

**РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОЙ ВЕРСИИ ИНСТРУМЕНТА СОЦИАЛЬНО -
ПОВЕДЕНЧЕСКОГО МОДЕЛИРОВАНИЯ НА ОСНОВЕ ОНТОЛОГИИ ПРЕДМЕТНОЙ
ОБЛАСТИ И ГРАФОВОЙ БАЗЫ ДАННЫХ****Б.И. Клебанов, Н.В. Кабиров, Д.С. Ломотков, И.А. Юрасов (Екатеринбург)**

При планировании и прогнозировании развития городской среды, анализе поведения различных групп населения обычно используются методы, основанные на сборе статистических данных, которые анализируются различными экспертами. Для помощи в принятии решений и прогнозировании эксперты могут использовать системы моделирования, которые используют полученную статистику. Однако создание таких систем требует участия программистов, что снижает доверие к результатам моделирования. В отчете известной американской организации RAND[1], посвященном приоритетным задачам социально-поведенческих исследований и их моделирования, отмечаются препятствия существующей практики исследований в данной отрасли, к числу которых отнесены: фрагментация науки; недостаточная понятность, рецензируемость и воспроизводимость ранее разработанных моделей; неполное соответствие науки и моделей; потребность в модульности и композиции; слабое внимание к оценке достоверности моделей.

Цель данной работы является создание экспериментальной версии инструмента разработки моделей развития общества без использования программирования, что позволит снизить трудоемкость создания моделей и приблизить язык разработчика модели к языку экспертов в данной предметной области.

Теоретической основой создания инструмента является математическая модель расширенного гибридного автомата (РГА), которой представляются все объекты действительности [2][3]

Основой для создания модели является онтологическое описание предметной области. Процесс создания модели с использованием данного инструмента включает следующие этапы:

1. Определение основных понятий для онтологии верхнего уровня;
2. Создание онтологии верхнего уровня в системе описания онтологии;
3. Создание прототипа базы знаний на основе онтологии верхнего уровня;
4. Экспорт базы знаний (БЗ) в СУБД, с которой взаимодействует движок модели.
5. Автоматическая генерация и пространственное распределение заданного множества экземпляров объектов.
6. Проведение экспериментов с моделью.

В качестве системы описания онтологии (базы знаний) выбран известный инструмент Protégé [4]. Для представления знаний и данных в процессе имитации выбрана графовая СУБД Neo4j [5]. Эта СУБД наиболее соответствует алгоритмам обработки связей и отношений в рассматриваемой предметной области. Для экспорта БЗ из Protégé в Neo4j использовался плагин neosemantics [6]. Для создания «движка» использован язык программирования C#, на основе фреймворка .NET. Интерфейс приложения создан на WindowsForms.

Применение указанной технологии покажем на простом примере моделирования поведения группы людей при удовлетворении простейших потребностей в пище, покое и т.п.

В качестве примера использовалась иерархия классов предметной области, представленная на рис.1.

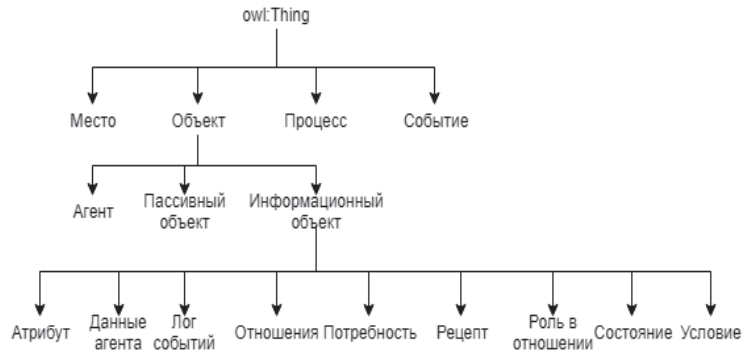


Рис. 1. Иерархия классов базы знаний

На рис. 2 представлена соответствующая иерархия классов базы знаний, созданная в Protégé.

