

Проектирование инструментов визуализации данных на основе интеграции возможностей предметно-ориентированного моделирования и генеративного искусственного интеллекта

Design of Data Visualization Tools Based on the Integrating Domain-Specific Modeling and Generative Artificial Intelligence Capabilities

А. Д. Джейранян, Я. А. Ларионова, Л. Н. Лядова

Национальный исследовательский университет «Высшая школа экономики», Пермь, Россия

A. D. Dzheiranian, Ya. A. Larionova, L. N. Lyadova

National Research University – Higher School of Economics (HSE University), Perm, Russia

Аннотация. Средства визуализации данных представляют собой ключевые инструменты аналитиков, облегчающие выявление зависимостей, тенденций и закономерностей. Эти средства находят применение у широкого круга пользователей (аналитиков, учёных, руководителей предприятий, менеджеров, преподавателей и специалистов других областей, где точное и понятное представление информации критически важно для повышения эффективности анализа данных). Существующие инструменты визуализации ограничивают возможности кастомизации, предлагают лишь базовые настройки параметров, что не всегда удовлетворяет потребностям пользователей, т. к. некоторые сценарии требуют более глубоких модификаций, включая возможность создания собственных типов диаграмм. Такая глубокая настройка зачастую требует использования языков программирования, что создаёт барьер для пользователей без технической подготовки. Предлагается новая методика разработки пользовательских визуализаций данных, которая объединяет преимущества трёх подходов: языково-ориентированный подход к разработке эффективных моделей визуализации с использованием средств автоматизации разработки языков и генерации кода для реализации визуализаций; расширение существующих библиотек пользовательскими пакетами визуализации данных; автоматическая генерация скриптов для реализации пользовательских визуализаций на основе текстовых запросов к языковым моделям, возможностей генеративного искусственного интеллекта. Представлено описание архитектуры системы визуализации данных, многоаспектной онтологии, являющейся ядром системы. Приведены примеры применения разработанного исследовательского прототипа средств создания пользовательских визуализаций, показывающие практическую значимость предлагаемого подхода. Даны оценки полученных результатов и показаны перспективы развития описанных средств.

Ключевые слова: визуализация данных, модели визуализации, классификация методов визуализации, многоаспектная онтология, предметно-ориентированные языки, генерация кода, генеративный искусственный интеллект.

Abstract. Data visualization tools are key analytics tools that make it easier to identify dependencies, trends, and patterns. These tools are used by a broad range of specialists (analysts, scientists, business leaders, managers, teachers, and specialists in other fields where accurate and understandable presentation of information is critically important to improve the effectiveness of data analysis). Existing visualization tools limit customization options and offer only basic parameter settings, which do not always meet the needs of users, as some scenarios require more in-depth modifications, including the ability to create custom chart types. Such deep customization often requires the use of programming languages, which creates a barrier for users without technical skills. A new technique for developing custom data visualizations is proposed, which combines the advantages of three approaches: a language-oriented approach to developing effective visualization models using automation tools for developing languages and generating code for implementing visualizations; existing libraries addition with custom data visualization packages; automatic script generation for implementing custom visualizations based on prompts to language models, capabilities of generative artificial intelligence. The article describes the architecture of data visualization system and the multifaceted ontology that is the core of the system. Examples of using the developed research prototype of tools for creating custom visualizations are given, showing the practical significance of the proposed approach. The results obtained are evaluated and the prospects for the development of the tools described are shown.

Keywords: data visualization, visualization models, classification of visualization methods, multifaceted ontology, domain-specific languages, code generation, generative artificial intelligence.

I. Введение

При принятии решений на основе данных системы визуализации представляют собой ключевой инструмент, позволяющий преобразовать сырые наборы данных в графические представления, с помощью которых можно выявлять скрытые зависимости, тенденции и закономерности. Эти средства находят применение у широкого круга пользователей (руководителей предприятий и менеджеров, учёных и преподавателей и др.), для которых понятное представление информации критически важно для повышения эффективности анализа данных.

Сфера визуализации данных является объектом интенсивного научного интереса, что подтверждается динамикой публикационной активности. Для анализа научных работ был разработан Telegram-бот, обеспечивающий автоматизированное получение данных из крупнейшей базы научных публикаций – Scopus. В результате поиска по ключевому слову «data visualization» было найдено 46 625 научных публикаций за последнее десятилетие, с ростом на 77%: с 3 232 работ в 2015 г. до 5 724 в 2024 г.

В статьях [1, 2, 3] был предложен языково-ориентированный подход к разработке средств визуализации данных, основанный на создании предметно-ориентированных языков для реализации пользовательских моделей визуализации данных. Частотный анализ заголовков найденных статей подтвердил универсальность методов визуализации данных и предметно-ориентированного моделирования (Domain Specific Modeling, DSM), применяемых в различных сферах, включая финансы и бизнес, транспорт, энергетику, здравоохранение и медицину, общественное питание, образование и науку, умные дома и Интернет вещей (IoT) и пр. Однако этот подход пока не находит широкого применения в силу сложности разработки самих языков.

Несмотря на возрастающее значение визуализации, многие пользователи сталкиваются с серьёзными трудностями при создании качественных и информативных графических представлений. Одной из основных проблем является распространение визуализаций низкого качества, которые не только упускают важные аспекты исходных данных, но и могут вводить в заблуждение или исказить информацию [4]. Часто это происходит из-за того, что существующие виды диаграмм ограничены базовыми и чрезмерно обобщёнными геометрическими формами, что приводит к семантическому разрыву между спецификой предметных областей и создаваемыми моделями визуализации.

В связи с этим возникает необходимость внедрения пользовательских спецификаций, позволяющих адаптировать диаграммы к конкретным задачам и особенностям предметной области [5]. Такой подход способствует созданию адаптированных визуализаций, которые учитывают специфические требования пользователей и обеспечивают более эффективное представление данных, тем самым повышая качество управленческих решений и исследовательской работы.

Существующие инструменты визуализации отличаются степенью кастомизации и уровнем реализации спецификаций. В большинстве случаев предлагается лишь базовая настройка параметров, что не всегда удовлетворяет потребностям пользователей. Некоторые сценарии требуют более глубоких модификаций, включая возможность создания собственных типов диаграмм. Такая глубокая настройка зачастую требует использования языков программирования, что создаёт барьер для пользователей без технической подготовки.

Важность совершенствования методов работы с данными также подтверждается официальными документами и национальными программами. Согласно отчёту Gartner «100 предположений стратегического планирования на период до 2028 года», данные и аналитика остаются приоритетными направлениями (<https://www.gartner.com.au/en/doc/over-100-data-and-analytics-predictions-through-2028>).

Внимание многих исследователей сфокусировано на создании диаграмм, предназначенных специально для визуализации данных в контексте их анализа, на проблемах и подходах к их решению [2, 3, 6, 7]. Традиционные инструменты (электронные таблицы и BI-платформы), обеспечивают быстрое создание визуализаций через drag-and-drop-интерфейсы, но возможности кастомизации остаются ограниченными.

Применение языков программирования общего назначения обеспечивает наивысший уровень гибкости и настройки создаваемых моделей, но требует значительной технической подготовки.

Современные исследования [8, 9, 10] и др. демонстрируют потенциал разработки предметно-ориентированных языков (Domain Specific Language, DSL) для визуализации данных, однако существующие решения зачастую ограничиваются небольшим набором стандартных диаграмм (например, круговых диаграмм и гистограмм) или конкретными типами данных (например, геопространственными и т. п.), что не позволяет полноценно настроить диаграмму на разнообразные задачи визуализации.

