УДК 004: 378.147

**Обзор литературных источников по теме автоматизации процессов проектирования и визуализации компьютерных схем и диаграмм \***

**М.Л. Лапардин[[1]](#footnote-1)\*\***

Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

**Review of literature sources on technologies for automating design processes and visualizing computer circuits and diagrams\*\*\***

**M.L. Lapardin**

Don State Technical University, Rostov-on-Don, Russian Federation

|  |  |  |
| --- | --- | --- |
| **Аннотация**  *Введение.* В последние годы наблюдается значительный рост интереса к автоматизации проектирования и визуализации схем и диаграмм. Это связано с необходимостью повышения точности, гибкости и эффективности программных средств для представления сложных структурных моделей. Для решения проблемы автоматизации проектирования и визуализации схем и диаграмм активно используются современные математические модели, методы визуализации и алгоритмы обработки графов. Основное внимание в отечественных исследованиях уделяется разработке инструментов, позволяющих проектировать и анализировать схемы и диаграммы, учитывая специфику и потребности российских пользователей. Целью данного обзора является анализ существующих российских и зарубежных исследований в области создания автоматизированных средств проектирования и визуализации компьютерных схем и диаграмм. Рассматриваются основные подходы, используемые алгоритмы и модели, разработанные для реализации этих задач.  *Материалы и методы.*  В данном исследовании проанализированы как российские, так и зарубежные источники, охватывающие аспекты проектирования и визуализации схем и диаграмм. Включены статьи и диссертации, посвященные теоретическим и практическим вопросам автоматизации этих процессов. Графовые алгоритмы, методы машинного обучения и вычислительные модели, адаптированные для использования в графических редакторах, являются одними из наиболее распространенных в исследовательской литературе. Значительное внимание также уделено сравнению различных методов, что позволяет определить наиболее эффективные подходы к проектированию интерфейсов и визуализации крупных схем.  *Результаты исследования*  Исследования показали, что многие из существующих методов для проектирования и визуализации диаграмм требуют адаптации к конкретным требованиям пользователя и условиям эксплуатации. Рассмотренные работы исследователей подтверждают важность дальнейшего развития инновационных методов и алгоритмов для повышения адаптивности интерфейса и упрощения проектирования диаграмм.  *Обсуждения и заключения*  На основе проведенного обзора можно сделать вывод, что задачи проектирования и визуализации диаграмм во многом схожи, но различия в методах компоновки, адаптивности интерфейсов и оптимизации процесса автоматизации делают каждую задачу уникальной. Российские и зарубежные исследователи рассматривают автоматизацию проектирования диаграмм как перспективное направление, востребованность которого будет только увеличиваться по мере роста требований к скорости и эффективности разработки.  ***Не менее 250 слов***  **Ключевые слова:** Автоматизация проектирования, визуализация диаграмм, графический редактор, схемы и диаграммы, адаптивный интерфейс, алгоритмы визуализации. |  | Based on the review, it can be concluded that the tasks of designing and visualizing diagrams are similar in many ways, but differences in layout methods, adaptability of interfaces and optimization of the automation process make each task unique. Russian and foreign researchers consider automation of diagram design as a promising area, the demand for which will only increase as the requirements for speed and efficiency of development increase.  **Keywords:** Design automation, diagram visualization, graphic editors, adaptive interface, visualization algorithms. |

**Введение.**

Задача автоматизации проектирования и визуализации компьютерных схем и диаграмм является комплексной и трудоемкой. В связи с развитием сложных технических и информационных систем, возникает потребность в ПО, обеспечивающим создание, редактирование и анализ схем и диаграмм различной сложности. Данная задача занимает важное место в бизнесе России, поскольку создание современного редактора компьютерных схем и диаграмм невозможно без рассмотрения некоторых иностранных и российских источников.

