|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | |
|  |  | | |  |
| Пермский государственный национальный  исследовательский университет | | | | |
|  |  | | |  |
|  | **разработка ОНТОЛОГИЧЕСКОЙ БАЗЫ ЗНАНИЙ**  *Практическая работа по дисциплине «Инструментальные средства создания оболочек экспертных систем»* | | |  |
|  |  | | |  |
|  |  |  | Работу выполнил  студент гр. ПМИ-2 4 курса механико-математического факультета  \_\_\_\_(подпись)  «3» декабря 2019 |  |
|  |  |  | Работу проверил  ассистент кафедры МОВС  Леонтьева Т.А. \_\_\_\_\_ (подпись)  «\_\_» \_\_\_\_\_\_ 201\_ |  |
|  |  |  |  |  |
| Пермь 2019 | | | | |

СОДЕРЖАНИЕ

[Введение 3](#_Toc26253558)

[Глава 1. Этап идентификации 4](#_Toc26253559)

[Глава 2. Этап концептуализации 5](#_Toc26253560)

[Глава 3. Этап формализации 10](#_Toc26253561)

[Глава 4. Построение онтологии 12](#_Toc26253562)

[Заключение 14](#_Toc26253563)

Введение

Цель: разработка онтологической базы знаний для проблемной области задачи, решаемой в рамках выпускной работы.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Выполнить этап идентификации.
2. Выполнить концептуализацию онтологической БЗ.
3. Выполнить формализацию онтологической БЗ.
4. Реализовать онтологическую БД в среде визуального инструментального средства построения онтологических БЗ ОНТОЛИС.

Отметим, что по условиям задания разрабатываемая онтологическая БЗ должна дополнять одну из двух предоставленных онтологических БЗ.

# Этап идентификации

В разрабатываемой онтологической БЗ должны быть представлены знания о проблемной области задачи, решаемой в рамках выпускной работы.

В рамках выпускной работы решается задача разработки оболочки систем визуального моделирования на основе DSM-платформы MetaLanguage, модифицированной для использования гиперграфов с полюсами в качестве математической модели.

Следовательно, онтологическую БЗ необходимо разработать для проблемной области «Предметно-ориентированное моделирование».

Выделим основные понятия, связанные с рассматриваемой проблемной областью.

# Этап концептуализации

Выделим основные понятия в проблемной области, опираясь на описание решаемой задачи, методов решения задачи и связанных с ними понятий.

Начнём с того, что разрабатывается программная оболочка систем визуального моделирования (далее — ОСВМ). Следовательно, два центральных понятия — «Оболочка систем визуального моделирования» и «Программная оболочка».

Оболочка систем визуального моделирования предназначена для упрощения работы с системами визуального моделирования, поэтому необходимо ввести и понятие «Система визуального моделирования».

Разрабатываемая ОСВМ использует DSM-платформу MetaLanguage, поэтому вводим ещё два понятия — «MetaLanguage» и «DSM-платформа».

DSM-платформа используется для разработки предметно-ориентированных языков (Domain Specific Languages, сокращённо DSL) для предметно-ориентированного моделирования (Domain Specific Modeling, сокращённо DSM). Для отражения этого факта, вводим ещё три понятия: «Разработка DSL», «DSL» и «Domain Specific Modeling».

В некоторой степени, DSM — частный случай визуального моделирования, поэтому введём общее понятие «Визуальное моделирование».

Результатом визуального моделирования является визуальная модель, поэтому для дополнения картины введём понятие «Визуальная модель».

Визуальная модель — это модель объекта или системы (а иногда и процесса), представленная в графическом виде. Два основных компонента визуальной модели — представление модели и структура модели (иногда являющихся двумя полностью независимыми частями модели, а иногда объединяемыми в одну часть). Введём понятия «Модель», «Представление визуальной модели», «Представление» и «Структура визуальной модели».