Наконец, наблюдается активное применение методов искусственного интеллекта (ИИ) для визуализации данных. Современные исследования [11, 12, 13, 14] показывают, что предварительно обученные языковые модели успешно используются для различных этапов анализа данных – от предобработки и генерации рекомендаций до непосредственного построения диаграмм. Однако генерация визуализаций не поддерживает сложные сценарии, ограничиваясь шаблонными решениями [12, 14].

Ещё одна проблема для неопытных пользователей – сложность выбора подходящего типа диаграмм,

который позволил бы максимально точно отразить те выводы, идеи, которые хотел бы донести пользователь при визуализации данных из различных источников, даже при использовании стандартных видов диаграмм, предполагающих минимальные настройки параметров. Несмотря на существование различных рекомендаций [4, 5, 15, 16, 17] по выбору типа диаграмм в зависимости от целей анализа, типа данных и характера задач, для неопытных пользователей, не обладающих специальной подготовкой в области визуализации данных, применение этих рекомендаций остаётся затруднительным. В связи с этим возрастает потребность в средствах, которые могли бы поддерживать пользователя в процессе выбора диаграмм [18, 19].

Таким образом, несмотря на наличие широкого спектра подходов и инструментов для визуализации данных, ни одно из существующих решений не сочетает в себе высокую гибкость настройки на конкретные потребности (включая возможность создания собственных моделей визуализации) и доступность для пользователей, не обладающих навыками программирования.

Можно выделить следующие *проблемы*, с которыми сталкиваются разработчики визуализаций:

1. Использование *неэффективных методов визуализации*. Неправильный подбор метода отображения данных может привести к потере их смысла. Определение подходящего типа визуализации остаётся сложной задачей даже для экспертов, поскольку понимание того, какие именно данные отобразить, какую идею передать и как её оформить, является итеративным процессом. Строгие критерии отбора методов требуют глубокого понимания предметной области, организации данных, что для неопытных пользователей ведёт к ошибкам.

2. *Ограниченность существующих методов визуализации*. Существующие методы визуализации данных дают возможность строить, с одной стороны, простые и чрезмерно обобщенные визуализации, которым, с другой стороны, часто не хватает универсальности, средств настройки для применения в различных областях.

3. *Техническая сложность создания эффективных пользовательских визуализаций*. Использование библиотек для разработки нестандартных, интерактивных визуализаций, требует навыков программирования, понимания структур данных и ручной работы с кодом, разработчикам необходимо уметь работать с асинхронными событиями и т. д., особенно при работе с распределёнными или потоковыми данными [20].

4. *Риск неверной интерпретации данных*. При создании визуализаций для широкой аудитории необходимо учитывать различные уровни «визуальной грамотности» аудитории (способность интерпретировать изображения) [21] – даже корректно построенная диаграмма может привести к ошибочным выводам [7].

5. *Отсутствие масштабируемости для больших объёмов данных*. При росте размеров наборов данных должны меняться и паттерны отображения; т. к. существуют трудности в сохранении читаемости, а также в обеспечении производительности при сравнении и кластеризации больших массивов информации.

6. *Отсутствие унифицированных инструментов для оценки качества визуализаций*. Это создаёт трудности оценки, сравнения качества разных визуализаций при их разработке.

7. *Отсутствие устоявшихся подходов к обучению* созданию эффективных визуализаций, сформированных принципов обучения и преподавания, формирования «визуальной грамотности» аудитории [22].

Объектом исследования, рассматриваемым в данной статье, является процесс создания эффективных визуализаций данных. *Предмет исследования* – средства создания пользовательских визуализаций данных, обеспечивающих отражение специфики решаемых задач и предметных областей, не требующих от пользователей профессиональных навыков в области программирования.

Предлагается *новая методика разработки пользовательских визуализаций данных*, которая объединяет преимущества трёх подходов: разработка *предметно-ориентированных языков* (DSL, Domain Specific Language) для создания моделей визуализации, соответствующих потребностям пользователей; *расширение существующих библиотек языков программирования* пользовательскими пакетами; *автоматическая генерация скриптов для реализации пользовательских визуализаций* на основе текстовых запросов к языковым моделям (LLM, Large Language Model), возможностей генеративного искусственного интеллекта (GAI, General Artificial Intelligence). Использование многоаспектной онтологии в качестве ядра проектируемой системы позволяет хранить все знания, необходимые для функционирования платформы. В представленном проекте средства разработки пользовательских визуализаций дополняются возможностью подбора подходящих типов визуализации на основе образцов, предлагаемых пользователями.

II. Постановка задачи

Основные задачи исследования определяются в ходе анализа требований к средствам визуализации данных, используемыми в различных областях.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Обзор применения визуализации данных, задач и методов визуализации.
2. Анализ требований к средствам решения задач визуализации данных.
3. Проектирование средств визуализации данных, управляемых знаниями, интегрированных с

возможностями генеративного ИИ, и реализация исследовательского прототипа средств визуализации данных.

4. Аprobация подхода – проведение экспериментов и оценка полученных результатов.

Для проектирования средств визуализации данных необходимо выполнить их *классификацию*, которая должна послужить основой для разработки онтологии языков, используемых для разработки пользовательских визуализаций. Среди известных подходов к классификации можно выделить следующие: по типу данных, по форме представления [23], по цели создания [24], по интерактивности [23] и др. Схема выбора диаграмм (Chart Chooser), предложенная в [1], основана на разделении диаграмм по видам задач анализа (сравнение, распределение, соотношение, структура) и параметрам данных (количеству категорий, переменных и периодов). На основе таксономии методов визуализации данных, включающей пять категорий [25], описанных в [2], разработана расширенная классификация, включающая 22 метода визуализации, объединенных в 7 групп в зависимости от цели создания диаграммы и особенностей данных, что обеспечивает упрощение выбора диаграмм пользователями системы при разработке своих моделей визуализации.

Одним из ключевых трендов при разработке средств визуализации является интеграция визуализации в сторителлинг (повествование) [26]. Этот подход позволяет преобразовать сложные данные в запоминающиеся истории, усиливая эмоциональную вовлечённость аудитории. Основные аспекты дата-сторителлинга:

- Преодоление разрыва между данными и их пониманием. Это делает информацию доступной для широкой аудитории, например, в медицине для объяснения сложных концепций пациентам [27].
- Повышение эффективности коммуникации. Визуализации подкрепляют утверждения фактами, улучшают запоминаемость и передачу скрытых в данных смыслов, закономерностей [27, 28].
- Иммерсивные технологии. Применение дополненной и виртуальной реальности создает пространственную связь с данными, усиливая восприятие и вовлечённость [29].

На основе описанных ранее проблем и ключевых трендов разработки можно предъявить *общие требования*, которым должны соответствовать средства визуализации:

1. Обеспечение адаптации графических представлений к специфике предметной области, в том числе возможность комбинирования и расширения существующих типов диаграмм для создания новых пользовательских визуализаций [5].

2. Обеспечение создания и настройки визуализаций без необходимости программирования [20].

3. Обеспечение создания интерактивных представлений данных (с возможностью детализации изображений, их масштабирования, отображения подсказок) [9].

4. Обеспечение редактирования ранее созданных диаграмм, возможность вносить изменения «на лету»: перенастройка параметров, элементов дизайна и структуры диаграммы.

5. Обеспечение широкого набора базовых методов визуализации для представления одних и тех же данных в разных формах, плоскостях и измерениях в зависимости от решаемых задач.