Редактор схем и диаграмм является полезным инструментом для бизнес-процессов в компании в виду наглядного представления и оптимизации рабочих процессов. Он позволяет визуализировать все этапы, задачи и связи между подразделениями и ресурсами, обеспечивая понимание сложных процессов на интуитивном уровне. Это упрощает коммуникацию между отделами, улучшает управление проектами и помогает выявить неэффективные действия в процессах, требующих улучшения.

В целом, процесс автоматизации построения схем и диаграмм можно отнести к классу многофакторных задач, требующих разработки специализированных методов и алгоритмов, которые способны учитывать индивидуальные требования пользователей и позволяющих адаптироваться к любым изменениям.

Проанализировав степень разработанности изучаемой темы в России, я пришел к выводу, что качественных исследований автоматизации проектирования и визуализации схем и диаграмм на данный момент критически мало.

**Объектом исследования** диссертационной работы являются процессы автоматизации проектирования и визуализации схем и диаграмм, связанные с построением и сопровождением схем в различных областях проектирования.

В условиях быстрого развития информационных технологий и потребности в эффективных средствах визуализации и проектирования различных систем, задачи автоматизации создания и редактирования схем и диаграмм становятся важным направлением в области программного обеспечения. Важность таких систем заключается в их применении в разных сферах — от проектирования компьютерных схем и диаграмм до автоматизации рабочих процессов в инженерии и менеджменте. Разработка и совершенствование таких инструментов требует учета разнообразных требований, включая удобство пользовательского интерфейса, гибкость и масштабируемость системы, что подчеркивает актуальность и практическую значимость данного направления.

При создании редактора схем и диаграмм можно столкнуться с рядом сложностей, которые нужно учитывать при проектировании и разработке:

* гибкость интерфейса и настройка. Редактор должен предоставлять возможности для адаптации интерфейса к различным потребностям пользователей. Это включает возможность настройки элементов, выбора стилей, добавления анимаций и поддержки уникальных графических представлений. Подобная гибкость увеличивает сложность разработки, так как требует поддержки широкого набора инструментов и стилей.
* управление сложными диаграммами. Для поддержки масштабных проектов редактор должен эффективно обрабатывать диаграммы, состоящие из множества взаимосвязанных элементов. Это требует применения алгоритмов для оптимизации структуры диаграмм, автоматического выравнивания элементов и управления иерархиями. В противном случае работа с крупными проектами становится неудобной и менее эффективной.
* совместимость и интеграция. Редактор должен интегрироваться с другими системами и поддерживать импорт и экспорт файлов в различных форматах (например, XML, SVG, PNG). Это особенно важно для приложений, где необходимо обмениваться данными с другими инструментами, такими как системы CAD, BPM или базы данных.
* поддержка разных типов диаграмм и схем. Пользователи могут использовать редактор для создания различных типов диаграмм (например, диаграммы последовательностей, UML-диаграммы, блок-схемы, ER-диаграммы, схемы бизнес-процессов и т. д.). Каждый тип диаграммы имеет свои особенности, что увеличивает сложность реализации и требует универсальности в подходах к визуализации и взаимодействию.
* эффективность и производительность. С ростом объема данных в диаграмме повышаются требования к производительности редактора. Важно, чтобы интерфейс работал плавно даже с большим числом объектов и линий связи. Это требует оптимизации рендеринга и обработки графических элементов.
* обработка ошибок и исключений. При работе с редакторами диаграмм пользователи часто совершают ошибки, например, создавая циклы там, где их не должно быть, или нарушая логические связи. Редактор должен обрабатывать такие ситуации, предупреждать о них и, по возможности, помогать исправлять ошибки.
* интуитивное взаимодействие и удобство работы. Разработка интуитивного интерфейса, удобного для пользователей с разным уровнем подготовки, также представляет собой вызов. Пользователь должен легко освоить базовые функции, а опытные пользователи — быстро находить продвинутые возможности.
* безопасность и контроль версий. В случаях, когда схемы создаются и редактируются коллективно или содержат конфиденциальные данные, необходимо предусмотреть функции контроля версий, возможность отслеживания изменений и защиты данных от несанкционированного доступа.
* поддержка различных платформ. Если редактор предназначен для многоплатформенного использования, он должен одинаково хорошо работать на различных устройствах и операционных системах, что усложняет процесс разработки и тестирования.

