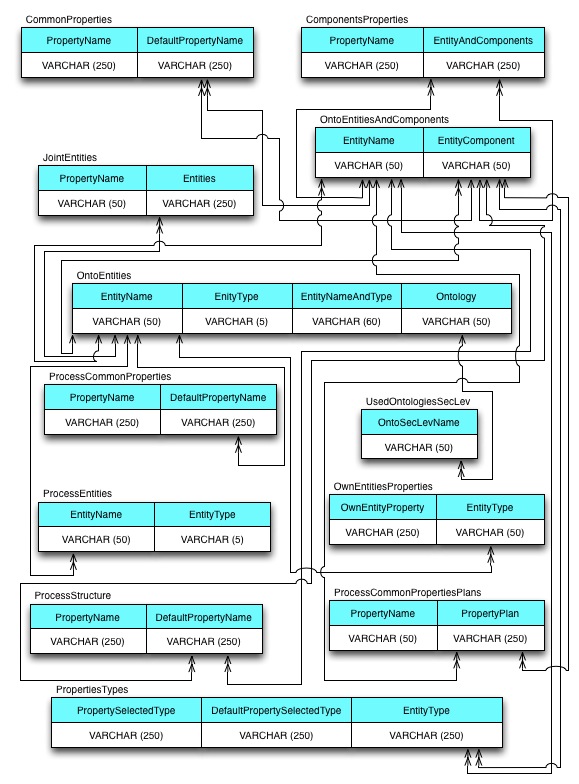


Рисунок \*\*\*. Проект базы данных, содержащей структуру метаонтологии.

**Описание структуры базы данных**

Таблица CommonProperties предназначена для хранения общих свойств сущностей и их компонентов.

Таблица \*\*\*. CommonProperties.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (250) | Хранит имя общего свойства сущности и ее компоненты. |
| DefaultPropertyName | VARCHAR (250) | Хранит имя общего свойства сущности и ее компоненты, заданного по умолчанию. |

Таблица ComponentProperties предназначена для хранения свойств компонентов нескольких типов.

Таблица \*\*\*. ComponentsProperties.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (250) | Хранит имя свойства компонентов нескольких типов. |
| EntityAndComponents | VARCHAR (250) | Хранит имена сущности и ее компонентов. |

Таблица JointEntitiesPropertios предназначена для хранения совместных свойств сущностей.

Таблица \*\*\*. JointEntitiesProperties.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (50) | Хранит имя совместного свойства сущностей. |
| DefaultPropertyName | VARCHAR (250) | Хранит имена сущностей. |

Таблица OntoEntitiesAndComponents предназначена для хранения компонентов сущностей.

Таблица \*\*\*. OntoEntitiesAndComponents.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| EntityName | VARCHAR (50) | Хранит имя сущности. |
| EntityComponent | VARCHAR (50) | Хранит компонент сущности. |

Таблица OntoEntities предназначена для хранения сущностей и их типов.

Таблица \*\*\*. OntoEntites.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| EntityName | VARCHAR (2=50) | Хранит имя сущности. |
| EntityType | VARCHAR (5) | Хранит тип сущности. |
| EntityNameAndType | VARCHAR (60) | Хранит имя и тип сущности. |
| Ontology | VARCHAR (50) | Хранит название онтологии, к которой относится. |

Таблица UsedOntologiesSecLev предназначена для хранения названий используемых метаонтологий.

Таблица \*\*\*. UsedOntologiesSecLev.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| OntoLevName | VARCHAR (50) | Хранит название метаонтологии. |

Таблица OwnEntitiesProperties предназначена для хранения собственных свойств сущностей.

Таблица \*\*\*. OwnEntitiesProperties.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| OwnEntityProperty | VARCHAR (250) | Хранит названия собственных свойств сущностей. |
| EntityType | VARCHAR (50) | Хранит имя сущности. |

Таблица ProcessCommonProperties предназначена для хранения общих свойств процесса и их компонентов.

Таблица \*\*\*. ProcessCommonProperties.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (250) | Хранит названия общих свойств процесса и его компонентов. |
| DefaultPropertyName | VARCHAR (250) | Хранит названия общих свойств процессов и его компонентов, заданных по умолчанию. |

Таблица ProcessCommonPropertiesPlans предназначена для хранения общих свойств процесса и их схем.

Таблица \*\*\*. ProcessCommonPropertiesPlans.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (50) | Хранит названия общих свойств процесса. |
| PropertyPlan | VARCHAR (250) | Хранит схемы общих свойств процесса. |

Таблица PropertiesEntities предназначена для хранения сущностей процесса.

Таблица \*\*\*. PropertiesEntities.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| EntityName | VARCHAR (50) | Хранит имена сущностей. |
| EntityType | VARCHAR (5) | Хранит типы сущностей. |

Таблица ProcessStructure предназначена для хранения свойств компонентов сущностей процесса.

