

Применение агентной модели интерактивной визуализации для создания средств визуального управления онтологическими данными

Д. С. Раздьяконов¹, Д. И. Муромцев²

Факультет Программной Инженерии и Компьютерных Технологий, Университет ИТМО, Санкт-Петербург, Россия

¹ ORCID: 0009-0009-9668-207X, ladone3@gmail.com

² ORCID: 0000-0002-0644-9242, d.muromtsev@gmail.com

Аннотация

В работе представлена агентная модель интерактивной визуализации и предложен способ её применения для разработки средств визуального управления онтологическими данными. Процесс взаимодействия пользователя с визуализацией представляется в виде графа состояний, где каждый узел является отдельной визуализацией, начальное состояние известно, а конечное формируется в процессе взаимодействия пользователя со средством визуального управления онтологическими данными. Каждая промежуточная визуализация графа состояний помещается в многомерное евклидово пространство данных, формируемое на основе визуализируемой онтологии, что даёт возможность рассчитать весовые коэффициенты на ребрах графа состояний и выполнить поиск по графу с использованием известных алгоритмов. Применение модели позволяет снизить трудоемкость выполнения пользовательских сценариев за счет сведения пошагового построения визуализаций к задаче поиска в пространстве состояний.

Ключевые слова: онтологии, управление онтологиями, методы визуализации онтологий, модели визуализации, интерактивная визуализация.

1. Введение

Эффективность процесса управления онтологическими данными (УОД) напрямую зависит от программных средств, которые обеспечивают данный процесс, где эффективность измеряется как в человеко-часах для бизнес-задач, так и во времени обработки SPARQL/UPDATE запросов для задач межсервисного взаимодействия.

В зависимости от целей управления для улучшения эффективности процесса УОД применяются различные программные средства. Для разработки малых онтологий, представленных в виде десятка зависимых файлов, инженеры данных предпочитают использовать Protégé [12] или WebVOWL [21], однако для управления онтологиями, которые относятся к LOD (Linked Open Data), используют web-платформы, такие как Metaphactory [10] или web-интерфейсы TopBraidComposer [25]. Такие платформы способны выполнять ленивые запросы к данным, хранящимся в общественных базах знаний (БЗ).

Важной составляющей коммерческих средств разработки онтологий является функция визуального управления онтологическими данными (ОД), которая значительно повышает эффективность процессов УОД, автоматизируя составление SPARQL-запросов и обеспечивая прозрачную работу со схемами данных онтологий.

Визуальное управление ОД – процесс управления ОД, выполняемый с использованием программных средств, совмещающих функции интерфейсов

«человек-машина» и «машина-человек». То есть таких, где средство визуализации ОД одновременно является и средством редактирования ОД.

При этом недостаточно, чтобы одно и то же средство УОД осуществляло две функции разом (см. Рис. 1). Например, программа, отображающая онтологический граф и предоставляющая форму для создания новых сущностей онтологии, не будет считаться средством визуального управления, если два графических интерфейса не будут связаны вместе, то есть не будут разделять общее состояние.

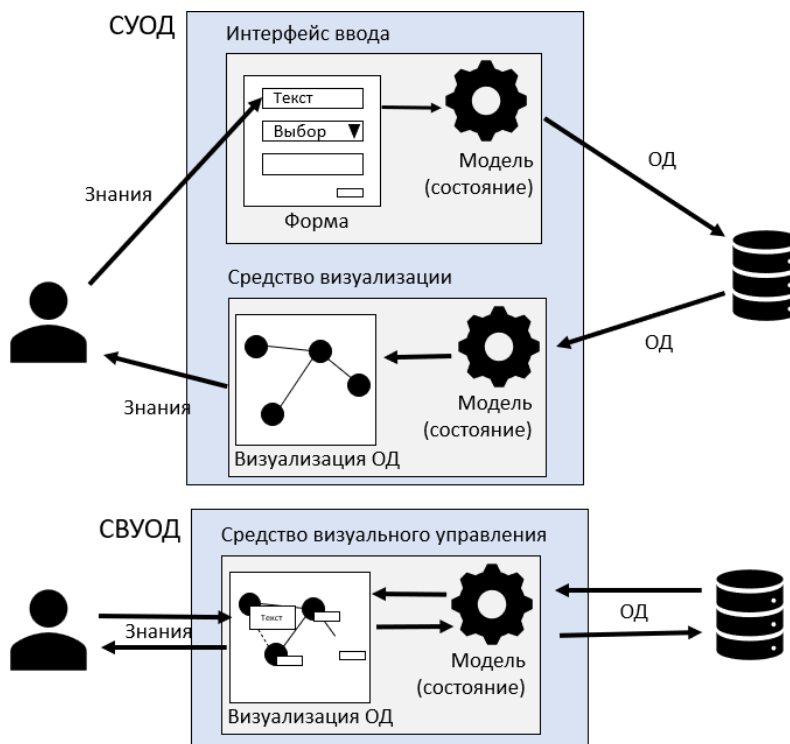


Рис. 1: Средство управления ОД (СУОД) и средство визуального управления ОД (СВУОД)

Сложностью при разработке средств визуального УОД (СВУОД) является то, что приходится решать задачи не только поставленные перед средствами редактирования ОД, но и задачи, поставленные перед инструментами визуализации ОД. Так, например, если возникает необходимость работы с ОД, которые содержат анонимные узлы, то средство должно иметь возможность не только отредактировать или создать структуры, содержащие анонимные узлы (ССАУ), но и должно быть способно предоставить соответствующую визуализацию для ССАУ.

При разработке СВУОД важен порядок взаимодействия пользователя с элементами управления. Если для СУОД порядок взаимодействия может изменяться свободно без привязки к визуализации, то для СВУОД, где одно и то же действие может быть выполнено несколькими способами, а порядок взаимодействия с элементами управления напрямую влияет на читабельность визуализации и эффективность взаимодействия, разработка может вызывать значительные затруднения.

В СВУОД часть визуализации неизбежно становится интерактивной, так как её состояние зависит от действий пользователя, поэтому в рамках нашей работы мы предлагаем использовать модель интерактивной визуализации для оптимизации процесса разработки СВУОД. В данном подходе модели ИВ позволяет формально описать целевую функцию процесса ИВ и обеспечить целенаправленность и ускорение процессов ВУОД.

Интерактивная визуализация (ИВ) – это визуализация, которая имеет изменяемое состояние, где состояние визуализации может изменяться как со

временем, так и в ответ на внешние события (операции ввода или изменения параметров визуализации).

2. Обзор предметной области

Существующие средства управления онтологическими данными так или иначе реализуют методы УОД, где методы разбиваются на методы разработки, визуализации и хранения. На данный момент существует много работ, таких как «Ontology visualization methods – a survey» [14], «Ontology visualization methods and tools: a survey of the state of the art» [7] и «Ontology visualization PROTE´GE´ tools—a review» [18], которые дают подробные и многосторонние обзоры на средства и методы визуализации ОД.

С другой стороны, есть такие работы, как «Ontology development methods» [9], «Methodologies for ontology development» [13], «A review on ontology development methodologies for developing ontological knowledge representation systems for various domains» [2] и «Methods for ontology development» [5], которые разбирают и сравнивают методики разработки онтологий. В дополнение к перечисленным методам есть работы «Managing ontologies: a comparative study of ontology servers» [1], «Hybrid method for storing and querying ontologies in databases» [20], «Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies» [15] и «On storing ontologies including fuzzy datatypes in relational databases» [3], которые рассматривают проблемы эффективного хранения ОД в БЗ и в других доступных форматах.