Анализ литературных источников и исследований показывает, что задачи автоматизации проектирования и визуализации схем и диаграмм можно разделить на две основные категории. Первая группа включает задачи, ориентированные на обработку большого объема данных, представленных в структурированном виде, таких как параметры элементов схем и стандартные шаблоны визуализации. Эти задачи, как правило, легко поддаются автоматизации и могут быть решены с помощью применения формальных алгоритмов для визуального представления и обновления схем.

Вторая группа задач связана с обработкой неполных, изменяющихся и зачастую плохо структурированных данных, которые часто зависят от контекста применения, таких как специфические настройки для отдельных бизнес-процессов или сложные пользовательские взаимодействия. Эти задачи требуют применения гибких и адаптивных алгоритмов, способных учитывать изменчивость входных данных и специфические требования пользователя. Решение задач второй группы обычно связано с наибольшими трудностями, так как требует комплексного подхода, включающего как формальные методы, так и эвристические алгоритмы для реализации адаптивных интерфейсов и персонализированных решений.

В рамках автоматизации процесса проектирования и визуализации схем можно выделить ключевые критерии успешности:

* адаптивность алгоритмов. Возможность масштабирования для разных типов и объемов данных;
* гибкость интерфейса. Обеспечение интерактивности и возможности индивидуальной настройки элементов;
* эффективность использования вычислительных ресурсов при работе с крупномасштабными схемами и диаграммами.

**Модели и алгоритмы автоматизации проектирования и визуализации схем и диаграмм.**

Автоматизация проектирования и визуализации схем и диаграмм — это многогранная задача, требующая применения различных моделей, методов, алгоритмов и подходов. В данной области накоплен значительный объем исследований, посвященных как теоретическим аспектам, так и практическим решениям, включая разработку программных средств. Среди популярных подходов к автоматизации проектирования стоит отметить объектно-ориентированные, иерархические и графовые модели.

Объектно-ориентированные модели позволяют представить элементы системы в виде объектов, которые обладают свойствами и методами. Это облегчает управление сложными структурами, поддерживает модульность и обеспечивает повторное использование компонентов. Например, каждый элемент диаграммы (узел, блок, соединитель) может быть представлен как объект с чётко определёнными атрибутами, такими как размеры, положение, цвет или логическое поведение. Наследование и полиморфизм дают возможность создавать специализированные элементы, сохраняя при этом их интеграцию в общую структуру.

Объектно-ориентированное моделирование применимо, например, в системах автоматизации проектирования инженерных сетей. При этом можно описать систему в виде набора классов, где каждый класс представляет определённый тип оборудования или соединения. Это позволяет строить сложные структуры из модульных компонентов, поддерживать их обновление и масштабирование.

Иерархические модели, в свою очередь, обеспечивают структурированное представление данных. В них элементы верхнего уровня объединяют связанные элементы нижнего уровня, что позволяет разделять сложные схемы на логические подсистемы. Это особенно важно для крупных проектов, где необходима наглядность и возможность работы на разных уровнях детализации. Например, в проектировании электронных схем иерархия позволяет группировать компоненты по подсистемам, обеспечивая как локальную, так и глобальную оптимизацию.

В контексте автоматизации проектирования схем и диаграмм иерархические модели позволяют:

* упрощать визуализацию. Элементы верхнего уровня могут представлять агрегаты, такие как подсистемы или группы, а элементы нижнего уровня — их детали.
* оптимизировать работу с данными. Переход на более высокий уровень позволяет скрыть незначимые детали и сосредоточиться на общем представлении системы.
* поддерживать модульность. Иерархические подходы облегчают повторное использование элементов и упрощают внесение изменений.

Примером использования иерархических моделей может служить создание UML-диаграмм, где различные уровни иерархии (классы, методы, свойства) описывают архитектуру программного обеспечения. Аналогичный подход применим к проектированию электронных схем, где верхний уровень может представлять схему в целом, а нижние уровни — отдельные компоненты, такие как транзисторы или резисторы.