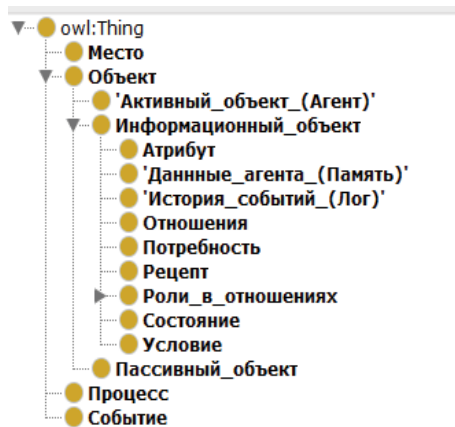


Рис. 2. Иерархия классов в Protégé

Возможные отношения между экземплярами классов представлены на рис. 3.

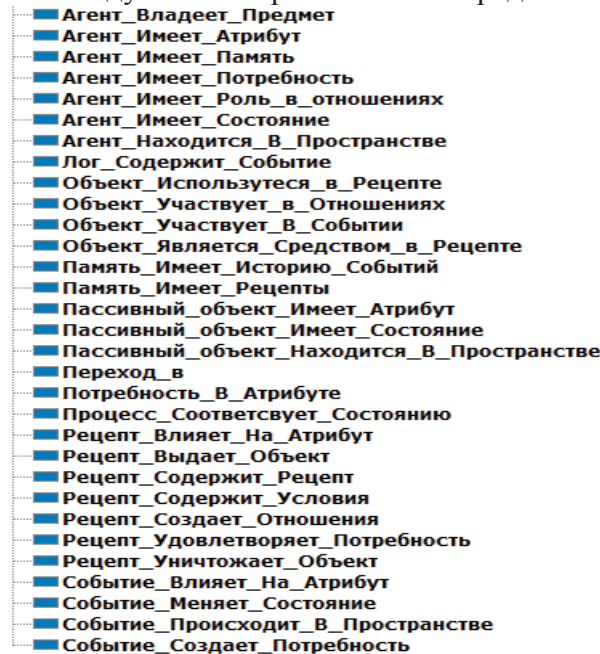


Рис. 3. Возможные отношения между экземплярами

Простой граф переходов РГА, представляющего поведение агента представлен на рис.4

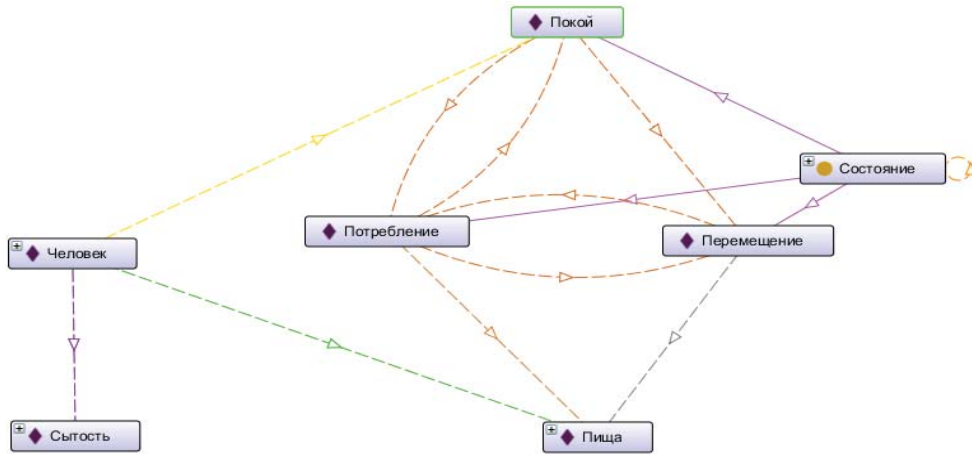


Рис. 4. Граф базы состояний человека

Приведенная выше схема была реализована в Protégé с помощью механизма создания экземпляров классов. Информация об атрибутах и условиях перехода привязана с помощью комментариев. На рисунке 5 представлен результат переноса базы знаний вышеприведенного примера в Neo4j.