В то время как существует довольно много работ, посвященных средствам, поддерживающим процесс УОД, разработка подобных средств – это отдельная большая область знаний, которая редко связана с областью семантических технологий и ОД. С точки зрения разработки пользовательских интерфейсов могут быть интересны такие статьи, как «Integrating human-centered and model-driven methods in Agile UI development» [8] и «Paprika: Rapid UI Development of Scientific Dataset Editors for High Performance Computing» [16], а для погружения в область разработки средств интерактивной визуализации многомерных данных может быть полезна работа «Разработка системы интерактивного визуального анализа многомерных данных» [23].

3. Агентная модель интерактивной визуализации

В рамках данной работы мы формулируем подход к разработке СВУОД. Наш метод мы основываем на модели интерактивной визуализации. Опишем модель.

Состояние интерактивной визуализации меняется в ответ на действия пользователя, где действия пользователя могут быть направлены как на редактирование ОД, так и на изучение ОД.

Модель интерактивной визуализации построена вокруг процесса интерактивного поиска – наиболее востребованного «подпроцессом» управления при работе с большими данными в рамках порционной визуализации. Интерактивный поиск – это процесс, при котором пользователь изменяет состояние визуализации так, чтобы визуализация отвечала на заданный поисковый запрос. Подробно данная тема рассмотрена в работах «An ontology-driven visual question-answering framework» [4] и «VQASTO: Visual question answering system for action surveillance based on task ontology» [19].

Основная составляющая модели – это онтологический граф G , который является множеством троек «субъект-предикат-объект» $\langle s, p, o \rangle$ (1). Онтологический граф формирует многомерное пространство данных, измерения которого соответствуют области значений предикатов из онтологии, то есть измерение – это множество возможных объектов, встречающихся в триплетах с соответствующим предикатом.

U – Множество URI, где каждый URI определяет логический или физический ресурс.

B – Множество анонимных узлов.

L – Множество литеральных значений, таких как string, integer или boolean.

G – Онтологический граф.

$$s \in B \cup U; p \in U; o \in B \cup L \cup U. \quad (1)$$

Данные формируют пространство D (2), которое состоит из измерений D_i (Рис. 2).

$$D = \langle D_1, D_2, D_3, \dots, D_i \rangle \mid D_i = o \in (B \cup L \cup U) : \langle s, p, o \rangle \in G \mid i = p. \quad (2)$$

Пользователь на экране наблюдает пространство данных или его часть, где пространство данных проецируется на экран пользователя с помощью функции визуализации, которая на вход принимает положение и размер окна поиска в пространстве данных, а на выход отдаёт двухмерную статическую визуализацию.

Окно поиска SF – это область пространства данных, доступная пользователю для наблюдения или положение агента в пространстве данных. В процессе визуализации данные из области, ограниченной окном поиска, посредством функции визуализации $V_D(SF)$ преобразуются в набор параметров, которые поступают на вход в инструмент визуализации.

Точка поиска SP – позиция фокуса внимания пользователя в пространстве данных при использовании инструмента интерактивной визуализации.

Пользователь – человек, взаимодействующий с интерактивной визуализацией через человеко-машинный интерфейс (монитор/VR-очки + инструмент визуализации).

Агент – проекция пользователя внутрь инструмента визуализации, то есть абстракция, описывающая состояние инструмента визуализации на каждом шаге интерактивной визуализации.

$Агент = SF$.

$IVis_D$ – интерактивная визуализация пространства данных D (Рис. 3) представляет собой кортеж из двух элементов: функция визуализации данных $V_D(SF)$ и функция управления визуализацией $T_D(SF, INPUT)$. Где SF – это окно поиска, $INPUT$ – набор команд от интерфейса инструмента визуализации, а SF' – новое положение окна поиска. v_D – это набор параметров, которые передаются на вход инструменту визуализации.

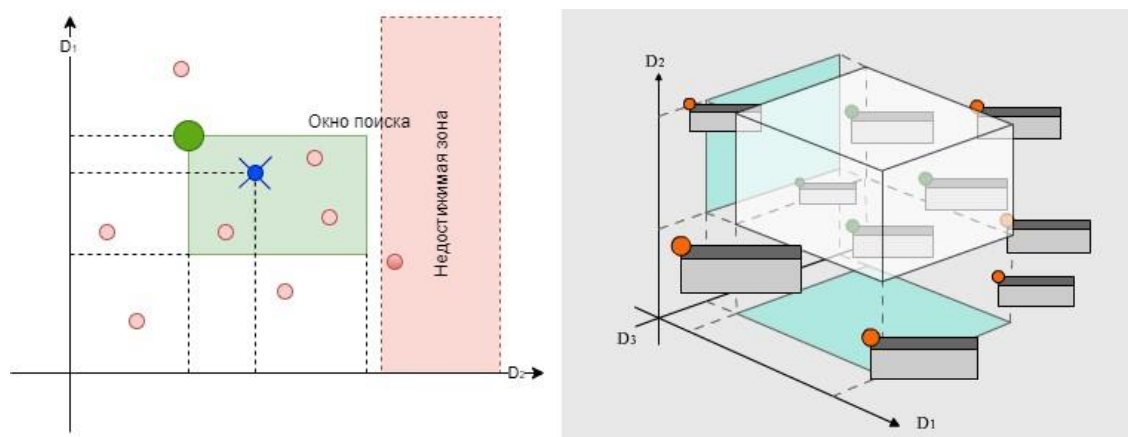


Рис. 2: 2D/3D пространства данных

В нашем случае v_D – это набор графических примитивов, а также параметры камеры и настройки освещения, поступающие на обработку в графический движок.

Выполняя заданное число простейших действий (T_{sg}), пользователь с помощью инструментов формирует команды к интерактивной визуализации, которые изменяют положение окна поиска (SF), тем самым изменяя состояние визуализации.

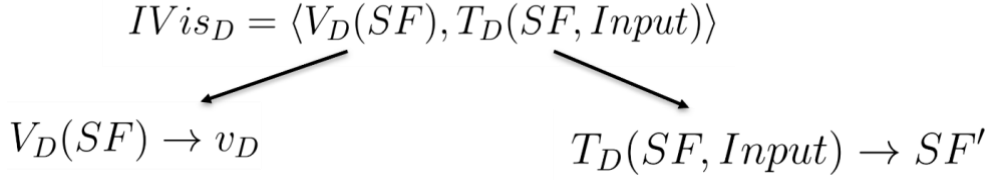


Рис. 3: Интерактивная Визуализация пространства данных D

Исследуя результат визуализации, пользователь изменяет позицию фокуса внимания (SP).

Если принять число операций ввода (T_s) как время, а дистанцию в пространстве (S_s) данных (d_i) как путь, мы можем посчитать скорость перемещения агента в пространстве данных.

$$V_s = \frac{\sqrt{(d_{1a}-d_{1b})^2 + (d_{2a}-d_{2b})^2 + \dots + (d_{na}-d_{nb})^2}}{\sum_{g=1}^m T_{sg}} = \frac{S_s}{T_s}, \quad (3)$$

a, b – начальная и конечная позиция в пространстве данных; d_n – координата позиции в многомерном пространстве; m – число шагов поиска.

Главным преимуществом нашей модели является то, что она позволяет свести пошаговое построение визуализации к задаче поиска в пространстве состояний, формируемом на основе многомерных онтологических данных и атомарных действий инструментов визуализации. Модель описывает функцию процесса интерактивной визуализации в терминах скорости, трудоемкости и расстояния, что позволяет обеспечить целенаправленность и ускорение интерактивного поиска по онтологическим базам знаний в условиях ограниченной наблюдаемости пространства данных.