Для описания структуры визуальных моделей в разрабатываемой ОСВМ будут использоваться гиперграфы с полюсами (являющимися частным случаем гиперграфов), поэтому введём понятия «Гиперграф с полюсами» и «Гиперграф».

Обычные гиперграфы состоят из множества вершин и множества гиперрёбер. Гиперграфы с полюсами отличаются тем, что к этим двум множествам добавляется ещё одно: множество полюсов. Для отображения такого положения дел необходимо ввести понятия «Множество вершин», «Вершина», «Множество гиперрёбер», «Гиперребро», «Множество полюсов», «Полюс». Также введём общее понятие «Множество», объединяющее понятия «Множество вершин», «Множество гиперрёбер» и «Множество полюсов».

Вернёмся к рассмотрению подробностей реализации ОСВМ. Для экспорта и импорта визуальных моделей ОСВМ будет использовать разработанную ранее систему экспорта-импорта визуальных моделей, состоящую из двух независимых модулей: экспортёр визуальных моделей (используемый для экспорта визуальных моделей) и импортёр визуальных моделей (используемый для импорта визуальных моделей). В текущий момент оба этих модуля могут использовать XML (Extensible Markup Language) и JSON для передачи визуальных моделей. Для отражения всей этой ситуации, введём понятия «Система экспорта-импорта визуальных моделей», «Экспортёр визуальных моделей», «Экспорт визуальных моделей», «Импортёр визуальных моделей», «Импорт визуальных моделей», «Extensible Markup Language», «JSON». Помимо упомянутых понятий, также введём понятия, определяющие или обобщающие перечисленные выше понятия: «Программная система» (объединяющее понятия «Программная оболочка» и «Система экспорта-импорта визуальных моделей»), «Язык разметки» (определяющее понятие «XML») и «Текстовый формат обмена данными» (определяющее понятие «JSON»).

ОСВМ разрабатывается как настольное приложение с графическим интерфейсом (далее для обозначения графического интерфейса будет использоваться сокращение GUI — Graphical User Interface). Двумя основными компонентами GUI будет рабочее полотно и набор элементов, поддерживаемых визуальной моделью. Для того, чтобы разработать GUI, будет использоваться декларативный подход к разработке GUI. Чтобы обозначить такое положение дел, введём понятия «Настольное приложение», «GUI», «Рабочее полотно», «Набор элементов», «Декларативный подход к разработке GUI».

Перейдём к инструментальной стороне разработки ОСВМ. Для разработки ОСВМ будет использоваться Liberica JDK — одна из имеющихся реализаций JDK, а для запуска ОСВМ может быть использована любая JRE нужной версии. Заметим, что JDK — это т.н. набор разработчика (SDK) для разработки на Java. Т.о., для отображение этой ситуации необходимо ввести понятия «Liberica JDK», «JDK», «Набор инструментов разработчика», «JRE», «Java». Также необходимо ввести понятие, определяющее Java — «Объектно-ориентированный ЯП».

Для декларативной разработки GUI будет использоваться RAD-средство Scene builder, с которым будут использоваться два фреймворка для разработки GUI: JavaFX (использующий Java) и TornadoFX (использующий Kotlin). Чтобы зафиксировать такую ситуацию, введём понятия «Scene builder», «RAD-средство», «JavaFX», «TornadoFX», «Фреймворк для разработки GUI», «Kotlin». Также чтобы отметить, что RAD-средства используются для быстрой разработки приложений (Rapid Application Development — RAD), введём понятие «RAD».

Дадим определения всем введённым понятиям.

Все введённые понятия и их определения могут быть структурированы в виде списка.

