

УДК 004.925

## ПРЕОДОЛЕНИЕ ОГРАНИЧЕНИЙ СОВРЕМЕННЫХ ИНСТРУМЕНТОВ ВИЗУАЛИЗАЦИИ ДАННЫХ: НОВАЯ МЕТОДОЛОГИЯ



**А.Д. Жейранян**  
Студент 4 года обучения  
образовательной  
программы «Программная  
инженерия» НИУ ВШЭ-  
Пермь  
addzheytanyan@edu.hse.ru



**Л.Н. Лядова**  
Доцент кафедры  
информационных технологий  
в бизнесе НИУ ВШЭ-Пермь,  
кандидат физико-  
математических наук, доцент  
LNLyadova@gmail.com

### **А.Д. Жейранян**

Студент 4 курса бакалавриата Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ–Пермь), образовательная программа «Программная инженерия». Область научных интересов связана с анализом и визуализацией данных, искусственным интеллектом, генеративными моделями, предметно-ориентированным моделированием.

### **Л.Н. Лядова**

Кандидат физико-математических наук, доцент, доцент кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ–Пермь). Область научных интересов включает языки моделирования, предметно-ориентированное моделирование, языковые инструментариумы, CASE-средства, системы имитационного моделирования.

**Аннотация.** Выполнен анализ существующих методов и инструментов визуализации данных, выявлены их ограничения, связанные с необходимостью продвинутых навыков программирования или недостаточной гибкостью настройки. Предложена методология разработки инструментов визуализации данных на основе экспертных знаний, обеспечивающая возможность эффективного создания настраиваемых визуальных представлений. Методология включает три ключевых подхода: (1) разработку предметно-ориентированных языков (*DSLs*) для визуализации данных, (2) автоматизированную генерацию скриптов визуализации с использованием библиотек *Python* и больших языковых моделей (*LLMs*) и (3) расширение существующих библиотек визуализации пакетами с пользовательскими элементами. Ожидается, что предложенная методология повысит доступность и эффективность инструментов визуализации данных, снизив требования к программированию при сохранении высокой точности представления данных в различных предметных областях.

**Ключевые слова:** визуализация данных, многоаспектная онтология, искусственный интеллект, предметно-ориентированный язык, *python*, скрипт, библиотека.

**Введение.** Область визуализации данных вызывает возрастающий интерес у исследователей, что подтверждается увеличением числа публикаций по этой тематике в базе данных *Scopus* – с 3232 работ в 2015 году до 5724 в 2024, что соответствует росту на 77 %.

При принятии решений на основе анализа данных визуализация становится неотъемлемым компонентом, позволяющим понять сложные наборы информации, выявлять скрытые закономерности и делать обоснованные выводы. Такие инструменты

находят применение у широкого круга пользователей, включая аналитиков, учёных, руководителей предприятий, менеджеров, преподавателей, студентов и т. д.

Несмотря на возрастающее значение визуализации, многие пользователи сталкиваются с серьёзными трудностями при создании качественных и информативных представлений данных. Одной из главных проблем является распространение визуализаций низкого качества, вводящих в заблуждение и не отражающих важные аспекты данных, а порой даже искажающих исходную информацию [1]. Эта проблема широко обсуждается в онлайн-сообществах, таких как *Data is Ugly* ([www.reddit.com/r/dataisugly/](http://www.reddit.com/r/dataisugly/)), где участники демонстрируют и критикуют примеры неудачных визуальных решений – от неверных подписей осей и неправильного масштабирования до избыточного использования графических элементов, которые скорее затрудняют восприятие, чем способствуют его улучшению.

Для преодоления ограничений в передаче информации о визуализируемых объектах и процессах необходимо внедрение пользовательских спецификаций визуализации [2]. Они дают возможность адаптировать диаграммы под конкретные задачи и особенности предметной области, обеспечивая более эффективное представление данных.

Существующие инструменты визуализации отличаются степенью кастомизации и уровнем реализации спецификаций. В большинстве случаев предлагается лишь базовая настройка параметров, что не всегда удовлетворяет потребности пользователей. Некоторые сценарии требуют более глубоких модификаций, включая возможность создания собственных типов диаграмм. Так, например, на рисунке 1 показаны два случая, характерных для баскетбольной статистики: первая диаграмма адаптирует турнирные деревья, а вторая иллюстрирует эффективность бросков с разных участков площадки.