Процесс, который мы сводим к задаче поиска в пространстве состояний, – это процесс интерактивной визуализации, то есть пошаговое построение визуализации. Соответственно целевое состояние – это набор значений параметров визуализации.

С точки зрения модели конечное и начальное состояния, то есть конечная и начальная визуализации, отличаются положением окна поиска в пространстве данных, где положение описывается точкой в многомерном пространстве данных, а также шириной окна по каждому измерению. Из этого следует, что оптимизация может идти по двум параметрам: по ширине и по положению окна поиска в многомерном пространстве. Цель оптимизации – это минимум функции расстояния между целевым положением окна поиска и текущим положением окна поиска в пространстве данных, а также минимум разности между искомой шириной окна поиска по каждому измерению и текущей шириной окна поиска для этих измерений. Если рассматривать ширину окна поиска при расчёте целевой функции как набор дополнительных измерений, мы получаем пространство из $2n$ измерений и функцию дистанции в качестве целевой функции.

Целевая функция, которая используется для оптимизации процесса поиска в пространстве состояний, представляет собой евклидово расстояние в пространстве размерностью $2n$ пространства данных, между двумя векторами (p, q), описывающими целевое состояние и текущее.

$$f(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{2n} (p_i - q_i)^2}. \quad (4)$$

Каждый вектор имеет n значений координат измерений в пространстве данных и n значений ширин для этих же измерений, то есть $2n$ значений. Поскольку процесс, который мы сводим к задаче поиска в пространстве состояний – это процесс пошаговой визуализации, каждый вектор состояния описывает состояние визуализации на одном шаге, то есть описывает набор значений параметров, которые были использованы для построения визуализации, где визуализация – это набор графических примитивов,

отображающих часть пространства данных. При этом целевая визуализация в начале поиска может быть неизвестна и формироваться в ходе интерактивной работы пользователя с инструментами визуализации. Если описывать процесс с точки зрения пространства состояний, то можно сказать, что путь к целевому состоянию на момент начала поиска неизвестен и вычисляется в ходе эвристического поиска.

Пространство состояний представляется четвёркой $[N, A, S, GD]$, где

N – множество последовательных визуализаций или множество положений окна поиска в пространстве данных.

A – множество шагов ввода (изменения параметров визуализации) в процессе построения целевой визуализации.

S – непустое множество начальных состояний, то есть возможные стартовые позиции окна поиска.

GD – непустое множество целевых состояний, то есть набор конечных визуализаций, которые могут описываться одним из следующих способов:

- Измеряемыми свойствами построенных визуализаций, встречающихся в процессе поиска. (Например, количество элементов визуализации).
- Порядком и характером простейших действий ввода, с помощью которых была получена целевая визуализация.

Допустимый путь – это путь из состояния множества S в состояние из множества GD . То есть набор шагов ввода и состояний визуализации (положений окна поиска), которые последовательно были осуществлены в процессе поиска целевой визуализации.

Переходы в пространстве состояний соответствуют шагам процесса построения визуализации и описывают возможность перехода из одного состояния визуализации в другое.

Параметры визуализации – положение и ширина окна поиска в пространстве данных по каждому измерению данных (состояние при визуализации проецируются в набор графических примитивов).

Таблица 1: Состояния ИВ и их параметры

Состояние визуализации	Координаты (hasColor, hasSugar)	Ширина окна поиска (hasColor, hasSugar)	Дистанция до цели
Состояние визуализации 1	(1, 1)	(3, 4)	4.123
Состояние визуализации 2	(1, 1)	(1, 4)	3.605
Состояние визуализации 3	(1, 3)	(1, 1)	0
Состояние визуализации 1.1	(1, 1)	(2, 4)	3.741
Состояние визуализации 2.1	(1, 4)	(1, 1)	1

Таблица 2: Расчёт длин пройденных путей

Путь	Общая трудоемкость	Длина пути
1(1) → C2(2) → 3	3 + 3 = 6	2 + 3.605 = 5.605
1(1.1) → 1.1 → (1.2) → C2(2) → 3	4 + 3 + 3 = 10	1 + 1 + 3.605 = 5.605
1(1) → C2(2.1) → 2.1	3 + 3 = 6	2 + 4.242 = 6.242

Модель описывает пространство состояний и пространство данных, в котором расположен граф состояний, а также скорость, с которой можно двигаться от одного состояния (узла графа) к другому с использованием доступных элементов пользовательского интерфейса. Это позволяет свести задачу разработки средства визуализации к поиску оптимального пути на взвешенном графе состояний, где вес на ребрах – это трудоемкость перехода из одного состояния в другое, которая

рассчитывается на основе скорости и расстояния в пространстве данных. Для решения задачи поиска можно использовать различные алгоритмы поиска и функции оптимизации.

4. Пример расчёта пути в двумерном пространстве состояний

Приведем пример процесса интерактивного поиска, и как его описывает наша модель. В качестве целевой онтологии взята урезанная онтология вин, содержащая по два атрибута для каждого вина (цвет и степень содержания сахара).