1. «DSL» — искусственный язык, специализированный для конкретной ПрО.
2. «Domain Specific Modeling» — методология моделирования с использованием предметно-специфичных языков (DSL).
3. «DSM-платформа» — программное средство для разработки DSL.
4. «DSM-платформа MetaLanguage» — DSM-платформа для разработки визуальных DSL.
5. «GUI» — (графический интерфейс пользователя) вид пользовательского интерфейса, который позволяет пользователю взаимодействовать с электронным устройством посредством манипуляции отображаемыми на экране графическими элементами управления.
6. «Java» — объектно-ориентированный ЯП, разработанный компанией Sun Microsystems.
7. «JavaFX» — фреймворк для разработки GUI на ЯП Java.
8. «JDK» — набор из стандартных библиотек, предназначенных для разработки на Java (Java Development Kit).
9. «JRE» — среда выполнения Java (Java Runtime Environment).
10. «JSON» — текстовый формат обмена данными, основанный на JavaScript.
11. «Kotlin» — объектно-ориентированный ЯП
12. «Liberica JDK» — реализация JDK от компании ООО Беллсофт.
13. «RAD» — (Rapid Application Development) адаптивные подходы к разработке ПО с меньшим упором на планирование и большим на адаптивный процесс разработки. Зачастую, в RAD прототипы заменяют макеты.
14. «RAD-средство» — программное средство для разработки в соответствии с подходами RAD.
15. «Scene builder» — RAD-средство для разработки на JavaFX и TornadoFX, разработанное компанией Gluon.
16. «TornadoFX» — реализация JavaFX для ЯП Kotlin.
17. «XML» **—** язык разметки, задающий формат для записи документов в человеко- и машинно-читаемом форматах. Определён в первую очередь спецификацией W3C XML 1.0 (а также некоторыми другими).
18. «Библиотека» — сборник подпрограмм или объектов, используемых при разработке ПО.
19. «Вершина» — множество взаимно не пересекающихся подмножеств полюсов.
20. «Визуальная модель» — графическое представление модели объекта и/или системы.
21. «Визуальное моделирование» — представление объектов и систем в графическом виде с помощью графических языков.
22. «Гиперграф» — упорядоченная двойка G=(V,W), где V — непустое множество вершин, W — множество гиперрёбер.
23. «Гиперграф с полюсами» — упорядоченная тройка G=(P,V,W), где P —абстрактное множество полюсов, V — непустое множество вершин, W — множество гиперрёбер.
24. «Гиперребро» — подмножество множества всех подмножеств полюсов, определяющее связи между вершинами.
25. «Графический язык моделирования» — язык моделирования для описания объектов/систем и связей между ними в виде диаграммы с именованными символами, обозначающими понятия, и линиями, обозначающими связи.
26. «Декларативная разработка GUI» — способ полностью описать GUI и его поведение на некотором языке описания данных.
27. «Импорт визуальных моделей» — процесс передачи визуальных моделей в программную систему для работы с ними.
28. «Импортёр визуальных моделей» — средство для импорта визуальных моделей.
29. «Множество» — набор уникальных однотипных элементов.
30. «Множество вершин» — множество, состоящее из вершин.
31. «Множество гиперрёбер» — множество, состоящее из гиперрёбер.
32. «Множество полюсов» — множество, состоящее из полюсов.
33. «Модель» — представление объекта/системы, отображающее его основные свойства, компоненты и связи между ними.
34. «Набор разработчика» — набор библиотек для разработки с применением той или иной технологии.
35. «Набор элементов» — набор допустимых для формирования визуальной модели элементов.
36. «Настольное приложение» — ПО, предназначенное для запуска на ПК и выполняемое вне веб-браузера.
37. «Обмен данными» — физический перенос данных в виде сигналов от точки к точке по каналу передачи данных.
38. «Оболочка систем визуального моделирования» — программная оболочка, предназначенная для работы с системами визуального моделирования.
39. «Полюс» — базовая структурная единица гиперграфа с полюсами.
40. «Представление» — отображение объекта/системы в форму, отличную от изначальной.
41. «Представление визуальной модели» — графическое представление всех элементов визуальной модели и связей между ними.
42. «Программная оболочка» — программа, предоставляющая интерфейс для работы со сложными программами системами, тем самым упрощая работу с ними.
43. «Программная система» — совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.
44. «Рабочее полотно» — область GUI, где отображается результат работы пользователя.
45. «Разработка DSL» — процесс создания DSL.
46. «Система экспорта-импорта визуальных моделей» — контейнер для экспортёра и импортёра визуальных моделей.
47. «Структура визуальной модели» — набор элементов визуальной модели и связей между ними.
48. «Текстовый формат обмена данными» — формат представления информации строкового типа, используемый при обмене данными.
49. «Фреймворк для разработки GUI» — набор библиотек, составляющих единую экосистему и предназначенных для разработки GUI.
50. «Экспорт визуальных моделей» — процесс передачи визуальных моделей вовне программной системы, работающей с ними.
51. «Экспортёр визуальных моделей» — средство для экспорта визуальных моделей.
52. «Язык моделирования» — искусственный язык для представления информации, знаний или систем структурой, заданной набором определённых правил.