6. Обеспечение рекомендаций по выбору метода визуализации на основе анализа введённых данных и цели создания визуализации [5], на основе образцов, предлагаемых пользователями.

7. Обеспечение достаточного уровня автоматизации при построении диаграмм при сохранении полного пользовательского контроля над настройкой дизайна и параметров визуализации [21].

Требования к функциональности средств визуализации данных [2] расширены. Для предоставления *конечному пользователю* возможностей, необходимых для решения специфичных задач аналитики, необходима тесная коллаборация специалистов в разных областях:

– *Эксперт в предметной области* обладает специализированными знаниями в своей сфере. Он активно задействуется при разработке онтологии предметной области посредством интервьюирования.

– *DSM-эксперт* способен разрабатывать новые предметно-ориентированные языки, правила трансформации моделей.

– *Программный инженер* отвечает за расширение существующих библиотек визуализации пользовательскими пакетами и компонентами.

– *ML-инженер* задействуется на этапе дообучения языковой модели тем пакетам, которые были разработаны ранее программным инженером: он создает специализированный набор данных и обучает на нём модель под разные сценарии для эффективной работы в определённой предметной области.

При использовании прототипа конечный пользователь – разработчик визуализации – может создать стандартную или нестандартную визуализацию. Процесс создания стандартной визуализации может быть выполнен двумя способами: настройкой параметров существующего DSL, разработанного для описания пользовательской модели визуализации, или написанием текстового запроса для генерации скрипта визуализации данных на языке программирования. Эксперт в предметной области отвечает за разработку онтологии, которая служит основой для решения задач пользователями – разработчиками визуализаций. DSM-эксперт занимается созданием метамodelей базовых языков. Генерация новых языков автоматизирована

за счёт применения правил отображения онтологии предметной области на метамодель базового языка.

Создание нестандартной визуализации становится доступным также благодаря написанию текстового запроса, но отличие в том, что модель предварительно обучается ML-инженером на написание кода с помощью кастомных пакетов визуализации, которые разрабатываются программным инженером. Кроме того, нетипичные задачи могут быть решены при помощи разработки нового DSL, соответствующего задачам пользователя. Процесс трансформации моделей в код позволяет реализовать созданные модели. Правила трансформации, необходимые для генерации кода визуализации (в том числе и интерактивной), разрабатываются DSM-экспертом.

III. Теория: моделирование средства визуализации данных

Проектирование средств визуализации данных основано на разработке многоаспектной онтологии – ядра системы, управляемой знаниями [1, 2, 3, 6]. Она является центральным элементом архитектуры системы.

Многоаспектная онтология – это совокупность онтологий, которые предназначены для независимого моделирования и объединения различных аспектов интеллектуальной системы (рис. 1):

1. *Онтология источников данных* включает информацию о структуре, типах данных и т. д.
2. *Онтология предметных областей* содержит субъективные знания экспертного сообщества: концепты предметной области, их взаимоотношения и ограничения (в том числе правила).
3. *Онтология DSM-знаний* – ядро логики и визуального конструирования DSL, которое состоит из онтологий, которые содержат описание моделей, визуальных языков, предметных областей, правил трансформации и генерации DSL и т. д.
4. *Онтология скриптов* фиксирует результаты генерации: скрипты, используемые промпты (текстовые запросы) и метаданные (версию модели, имя файла источника данных и т. д.).
5. *Онтология пакетов* описывает расширения существующих библиотек языков программирования общего назначения специализированными пакетами с набором функций и компонентов, необходимых для визуализации нового типа диаграммы или для решения задач в определенной предметной области.
6. *Репозиторий пользовательских задач* фиксирует задачи визуализации и позволяет находить и переиспользовать существующие решения.

Помимо онтологий, проектируемый граф знаний системы включает также *функции*, реализующие базовые алгоритмы: F1 – функция *автоматизации разработки новых языков* по правилам, заданным пользователем для отображения элементов онтологии предметной области на элементы метамодели выбранного базового языка; F2 – функция *формирования правил трансформации* – генерации кода на целевом текстовом языке по построенной на визуальном языке модели (элементам метамодели ставятся в соответствие символы языка); F3 – функция *генерации нового DSL* в соответствии с правилами, заданными через F1; F4 – функция *трансформации моделей (генерации кода)* по правилам, заданным через F2; F5 – функция *определения правил интерпретации* для интерактивных моделей; F6 – функция *реализации взаимодействия пользователей с визуальными моделями* в интерактивных визуализациях по правилам, заданным через F5; F7 – функция, *связывающая модель предметной области в онтологии с соответствующим ей пользовательским пакетом визуализации*, в котором реализованы пользовательские функции визуализации; F8 – функция *связывания пользовательских задач с источниками данных*, используемыми в решениях; F9 – функция *верификации соответствия скрипта пакету* (сопоставляет сгенерированный скрипт с существующим пакетом визуализации и сохраняет в метаданных скрипта свойство соответствия пакету); F10 – функция *создания в репозитории пользовательских задач объекта «Решение»*, представляющего программный код созданной модели (как результат выполнения функций F4 или F6); F11 – функция *создания нового объекта «Задача» в репозитории пользовательских задач* по текстовому запросу пользователя (автоматически связывает его с порождённым скриптом как с «Решением»).

Проектирование архитектуры системы основано на подходе, ориентированном на знания, где ядром выступает многоаспектная онтология (рис. 1). Платформа разбита на шесть ключевых компонентов (рис. 2).

Пользовательский интерфейс – единая точка входа, реализованная через паттерн «Фасад». *Подсистема расширения библиотек визуализации* отвечает за создание новых пакетов над существующими библиотеками (например, Matplotlib, Seaborn, D3 и др.) и обучение языковой модели этим пакетам. *Подсистема генерации скриптов* автоматически превращает естественно-языковые запросы в исполняемые Python-скрипты. *Подсистема предметно-ориентированных языков* – языковой инструментарий для разработки DSL и моделей визуализации с их использованием, а также генерации кода, реализующего визуализации по разработанным моделям [1, 2, 3]. *Модуль интеграции с источниками данных* отвечает за загрузку данных в форматах CSV, XLSX и др. *Подсистема генерации скриптов* – новый компонент – сочетает семантические технологии и большие языковые модели для автоматизации генерации кода визуализации при сохранении истории

взаимодействий с пользователем, реализован по принципам многослойной архитектуры, разделяющей функции представления, бизнес-логики и управления данными (рис. 3).

Уровень представления. *DashComponent* выполняет роль корневого контейнера, реализуя пользовательский интерфейс на базе фреймворка *Dash*. Построен по модели реактивного программирования: обратные вызовы *Dash* связывают действия пользователя с бэкенд-сервисами. Он координирует работу трёх подкомпонентов: 1) *ChatComponent* управляет взаимодействием пользователя с системой в чате в реальном времени через интерфейс *IChatLogic*, отображает историю пользовательских запросов и передаёт ответы модели; 2) *GraphComponent* обрабатывает результаты визуализации, координируя зависимости через три ключевых интерфейса (*IDataParser* – приём и парсинг структурированных данных, *IAIRResponse* – отображение сгенерированного ответа модели, *IVisualizationExecutor* – динамическое выполнение сгенерированного кода), реализует интерфейс *IUserRequest* для представления введённых пользователем запросов; 3) *dcc.Store* – это компонент для клиентского хранения состояния прямо в браузере (предоставляет интерфейс *IStateStorage*, позволяющий любым компонентам пользовательского интерфейса читать и записывать временные JSON-данные (например, текущие параметры запроса, настройки графиков или промежуточные результаты). За счёт хранения часто используемого состояния в браузере *dcc.Store* устраняет лишние обращения к серверу и ускоряет работу приложения.