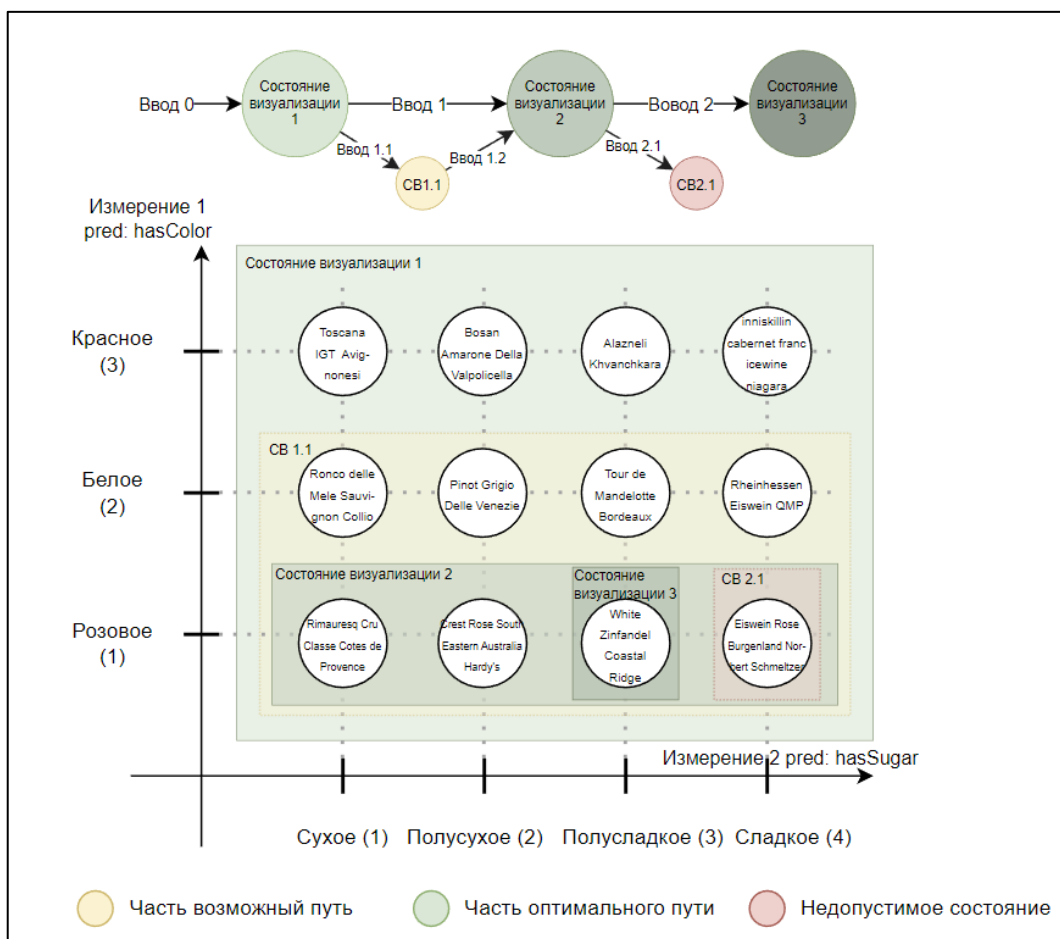


Рис. 4: Двухмерное пространство данных и граф состояний визуализации

Целью поиска является вино со следующими параметрами: {цвет} – «розовое», {содержание сахара} – «полусладкое». Пользователю необходимо изменить состояние ИВ так, чтобы на экране были только те вина, которые удовлетворяют условиям поиска. Пространство данных и граф состояний визуализации представлены на рисунке 4. Пространство содержит 12 различных экземпляров класса «Вино». Основные параметры состояния интерактивной визуализации представлены в таблицах 1, 2, а визуализация состояния поиска с использованием инструмента Ontodiazd [6] представлена на рисунке 5. Анимированный процесс взаимодействия с инструментом Ontodiazd в VR представлен на рисунке 6.

Данный пример показывает, как представленная нами модель описывает пространство данных и граф состояний интерактивной визуализации, что позволяет нам свести пошаговое построение визуализации к задаче поиска в пространстве состояний, которое формируется на основе двух параметров (в данном примере), описывающих все классы онтологии. Подобное описание позволяет отбросить

неоптимальные пути формирования целевой визуализации и выявить узкие места интерфейса средства визуализации.



Рис. 5: Состояния визуализации 1 – 4

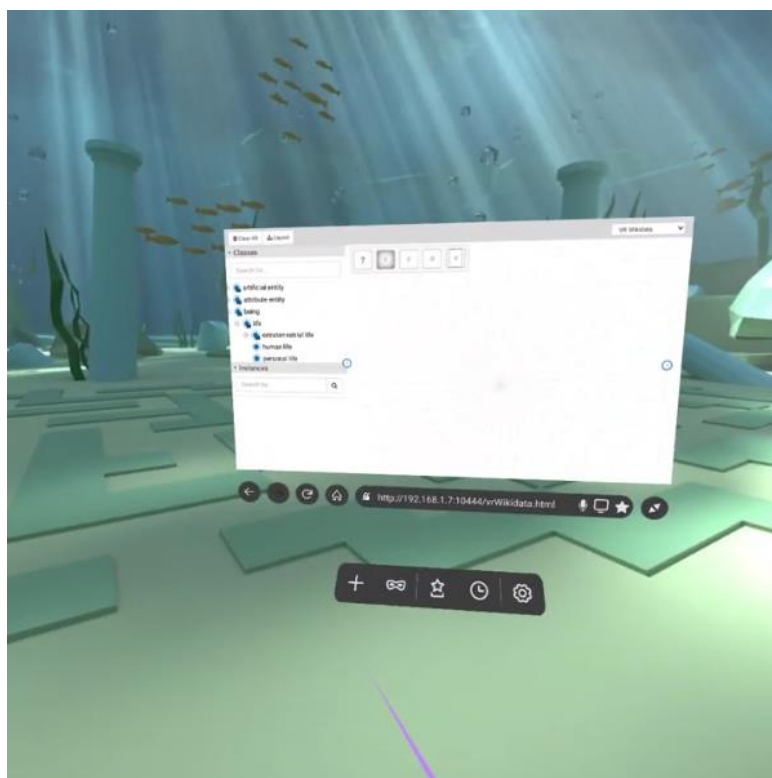


Рис. 6: Процесс взаимодействия с Ontodia3d в VR

5. Метод визуального управления онтологическими данными

Ранее мы определили, что метод – это технический способ разработки средств визуального управления ОД. Наш метод мы основываем на модели интерактивной визуализации и предлагаем активно применять её на этапе разработки интерфейса средств визуального управления ОД.

Важной характеристикой нашей модели является то, что модель описывает граф состояний ИВ и позволяет построить целевую функцию процесса интерактивной визуализации (4).

Шаги метода:

1. Формирование целевого набора данных для тестирования выбранных сценариев и составление пространства данных на его основе. Целевой набор данных, согласно модели – это граф $G = \langle s, p, o \rangle$.

(a) Выделение всех возможных свойств элементов онтологии, по которым может вестись интерактивный поиск. То есть выделение измерений D_i из графа G , где каждое измерение $D_i = o \in (B \cup L \cup U) : \langle s, p, o \rangle \in G \mid i = p$.

(b) Формирование многомерного пространства данных онтологии, где количество измерений пространства равно количеству выделенных в предыдущем шаге свойств $D = \langle D_1, D_2, D_3, \dots D_i \rangle$.

Формирование пространства по каждому измерению для различных сценариев использования – это отдельная большая тема для исследования, однако в рамках нашей визуализации, где визуализация производится в виде трёхмерного графа (инструмент описан в работе [6]) с семантическим распределением узлов в пространстве визуализации, мы предлагаем использовать word embedding алгоритмы, которые подробно описаны в статьях «Using word embeddings for visual data exploration with ontodia and wikidata» [22] и «Word embeddings as metric recovery in semantic spaces» [11].

2. Составление набора сценариев использования (C) средства УОД. $C = \langle C_1, C_2, \dots C_i \rangle$.

3. Составление графа состояний ИВ (SG). На этом шаге вычлняются основные состояния интерфейса средства УОД, и составляется граф состояний, где вершины графа – это состояния ИВ, которые имеют фиксированное положение в пространстве данных, а ребра обозначают возможности перехода между состояниями $SG(V, E) = \langle V, E \rangle$ (V – множество вершин, а E – множество рёбер). Для вычисления весовых коэффициентов на ребрах графа используется функция евклидова расстояния (4).

4. Построение наборов допустимых путей в графе состояний для каждого сценария использования $P = \langle P_{c1}, P_{c2} \dots P_{ci} \rangle \mid P_{ci} = \langle (v_1, v_2), (v_2, v_3) \dots (v_{i-1}, v_i) \rangle$, а также определение начальных $S = \langle \dots S_{ci} \dots \rangle$ и конечных состояний для каждого сценария $GD = \langle \dots GD_{ci} \dots \rangle$. На этом этапе могут применяться стандартные алгоритмы поиска путей в графе, например, алгоритм Дейкстры или алгоритм A^* .

5. Расчет трудоемкости прохождения возможных путей с учётом весовых коэффициентов $L(P_{ci}) = Difficulty(v_1, v_2) + \dots + Difficulty(v_{i-1}, v_i)$.

6. Сортировка путей по длинам и удаление избыточных путей

$RemoveDuplicates(Sort(L)) \mid L = \langle L(P_{c1}) \dots L(P_{ci}) \rangle$.

Таким образом можно исключить избыточные элементы интерфейса и оптимизировать работу пользователя с интерфейсом СУОД.

Для случаев, когда невозможно определить фиксированный набор состояний визуализации, например положение камеры в пространстве состояний имеет тысячи возможных позиций, можно оценивать средства визуализации по скорости перемещения в пространстве состояний. В рамках этого подхода необходимо выполнить следующие шаги:

1. Составить перечень элементов управления.