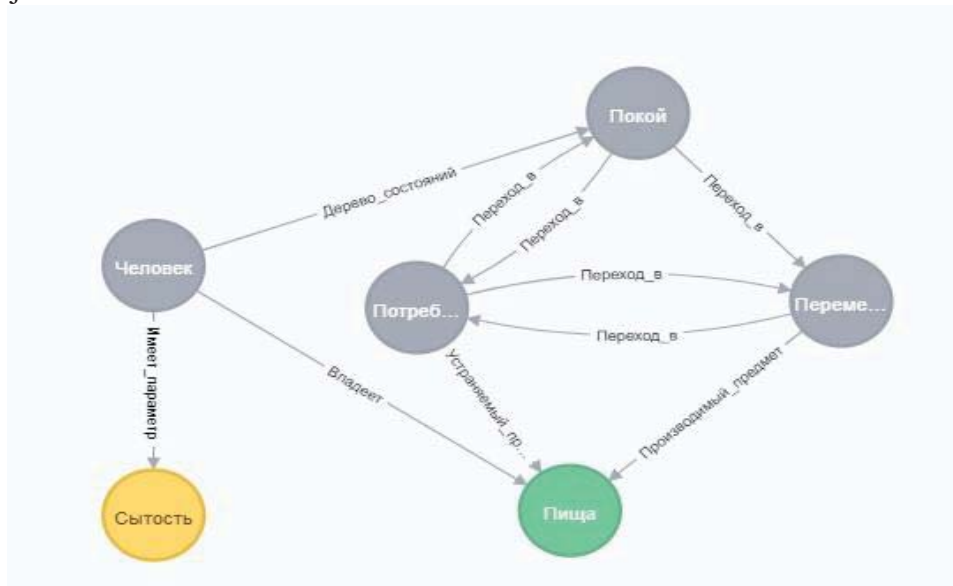


Рис. 5. Граф базы состояний в в Neo4j

Для того чтоб создать экземпляры объектов и/или агентов, необходимо задать количество создаваемых экземпляров, список и домены значений и вид распределения характеристик объектов и агентов, а также существующие отношения между ними.

Примеры результатов работы приложения генерации экземпляров на основе базы знаний представлены на рисунках 6 и 7.