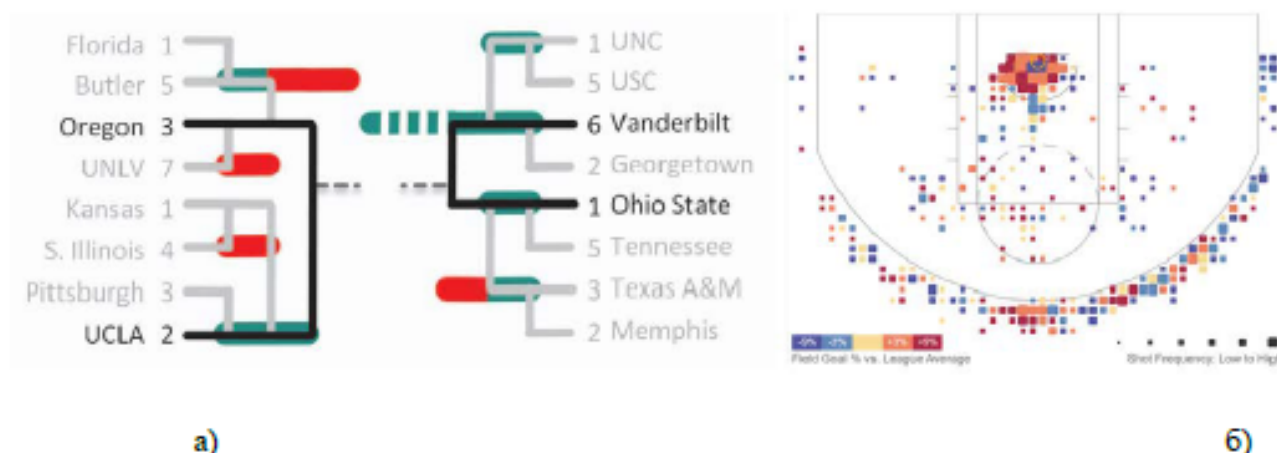


Рисунок 1. а – Адаптивная визуализация дерева для турнирных сеток [3]. б – Визуализация баскетбольного броска [4]

Такая глубокая настройка зачастую требует использования языков программирования. Это создает барьеры для пользователей без технической подготовки и подчеркивает необходимость *low-code* решений.

Кроме того, многие исследователи отмечают сложность создания визуализаций: традиционные методы часто требуют глубоких знаний в области программирования, что представляет значительные трудности для нетехнических специалистов [5]. Аналогичным образом, в исследовании [2] подчеркивается, что даже для опытных специалистов разработка эффективных визуальных представлений остается трудоемкой задачей.

В связи с этим возрастает потребность в новом подходе, позволяющем адаптировать диаграммы под специфику предметных областей и решаемых задач без высоких требований к навыкам программирования.

В данном исследовании сформулированы следующие исследовательские вопросы:

– (RQ1) какие типы диаграмм достаточны для передачи большинства идей визуализации?

– (RQ2) какие сильные и слабые стороны присущи существующим средствам визуализации данных?

– (RQ3) какие методы могут быть интегрированы для снижения требований к навыкам программирования конечных пользователей?

Цель исследования заключается в разработке собственной методологии визуализации данных, способной преодолеть ограничения современных средств, а именно, позволяющей гибко настраивать визуализацию данных под специфику решаемых задач и предметных областей, при этом минимизируя требования к навыкам программирования пользователей. Настройка будет осуществляться на основе экспертных знаний, аккумулированных в базе знаний системы – многоаспектной онтологии.

**Классификация методов визуализации.** В следующих разделах будет представлен анализ существующих типов диаграмм и инструментов визуализации данных с целью устранения существующего разрыва между пользователями без технических навыков и процессом создания настраиваемых визуализаций.

По мере развития области визуализации данных появляются новые типы диаграмм. На сегодняшний день общее количество типов диаграмм превышает 60. В связи с расширением их числа исследователи разделились на две группы.

Одна группа исследователей [6] считает, что для иллюстрации количественных данных в 90% случаев достаточно 5 типов диаграмм: круговая диаграмма, линейчатая диаграмма, гистограмма, линейный график и точечная диаграмма. А остальные 10% – это использование их комбинаций, таких как линейный график с гистограммой и т. д.