2. Рассчитать скорость перемещения в пространстве состояний для каждого элемента управления и определить зону пространства данных, в которой может использоваться данный инструмент.

3. Исключить дублирующие инструменты, которые позволяют перемещаться в одной и той же зоне пространства данных.

Интересным также является возможность использования математической оптимизации для определения потенциальных переходов между состояниями, когда элементы интерфейса, позволяющие осуществить переход между состояниями, ещё не

реализованы. В таких случаях оптимизация может показать оптимальный путь между состояниями, чтобы программисты затем могли реализовать элементы интерфейса, позволяющие осуществить переходы.

В результате применения метода к средству визуализации ОД в трехмерном и VR пространствах [6] средняя трудоемкость использования средства снизилась на 21.8% (Рис. 7). Где под трудоемкостью понимается количество простейших действий (клик кнопки мыши, прокрутки колёсиком, ввод символа с использованием клавиатуры), которое должен совершить пользователь, чтобы выполнить сценарий. На графике представлены средние значения трудоемкостей, полученные в ходе испытаний, поэтому числа на рисунке 7 дробные.

Уменьшение трудоемкости было достигнуто за счёт удаления неоптимальных путей из графа состояний ИВ согласно предложенному методу. Процесс взаимодействия с панелью фильтров (Рис. 8) был оптимизирован за счёт упорядочивания шагов ввода средствами пользовательского интерфейса, что сократило путь в пространстве данных. При первом сценарии пользователям было разрешено применять фильтры в произвольном порядке, а после изменения применение фильтров было приведено к строгому порядку. На рисунке 9 представлены графики сближения с целью (целевым состоянием). Графики построены только для шагов, связанных с применением фильтров.

Второй составляющей, которая была подвергнута оптимизации, была панель связей элемента (рис. 10).

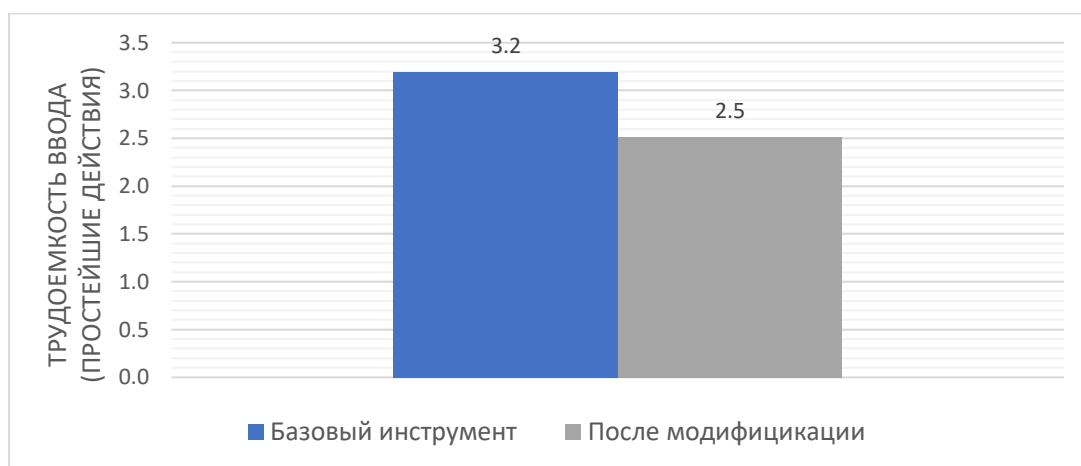


Рис. 7: Трудоемкость использования средства УОД до и после применения метода

Изначально панель выдавала полный перечень всех связанных элементов, то есть отображала все измерения, в которых элемент имеет позицию, отличную от нуля, что, с точки зрения позиции окна поиска в пространстве данных, выглядело как отображение всего пространства вокруг целевого элемента, а затем фокусировка на определенной позиции в этом подпространстве. Мы оптимизировали интерфейс так, чтобы пользователь сначала выбирал интересующий его тип связи, то есть измерение, а затем значение в этом измерении.

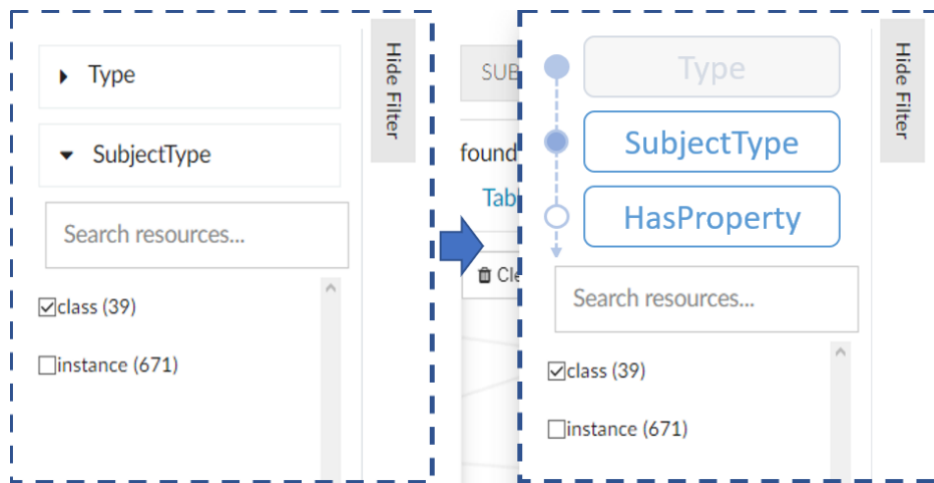


Рис. 8: Панель фильтров до и после оптимизации

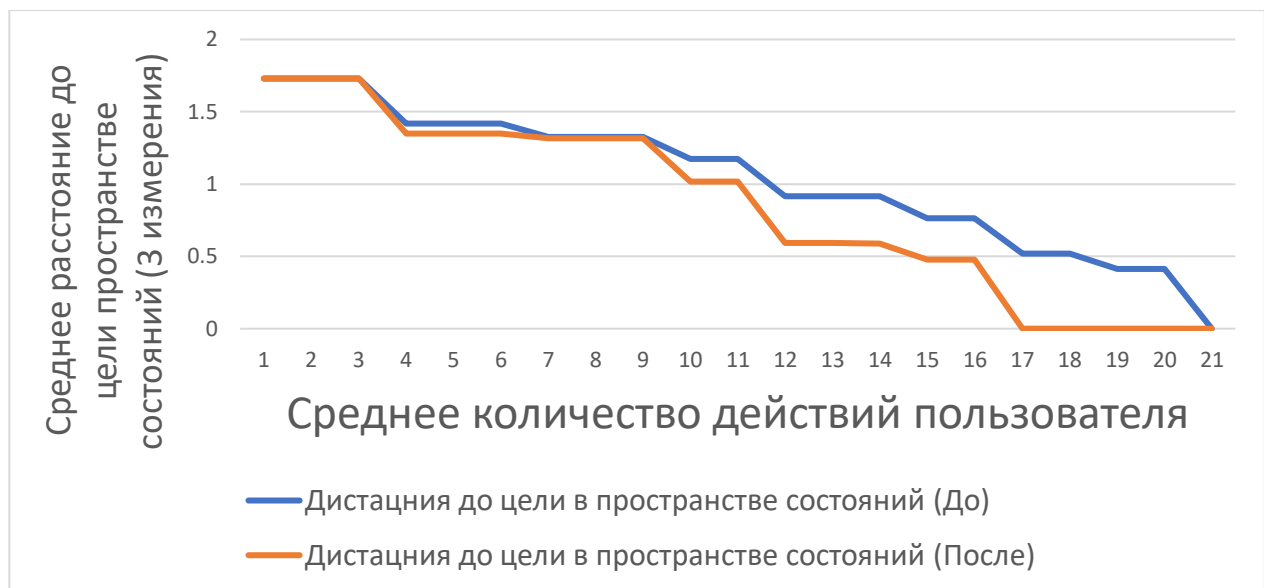


Рис. 9: Графики сближения с целью (целевым состоянием)