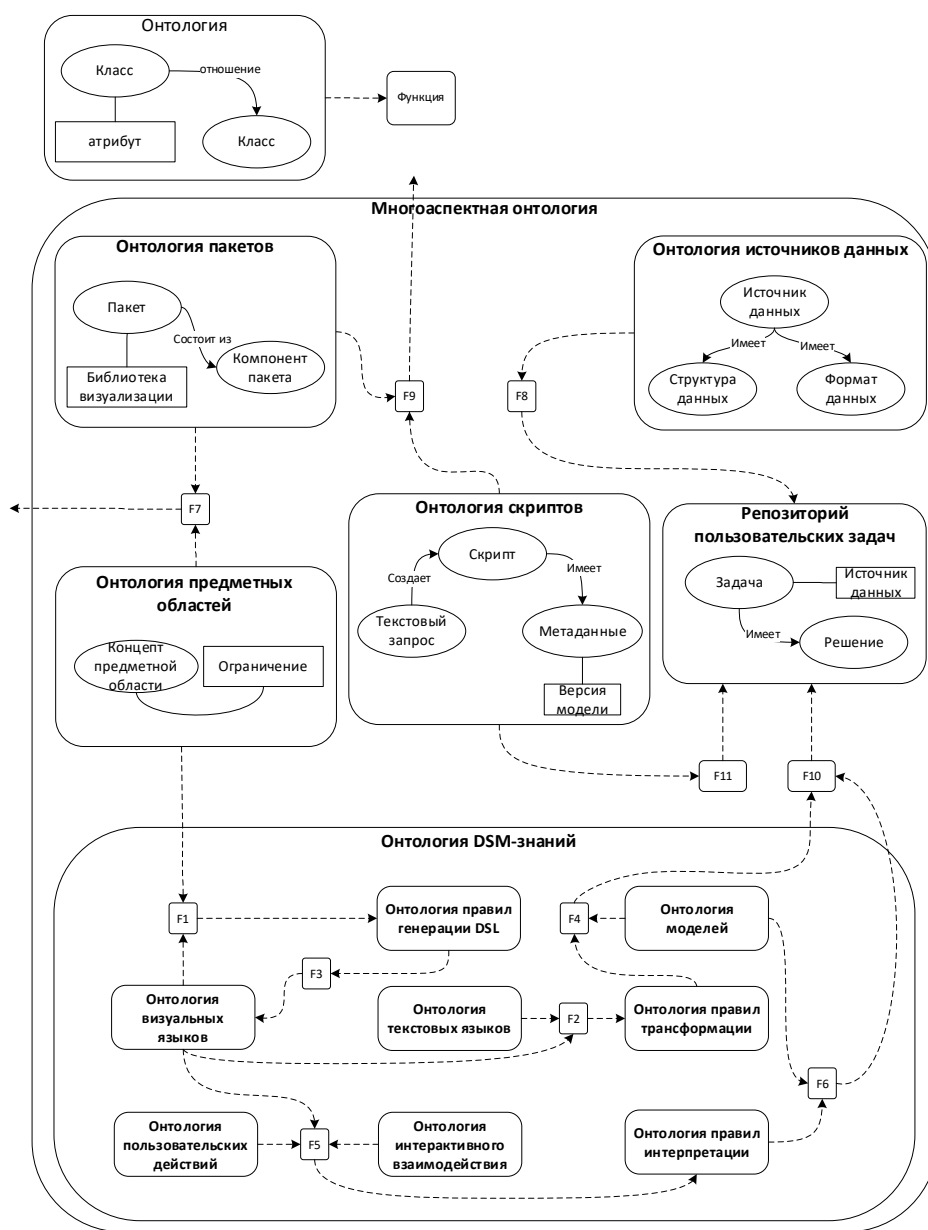


Рис. 1. Структура многоаспектной онтологии

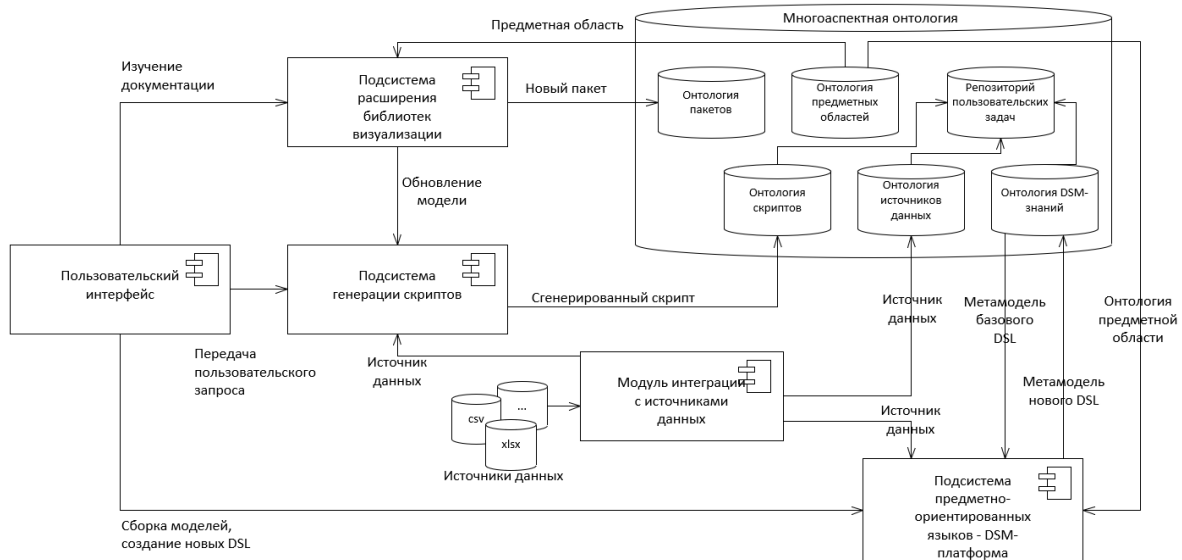


Рис. 2. Обобщённая структура средств визуализации данных, управляемых знаниями

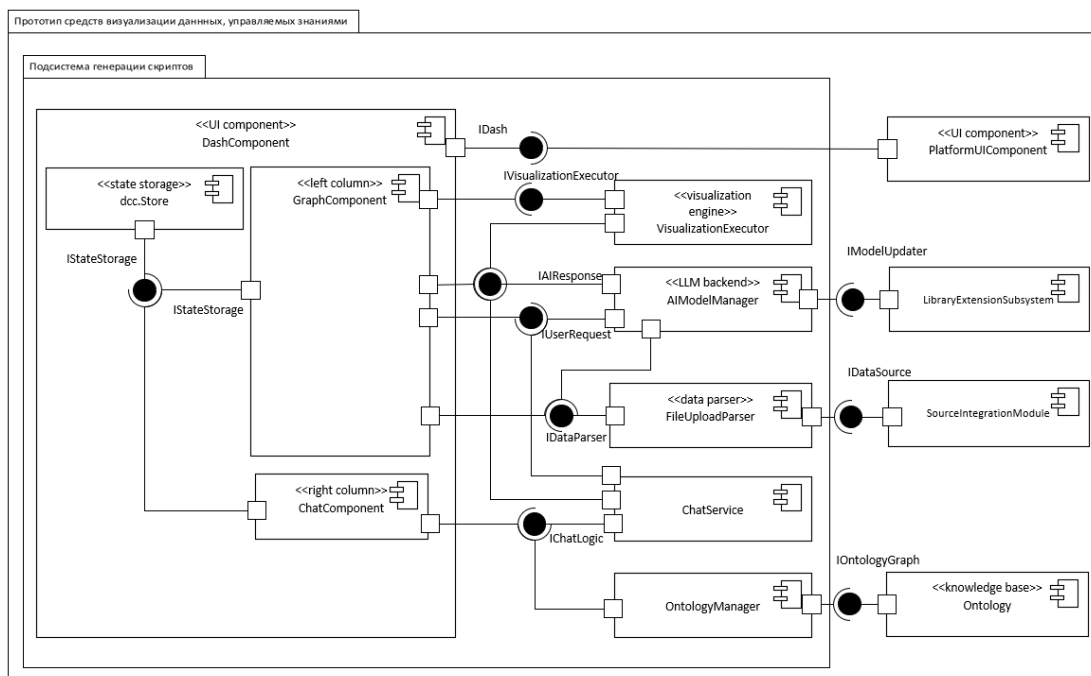


Рис. 3. Архитектура подсистемы генерации скриптов

Уровень бизнес-логики включает пять компонентов, реализующих предметно-ориентированную логику: 1) *FileUploadParser* (*IDataParser*) выполняет парсинг файлов CSV/Excel с помощью *Pandas*, извлекая табличные данные в *DataFrame*; 2) *ChatService* (*IChatLogic*) управляет состоянием сессии чата, обновляет чат при помощи сохраненного контекста через *IStateStorage*; 3) *AIModelManager* (*IAIResponse*) использует пайплайны *LangChain* и API *ChatGroq* (Llama3-70B) для генерации кода визуализации на *Plotly* на основе загруженных данных (*IDataParser*) и пользовательского запроса (*IUserRequest*); 4) *VisualizationExecutor* (*IVisualizationExecutor*) безопасно выполняет сгенерированный *Python*-код в *sandboxed*-средах; 5) *OntologyManager* реализует операции с онтологией: создание экземпляров сущностей (промттов/скриптов), версионирование моделей и др. Уровень данных (компонент *Ontology*), предоставляющий интерфейс *IOntologyGraph*, выступает в качестве постоянной базы знаний системы. Внешние компоненты системы: *PlatformUIComponent* является основным пользовательским интерфейсом платформы. Он интегрирует *DashComponent* для отображения визуализации и взаимодействия в чате через требуемый интерфейс (*IDash*); *LibraryExtensionSubsystem* отвечает за разработку новых пакетов визуализации и обучение языковой модели этим пакетам. Он реализует интерфейс *IModelUpdater*, через который *AIModelManager* инициирует обновление или селекцию версии LLM-модели при подготовке запроса; *SourceIntegrationModule* передаёт компоненту *FileUploadParser* загруженный в систему файл через интерфейс *IDataSource* для дальнейшей обработки.

Ещё один компонент, который может быть интегрирован в систему, – прототип компонента получения рекомендаций, который на основе сформированной онтологии визуальных языков и образца, представленного пользователем, формирует подсказку по выбору подходящей модели визуализации. Обобщённая структура компонента показана на рис. 4. Работа компонента основана на использовании описанной онтологии (рис. 1).

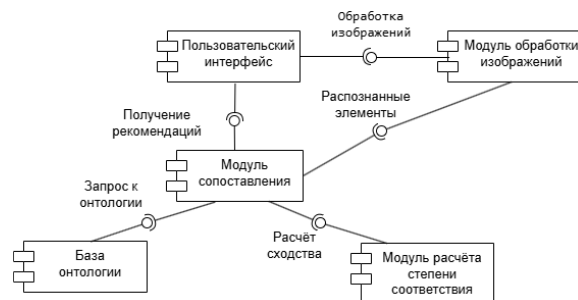


Рис. 4. Структура компонента получения рекомендаций

Для реализации прототипа использованы язык программирования *Python*, библиотека *RDFlib* для работы с онтологиями и модель детекции объектов *YOLOv8*. Все разработки велись в интегрированной среде *PyCharm*. Для извлечения из онтологии информации о типах диаграмм, их характерных элементах и соответствующих весах используются *SPARQL*-запросы.

IV. Результаты экспериментов

Описанные средства прошли апробацию:

1. Разработаны пользовательские модели визуализации с помощью DSL, созданных на основе базового языка и описания предметной области. С использованием моделей построены визуализации.