В противоположность этому, другая группа исследователей считает, что традиционные типы диаграмм могут быть недостаточными для эффективной визуализации данных. Соответственно, они разрабатывают новые методы визуализации и исследуют инновационные подходы к классификации диаграмм. Среди широко известных подходов к классификации можно выделить следующие: по типу данных, по форме представления [7], по цели создания [8-9], по интерактивности [7] и др.

Заметна разрозненность мнений (RQ1). Однако остаётся фактом, что разнообразие применяемых методов велико, и пользователи постоянно сталкиваются с задачами, требующими разработки новых техник визуализации данных.

**Сравнительный анализ подходов к визуализации данных.** Ряд исследований [9] показал, что инструменты визуализации данных играют решающую роль в улучшении анализа и представления данных. Перед оценкой различных инструментов необходимо установить критерии сравнения:

1 Гибкость в адаптации диаграмм к конкретным предметным областям и задачам.

2 Доступность для пользователей без знаний программирования.

Далее рассматриваются сильные и слабые стороны пяти групп средств визуализации (RQ2): электронные таблицы, BI-платформы, языки программирования, предметно-ориентированные языки и большие языковые модели.

Электронные таблицы, вероятно, являются наиболее распространенным инструментом для вычислений и анализа данных в настоящее время, поскольку они стали первыми инструментами, используемыми пользователями микрокомпьютеров. Электронные таблицы, особенно такие инструменты, как *Excel* и *Google Sheets*, широко применяются для визуализации данных благодаря их доступности, простоте использования и широкому набору стандартных типов диаграмм. Эти инструменты предлагают базовые возможности настройки, включая изменение осей, модификацию заголовков, изменение цветов и добавление подписей данных, что делает их удобными для выполнения рутинных аналитических задач [10]. Однако электронные таблицы имеют существенные ограничения, такие как базовая настройка параметров и невозможность создания новых типов диаграмм,

что снижает их эффективность при проведении анализа данных. Для устранения этих недостатков пользователи могут использовать надстройки или макросы, такие как *RExcel*, который интегрирует *Excel* с языком программирования *R*, расширяя его функциональные возможности. Однако такие решения требуют навыков программирования, что создает барьер для пользователей без технической подготовки.

Следующим традиционным подходом является использование *BI*-платформ. Лидерами в этой области, согласно *2024 Gartner Magic Quadrant for Analytics and Business Intelligence*, являются *Power BI*, *Tableau*, *Qlik* и др. Эти платформы предлагают удобные интерфейсы и интеграцию с различными источниками данных, что делает их доступными для пользователей без знаний программирования [11]. *Power BI*, например, широко используется благодаря бесшовной интеграции с инструментами *Microsoft*, но его возможности настройки для узкоспециализированных визуализаций остаются ограниченными [12-13]. Аналогично, *Tableau* предоставляет интуитивно понятные функции обработки данных, однако для создания сложных визуализаций часто требуется знание *R*, а некоторые параметры настройки ограничены предустановленными форматами [8].

В отличие от рассмотренных ранее двух групп, языки программирования предоставляют широкие возможности для создания эффективных и настраиваемых визуальных представлений данных. Наиболее популярными языками в этой области являются *Python*, *R* и *JavaScript*, каждый из которых обладает богатой системой библиотек для визуализации [9]. Кроме того, наличие открытого исходного кода позволяет пользователям разрабатывать собственные библиотеки и расширять существующие инструменты. Например, в среде *R* был создан специализированный пакет для интегративного пан-ракового анализа и визуализации мультиомных данных *TCGA* [14]. Несмотря на то, что языки программирования обеспечивают максимальную гибкость в настройке визуализации, их использование требует навыков кодирования. Специализированные библиотеки снижают это требование за счет предопределенных функций, однако базовое понимание программирования остается необходимым.

Предметно-ориентированные визуальные языки (*DSLs*) значительно упрощают разработку, позволяя пользователям создавать модели с помощью *drag-and-drop* вместо написания кода. Существующие визуальные *DSLs* [15-16] преимущественно ориентированы либо на стандартные диаграммы (например, круговые диаграммы, гистограммы), либо на специализированные типы данных, такие как геопространственные данные. В то же время *DSLs* на основе *JSON* демонстрируют сложность нахождения баланса между удобством использования и выразительностью, подчеркивая противоречие между формальными и неформальными моделями [17]. Однако возможности адаптации *DSL* под специфические области остаются ограниченными, зачастую требуя профессиональных навыков программирования. Недавние разработки, такие как *ModelViz*, показывают, как *DSLs* могут интегрировать предметные знания в визуализацию, упрощая создание дашбордов для непрофессионалов при сохранении ориентации на бизнес-цели [18]. Эти достижения свидетельствуют о тенденции к более удобным, основанным на моделях подходам в разработке инструментов визуализации данных.

