Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №1-2**

по дисциплине

Системы искусственного интеллекта

Тема – Dungeon & Dragon’s: Eberron

*Выполнил:*

Студент группы P3308

Иванов Леонид  
Антонович

*Преподаватель:*

Кугаевских Александр Владимирович



Санкт-Петербург, 2024

Введение

Целью первого блока лабораторных работ является разработка базы знаний, онтологии и программы, которая будет использовать базу знаний или онтологию для предоставления рекомендаций на основе введенных пользователем данных.

Значимость данного проекта заключается в том, что он помогает автоматизировать процесс принятия решений, повышает эффективность бизнес-процессов и создает персонализированные рекомендации.

Анализ требований

Для корректной работы система поддержки принятия решений должна:

* Правильно обрабатывать пользовательский ввод.
* Поддерживать различные форматы входных данных.
* Предупреждать пользователя при некорректном вводе.
* Обладать возможностью расширения базы знаний и инфраструктуры.
* Автоматически инициализировать базу знаний при запуске.

Требования к базе знаний:  
  
База знаний должна основываться на данных из сеттинга D&D Eberron и включать как факты, так и правила. Она должна уточнять требования к онтологии, описывая классы, свойства и отношения между сущностями.

Изучение основных концепций и инструментов

**База знаний (Knowledge Base)** — это структурированное хранилище информации, которое содержит знания о предметной области в форме фактов и правил. Она используется для организации и управления информацией, позволяя компьютерным системам эффективно извлекать, обрабатывать и предоставлять данные в удобной для пользователя форме. База знаний выполняет функцию хранения и обработки информации, необходимой для принятия решений и выполнения задач, связанных с искусственным интеллектом.

* **Знания** — это совокупность фактов и концепций, относящихся к определённой области, которые хранятся в базе знаний.
* **Факты** — это утверждения, описывающие свойства и характеристики объектов в предметной области. Они служат основой для работы системы, предоставляя конкретную информацию о сущностях и их состояниях.
* **Правила** — это логические конструкции, описывающие взаимоотношения между фактами. Система использует их для вывода новой информации и принятия решений на основе существующих данных.
* **Запросы** — это механизмы получения информации из базы знаний. Пользователь формирует запрос, указывая условия, а система возвращает данные, удовлетворяющие этим условиям.

**Онтология (Ontology)** — это формальная модель, описывающая предметную область и связи между понятиями, которые её составляют. Онтология структурирует данные в виде классов, свойств и отношений, создавая логическую основу для интерпретации информации. Она предоставляет возможность не только хранить данные, но и понимать их смысл на более глубоком уровне, что особенно важно для сложных систем искусственного интеллекта.

* **Классы** — это категории, к которым относятся сущности в предметной области. Классы описывают общие характеристики объектов, принадлежащих к одному типу. Например, в системе D&D Eberron, классами могут быть персонажи, оружие, расы и заклинания.
* **Свойства** — это характеристики или атрибуты сущностей, которые описывают их конкретные особенности. Например, персонажи могут иметь свойства, такие как "сила" или "здоровье", а заклинания могут иметь свойства "урон" или "тип".
* **Инстансы** — это конкретные объекты, являющиеся экземплярами классов. Например, персонаж "Эльф-Воин" будет инстансом классов "Эльф" и "Воин".
* **Отношения** — это связи между сущностями или классами. Например, персонаж может быть связан с классом "воин" и обладать отношением к оружию "меч". Отношения в онтологии играют ключевую роль для понимания взаимодействий между объектами и их характеристиками.

### Сходства и различия между базой знаний и онтологией

**Сходства**:

1. **Организация информации**: И база знаний, и онтология служат для структурирования информации. База знаний концентрируется на фактах и правилах, а онтология формирует семантическую структуру данных, организуя их в понятия и отношения.
2. **Использование в информационных системах**: Обе концепции активно применяются в системах искусственного интеллекта для управления данными и их анализа. Они дополняют друг друга, обеспечивая как хранение данных, так и их интерпретацию.

**Различия**:

1. **Абстракция и структура**: Онтология более абстрактна и описывает концепции и отношения на семантическом уровне, тогда как база знаний содержит конкретные факты и правила.
2. **Семантика**: Онтология придаёт особое значение смыслу данных, структурируя их так, чтобы компьютерные системы могли интерпретировать их. База знаний же может просто хранить факты без явного описания семантики.
3. **Цель**: Основная задача онтологии — обеспечить понимание и интерпретацию данных, в то время как база знаний предназначена для хранения и извлечения информации для выполнения логических операций и поддержки принятия решений.

Основное отличие между базами знаний и онтологиями заключается в способе представления и организации знаний. Базы знаний представляют собой собрание фактов и правил, в то время как онтологии моделируют понятия и отношения между ними.

В итоге, онтология и база знаний взаимосвязаны, но выполняют разные функции в организации и управлении информацией в различных предметных областях.

Prolog (Programming in Logic) - это декларативный язык программирования, который используется для решения задач искусственного интеллекта и логического программирования. Prolog основан на логике первого порядка и предназначен для решения задач, в которых логические отношения и правила играют ключевую роль.