Определим связи между всеми введёнными понятиями.

# Этап формализации

Связи между всеми введёнными понятиями могут быть структурированы в виде таблицы (см. табл. 3.1). Так как разрабатываемая онтология будет интегрирована с предоставленной онтологией, то отобразим и внешние по отношению к разрабатываемой онтологии понятия (будут обозначены жирным выделением).

Таблица 3.1 — Связи между основными понятиями

| Понятие 1 | Связь | Понятие 2 |
| --- | --- | --- |
| DSL | useFor | Domain Specific Modeling |
| DSL | sameAs | **Domain specific languages** |
| Domain Specific Modeling | isA | Визуальное моделирование |
| DSM-платформа | useFor | Разработка DSL |
| DSM-платформа MetaLanguage | isA | DSM-платформа |
| Java | instanceOf | **Object oriented languages** |
| JavaFX | isA | Фреймворк для разработки GUI |
| JavaFX | uses | Java |
| JDK | isA | Набор разработчика |
| JDK | uses | Java |
| JRE | uses | Java |
| JSON | instanceOf | Текстовый формат обмена данными |
| Kotlin | instanceOf | **Object oriented languages** |
| Liberica JDK | instanceOf | JDK |
| RAD-средство | useFor | RAD |
| Scene builder | instanceOf | RAD-средство |
| Scene builder | useFor | Декларативная разработка GUI |
| Scene builder | uses | TornadoFX |
| Scene builder | uses | JavaFX |
| TornadoFX | instanceOf | JavaFX |
| TornadoFX | uses | Kotlin |
| XML | instanceOf | **Extensible Markup Language** |
| Библиотека | aPartOf | Набор разработчика |
| Библиотека | aPartOf | Фреймворк для разработки GUI |
| Вершина | aPartOf | Множество вершин |
| Вершина | isA | Множество полюсов |
| Визуальная модель | isA | Модель |
| Визуальная модель | has | Представление визуальной модели |
| Визуальная модель | has | Структура визуальной модели |
| Визуальное моделирование | output | Визуальная модель |
| Гиперграф | sameAs | **Hypergraphs** |
| Гиперграф с полюсами | isA | Гиперграф |
| Гиперребро | aPartOf | Множество гиперрёбер |
| Гиперребро | isA | Множество полюсов |
| Графический язык моделирования | useFor | Визуальное моделирование |
| Декларативная разработка GUI | isA | Разработка GUI |
| Декларативная разработка GUI | output | GUI |
| Импорт визуальных моделей | output | Визуальная модель |
| Импортёр визуальных моделей | useFor | Импорт визуальных моделей |
| Импортёр визуальных моделей | uses | JSON |
| Импортёр визуальных моделей | uses | XML |
| Импортёр визуальных моделей | aPartOf | Система экспорта-импорта визуальных моделей |
| Множество вершин | isA | Множество |
| Множество вершин | aPartOf | Гиперграф |
| Множество гиперрёбер | isA | Множество |
| Множество гиперрёбер | aPartOf | Гиперграф |
| Множество полюсов | isA | Множество |
| Множество полюсов | aPartOf | Гиперграф |
| Набор элементов | aPartOf | GUI |
| Оболочка систем визуального моделирования | has | GUI |
| Оболочка систем визуального моделирования | isA | Настольное приложение |
| Оболочка систем визуального моделирования | uses | Liberica JDK |
| Оболочка систем визуального моделирования | instanceOf | Программная оболочка |
| Оболочка систем визуального моделировани | uses | Система экспорта-импорта визуальных моделей |
| Оболочка систем визуального моделирования | uses | Система визуального моделирования |
| Оболочка систем визуального моделирования | uses | DSM-платформа MetaLanguage |
| Полюс | aPartOf | Множество полюсов |
| Представление визуальной модели | instanceOf | Представление |
| Программная оболочка | isA | Программная система |
| Рабочее полотно | aPartOf | GUI |
| Разработка DSL | output | DSL |
| Разработка GUI | sameAs | **User interface programming** |
| Система экспорта-импорта визуальных моделей | isA | Программная система |
| Структура визуальной модели | uses | Гиперграф с полюсами |
| Текстовый формат обмена данными | useFor | Обмен данными |
| Экспортёр визуальных моделей | useFor | Экспорт визуальных моделей |