Таблица \*\*\*. ProcessStructure.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertyName | VARCHAR (250) | Хранит имена свойств. |
| DefaultPropertyName | VARCHAR (250) | Хранит имена свойств, заданных по умолчанию. |

Таблица PropertiesTypes предназначена для хранения свойств указанных типов.

Таблица \*\*\*. PropertiesTypes.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| PropertySelectedType | VARCHAR (250) | Хранит имена свойств. |
| DefaultPropertySelectedType | VARCHAR (250) | Хранит имена свойств, заданных по умолчанию. |
| EntityType | VARCHAR (250) | Хранит имена сущностей. |

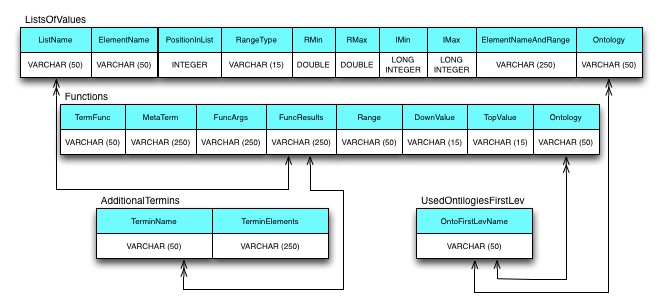


Рисунок \*\*\*. Проект базы данных, содержащей структуру онтологии.

**Описание структуры базы данных**

Таблица AdditionalTermins предназначена для хранения вспомогательных терминов онтологии.

Таблица \*\*\*. AdditionalTermins.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| TerminName | VARCHAR (50) | Хранит имена вспомогательных терминов. |
| TerminElements | VARCHAR (250) | Хранит имена элементов вспомогательных терминов. |

Таблица UsedOntologiesFirstLev предназначена для хранения названий используемых онтологий.

Таблица \*\*\*. UsedOntologiesFirstLev.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| OntoFirstLevName | VARCHAR (50) | Хранит имена используемых онтологий. |

Таблица Functions предназначена для хранения терминов-функций.

Таблица \*\*\*. Functions.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| TermFunc | VARCHAR (50) | Хранит имена терминов-функций. |
| MetaTerm | VARCHAR (250) | Хранит имена метатерминов, используемых функциями. |
| FuncArgs | VARCHAR (250) | Хранит аргументы терминов-функций. |
| FuncResults | VARCHAR (250) | Хранит результаты терминов-функций. |
| Range | VARCHAR (50) | Хранит диапазон значений терминов-функций. |
| DownValue | VARCHAR (15) | Хранит нижнюю границу значений. |
| TopValue | VARCHAR (15) | Хранит верхнюю границу значений. |
| Ontology | VARCHAR (50) | Хранит имя онтологии, к которой относится термин-функция. |

Таблица ListOfValues предназначена для хранения кортежей значений.

Таблица \*\*\*. ListOfValues.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| ListName | VARCHAR (50) | Хранит имена кортежей. |
| ElementName | VARCHAR (50) | Хранит имена элементов кортежей. |
| PositionInList | INTEGER | Хранит позиции элементов кортежей. |
| RangeType | VARCHAR (15) | Хранит тип диапазона значений. |
| RMin | DOUBLE | Хранит нижнюю границу значений типа R. |
| RMax | DOUBLE | Хранит верхнюю границу значений типа R. |
| IMin | LONG INTEGER | Хранит нижнюю границу значений типа I. |
| IMax | LONG INTEGER | Хранит верхнюю границу значений типа I. |
| ElementNameAndRange | VARCHAR (250) | Хранит имя элемента и его диапазон значений. |
| Ontology | VARCHAR (50) | Хранит онтологию, к которой относится кортеж. |

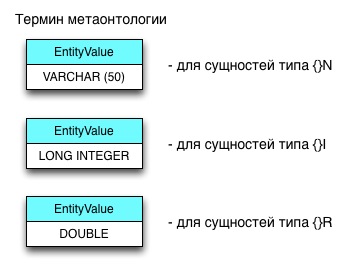


Рисунок \*\*\*. Проект базы данных, содержащей структуру знаний метаонтологии.

Таблица с именем сущности метаонтологии предназначена для хранения значений терминов метаонтологий. Каждая таблица имеет только одно поле EntityValue, тип которого зависит от типа сущности метаонтологии. Каждая таблица соответствует сущности метаонтологии.