Одним из ключевых подходов к автоматизации проектирования схем является использование графовых моделей, позволяющих представлять элементы диаграмм в виде вершин, а связи между ними — в виде рёбер графа. Эти модели упрощают структурирование схем, делая их оптимальными и легко читаемыми. Задача построения и редактирования диаграмм в графовых моделях часто сводится к решению задач размещения и маршрутизации. Графовые модели позволяют применить разнообразные методы, такие как раскраска графов для минимизации пересечений, и алгоритмы поиска кратчайшего пути для оптимизации связей.

Графовые модели имеют несколько ключевых преимуществ:

* гибкость представления данных. Практически любой тип связей и элементов схемы можно описать с помощью графа;
* масштабируемость. Графы позволяют работать с большими и сложными структурами, которые возникают в реальных приложениях;
* интеграция алгоритмов. Использование проверенных алгоритмов из теории графов (например, поиска кратчайшего пути или раскраски графов) делает решения эффективными и удобными для реализации.

В редакторе схем и диаграмм элементы (такие как узлы в электрической цепи или блоки в блок-схеме) могут быть вершинами графа, а соединения между ними — рёбрами. Если элемент связан с несколькими другими, это отображается как многосвязный граф. Дополнительные атрибуты рёбер, такие как вес или направление, позволяют учитывать физические или логические характеристики связей.

Для улучшения читаемости и упрощения восприятия диаграмм часто используются алгоритмы раскраски графов. Раскраска позволяет минимизировать пересечения связей и избежать визуального беспорядка. Например, при проектировании электрических схем можно использовать раскраску для разделения токовых цепей. Задача минимизации пересечений связей сводится к поиску раскраски вершин или рёбер графа так, чтобы соседние элементы не пересекались.

Формально, если — граф, где V — множество вершин, а E — множество рёбер, то задача раскраски заключается в нахождении функции , такой что для любого ребра :

Минимальное число цветов необходимое для раскраски графа, называется его хроматическим числом.

Если необходимо провести линии связи между компонентами схемы, это сводится к задаче поиска кратчайшего пути на графе. Алгоритмы Дейкстры или Беллмана-Форда применяются для определения оптимальных маршрутов, учитывая такие параметры, как длина связи или пропускная способность.

Пусть — вес ребра между вершинами и . Задача поиска кратчайшего пути между двумя вершинами и может быть выражена как:

где — путь в графе, соединяющий начальную вершину и конечную вершину , — ребро графа, входящее в путь , — вес ребра, который может представлять расстояние, стоимость или любую другую метрику.

В сложных диаграммах может возникнуть необходимость внесения изменений в режиме реального времени, например, при добавлении новых элементов или связей. Для этого применяются алгоритмы динамической обработки графов, которые позволяют быстро пересчитывать раскраску или маршруты без необходимости полной переработки всей схемы.

Допустим, задача — это визуализировать бизнес-процесс в виде графа. Вершины этого графа будут представлять отдельные процессы или задачи в рамках бизнес-операции, а рёбра — зависимости между ними, например, порядок выполнения задач или их взаимное влияние. Это типичная задача для схем бизнес-процессов, где важно наглядно показать связи между элементами и минимизировать перекрестные линии и сложность восприятия.

На первом этапе важно расположить вершины графа таким образом, чтобы избежать перегрузки визуальной информации и минимизировать перекрытия рёбер. Для этого используется алгоритм force-directed layout. Этот алгоритм моделирует вершины как частицы, между которыми действуют силы: сила отталкивания (аналогичная электростатическим силам) и сила притяжения (аналогичная пружинной силе между вершинами, соединенными ребром).

Алгоритм находит равновесие, в котором минимизируются пересечения рёбер и вершины распределяются по пространству, обычно равномерно. Сначала вершины расположены случайным образом, но по мере выполнения алгоритма они начинают "отталкиваться" друг от друга, пока не находят оптимальное положение. В результате, граф становится менее запутанным и легко воспринимаемым.