2. Проведены эксперименты по генерации скриптов для визуализации данных в соответствии с запросами пользователей различной сложности для данных с разной структурой.
3. Выполнены исследования возможностей подбора подходящих типов визуализации на основе распознавания образцов, предложенных пользователем.

Метамодели языков формируются как наследники соответствующих классов диаграмм в онтологии. На рис. 5 показана метамодель базового DSL для создания пользовательских моделей *пузырьковых диаграмм*.

Модель пузырьковой диаграммы, созданная с использованием разработанного языка, приведена на рис. 6, а. Разработанный язык для создания пузырьковых диаграмм использован в качестве базового для создания нового DSL для визуализации данных о поведении посетителей кафе в зависимости от их предпочтений в кофе и уровня шума в помещении (цель – понять, как шум влияет на выбор кофе и общее время пребывания в кафе). Кастомизированная модель визуализации для этой задачи показана на рис. 6, б. Диаграммы, разработанные с использованием моделей визуализаций, показанных на рис. 6, приведены на рис. 7.

Выполненные при разработке новых DSLs и моделей настройки не требуют от пользователей навыков программирования – для их реализации созданы визуальные конструкторы правил, обеспечивающие как генерацию метамоделей пользовательских DSLs на основе знаний о предметной области, имеющихся в онтологии, так и генерацию кода для реализации визуализаций (в том числе и интерактивных).

С использованием разработанных компонентов проведена серия экспериментов с обработкой данных, имеющих различную структуру, и запросов различной сложности. В качестве примера рассмотрим сложную задачу с выполнением запроса без указания типа диаграммы: «Покажи мне визуализацию, из которой можно сделать много выводов». На рис. 8 показан пользовательский интерфейс ИИ-ассистента для генерации визуализаций. Левая панель объединяет загрузчик файлов с поддержкой форматов CSV/Excel; интерактивную таблицу для предварительного просмотра загруженных данных; текстовое поле для ввода запросов на естественном языке и кнопку для запуска обработки ИИ; динамическое отображение графиков Plotly и сгенерированных фрагментов кода. Правая панель отображает историю чата для текущей сессии с разным оформлением для пользовательских сообщений и ответов модели. Приложение на основе введенных данных формирует контекстно-ориентированный промпт. После формирования запрос отправляется на обработку через API Groq с использованием модели LLaMA. Далее код, сгенерированный ИИ, преобразуется в исполняемые рабочие процессы визуализации на Python. Компонент истории чата обеспечивает сохранение диалога.

Учитывая открытый формат задачи, модель сгенерировала точечную диаграмму (рис. 9). Эта визуализация демонстрирует взаимосвязь между предложенными данными: продолжительностью сна, уровнем физической активности и уровнем стресса для каждого участника. Вместе со сгенерированным программным кодом в ответе предложены вопросы для облегчения интерпретации, что помогает анализу данных пользователем.