Применение искусственного интеллекта в визуализации данных охватывает автоматическую генерацию, анализ информации и ее графическое представление [19-21]. Современные технологии используют большие языковые модели (*LLMs*), такие как *GPT-3* и *ChatGPT*, для генерации визуализаций на основе естественного языка, что повышает точность по сравнению с традиционными *NLP*-интерфейсами [22]. Например, система *Chat2VIS* демонстрирует, что эффективная промпт-инженерия может улучшить точность интерпретации визуализаций, сохраняя при этом безопасность данных. Однако сами разработчики системы отмечают, что генерируемые визуализации часто недостаточно гибки для сложных аналитических задач и плохо учитывают пользовательский замысел,

особенно при неоднозначных запросах [22]. Кроме того, у языковых моделей возникают сложности с креативным дизайном визуализаций, что подчеркивает необходимость лучшей интеграции человеческого опыта в машинный процесс для создания осмысленных и контекстно значимых решений.

Таблица 1 представляет сравнительную оценку всех пяти групп инструментов на основе ранее определенных критериев. Оценки выставлены по шкале, используемой в олимпийском стиле: --, +, +-, ++.

Таблица 1. Сравнение подходов к визуализации данных

Подход	Гибкость настройки диаграмм на специфику решаемых задач и предметных областей	Доступность для пользователей без знаний программирования
Электронные таблицы	-+	++
BI-платформы	+-	++
Языки программирования	++	--
DSLs	++	+-
LLMs	+-	+-

Оценка существующих инструментов визуализации данных подчеркивает ключевой разрыв: ни один метод не обеспечивает полного баланса между глубиной кастомизации и доступностью:

- электронные таблицы просты в использовании, но ограничены в возможностях настройки;
- BI-платформы предлагают большую гибкость, но по-прежнему имеют ограничения в создании новых типов диаграмм;
- языки программирования обеспечивают максимальную настройку, но требуют навыков программирования.
- предметно-ориентированные языки балансируют между гибкостью и доступностью, но являются сложными для разработки;
- большие языковые модели могут генерировать код для визуализации, но не способны самостоятельно разрабатывать принципиально новые типы диаграмм. Их эффективное использование требует умения формулировать точные текстовые запросы.

Гибридные подходы: интеграция искусственного интеллекта с существующими инструментами. Для преодоления существующих ограничений применяются гибридные подходы, объединяющие искусственный интеллект с уже используемыми инструментами (RQ3). В частности, электронные таблицы используют большие языковые модели (LLMs) для автоматической генерации статичных визуализаций. Например, *Google Sheets* уже интегрировал *Gemini* для создания графиков по пользовательским запросам, однако полученные изображения статичные и неизменяемые. Аналогично, интегрированные среды разработки (IDE) внедряют искусственный интеллект для упрощения кодирования пользователями, не имеющими опыта программирования. Подход, основанный на предметно-ориентированном моделировании (DSM), в сочетании с автоматизацией на основе генеративных моделей (как показано в исследовании [23]), способен снизить трудозатраты на разработку метамodelей языков.

Хотя развитие LLMs демонстрирует перспективные возможности, существующие решения все еще ограничены в плане кастомизации диаграмм, что требует дальнейших исследований для их более глубокой интеграции в процессы визуализации данных.

Предлагаемая методология. Методология визуализации данных начинается с классификации передовых методов визуализации, основанной на анализе литературных

обзоров. Эта классификация представлена на рисунке 2, она послужит основой для разработки метамоделей базовых языков.