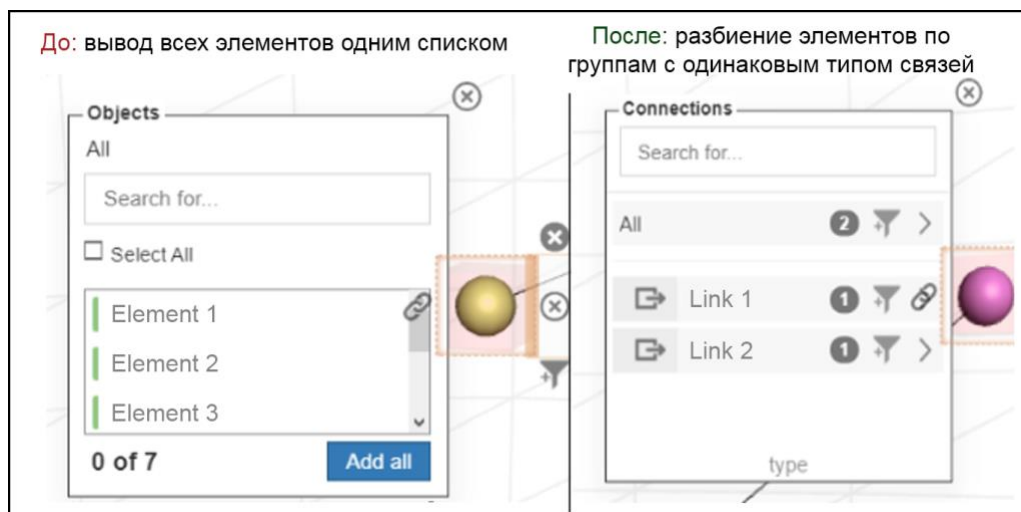


Рис. 10: Панель связей элемента до и после оптимизации

После расчёта путей для первого и второго варианта интерфейса новый путь в пространстве состояний оказался короче. На рисунке 11 представлен взвешенный граф состояний с весовыми коэффициентами, обозначающими дистанцию в пространстве

состояний между узлами (визуализациями) (граф построен с использованием сервиса <https://graphonline.ru>).

6. Инструменты

Процесс разработки и применения метода осуществлялся при помощи коммерческого продукта Metaphactory [10] с закрытой лицензией (<https://metaphacts.com/metaphactory-software-license-agreement>). Платформа Metaphactory предоставляет возможность модифицировать пользовательский интерфейс web-приложения в режиме реального времени. На основе описанного метода был разработан вспомогательный компонент, который осуществляет сбор и обработку статистических данных, и позволяет давать оценку конфигурации пользовательского интерфейса в реальном времени. На данный момент компонент находится в статусе прототипа и не включён в релизную сборку.

Основным компонентом визуализации онтологических графов Metaphactory является компонент Ontodia, упомянутый в работах [17, 24]. Компонент осуществляет визуализацию ОД в виде двухмерного графа, однако для проверки гипотез был разработан прототип компонента, который осуществляет визуализацию в виде трёхмерного графа. Публичная версия компонента Ontodia (GNU General Public License) доступна по ссылке <https://github.com/metaphacts/ontodia>. Реализация Ontodia3d использует функционал Metaphactory, однако базовая часть компонента вынесена в открытый репозиторий <https://github.com/Ladone3/L3-graph> и распространяется свободно.

В рамках работы проводились замеры параметров трудоемкости и скорости перемещения в пространстве данных при использовании компонентов Ontodia и Ontodia3d, на основе которых формировались соответствующие layout-алгоритмы, минимизирующие трудоемкость построения визуализаций.

Иной сценарий использования предполагает, что метод может быть применен для разработки СВУОД с нуля в рамках циклической модели разработки, где на каждом цикле разработки применяются шаги метода и модифицируется интерфейс.

7. Заключение

В рамках данной работы мы даём описание понятий и определений, лежащих в основе процессов управления ОД и ИВ, и описываем агентную модель интерактивной визуализации. Основным направлением для продолжения исследования является разработка метрик ИВ для оценки средств управления ОД.

В работе «A new tool for linked data visualization and exploration in 3D/VR space» [6] на основе описанной модели ИВ мы строим средство визуализации ОД в 3D и VR пространствах для осуществления процесса интерактивного поиска (data exploration) и используем это средство для визуального редактирования ОД. Модель интерактивной визуализации и построенное на её основе средство визуального управления лежат в основе метода визуального управления ОД, описанного в данной работе.

В рамках данного метода был также разработан подход к визуализации анонимных узлов, который описан в статьях «Approach to Blank Node Processing in Incremental Data Visualization by the Example of Ontodia» [17] и «Подход к обработке пустых узлов при порционной визуализации данных на примере инструмента ontodia» [24]. Данный подход к визуализации позволяет устранить узкие места методов визуального управления, то есть позволяет визуально управлять (визуализировать и редактировать) ОД, содержащих анонимные узлы.

Идентификаторы анонимных узлов нельзя использовать в SPARQL и UPDATE запросах, что делает невозможным автоматическое построение визуализаций фрагментов графа, содержащих анонимные узлы, а также пошаговое редактирование структур, содержащих анонимные узлы. Однако описанный в [17, 24] подход

предполагает использование контекстно зависимых идентификаторов для анонимных узлов, которые могут быть использованы для автоматизированного составления SPARQL и UPDATE запросов, а также для пошаговой визуализации CCAU.