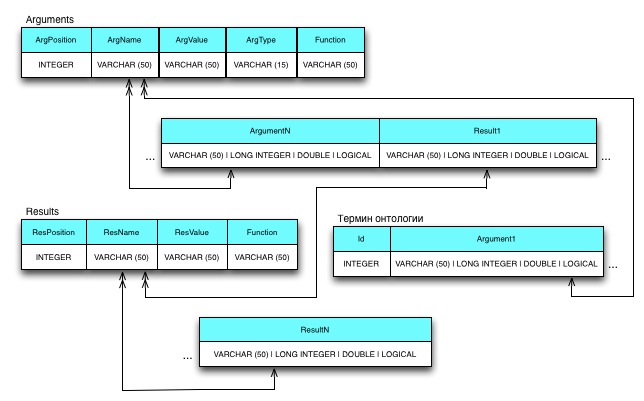


Рисунок \*\*\*. Проект базы данных, содержащей структуру знаний онтологии.

Таблица Arguments предназначена для хранения списка аргументов каждого термина-функции онтологии.

Таблица \*\*\*. Arguments.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| ArgPosition | INTEGER | Хранит номер аргумента в списке аргументов термина-функции. |
| ArgName | VARCHAR (50) | Хранит имя аргумента. |
| ArgValue | VARCHAR (50) | Хранит временное значение аргумента. |
| ArgType | VARCHAR (15) | Хранит тип аргумента. |
| Function | VARCHAR (50) | Хранит функцию, к которой относится аргумент. |

Таблица Results предназначена для хранения списка результатов каждого термина-функции онтологии.

Таблица \*\*\*. Results.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| ResPosition | INTEGER | Хранит номер результата в списке результатов термина-функции. |
| ResName | VARCHAR (50) | Хранит тип результата. |
| ResValue | VARCHAR (50) | Хранит временное значение результата. |
| Function | VARCHAR (50) | Хранит функцию, к которой относится результат. |

Таблица с именем термина-функции онтологии предназначена для хранения значений терминов-функций онтологии. Каждая таблица формируется с учетом аргументов и результатов каждой функции и их типов.

Таблица \*\*\*. Таблица с именем термина-функции онтологии.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Id | INTEGER | Хранит записи. |
| Argument1 | VARCHAR (50) | LONG INTEGER | DOUBLE | LOGICAL | Хранит значение аргумента1. |
| … | | |
| ArgumentN | VARCHAR (50) | LONG INTEGER | DOUBLE | LOGICAL | Хранит значение аргументаN. |
| Result1 | VARCHAR (50) | LONG INTEGER | DOUBLE | LOGICAL | Хранит значение результата1. |
| … |  |  |
| ResultN | VARCHAR (50) | LONG INTEGER | DOUBLE | LOGICAL | Хранит значение результатаN. |

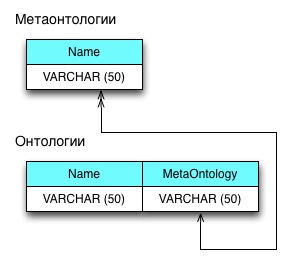


Рисунок \*\*\*. Проект базы данных, описывающей связи онтологий и метаонтологий.

Таблица Метаонтологии предназначена для хранения названий созданных метаонтологий.

Таблица \*\*\*. Метаонтологии.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Name | VARCHAR (50) | Хранит имена метаонтологий. |

Таблица Онтологии предназначена для хранения названий созданных онтологий и метаонтологий, на которых основывается каждая из созданных онтологий.

Таблица \*\*\*. Онтологии.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Name | VARCHAR (50) | Хранит имена онтологий. |
| MetaOntology | VARCHAR (50) | Хранит имена метаонтологий. |

**3.4 Архитектурно-контекстная диаграмма**

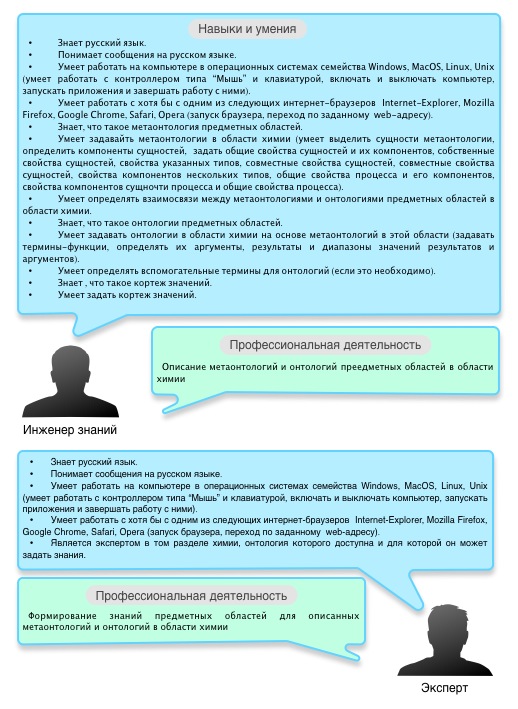
****

Рисунок \*\*\*. Профессиональная деятельность, навыки и умения пользователей.

**Список литературы**