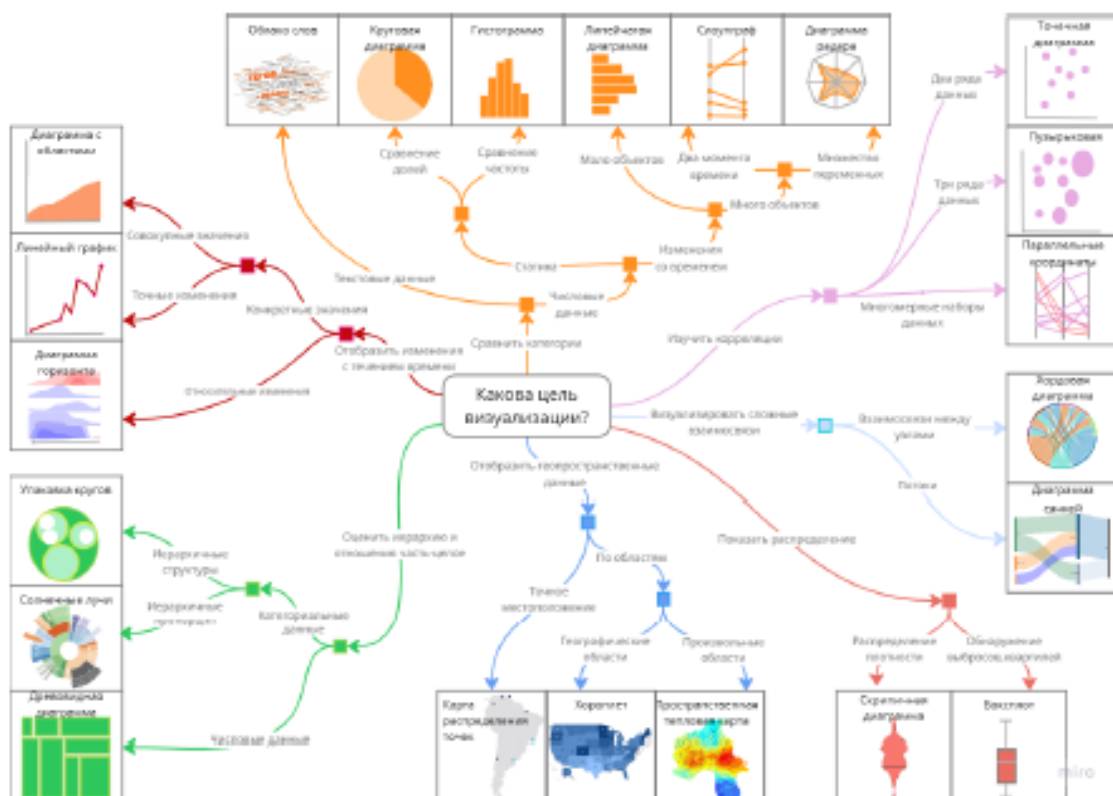


Рисунок 2. Классификация методов визуализации в зависимости от цели создания

Методология включает два типа задач: стандартные и нестандартные. Для стандартных задач применяется метод генерации кода на основе промтов в визуальной среде, что позволяет автоматически создавать и запускать скрипты. В случае нестандартных задач предполагается расширение существующих библиотек за счет пользовательских компонентов. Языково-ориентированный подход объединяет оба типа задач, обеспечивая гибкость и адаптивность. Общая схема методологии представлена на рисунке 3.



Рисунок 3. Схема методологии визуализации данных

Принятые архитектурные решения. Проектирование архитектуры системы основано на подходе, ориентированном на знания, где ядром платформы выступает база знаний. В отличие от традиционных баз данных, в качестве основы выбрана многоаспектная онтология, что обеспечивает гибкость в структурировании разнородных данных.

Онтология играет ключевую роль в хранении информации, необходимой для функционирования платформы. Она включает три основные группы онтологий:

– онтологии знаний *DSM* – содержат описание моделей, визуальных языков, предметных областей, правил трансформации и генерации *DSL* и т. д.;

– репозиторий пользовательских задач – включает онтологии пользовательских задач и применяемых методов, обеспечивая возможность многоразового использования создаваемых моделей;

– онтологии сгенерированных скриптов – охватывают онтологии промтов, результатов генерации и метаданных, что позволяет автоматизировать процесс создания и уточнения визуализаций.

Кроме того, архитектура должна интегрировать различные методы и алгоритмы, устанавливать правила *API* и обеспечивать совместимость с *DSM*-платформами.

Далее рассмотрим методы, включенные в предлагаемую методологию.

**Генерация скриптов.** Данный метод позволяет пользователям формулировать задачи визуализации на естественном языке, после чего система автоматически создает и выполняет *Python*-скрипты. Это значительно упрощает процесс работы с данными, снижая необходимость в глубоких знаниях программирования. В качестве примера можно рассмотреть генерацию визуализаций из *CSV*-файлов. В исследовании [5] была разработана платформа, которая интегрирует библиотеку *Pandas* с генеративным искусственным интеллектом и методами обработки естественного языка (*NLP*). В основе решения лежит модель *Generative AI*, основанная на архитектуре *GPT-3*, которая обучена на обширном корпусе материалов по анализу данных и их визуализации.