На 2 этапе, когда вершины графа уже размещены, необходимо заняться маршрутизацией рёбер. Это означает, что нужно прокладывать линии между вершинами так, чтобы они не пересекались, и их длина была как можно короче. Для этого используется алгоритм Дейкстры, который является эффективным методом поиска кратчайших путей в графе.

Алгоритм Дейкстры находит оптимальные маршруты от одной вершины к другой, минимизируя общую длину рёбер. В контексте схемы бизнес-процесса это может быть полезно, например, для минимизации затрат времени или других ресурсов на выполнение бизнес-процессов, где каждая связь между задачами имеет свою "стоимость" или важность.

На заключительном этапе проводится раскраска графа. Этот шаг помогает выделить важные элементы схемы, такие как критические пути или узлы с высокой нагрузкой. Раскраска может быть использована для выделения процессов, которые влияют на общий результат бизнес-процесса.

Критический путь — это последовательность взаимосвязанных процессов, где каждый процесс зависит от предыдущего. Отклонение в сроках выполнения любого процесса на критическом пути напрямую влияет на сроки выполнения всего проекта. Эти пути могут быть выделены особым цветом или стилем, чтобы акцентировать внимание на их важности.

Узлы с высокой нагрузкой — это такие процессы, которые требуют значительных ресурсов или времени, что делает их узкими местами в бизнес-процессе. Выделение этих узлов позволяет менеджерам сосредоточиться на оптимизации таких процессов и улучшении эффективности.

В результате после всех этих этапов схема бизнес-процесса становится наглядной и оптимизированной. Все элементы связаны по логике работы, а важные и критичные элементы выделены для дальнейшего анализа и возможной оптимизации.

Для повышения эффективности построения схем также используются линейное программирование и генетические алгоритмы.

Линейное программирование (ЛП) — это математический метод, используемый для оптимизации линейных функций при наличии ограничений, которые также являются линейными. В контексте редакторов схем линейное программирование применяется для задачи оптимизации размещения элементов на рабочем пространстве, где необходимо минимизировать или максимизировать некоторые параметры, такие как пространство, занимаемое элементами, или количество пересечений рёбер.

Предположим, что в редакторе схем имеется несколько объектов, которые должны быть размещены в двумерном пространстве. Пусть и ​ — это координаты элемента на плоскости. Задача заключается в минимизации функции стоимости, например, общей длины рёбер между объектами или количества пересечений рёбер.

Целевая функция может быть представлена в виде:

где ​ — вес (важность) рёбер между элементами и , а — расстояние между этими элементами. Это расстояние рассчитывается по формуле Евклида:

При этом для размещения элементов могут быть наложены различные ограничения. Например, для предотвращения наложения элементов или их выхода за пределы рабочего пространства, ограничения могут быть следующими:

, для всех ,

где и — размеры рабочей области (например, размеры экрана или страницы). Также можно учитывать расстояние между элементами для минимизации перекрытий:

, для всех .

Задача линейного программирования заключается в нахождении значений ​ и , которые минимизируют целевую функцию при соблюдении всех этих ограничений.

Линейное программирование эффективно работает для задач, где количество элементов не слишком велико, а сами ограничения достаточно просты и линейны. Однако при большом числе элементов задача может становиться вычислительно сложной, и тогда переходят к более сложным методам, таким как генетические алгоритмы.

Генетические алгоритмы (ГА) представляют собой класс эвристических методов оптимизации, вдохновлённых процессом естественного отбора в биологии. Они используются для поиска решений в сложных или многозадачных пространствах, где традиционные методы, такие как линейное программирование, могут быть недостаточно эффективными. Генетические алгоритмы являются подходящими для задач оптимизации размещения элементов на диаграмме, где нет явных аналитических решений или когда пространство поиска слишком велико.