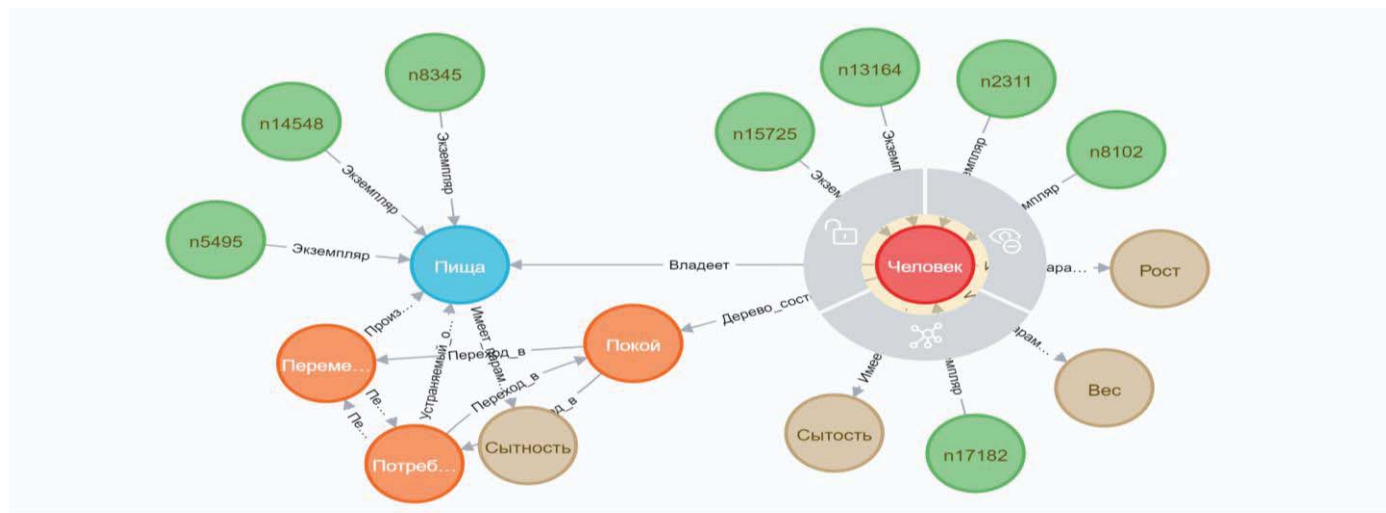


Рис.6. Классы и экземпляры связанных понятий

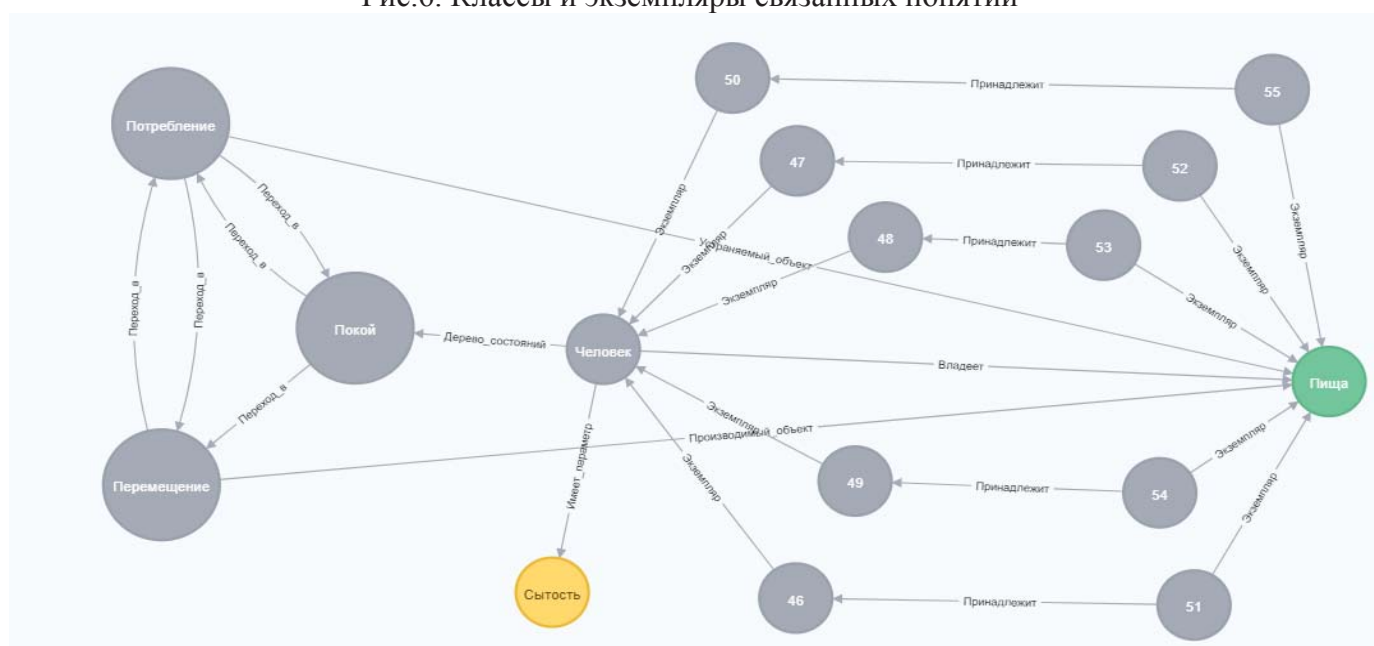


Рис. 7. Классы и экземпляры связанных понятий

Алгоритм работы движка представлен на рис. 8.