**Предметно-ориентированные языки.** Использование *DSL* реализуется двумя способами:

1 Сборка моделей на основе базовых языков – пользователи могут работать с готовыми *DSLs*, настраивая их параметры.

2 Создание новых предметно-ориентированных языков – разработка автоматизируется за счет отображения онтологии предметной области на метамодель базового языка. Данный подход подробно рассматривается в [24-26].

Рассмотрим пример разработки нового *DSL* для решения прикладной задачи – исследования поведения посетителей кафе в зависимости от их предпочтений в напитках и уровня шума в помещении. Предметная область: кафе, специализирующееся на подаче кофе различных сортов и других напитков. Данные собираются с помощью опросов и сенсоров, фиксирующих уровень шума, предпочтения посетителей и продолжительность их пребывания.

Онтология предметной области (объект – напитки) отображается на метамодель пузырьковой диаграммы. Результатом отображения является новый язык, метамодель которого имеет вид, представленный на рисунке 4.



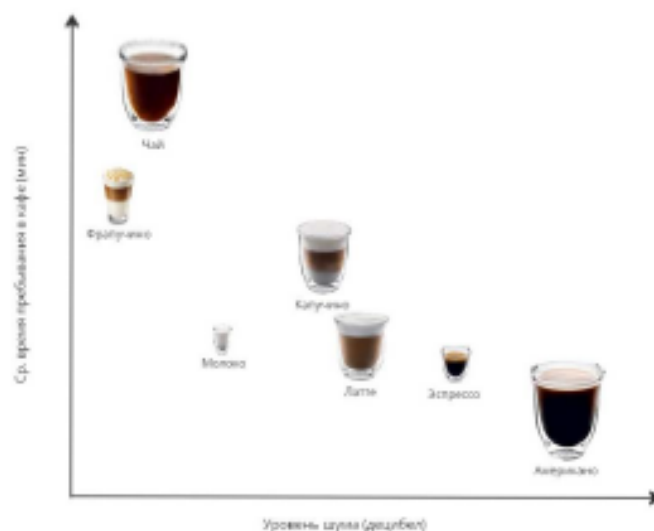


Рисунок 6. Диаграмма чашек

**Расширение библиотек визуализации.** Данный подход предлагает расширить существующую библиотеку визуализации, например, *Matplotlib* или *Plotly*, разработав пользовательские пакеты с новыми элементами и типами диаграмм. Процесс включает четыре этапа: создание, регистрация, загрузка и использование пакета.

Создание новых пакетов уже упрощает требования к навыкам программирования. Дополнительно в настоящее время изучается возможность предоставления визуальной среды – конструктора диаграмм, позволяющего пользователям собирать визуализации из отдельных элементов без написания кода.

**Направления дальнейшего исследования.** Дальнейшее развитие работы будет сосредоточено на проектировании архитектуры системы и реализации отдельных прототипов, ориентированных на автоматическую генерацию скриптов и расширение библиотек визуализации. С учетом перспектив расширения методологии в будущем необходимо создать гибкую архитектуру, позволяющую добавлять новые компоненты для реализации дополнительных методов, а также определить четкие требования к интеграции, обеспечивающие возможность разработки и вызова пользовательских алгоритмов.

Для оценки качества создаваемых визуализаций будут применяться установленные качественные метрики [27-28], связанные с когнитивными процессами. Среди метрик можно выделить: читаемость (ясность подписей и структуры), интерпретируемость (удобство извлечения выводов) и сопоставимость (эффективность сравнительных техник).

Кроме того, планируется исследование потенциала применения методов искусственного интеллекта для автоматизации построения предметно-ориентированных языков.

**Заключение.** В ходе исследования был проведен анализ существующих методов и инструментов автоматической генерации визуализаций. На основе этого анализа была предложена новая методология, включающая классификацию методов визуализации и три ключевых подхода: автоматическую генерацию скриптов на основе текстовых запросов к языковым моделям, разработку предметно-ориентированных языков и расширение существующих библиотек языков программирования пользовательскими элементами. Использование многоаспектной онтологии в качестве ядра системы позволяет хранить все знания, необходимые для функционирования платформы, и впоследствии интегрироваться с внешними аналитическими платформами, управляемыми знаниями (например, [29]). Результаты реализации подхода автоматизации создания DSL посредством отображения онтологии предметной области на метамодель базового языка прошли апробацию и были опубликованы в ряде статей [24-26]. Предложенная методология визуализации данных позволяет повысить гибкость и удобство инструментов визуализации, обеспечивая

возможность создания кастомизированных визуализаций без необходимости глубоких знаний в программировании.