Представим выделенные понятия и связи между ними в виде связного графа. Этот граф будет представлять онтологию.

Для построения онтологии используем средство построения онтологических БЗ ОНТОЛИС.

Глава 4. Построение онтологии

На основе всех вышеприведённых этапов разработки онтологии в средстве построения онтологических БЗ ОНТОЛИС была разработана онтология для проблемной области задачи, решаемой в рамках выпускной работы.

Результат приведён на рис. 4.1-4.2.

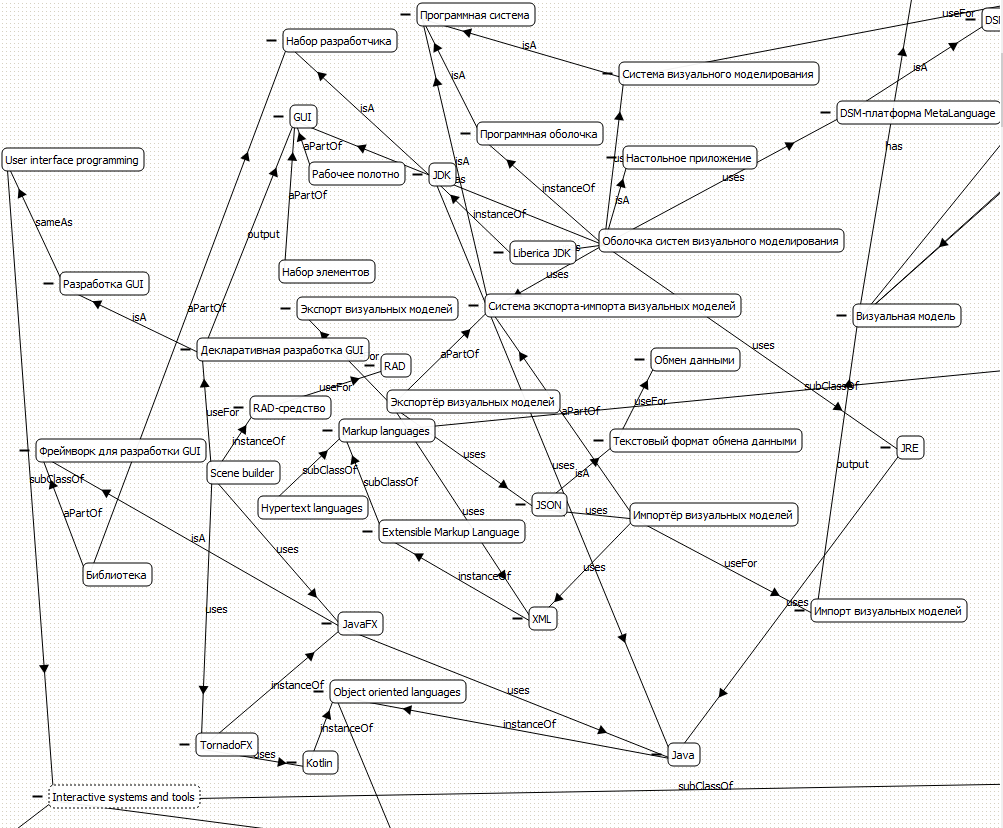


Рис. 4.1 — Часть онтологии №1

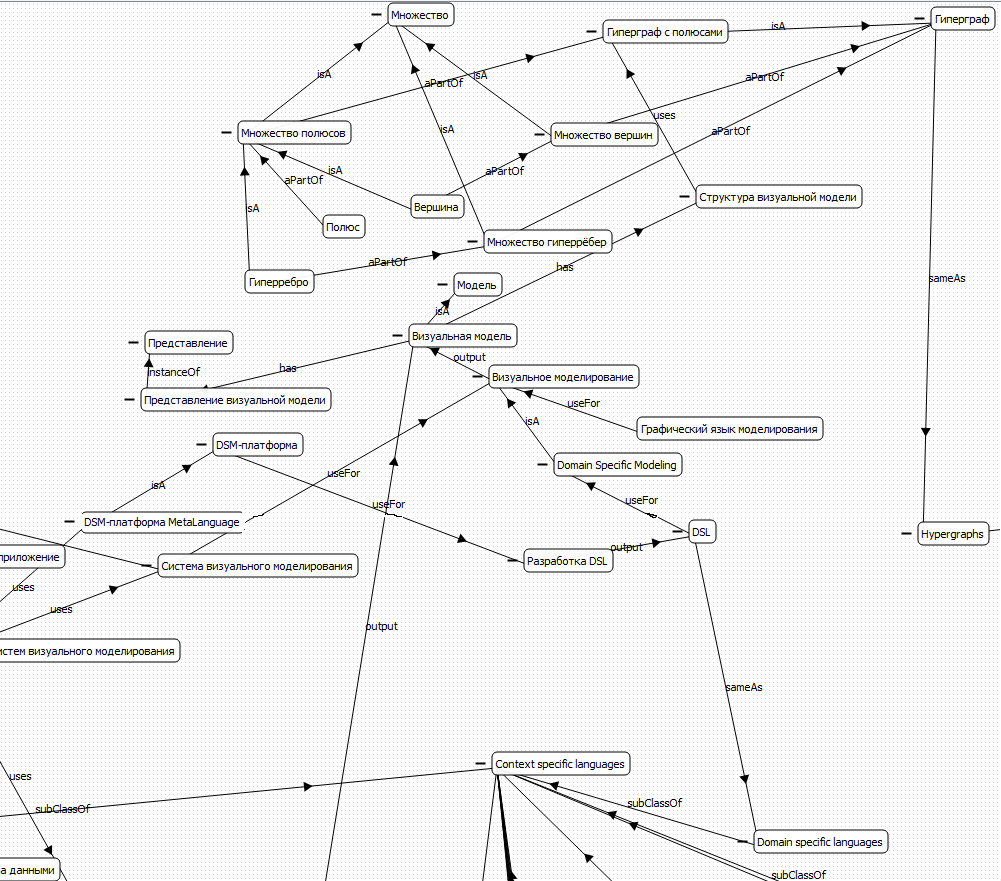


Рис. 4.2 — Часть онтологии №2

Заключение

В ходе данной работы была разработана онтология для проблемной области задачи, решаемой в рамках выпускной работы

Разработанная онтология полностью удовлетворяет предъявленным к ней требованиям.