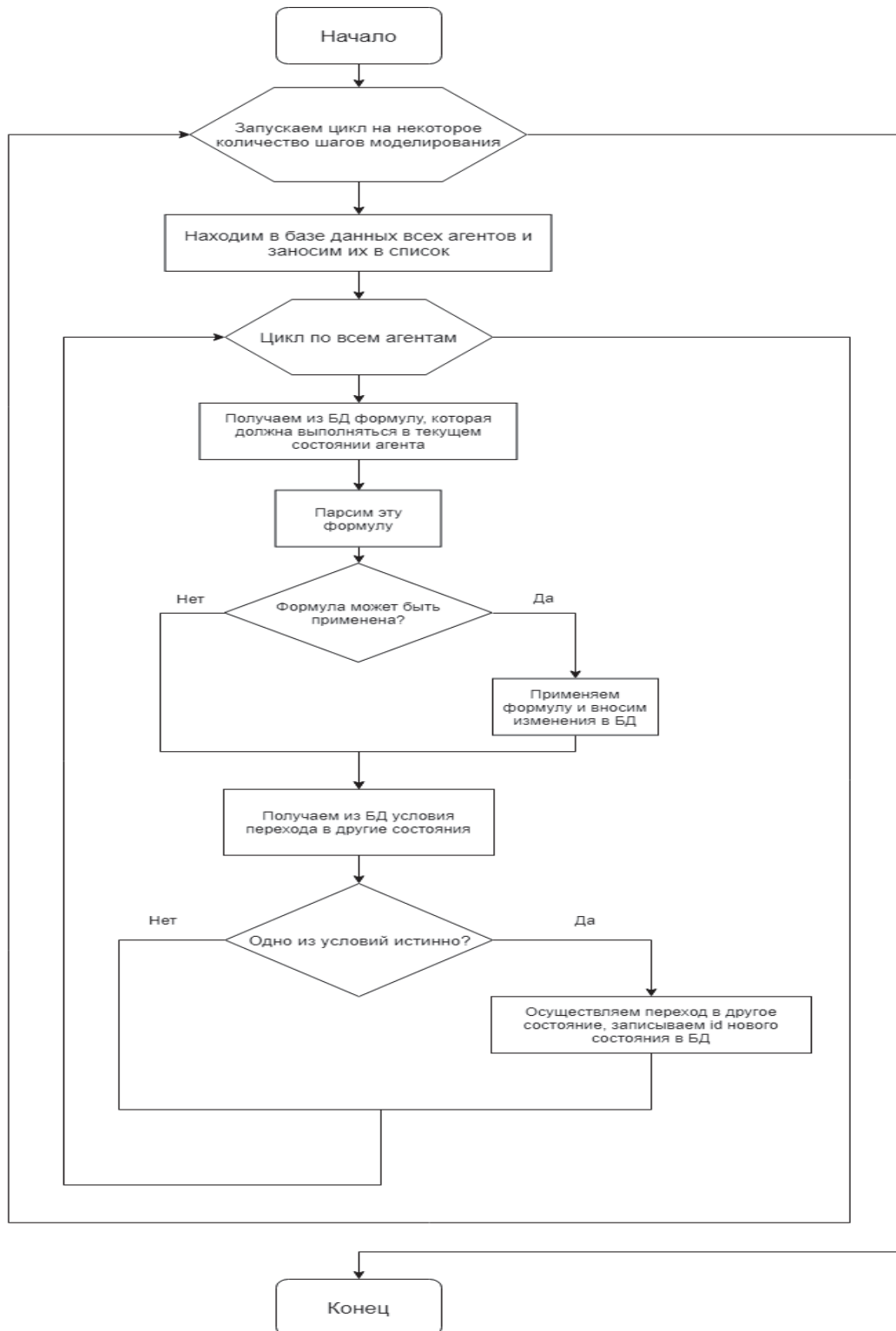


Рис. 8. Блок-схема алгоритма движка

Разработанный движок системы моделирования работает циклически. На каждой итерации алгоритма определяется список всех существующих на данный момент объектов в базе данных, и для каждого объекта выполняются действия согласно их состояниям, значениям атрибутов и принадлежащим объектам. После того, как для каждого агента будут выполнены операции, соответствующие текущему событию, проверены условия перехода и осуществлены все переходы в другие состояния, шаг моделирования завершается и начинается следующий. На рис.9 приведен пример выдачи результатов моделирования.