### Список литературы

- [1] Sawicki J., Burdukiewicz M. *VisQualdex: a Comprehensive Guide to Good Data Visualization*. *Scientific Visualization*. 2022;15(1):127–149. DOI: 10.26583/sv.15.1.11.
- [2] Qin X., Luo Y., Tang N., Li G. Making Data Visualization More Efficient and Effective: a Survey. *The VLDB Journal*. 2020;29(1):93–117. DOI: <https://doi.org/10.1007/s00778-019-00588-3>.
- [3] Tan D., Smith G., Lee B., Robertson G. *AdaptiviTree: Adaptive Tree Visualization for Tournament-Style Brackets*. *IEEE Transactions on Visualization and Computer Graphics*. 2007;13(6):1113–1120. DOI: 10.1109/TVCG.2007.70537.
- [4] Beshai P. *Buckets: Basketball shot visualization*. 2014.
- [5] 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;8(5):1–5.
- [6] Zelazny G. *Say It With Charts Complete toolkit*. McGraw Hill Professional. 2006.
- [7] Cuadrado-Gallego J. J., Demchenko Y., Losada M. A., Ormandjieva O. Classification and Analysis of Techniques and Tools for Data Visualization Teaching. 2021 IEEE Global Engineering Education Conference (EDUCON). 2021;1593–1599. DOI: 10.1109/EDUCON46332.2021.9453917.
- [8] 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;2(3). DOI: 10.20897/jisem.201719.
- [9] Shakeel H. M., Iram S., Al-Aqrabi H., Alsbowi T., Hill R. A Comprehensive State-of-the-Art Survey on Data Visualization Tools: Research Developments, Challenges and Future Domain Specific Visualization Framework. *IEEE Access*. 2022;10:96581–96601. DOI: 10.1109/ACCESS.2022.3205115.
- [10] Mendez-Carbajo D., Dellachiesa A. Choice of Data Visualization Tool: FRED or Spreadsheets? *International Review of Economics Education*. 2023;44:100275. DOI: <https://doi.org/10.1016/j.iree.2023.100275>.
- [11] Khatuwal V. S., Puri D. Business Intelligence Tools for Dashboard Development. 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM). 2022;128–131. DOI: 10.1109/ICIEM54221.2022.9853086.
- [12] Balasubramanian V. R., Vashishtha S., Yadav N. Integrating SAP Analytics Cloud and Power BI: Comparative Analysis for Business Intelligence in Large Enterprises. *International Journal of Multidisciplinary Innovation and Research Methodology*. 2024;3(4):111–140.
- [13] Nagy E., Ferenc G. T., Molnár G. Comparison of Data Visualization Platforms through European Traffic Data. 2024 IEEE 11th International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC). 2024;000043–000048. DOI: <https://doi.org/10.1109/iccc62278.2024.10582968>.
- [14] Liao C., Wang X. TCGAplot: an R Package for Integrative Pan-Cancer Analysis and Visualization of TCGA Multi-Omics Data. *BMC Bioinformatics*. 2023;24(1). DOI: <https://doi.org/10.1186/s12859-023-05615-3>.
- [15] Ledur C., Griebler D., Manssour I., Fernandes L. G. A High-Level DSL for Geospatial Visualizations with Multi-core Parallelism Support. *IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. 2017;298–304. DOI: 10.1109/COMPSAC.2017.18.
- [16] Smeltzer K., Erwig M. A Domain-Specific Language for Exploratory Data Visualization. 17th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences. 2018;1–13. DOI: 10.1145/3278122.3278138.
- [17] McNutt A. M. No grammar to rule them all: A survey of JSON-style DSLs for visualization. *IEEE Transactions on Visualization and Computer Graphics*. 2023;29(1):160–170. DOI: <https://doi.org/10.1109/tvcg.2022.3209460>.
- [18] Khakpour A., Vázquez-Ingelmo A., Colomo-Palacios R., García-Peñalvo F. J., Martini A. ModelViZ: A Model-Driven Engineering Approach for Visual Analytics System Design. *IEEE Access*. 2024;12:42667–42682. DOI: <https://doi.org/10.1109/access.2024.3379268>.
- [19] 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*. 2022;27(2):453–463. DOI: 10.1109/TVCG.2020.3030403.
- [20] 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;17(6). DOI: <https://doi.org/10.1007/s11704-023-2691-y>.
- [21] 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;28(12):5049–5070. DOI: <https://doi.org/10.1109/tvcg.2021.3099002>.

