**Итоговый отчет по лабораторной работе**

**Введение**

**Цель работы:**Разработать и сравнить два подхода к моделированию знаний на примере предметной области «Вселенная Видеоигры».

**Задачи:**

1. Создать формальную онтологию в редакторе Protégé, определив классы, свойства и правила логического вывода (аксиомы).
2. Проверить работоспособность онтологии с помощью механизма логического вывода (Reasoner/Drools).
3. Адаптировать разработанную онтологическую модель в графовую базу данных Neo4j.
4. Реализовать логику аксиом с помощью языка запросов Cypher и проверить результаты.
5. Провести сравнительный анализ двух подходов, выявив их сильные и слабые стороны, а также области применения.

**Описание Моделируемой Предметной Области**

Моделируемая вселенная представляет собой упрощенную структуру типичной ролевой видеоигры. Она включает в себя следующие основные сущности:

* **Персонажи:** Действующие лица, которые могут выполнять задания и находиться в различных локациях.
* **Квесты:** Задания, которые имеют определенные характеристики (например, уровень сложности).
* **Предметы:** Внутриигровые объекты (оружие, зелья и т.д.).
* **Локации:** Игровые зоны, в которых происходят события.

Ключевыми правилами, которые необходимо было смоделировать, являются:

* Персонаж считается «опытным», если он выполняет сложный квест.
* Персонаж получает статус «в опасности», если он находится в определенной локации.

**Реализация в Protege**

В Protégé используется декларативный подход на основе Языка Веб-Онтологий. Мы не описываем, как получить результат, а описываем факты и правила мира. Мы говорим: «Истина в том, что любой персонаж, выполняющий сложный квест, является опытным». Специальный механизм Reasoner сам использует эти правила, чтобы сделать выводы.

**Структура онтологии**

* **Классы:** Персонаж (с подклассом ОпытныйПерсонаж), Квест (с подклассом СложныйКвест), Предмет, Локация.
* **Объектные свойства (Отношения):** выполняетКвест, находитсяВ.
* **Свойства данных (Атрибуты):** имя, требуемыйУровень, типПредмета, размерЛокации, статус.

**Реализация логического вывода (Аксиомы)**

Логика была реализована с помощью правил SWRL. Эти правила не изменяют исходные данные, а позволяют Reasoner **динамически выводить** новые факты.

* **Правило 1 (Опытность):**  
  Персонаж(?p) ^ выполняетКвест(?p, ?q) ^ СложныйКвест(?q) -> ОпытныйПерсонаж(?p)
* **Правило 2 (Опасность):**  
  Персонаж(?p) ^ находитсяВ(?p, ?l) ^ Локация(?l) ^ имя(?l, "Темный Лес") -> статус(?p, "В опасности")

**Результат:** После запуска механизма Drools система корректно вывела, что Герой\_Артур принадлежит к классу ОпытныйПерсонаж, а Маг\_Мерлин имеет статус "В опасности". Эти выведенные знания являются "виртуальными" и существуют, пока работает Reasoner.

**Реализация в Neo4j**

**Подход к моделированию**

В Neo4j используется модель маркированного графа свойств (Labeled Property Graph) и императивный подход. Мы не просто описываем правила, а даем прямые команды: «Найди вот такой шаблон в данных и измени его вот так». Логический вывод — это не фоновый процесс, а явный запрос на модификацию данных.

**Структура графа**

* **Метки узлов (Классы):** :Персонаж, :ОпытныйПерсонаж, :Квест, :СложныйКвест, :Предмет, :Локация.
* **Отношения (Связи):** [:ВЫПОЛНЯЕТ\_КВЕСТ], [:НАХОДИТСЯ\_В].
* **Свойства узлов (Атрибуты):** id, имя, требуемыйУровень, типПредмета, размерЛокации, статус.

**Реализация логического вывода (Аксиомы)**

Логика была реализована с помощью запросов Cypher, которые физически изменяют (материализуют) граф.

**Правило 1 (Опытность):**

MATCH (p:Персонаж)-[:ВЫПОЛНЯЕТ\_КВЕСТ]->(q:СложныйКвест) SET p:ОпытныйПерсонаж (Этот запрос находит нужных персонажей и добавляет им новую метку, перманентно изменяя их тип.)

**Правило 2 (Опасность):**

MATCH (p:Персонаж)-[:НАХОДИТСЯ\_В]->(l:Локация {имя: 'Темный Лес'})

SET p.статус = 'В опасности'(Этот запрос находит персонажей в указанной локации и **добавляет им новое свойство**, перманентно изменяя их данные.)

**Результат:** После выполнения этих запросов граф был изменен. Проверочные запросы показали, что узел Герой\_Артур получил метку :ОпытныйПерсонаж, а узел Маг\_Мерлин — свойство статус: 'В опасности'.

**Заключение**

В ходе работы было установлено, что Protégé и Neo4j, хотя и могут решать схожие задачи, являются инструментами для разных этапов жизненного цикла модели знаний.

**Protégé — это про «что есть истина».** Он создает «чертеж» знаний и проверяет, что из этого следует по логике.

**Neo4j — это про «что нужно сделать».** Он хранит реальные данные и изменяет их по вашим командам.

**Итог в трех пунктах:**

**Protégé**

* 1. Что вы делаете? описываете правила мира
  2. Как он думает? делает логические выводы в "уме".
  3. Что получается? виртуальное знание. Данные не меняются.

**Neo4j**

* 1. Что вы делаете? Даете пошаговые команды.
  2. Как он думает? Выполняет ваши приказы.
  3. Что получается? Физическое изменение. Данные меняются навсегда.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.