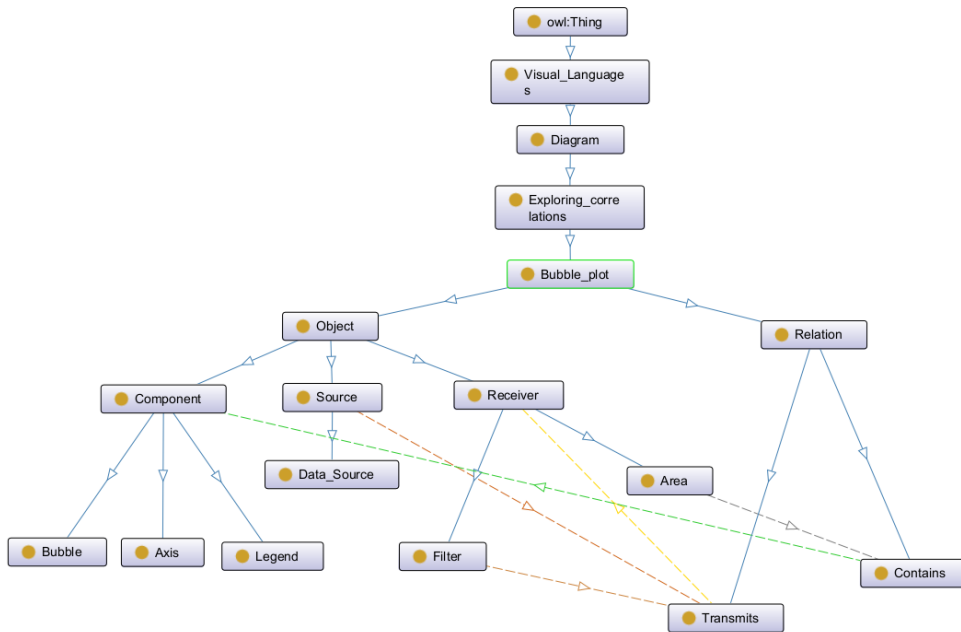


Рис. 5. Представление метамодели языка пузырьковой диаграммы в онтологии (фрагмент онтологии)

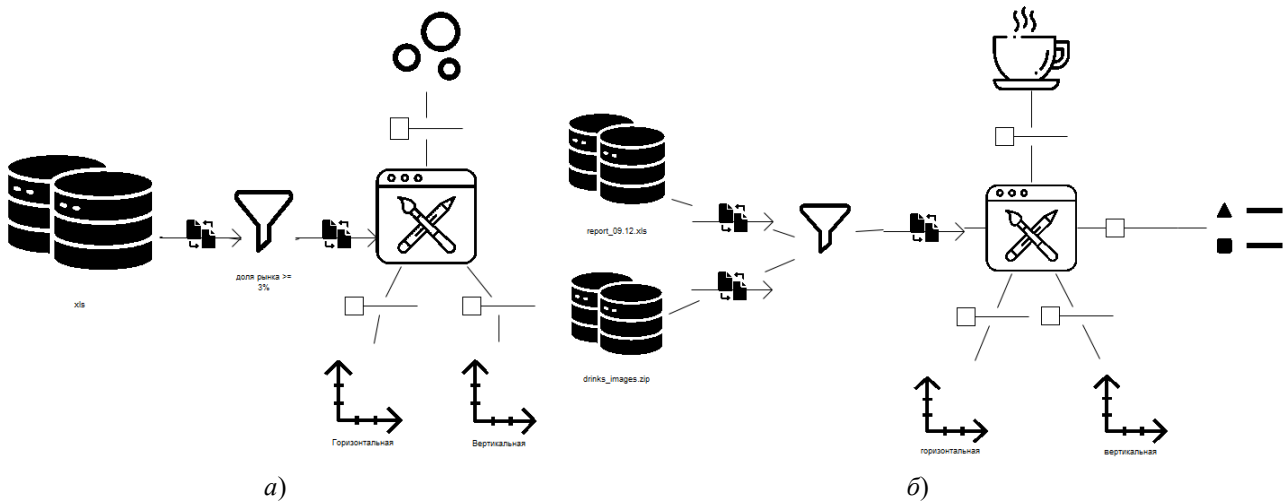


Рис. 6. Использование языка пузырьков диаграмм: а) модель диаграммы о доле рынка, занимаемой отдельными группами товаров, построенная на базовом языке; б) модель диаграммы чашек, созданная на пользовательском DSL

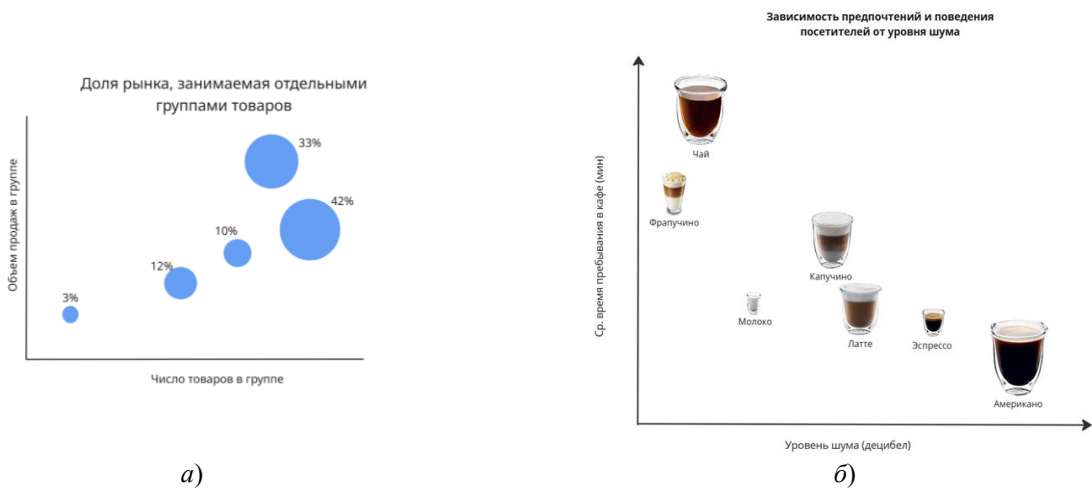


Рис. 7. Использование моделей пузырьков диаграмм: а) реализация модели, созданной на базовом языке; б) реализация модели, созданной на пользовательском DSL диаграммы чашек