[22] Maddigan P., Susnjak T. Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, CodeX and GPT-3 Large Language Models. *IEEE Access*. 2023;11:45181–45193. DOI: <https://doi.org/10.1109/access.2023.3274199>.

[23] Luo Y., Yang Y. Large Language Model and Domain-Specific Model Collaboration for Smart Education. *Frontiers of Information Technology & Electronic Engineering*. 2024;25(3):333–341. DOI: <https://doi.org/10.1631/fitee2300747>.

[24] Dzheiranian A. D., Ermakov I. D., Proskuryakov K. A., Lyadova L. N. Designing Data Visualization System Based on Language-Oriented Approach. *Trudy ISP RAN/Proc. ISP RAS*. 2024;36(2):127-140. DOI: 10.15514/ISPRAS-2024-36(2)-10.

[25] Джейранян А. Д., Ермаков И. Д., Проскуряков К. А., Лядова Л. Н. Разработка инструментов визуализации данных на основе предметно-ориентированного моделирования. *GraphiCon 2024: Материалы 34-й Международной конференции по компьютерной графике и машинному зрению*. 2024;300-314. DOI: 10.25206/978-5-8149-3873-2-2024-300-314.

[26] 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;16(4):82–101. DOI: 10.26583/sv.16.4.08.

[27] Behrisch M., Blumenschein M., Kim N. W., Shao L., El-Assady M., Fuchs J., Seebacher D., Diehl A., Brandes U., Pfister H., Schreck T., Weiskopf D., Keim D. A. Quality Metrics for Information Visualization. *Computer Graphics Forum*. 2018;37(3):625–662. DOI: 10.1111/cgf.13446.

[28] Midway S. R., Brum J. R., Robertson M. Show and Tell: Approaches for Effective Figures. *Limnology and Oceanography Letters*. 2022;8(2):213–219. DOI: <https://doi.org/10.1002/lol2.10288>.

[29] Zayakin V. S., Lyadova L. N., Rabchevskiy E. A. Design Patterns for a Knowledge-Driven Analytical Platform. *Proceedings of the Institute for System Programming of RAS (Proceedings of ISP RAS)*. 2022;34(2):43–56. DOI: [https://doi.org/10.15514/ISPRAS-2022-34\(2\)-4](https://doi.org/10.15514/ISPRAS-2022-34(2)-4).

#### Авторский вклад

Джейранян Анна Даниеловна – анализ существующих методов и инструментов визуализации данных, разработка методологии визуализации данных, классификация методов визуализации в зависимости от цели создания, описание кейса для анализа поведения посетителей кафе.

Лядова Людмила Николаевна – постановка задачи исследования, руководство исследованием.

## OVERCOMING THE LIMITATIONS OF MODERN DATA VISUALIZATION TOOLS: A NOVEL METHODOLOGY

*A.D. Dzheiranian*  
Undergraduate Student of  
Educational Program “Software  
Engineering” NRU HSE-Perm

*L.N. Lyadova*  
Associate Professor of the  
Department of Information  
Technologies in Business of NRU  
HSE-Perm, Cand. Sci. (Phys.-  
Math.), Associate Professor

**Abstract.** An analysis of existing data visualization methods and tools was conducted, revealing their limitations associated with the need for advanced programming skills or insufficient configuration flexibility. A methodology for developing data visualization tools based on expert knowledge is proposed, enabling the effective creation of customizable visual representations. The methodology comprises three key approaches: (1) the development of Domain-Specific Languages (DSLs) for data visualization; (2) the automated generation of visualization scripts using Python libraries and large language models (LLMs); and (3) the extension of existing visualization libraries with packages that incorporate custom elements. It is expected that the proposed methodology will improve the accessibility and efficiency of data visualization tools by reducing programming requirements while maintaining high accuracy in data representation across various domains.

**Keywords:** data visualization, multifaceted ontology, artificial intelligence, domain-specific language, python, script, library.