Инструмент для Базы знаний - <https://swish.swi-prolog.org/>

Инструмент для Онтологий - [https://protege.stanford.edu/software.php#desktop-protege](https://protege.stanford.edu/software.php" \l "desktop-protege)

Реализация системы искусственного интеллекта на Prolog

Код Prolog:

Основных классов 6:

* Character – основной класс, представляющий каждого персонажа
* Class – класс персонажа
  + Warrior, Priest, Druid, Archer, Berserker, Paladin
* Race – раса персонажа
  + Human, Elf, Dwarf, Orc, Goblin
* Weapon – класс для типов оружия
  + Sword, Staff, Dagger, Mace, Bow, Axe, Hammer
* Spell - класс для заклинаний
  + Fireball, Shield, Heal, Smite
* Alliance - класс для союзов
  + Knights of Valor, Circle of Mages и другие.

Правила:

% Правило о том, что персонаж является человеком и воином

is\_human\_warrior(Character) :-

character\_race\_of(Character, human),

character\_class\_of(Character, warrior).

% Правило о том, что персонаж является эльфом и магом или друидом

is\_elf\_magic\_user(Character) :-

character\_race\_of(Character, elf),

(character\_class\_of(Character, wizard); character\_class\_of(Character, druid)).

% Правило о том, что персонаж обладает двуручным оружием (мечом, топором или молотом)

has\_two\_handed\_weapon(Character) :-

character\_weapon(Character, sword);

character\_weapon(Character, axe);

character\_weapon(Character, hammer).

% Правило о том, что персонаж является воином и обладает двуручным оружием

is\_warrior\_with\_two\_handed\_weapon(Character) :-

character\_class\_of(Character, warrior),

has\_two\_handed\_weapon(Character).

% Правило о том, что персонаж может лечить (если он жрец или друид и знает заклинание "heal")

can\_heal(Character) :-

(character\_class\_of(Character, priest); character\_class\_of(Character, druid)),

character\_spell(Character, heal).

% Правило о том, что персонаж принадлежит к союзам рыцарей или магов

is\_knight\_or\_mage(Character) :-

character\_alliance(Character, knights\_of\_valor);

character\_alliance(Character, circle\_of\_mages).

% Правило о том, что персонаж является дворфом и носит молот

is\_dwarf\_with\_hammer(Character) :-

character\_race\_of(Character, dwarf),

character\_weapon(Character, hammer).

% Правило о том, что персонаж может использовать заклинания щита (если персонаж маг и знает заклинание "shield")

can\_cast\_shield(Character) :-

character\_class\_of(Character, wizard),

character\_spell(Character, shield).

% Правило о том, что персонаж принадлежит к союзу "Изумрудный Энклав" и может лечить

is\_emerald\_healer(Character) :-

character\_alliance(Character, emerald\_enclave),

can\_heal(Character).

% Правило о том, что персонаж является орком и берсерком, но не состоит в союзе

is\_orc\_berserker\_without\_alliance(Character) :-

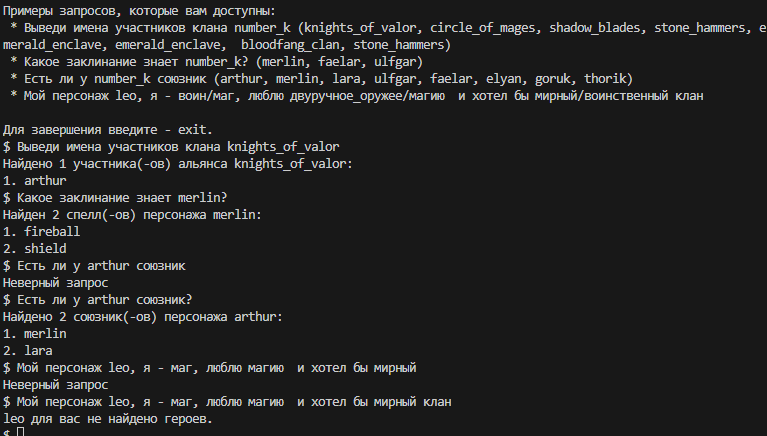
character\_race\_of(Character, orc),

character\_class\_of(Character, berserker),

\+ character\_alliance(Character, \_).

Оценка и интерпретация результатов

Запросы к нашей системе поддержки принятия решений



Наша система соответствует всем поставленным требованиям.

Цель была достигнута – была разработана база знаний и онтология по сеттинга D&D Eberron, на их основе было сделано приложение, которое помогает пользователю создавать запросы и узнавать новую информацию.

В дальнейшем можно дополнять базу знаний, прописывать новые связи. Также добавить больше шаблонов для пользовательского ввода и сделать ввод более гибким.

Заключение

Разработанное приложение может помочь игрокам глубже понять игровой процесс D&D, а также выбрать подходящего персонажа и оружие. Программа может быть интегрирована с игровыми сайтами для помощи игрокам в создании и улучшении своих игровых персонажей.

.