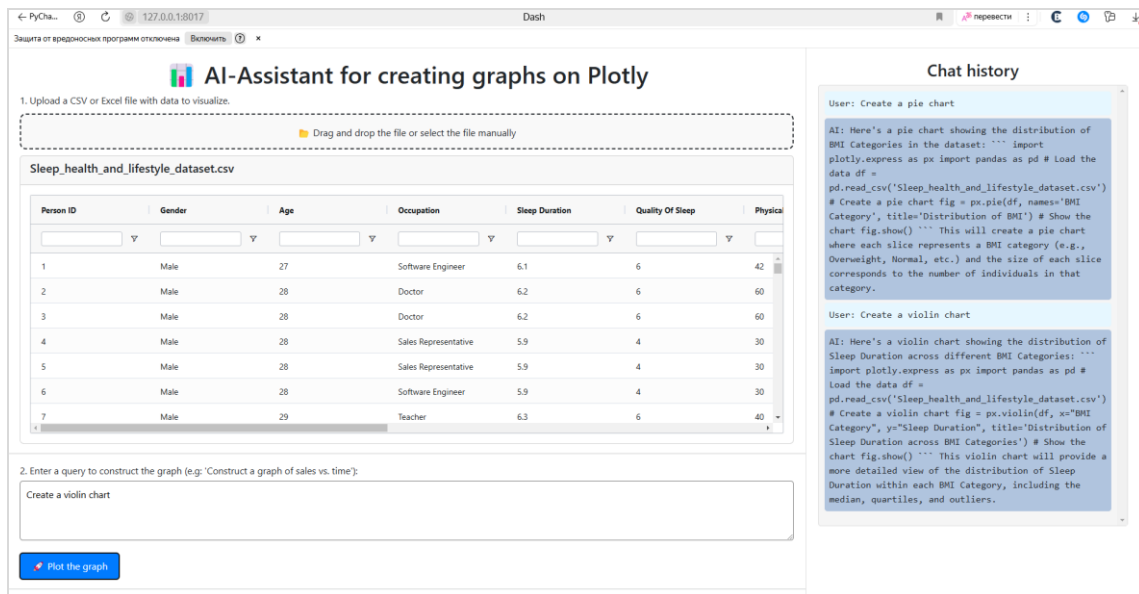


Рис. 8. Пользовательский интерфейс ИИ-ассистента

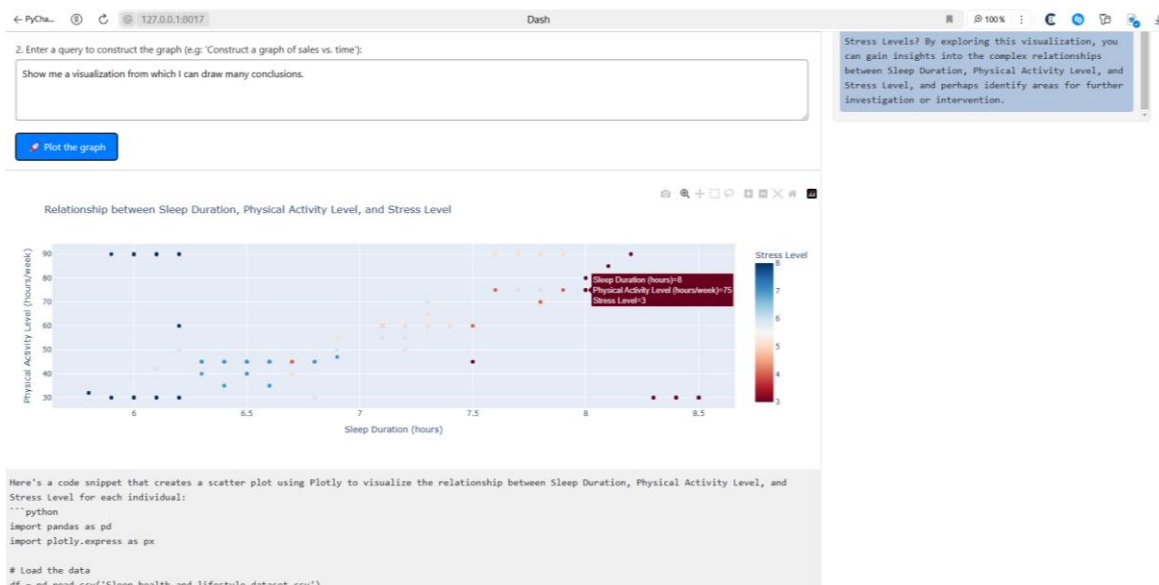


Рис. 9. Визуализация, сгенерированная по запросу ИИ-ассистентом

Анализ онтологии после проведённых экспериментов подтверждает, что она динамически обновлялась в ходе всех взаимодействий: цепочка «Запрос → Сгенерированный скрипт → Метаданные» сохраняется в виде экземпляров классов, что свидетельствует о надёжной интеграции артефактов в онтологию прототипа.

Была выполнена также *апробация подхода к созданию рекомендательной системы по выбору модели для эффективной визуализации данных*. Процесс определения типа диаграммы по целевому образцу, предоставленному пользователем, включает несколько этапов, при выполнении которых происходит обнаружение ключевых элементов диаграмм-образцов (решение основано на модели YOLOv8), сопоставление результатов распознавания элементов целевой диаграммы с типами диаграмм, описанными в онтологии, и расчёт степени соответствия. Результаты отображаются в виде текста, где перечислены все подходящие типы диаграмм с указанием степени их соответствия образцу и распознанных элементов, повлиявших на оценку.

V. Обсуждение результатов

Результаты проведённых ранее экспериментов по разработке пользовательских визуализаций на основе языково-ориентированного подхода [1, 2, 3] показали его практическую значимость при наличии средств автоматизации разработки DSLs и средств трансформации моделей типа «Модель–Текст» (генерации кода по визуальным моделям). Проведённые эксперименты в рамках предложенной методики [6] по интеграции ранее разработанных средств и возможностей ИИ показали хорошие результаты. Благодаря использованию онтологии обеспечивается сохранение и повторное использование метамоделей языков, моделей визуализации

и сгенерированного кода, что позволяет существенно сократить трудозатраты при создании новых визуализаций и гарантирует воспроизводимость полученных результатов. В отличие от традиционных инструментов, где глубокая настройка диаграмм требует ручного написания кода, разработанный подход сокращает время создания визуализаций и снижает требования к навыкам пользователей. Разработанный в рамках исследования ИИ-ассистент автоматизирует процесс формирования и исполнения Python-скриптов с использованием библиотеки Plotly. В ходе тестирования ИИ-ассистент успешно справился с различными задачами. При использовании ИИ-ассистента упрощается процесс выбора графического представления, подходящего для конкретных данных или целей анализа. Интеграция ИИ-ассистента с онтологией обеспечивает масштабируемое накопление знаний и расширяет возможности коллективной аналитики. При этом следует помнить, что применение ИИ требует внимательного отношения к вопросам качества и достоверности результатов, т. к. возможны искажения, неточности в сгенерированных визуализациях.

VI. Выводы и заключение

Предложена и прошла апробацию новая методика визуализации данных, которая, объединяя преимущества нескольких подходов, обеспечивает создание кастомизированных визуализаций без необходимости глубоких знаний в программировании. *Научной новизной* обладают следующие результаты: *интегрированная методика создания средств визуализации данных*, построенная на сочетании различных подходов; *модульная архитектура* системы, управляемой знаниями, обеспечивающая расширяемость средств визуализации, их настройки на новые сценарии использования; *новая классификация методов визуализации данных* – основа для разработки визуальных языков для создания визуализаций; *онтология скриптов*, аккумулирующая пользовательские запросы, генерируемые скрипты и метаданные, что позволяет применять шаблоны кода в решении аналогичных задач и расширяет возможности совместной аналитики.

Практическая значимость подтверждается результатами проведения серии экспериментов.

Направления дальнейших исследований включают разработку пользовательских пакетов для визуализации данных с функциями тонкой настройки диаграмм на конкретные предметные области, обучение языковой модели для автоматической генерации кода с использованием этих пакетов. Актуальны также исследования возможностей автоматизации разработки онтологий.