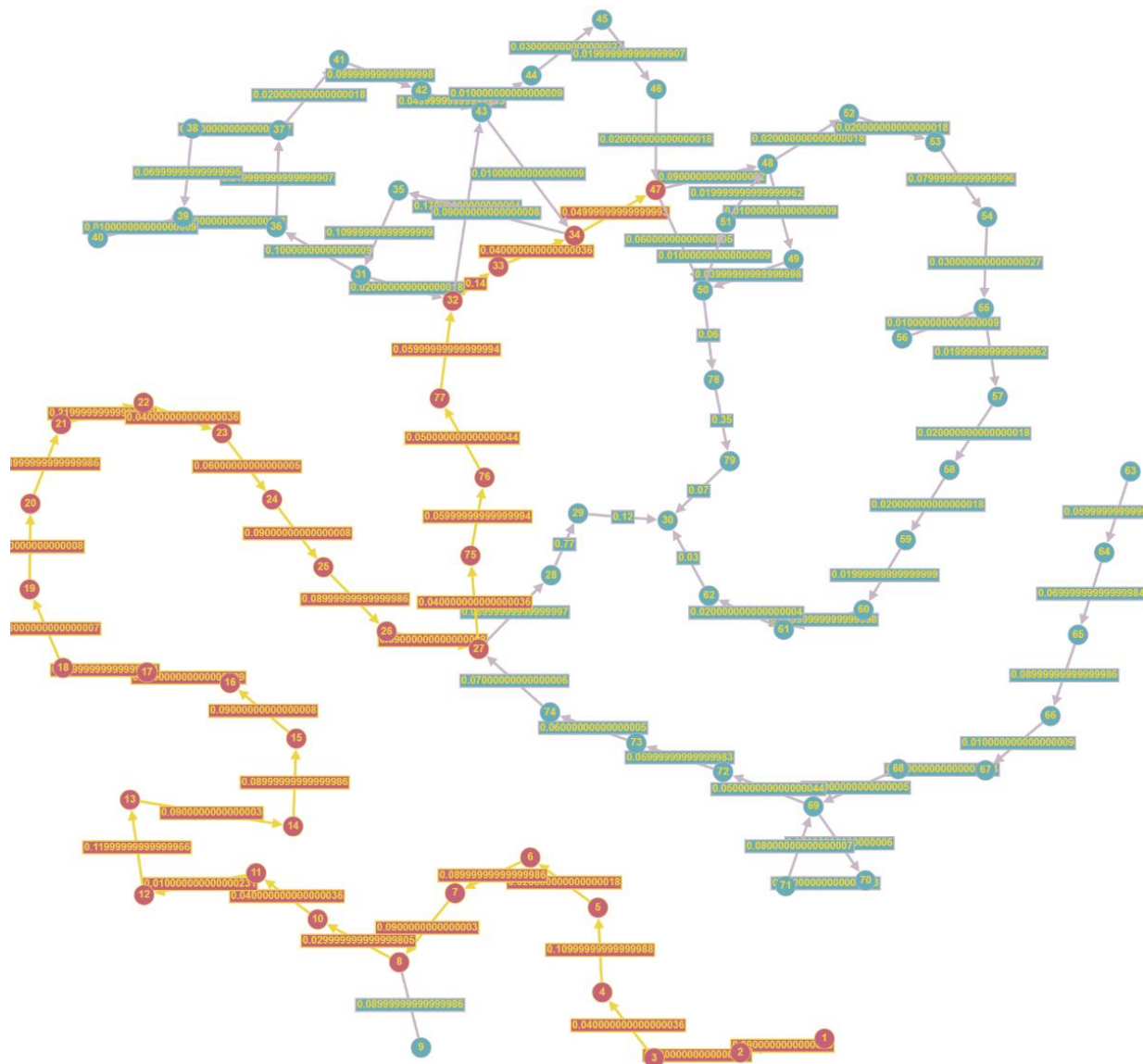


Рис. 11: Путь во взвешенном графе состояний

Другим естественным направлением для продолжения работ по данной тематике является исследование возможностей применения языковых моделей с механизмами внимания для вычисления наиболее релевантных для пользователя данных. Иными словами, применение нейросетей для автоматизации расчёта размера окна поиска. Размер окна поиска напрямую влияет на количество данных, попадающих на экран пользователя. При слишком маленьком размере окна поиска теряется связность данных, а при слишком большом размере – данные могут восприниматься как шум. Размер окна поиска является узким местом инструментов визуализации и визуального управления ОД, и автоматизированный расчёт способен повысить эффективность процессов управления ОД.

8. Термины и сокращения

Управление Данными (УД) – процесс синхронизации данных между несколькими ИД с участием человека.

Управление Онтологическими Данными (УОД) – управление данными, которые представлены в онтологическом виде.

Интерактивная визуализация (ИВ) – это визуализация, имеющая изменяемое состояние, где состояние визуализации может изменяться как со временем, так и в ответ на внешние события (операции ввода или изменение параметров визуализации). В рамках нашей работы мы предлагаем использовать модель интерактивной визуализации для оптимизации процесса разработки СВУОД.

Визуальное управление ОД (ВУОД) – процесс управления, который выполняется с использованием программных средств УОД, совмещающих функции интерфейсов «человек-машина» и «машина-человек», то есть таких, где средство визуализации ОД одновременно является и средством редактирования ОД.

ОД – Онтологические данные.

СУОД – Средство управления онтологическими данными.

СВУОД – Средство визуального управления онтологическими данными.

ССАУ – Структуры, содержащие анонимные узлы.

Список литературы

1. Ahmad M. N., Colomb R. M. Managing ontologies: a comparative study of ontology servers //ADC. – 2007. – Т. 7. – С. 13-22.
2. Aminu E. F. et al. A review on ontology development methodologies for developing ontological knowledge representation systems for various domains. – 2020.
3. Barranco C. D. et al. On storing ontologies including fuzzy datatypes in relational databases //2007 IEEE International Fuzzy Systems Conference. – IEEE, 2007. – С. 1-6.
4. Besbes G., Baazaoui-Zghal H., Ghezala H. B. An ontology-driven visual question-answering framework //2015 19th International Conference on Information Visualisation. – IEEE, 2015. – С. 127-132.
5. Breitman K. K., Casanova M. A., Truszkowski W. Methods for ontology development //Semantic Web: Concepts, Technologies and Applications. – 2007. – С. 155-173.
6. Daniil, R., Wohlgenannt, G., Pavlov, D., Emelyanov, Y., Mouromtsev, D. A new tool for linked data visualization and exploration in 3D/VR space //The Semantic Web: ESWC 2019 Satellite Events: ESWC 2019 Satellite Events, Portorož, Slovenia, June 2–6, 2019, Revised Selected Papers 16. – Springer International Publishing, 2019. – С. 167-171.
7. Dudáš M. et al. Ontology visualization methods and tools: a survey of the state of the art //The Knowledge Engineering Review. – 2018. – Т. 33. – С. e10.
8. Fischer H., Yigitbas E., Sauer S. Integrating human-centered and model-driven methods in Agile UI development //Proceedings of 15th IFIP TC. 13 International Conference on Human-Computer Interaction (INTERACT), S. – 2015. – С. 215-221.
9. Gokhale P., Deokattey S., Bhanumurthy K. Ontology development methods //DESIDOC Journal of Library & Information Technology. – 2011. – Т. 31. – №. 2.
10. Haase P. et al. metaphactory: A platform for knowledge graph management //Semantic Web. – 2019. – Т. 10. – №. 6. – С. 1109-1125.
11. Hashimoto T. B., Alvarez-Melis D., Jaakkola T. S. Word embeddings as metric recovery in semantic spaces //Transactions of the Association for Computational Linguistics. – 2016. – Т. 4. – С. 273-286.
12. Horridge M. et al. Webprotégé: A cloud-based ontology editor //Companion Proceedings of The 2019 World Wide Web Conference. – 2019. – С. 686-689.
13. Jones D., Bench-Capon T., Visser P. Methodologies for ontology development. – 1998.
14. Katifori A. et al. Ontology visualization methods—a survey //ACM Computing Surveys (CSUR). – 2007. – Т. 39. – №. 4. – С. 10-es.