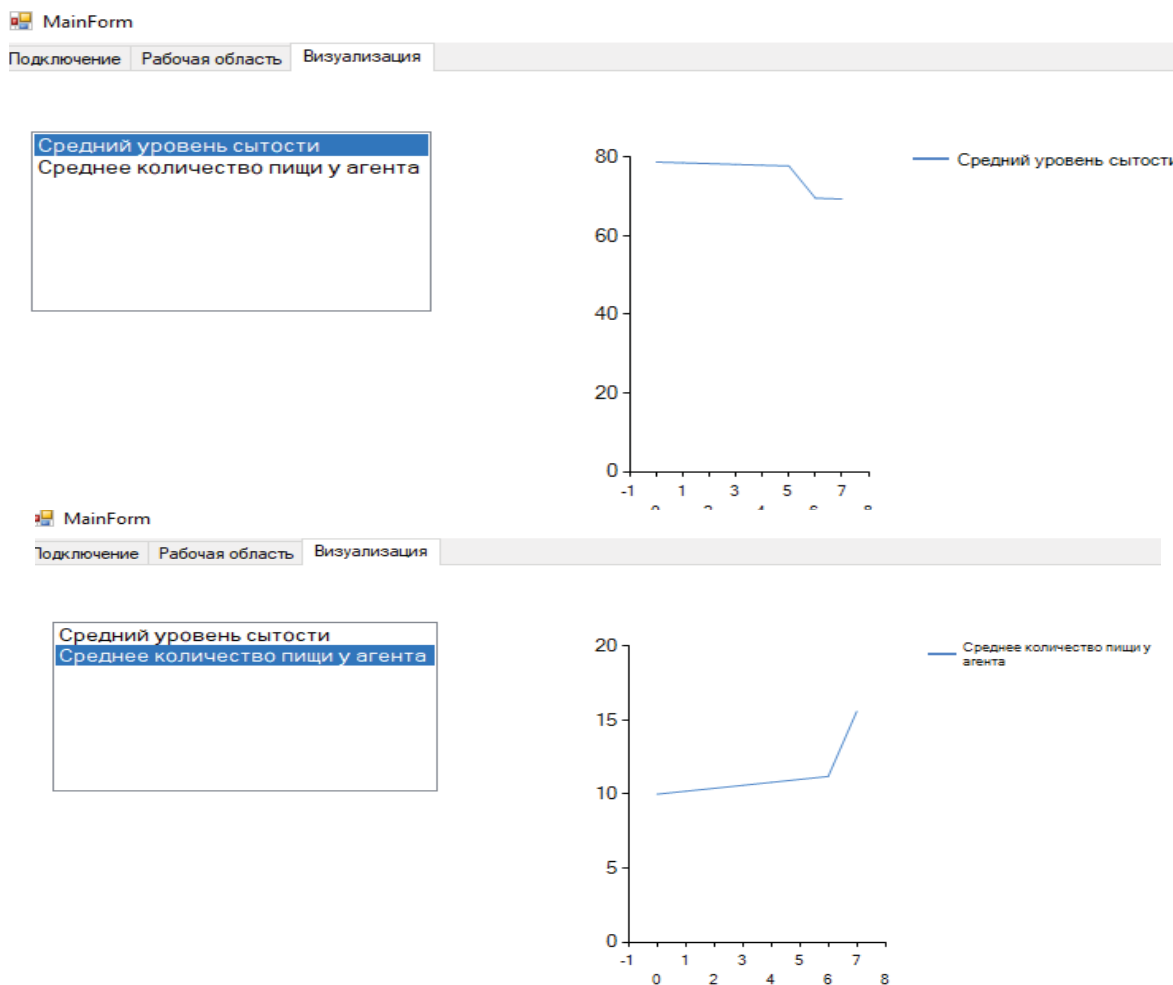


Рис. 9. Примеры выдачи результатов моделирования.

В результате проведенных исследований подтверждена возможность создания инструментария социально – поведенческого моделирования, основанного на графовом высокоуровневом представлении онтологии данной предметной области. Продемонстрированы удобства применения графовых баз данных для решения задач моделирования поведения множества связанных объектов.

Литература

1. Priority Challenges for Social and Behavioral Research and Its Modeling. Paul K. Davis, Angela O'Mahony, Timothy R. Gulden, Osonde A. Osoba, Katharine Sieck URL: https://www.rand.org/pubs/research_reports/RR2208.html
2. Клебанов Б.И., Антропов Т.В. Применение модели гибридного автомата для формализации поведения интеллектуальных агентов // Современные наукоемкие технологии. – 2019. – № 4. – С. 32-35; URL: <http://top-technologies.ru/ru/article/view?id=37487> (дата обращения: 18.09.2019).
3. Клебанов Б.И., Антропов Т.В. Учет ментальных свойств интеллектуальных агентов в модели гибридного автомата // Современные наукоемкие технологии. – 2019. – № 6. – С. 66-70; URL: <http://top-technologies.ru/ru/article/view?id=37551> (дата обращения: 18.09.2019).
4. Protégé URL: <https://protege.stanford.edu/> (Дата обращения: 18.09.2019)
5. Neo4j Developer manual URL: <https://neo4j.com/docs/developer-manual/current/cypher/> (Дата обращения: 18.09.2019)
6. Barrasa, J. Importing RDF data into Neo4j URL: <https://jbarrasa.com/2016/06/07/importing-rdf-data-into-neo4j/> (Дата обращения: 18.09.2019)