Генетический алгоритм работает через несколько итераций, называемых поколениями, на каждом из которых популяция решений эволюционирует, улучшая свои характеристики. Процесс включает несколько ключевых этапов:

* инициализация популяции. На начальном этапе генерируется случайная популяция возможных решений. Каждое решение представляет собой конфигурацию расположения элементов на рабочем пространстве.
* оценка приспособленности. Для каждого решения вычисляется функция приспособленности (fitness function), которая отражает качество данного расположения. Например, функция приспособленности может учитывать такие факторы, как минимизация длины рёбер и уменьшение числа пересечений.
* скрещивание (Crossover). Два "родительских" решения комбинируются для создания нового решения, называемого "потомком". Этот процесс напоминает биологическое скрещивание, когда наследуются характеристики двух родителей, что позволяет комбинировать сильные стороны разных решений.
* мутация. Существует вероятность, что потомок подвергнется мутации, что означает случайное изменение его характеристик. Мутации могут быть полезны для исследования новых областей пространства решений и предотвращения застревания в локальных оптимумах.
* селекция. Из текущей популяции выбираются лучшие решения, которые будут составлять следующее поколение. Это помогает сохранять решения с высокой приспособленностью, направляя эволюцию популяции к более качественным решениям.
* повторение процесса. Алгоритм повторяет шаги скрещивания, мутации и селекции до тех пор, пока не будет достигнут приемлемый результат или не будет выполнено заданное количество поколений.

Для задачи оптимизации размещения элементов на диаграмме функция приспособленности может быть определена как комбинация нескольких факторов. Например, для минимизации длины рёбер и уменьшения количества перекрытий можно использовать следующую функцию:

где и — веса, определяющие важность соответствующих факторов, — индикатор перекрытия элементов и , который принимает значение 1, если элементы перекрываются, и 0 в противном случае.

Генетический алгоритм будет пытаться минимизировать эту функцию, что позволит найти оптимальное расположение элементов на диаграмме.

Эвристические методы играют важную роль в автоматизации проектирования, особенно в задачах, связанных с размещением элементов на графах и диаграммах. В отличие от более сложных математических методов, таких как линейное программирование или генетические алгоритмы, эвристические подходы фокусируются на нахождении хороших, но не обязательно оптимальных, решений за минимальное время. Эти методы идеально подходят для задач, где необходим быстрый результат, и они широко применяются в редакторах схем и диаграмм для упрощения и ускорения процесса создания визуальных представлений информации. Два из наиболее распространённых эвристических методов — это алгоритм поиска ближайшего соседа и жадный алгоритм. Оба метода обеспечивают простоту реализации и высокую скорость.

Метод поиска ближайшего соседа (Nearest Neighbor Search) представляет собой эвристический алгоритм, который используется для последовательного размещения объектов на основе их близости друг к другу. Идея заключается в том, что на каждом шаге выбирается следующий элемент, минимизирующий расстояние до уже размещённых объектов.

Формально алгоритм можно описать следующим образом. Пусть и — координаты элемента на плоскости, а — множество всех ещё не размещённых элементов. На каждом шаге алгоритм выбирает элемент, который минимизирует расстояние до уже размещённого:

где — это расстояние между элементом и ближайшим элементом , а — Евклидово расстояние между координатами этих элементов.

В контексте редакторов схем и диаграмм этот метод может быть использован для размещения объектов на рабочем пространстве так, чтобы минимизировать пересечения рёбер. Например, при проектировании электрических схем или графов, элементы (например, узлы или объекты) часто должны располагаться с минимальными пересечениями рёбер, что облегчает восприятие схемы.

Хотя метод поиска ближайшего соседа не гарантирует глобально оптимальное решение, он может эффективно работать в случаях, когда важно быстро получить решение, особенно если количество объектов ограничено. Преимущество метода заключается в его простоте и скорости работы, что делает его удобным инструментом для первичной расстановки элементов.

Жадный алгоритм (Greedy Algorithm) является ещё одним популярным методом, широко используемым в оптимизации и автоматизации проектирования. В отличие от метода поиска ближайшего соседа, который ориентируется на минимизацию расстояний между ближайшими элементами, жадный алгоритм на каждом шаге принимает локально оптимальное решение для всей задачи.