15. Morente-Molinera J. A. et al. Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies //Information Sciences. – 2016. – Т. 328. – С. 418-434.
16. Nassiet D. et al. Paprika: Rapid UI development of scientific dataset editors for high performance computing //International SDL Forum. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. – С. 69-78.
17. Razd'yakonov D. S. et al. Approach to Blank Node Processing in Incremental Data Visualization by the Example of Ontodia //Programming and Computer Software. – 2020. – Т. 46. – С. 384-396.
18. Sivakumar R., Arivoli P. V. Ontology visualization PROTÉGÉ tools—a review //International Journal of Advanced Information Technology (IJAIT) Vol. – 2011. – Т. 1.
19. Vo H. Q., Phung T. H., Ly N. Q. VQASTO: Visual question answering system for action surveillance based on task ontology //2020 7th NAFOSTED Conference on Information and Computer Science (NICS). – IEEE, 2020. – С. 273-279.
20. Vysniauskas E., Nemuraite L., Paradauskas B. Hybrid method for storing and querying ontologies in databases //Elektronika ir Elektrotechnika. – 2011. – Т. 115. – №. 9. – С. 67-72.
21. Wiens V., Lohmann S., Auer S. WebVOWL Editor: Device-Independent Visual Ontology Modeling //ISWC (P&D/Industry/BlueSky). – 2018.
22. Wohlgenannt G. et al. Using word embeddings for visual data exploration with ontodia and wikidata //arXiv preprint arXiv:1903.01275. – 2019.
23. Масленников О. П. и др. Разработка системы интерактивного визуального анализа многомерных данных //Научная визуализация. – 2014. – Т. 6. – №. 4. – С. 30-49.
24. Раздьяконов Д. С. и др. ПОДХОД К ОБРАБОТКЕ ПУСТЫХ УЗЛОВ ПРИ ПОРЦИОННОЙ ВИЗУАЛИЗАЦИИ ДАННЫХ НА ПРИМЕРЕ ИНСТРУМЕНТА ONTODIA //Программирование. – 2020. – №. 6. – С. 16-29.
25. Alatrish E. S. Comparison some of ontology //Journal of Management Information Systems. – 2013. – Т. 8. – №. 2. – С. 018-024.

Application of an Agent-Based Model to Develop Ontological Data Visual Management Tool

D. S. Razdyakonov¹, D. I. Muromcev²
Faculty of Software Engineering and Computer Technologies, ITMO University,
St. Petersburg, Russia

¹ ORCID: 0009-0009-9668-207X, ladone3@gmail.com
² ORCID: 0000-0002-0644-9242, d.muromtsev@gmail.com

Abstract

The paper presents an agent-based model of interactive visualization and proposes a method for its application in the development of the ontological data visual management tool. The process of user interaction with the visualization is represented as a state graph, where each node is a separate visualization. The initial state is known, and the final state is formed in the process of user interaction with the ontological data visual management tool. Each intermediate visualization from the state graph is transferred into the multidimensional Euclidean data space formed on the basis of the visualized ontology, which allows to calculate weighting coefficients on the state graph edges and to search the graph using known algorithms. Application of the model allows for a reduction in the labor intensity of executing user scripts due to the reduction of step-by-step visualization creation to the task of searching in the state space.

Keywords: ontologies, ontology management, ontology visualization methods, visualization models, interactive visualization.

References

1. Ahmad, M. N., & Colomb, R. M. (2007, January). Managing ontologies: a comparative study of ontology servers. In ADC (Vol. 7, pp. 13-22).
2. Aminu, E. F., Oyefolahan, I. O., Abdullahi, M. B., & Salaudeen, M. T. (2020). A review on ontology development methodologies for developing ontological knowledge representation systems for various domains.
3. Barranco, C. D., Campaña, J. R., Medina, J. M., & Pons, O. (2007, July). On storing ontologies including fuzzy datatypes in relational databases. In 2007 IEEE International Fuzzy Systems Conference (pp. 1-6). IEEE.
4. Besbes, G., Baazaoui-Zghal, H., & Ghezela, H. B. (2015, July). An ontology-driven visual question-answering framework. In 2015 19th International Conference on Information Visualisation (pp. 127-132). IEEE.
5. Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). Methods for ontology development. *Semantic Web: Concepts, Technologies and Applications*, 155-173.
6. Daniil, R., Wohlgenannt, G., Pavlov, D., Emelyanov, Y., & Mouromtsev, D. (2019). A new tool for linked data visualization and exploration in 3D/VR space. In *The Semantic Web: ESWC 2019 Satellite Events: ESWC 2019 Satellite Events*, Portorož, Slovenia, June 2–6, 2019, Revised Selected Papers 16 (pp. 167-171). Springer International Publishing.
7. Dudáš, M., Lohmann, S., Svátek, V., & Pavlov, D. (2018). Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review*, 33, e10.
8. Fischer, H., Yigitbas, E., & Sauer, S. (2015, September). Integrating human-centered and model-driven methods in Agile UI development. In *Proceedings of 15th IFIP TC. 13 International Conference on Human-Computer Interaction (INTERACT)*, S (pp. 215-221).

9. Gokhale, P., Deokattey, S., & Bhanumurthy, K. (2011). Ontology development methods. *DESIDOC Journal of Library & Information Technology*, 31(2).
10. Haase, P., Herzig, D. M., Kozlov, A., Nikolov, A., & Trame, J. (2019). metaphactory: A platform for knowledge graph management. *Semantic Web*, 10(6), 1109-1125.
11. Hashimoto, T. B., Alvarez-Melis, D., & Jaakkola, T. S. (2016). Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, 4, 273-286.
12. Kamdar, M. R., Horridge, M., Wogulis, L., Fitzgerald, C., Hardi, J., Anderson, D., ... & Gonçalves, R. S. Interactive Exploration and Collaborative Curation of an Industry-Scale Healthcare Knowledge Graph Using the WebProtégé Cloud-Based Editor.
13. Jones, D., Bench-Capon, T., & Visser, P. (1998). Methodologies for ontology development.
14. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., & Giannopoulou, E. (2007). Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, 39(4), 10-es.
15. Morente-Molinera, J. A., Pérez, I. J., Ureña, M. R., & Herrera-Viedma, E. (2016). Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies. *Information Sciences*, 328, 418-434.
16. Nassiet, D., Livet, Y., Palyart, M., & Lugato, D. (2011, July). Paprika: Rapid UI development of scientific dataset editors for high performance computing. In *International SDL Forum* (pp. 69-78). Berlin, Heidelberg: Springer Berlin Heidelberg.
17. Razd'yakonov, D. S., Morozov, A. V., Pavlov, D. S., & Muromtsev, D. I. (2020). Approach to Blank Node Processing in Incremental Data Visualization by the Example of Ontodia. *Programming and Computer Software*, 46, 384-396.
18. Sivakumar, R., & Arivoli, P. V. (2011). Ontology visualization PROTÉGÉ tools—a review. *International Journal of Advanced Information Technology (IJAIT)* Vol, 1.
19. Vo, H. Q., Phung, T. H., & Ly, N. Q. (2020, November). VQASTO: Visual question answering system for action surveillance based on task ontology. In *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)* (pp. 273-279). IEEE.
20. Vysniauskas, E., Nemuraite, L., & Paradauskas, B. (2011). Hybrid method for storing and querying ontologies in databases. *Elektronika ir Elektrotechnika*, 115(9), 67-72.
21. Wiens, V., Lohmann, S., & Auer, S. (2018, August). WebVOWL Editor: Device-Independent Visual Ontology Modeling. In *ISWC (P&D/Industry/BlueSky)*.
22. Wohlgenannt, G., Klimov, N., Mouromtsev, D., Razdyakonov, D., Pavlov, D., & Emelyanov, Y. (2019). Using word embeddings for visual data exploration with ontodia and wikidata. *arXiv preprint arXiv:1903.01275*.
23. Maslennikov O.P., Milman I.E., Safiulin A.E., Bondarev A.E., Nizametdinov Sh.U.1, Pilyugin V.V. (2014). Development of a system for analyzing of multidimensional data. *Scientific Visualization*, 6(4), 30-49.
24. Razdyakonov D.S., Morozov, A. V., Pavlov, D. S., & Muromtsev, D. I. (2020) Approach to Empty Node Processing in Portionalization of Data Example Ontodia. *Programming*, (6), 16-29.
25. Alatrish, E. S. (2013). Comparison some of ontology. *Journal of Management Information Systems*, 8(2), 018-024.