Список литературы

1. Dzheiranian A. D., Ermakov I. D., Proskuryakov K. A., Lyadova L. N. Designing Data Visualization System Based on Language-Oriented Approach // Proceedings of ISP RAS. 2024. Vol. 36(2). P. 127–140. DOI: 10.15514/ISPRAS-2024-36(2)-10.
2. Dzheiranian A. D., Ermakov I. D., Proskuryakov K. A., Lyadova L. N. An Approach to Developing Data Visualization Tools Based on Domain Specific Modeling // Scientific Visualization. 2024. Vol. 16(4). DOI: 10.26583/sv.16.4.08.
3. Джейранян А. Д., Ермаков И. Д., Проскуряков К. А., Лядова Л. Н. Разработка инструментов визуализации данных на основе предметно-ориентированного моделирования // Материалы 34-й Международной конференции по компьютерной графике и машинному зрению (GraphiCon), Омск, 17–19 сентября 2024 г. С. 300–314. DOI: 10.25206/978-5-8149-3873-2-2024-300-314.
4. Sawicki J., Burdukiewicz M. VisQualdex : a Comprehensive Guide to Good Data Visualization // Scientific Visualization. 2023. Vol. 15(1). P. 127–149. DOI: 10.26583/sv.15.1.11.
5. Qin X., Luo Y., Tang N., Li G. Making Data Visualization More Efficient and Effective : a Survey // The VLDB Journal. 2020. Vol. 29. P. 93–117. DOI: 10.1007/s00778-019-00588-3.
6. Джейранян А. Д., Лядова Л. Н. Преодоление ограничений современных инструментов визуализации данных: новая методология // Big Data and Advanced Analytics : Сборник научных статей XI Международной научно-практической конференции, Минск (Республика Беларусь), 23–24 апреля 2025 г. С. 386–396.
7. Исаев Р. А., Подвесовский А. Г. Когнитивная ясность графовых моделей: подход к пониманию идеи и способ выявления влияющих факторов с использованием визуального анализа // Научная визуализация. 2022. Т. 14(4). С. 38–51. DOI: 10.26583/sv.14.4.04.
8. Smeltzer K., Erwig M. A Domain-Specific Language for Exploratory Data Visualization // GPCE 2018: Proceedings of the 17th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences. 2018. P. 1–13. DOI: 10.1145/3278122.3278138.
9. Morgan R., Grossmann G., Schrefl M., Stumptner M., Payne T. VizDSL: A Visual DSL for Interactive Information Visualization // Advanced Information Systems Engineering. CAiSE 2018. Lecture Notes in

- Computer Science(). Vol. 10816. Springer, Cham. P. 440–455. DOI: 10.1007/978-3-319-91563-0_27.
10. Khakpour A., Vazquez-Ingelmo A., Colomo-Palacios R., Garcia-Penalvo F. J., Martini A. ModelViz: A Model-Driven Engineering Approach for Visual Analytics System Design // IEEE Access. 2024. Vol. 12. P. 42667–42682. DOI: 10.1109/ACCESS.2024.3379268.
 11. Shi D., Xu X., Sun F., Shi Y., Cao N. Calliope: Automatic Visual Data Story Generation From a Spreadsheet // IEEE Transactions on Visualization and Computer Graphics. 2021. Vol. 27(2). P. 453–463. DOI: 10.1109/TVCG.2020.3030403.
 12. Wang X., Wu Z., Huang W., Wei Y., Huang Z., Xu M., Chen W. VIS+AI: Integrating Visualization with Artificial Intelligence for Efficient Data Analysis // Frontiers of Computer Science. 2023. Vol. 17 : 176709. DOI: 10.1007/s11704-023-2691-y.
 13. Wu A., Wang Y., Shu X., Moritz D., Cui W., Zhang H., Zhang D., Qu H. Ai4vis: Survey on Artificial Intelligence Approaches for Data Visualization // IEEE Transactions on Visualization and Computer Graphics. 2021. Vol. 28(12). P. 5049–5070. DOI: 10.1109/TVCG.2021.3099002.
 14. Maddigan P., Susnjak T. Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models // IEEE Access. 2023. Vol. 11. P. 45181–45193. DOI: 10.1109/ACCESS.2023.3274199.
 15. Midway S. R. Principles of Effective Data Visualization // Patterns. 2020. Vol. 1(9) : 100141. DOI: 10.1016/j.patter.2020.100141.
 16. Midway S. R., Brum J. R., Robertson M. Show and Tell: Approaches for Effective Figures // Limnology and Oceanography Letters (L&O Letters). 2023. Vol. 8(2). P. 213–219. DOI: 10.1002/lol2.10288.
 17. Schwabish J. A. Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks. New York : Columbia University Press, 2021.
 18. Hu K., Bakker M. A., Li S., Kraska T., Hidalgo C. Vizml: A Machine Learning Approach to Visualization Recommendation // CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019). 2019. Glasgow, Scotland UK. ACM. 18 p. DOI: 10.1145/3290605.33003581.
 19. Kubernátová P., Friedjungová M., van Duijn M. Constructing a Data Visualization Recommender System // Computer and Information Science. Vol. 862. Springer, 2019. Pp. 1–25. DOI: 10.1007/978-3-030-26636-3_1.
 20. Singh A., Goyal C., Parashar J. M. Democratizing Data Visualization and Insights Extraction with Pandas, Generative AI, and CSV Data // International Journal of Scientific Research in Engineering and Management. 2024. Vol. 8(5). P. 1–5. DOI: 10.55041/IJSREM33437.
 21. Dhawka P., Dasgupta S. The Social Construction of Visualizations: Practitioner Challenges and Experiences of Visualizing Race and Gender Demographic Data // Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '25). 2025. Yokohama. Available from: <https://arxiv.org/html/2502.09048v2>.
 22. Bach B., Keck M., Rajabiyazdi F., Losev T., Meirelles I., Dykes J., Carpendale S. Challenges and Opportunities in Data Visualization Education: A Call to Action // IEEE Transactions on visualization and computer graphics. 2023. Vol. 30(1). P. 649–660. DOI: 10.1109/TVCG.2023.3327378.
 23. Cuadrado-Gallego J. J., Demchenko Y., Losada M. A., Ormandjieva O. Classification and Analysis of Techniques and Tools for Data Visualization Teaching // Proceedings of the IEEE Global Engineering Education Conference (EDUCON). 2021. P. 1593–1599. DOI: 10.1109/EDUCON46332.2021.9453917.
 24. Cota M. P., Rodríguez M. D., González-Castro M. R., Gonçalves R. M. M. Analysis of Current Visualization Techniques and Main Challenges for the Future // Journal of Information Systems Engineering & Management. 2017. Vol. 2(3). P. 19. DOI: 10.20897/jisem.201719.
 25. Kirk A. Data Visualization: a Successful Design Process. Birmingham : Packt Publishing LTD, 2012. 206 p.
 26. Tong C., Roberts R., Borgo R., Walton S., Laramée R. S., Wegba K., Ma X. Storytelling and Visualization: An Extended Survey // Information. 2018. Vol. 9(3) : 65. DOI: 10.3390/info9030065.
 27. Dubey R., Auron M., Christenson C. L., Eichbaum Q., Friedman M., Tolich D., Bocquet C. The Art of Storytelling in Patient Blood Management: a Narrative Review on Bridging the Gap Between Complex Data and Understandable Information // Annals of Blood. 2024. Vol. 9. DOI: 10.21037/aob-24-24.
 28. Peng Z. Telling a Story with Graphs: How to Apply Narrative Visualization Strategies and Visualization Techniques in Journalistic Storytelling // The Palgrave Handbook of Global Digital Journalism. Palgrave Macmillan, Cham. 2025. P. 283–306. DOI: 10.1007/978-3-031-59379-6_18.
 29. Méndez J., Luo W., Rzayev R., Büschel W., Dachselt R. Immersive Data-Driven Storytelling: Scoping an Emerging Field Through the Lenses of Research, Journalism, and Games // IEEE Transactions on Visualization and Computer Graphics. 2025. DOI: 10.1109/TVCG.2025.3531138.