Основная идея жадного алгоритма заключается в том, что на каждом шаге выбирается элемент, который улучшает общую цель задачи, не учитывая возможные негативные последствия для других шагов. Например, для задачи минимизации длины рёбер между элементами, жадный алгоритм на каждом шаге выбирает позицию для нового элемента, минимизируя суммарную длину рёбер между ним и уже размещёнными объектами.

Формально это можно записать как:

где и — координаты элементов и , а — расстояние между ними.

Жадный алгоритм может быть использован для решения задач, где требуется минимизация определённых параметров, таких как длина рёбер в графах или перекрытия элементов в диаграммах. Например, в редакторах схем этот метод может помочь уменьшить длину соединений между объектами, что в свою очередь улучшает читаемость и визуальную ясность схемы.

Подобно методу поиска ближайшего соседа, жадный алгоритм не всегда приводит к глобально оптимальному решению, но его главная сила — в простоте и скорости. Он может быть полезен в задачах, где время отклика критично, а необходимость в точном решении не столь велика.

А.Ю. Ткаченко из Бурятского государственного университета представил подход к визуализации графов с использованием принципов объектно-ориентированного программирования. Исследование акцентируется на создании программного компонента, где классы объектов графа реализуют хранение и манипуляцию данными о вершинах и рёбрах, включая их визуализацию. Методы оптимизации, такие как буферизация и отображение только видимой части графа, позволили значительно снизить нагрузку на графическую подсистему, что увеличило производительность системы. Применение абстракции и инкапсуляции обеспечило модульность, а наследование и полиморфизм дали возможность легко расширять функционал компонента. Представленные методы успешно применимы в задачах проектирования сложных графовых структур, таких как сети и процессные диаграммы, минимизируя системные ресурсы и повышая эффективность пользовательского взаимодействия. Модульная структура и оптимизация системы могут применяться для повышения производительности интерфейса редактора диаграмм. Это напрямую связано с разработкой адаптивного редактора, где требуется эффективная обработка больших графов [3].

В свою очередь, И.В. Капустин из Тульского государственного университета в своей работе акцентировал внимание на разработке веб-ориентированных графических редакторов с использованием JavaScript. Основным инструментом разработки выбрана библиотека SigmaJS, позволяющая эффективно строить графы и визуализировать их в веб-приложениях. В работе подробно рассматриваются свойства узлов (например, количество портов и уникальные идентификаторы) и рёбер (такие как пропускная способность и тип среды передачи), которые отображаются в графе. Для улучшения интерактивности графического редактора использованы обработчики событий, такие как двойной клик по узлу или правый клик по рёбру, что позволило реализовать редактирование свойств элементов графа и добавление новых объектов. Его подход позволил внедрить алгоритмы маршрутизации линий связи в браузерных приложениях, обеспечив простоту интеграции таких инструментов в информационные системы. Это решение показало эффективность в области автоматизации проектирования сетевых схем, где требуется высокая динамичность изменений. Результаты этого исследования могут быть использованы для разработки пользовательского интерфейса в редакторе схем и диаграмм [4].

В статье Ю.М. Васильева и Г.М. Фридмана из Санкт-Петербургского государственного экономического университета представлен гибридный эвристический алгоритм, предназначенный для минимизации числа пересечений рёбер в иерархических графах. Исследование сосредоточено на решении задачи оптимального размещения вершин графа на слоях с учётом ограничений по времени вычислений. Для реализации метода применяются подходы, включающие кластеризацию графов по их характеристикам, последовательное использование неточных алгоритмов и подбор оптимальных цепочек таких алгоритмов для каждого кластера. Проведённые расчёты, охватывающие 2000 тестовых графов с различными параметрами, демонстрируют, что предложенный гибридный алгоритм уменьшает число пересечений рёбер на 10.9% по сравнению с лучшими отдельными эвристическими методами. Предложенный гибридный эвристический алгоритм улучшает читаемость диаграмм и повышает их визуальную ясность. Это особенно актуально для автоматизации построения сложных визуальных представлений [5].

Исследователи из Монреальского университета в Канаде разработали инновационный подход к автоматическому построению UML-диаграмм классов из текстовых спецификаций, написанных на естественном языке. В своей работе они использовали большой набор данных из 305 UML-диаграмм, извлеченных из открытого репозитория AtlanMod Zoo. Этот набор данных включал в себя 8706 фрагментов, каждый из которых был аннотирован вручную для создания текстовых описаний классов и отношений. Основной этап их метода включал классификацию предложений текста на описывающие либо классы, либо отношения с использованием алгоритмов машинного обучения. Полученные результаты показывают, что точность метода при строгих критериях остается ограниченной, но подход демонстрирует перспективы в автоматизации проектирования диаграмм. Этот подход не только упрощает процесс генерации диаграмм, но и позволяет сократить время, затрачиваемое на их создание вручную, что особенно важно в условиях работы с большими массивами документации [12].

Исследование, выполненное студентами из Гонконгского университета науки и технологий (Гуанчжоу, Китай), сосредоточено на использовании генеративного искусственного интеллекта (GenAI) для автоматизации визуализации. Авторы обобщили современные подходы к применению генеративных моделей, таких как GAN и диффузионные модели, в задачах визуализации данных. Они выделяют четыре основные стадии процесса: улучшение данных, генерация визуальных отображений, стилизация и интерактивность. Работа также акцентирует внимание на ограничениях современных подходов, таких как нехватка специализированных данных и сложность оценки качества генерируемых визуализаций. Исследование имеет важное значение для дальнейшей интеграции технологий GenAI в средства автоматизации проектирования и визуализации диаграмм [13].

Другое значительное исследование выполнено исследователями из Тунцзийского университета (Шанхай, Китай). В своей работе они проанализировали автоматизацию в процессе создания повествовательных визуализаций, включая использование инструментов, которые автоматически преобразуют структурированные данные в визуальные элементы, такие как диаграммы и схемы. Авторы выделяют уровни автоматизации и описывают, как современные инструменты помогают минимизировать человеческое вмешательство при создании сложных визуализаций, сохраняя при этом их читабельность и точность. Это исследование демонстрирует, как автоматизация может улучшить процессы визуализации в контексте больших массивов данных [14].

Исследование было проведено специалистами из Университета технологии Сиднея (University of Technology Sydney, Австралия), Тяньцзиньского университета (Tianjin University, Китай) и Университета Западного Сиднея (University of Western Sydney, Австралия). Работа посвящена визуализации крупных взвешенных графов, часто используемых для анализа корпоративных связей и структуры баз данных. Основная цель исследования заключалась в разработке гибридного метода, сочетающего кластеризацию узлов и улучшенные алгоритмы визуализации. Для кластеризации использовались методы на основе цепей Маркова (Markov Chain) и алгоритм DPCW, которые эффективно разделяют граф на логические группы. После этого применяется модифицированный алгоритм, основанный на силовом моделировании (force-directed layout), который минимизирует пересечения рёбер и равномерно распределяет узлы. Узлы графа моделируются как частицы с отталкивающимися зарядами, а рёбра — как пружины, притягивающие связанные узлы, что позволяет достичь равновесного состояния. Такой подход упрощает восприятие и анализ больших графов, обеспечивая удобство работы с социальными сетями, маршрутными схемами и сетевыми структурами [15].

**Выводы.** Таким образом, на основе обзора литературных источников по теме автоматизации проектирования и визуализации компьютерных схем и диаграмм, можно сделать следующие выводы:

* Объектно-ориентированные, иерархические и графовые модели являются основными подходами для проектирования современных систем автоматизации построения схем и диаграмм. Каждая из них предлагает уникальные подходы к организации и визуализации данных.
* Объектно-ориентированный подход способствует модульности и повторному использованию элементов, что особенно важно при проектировании сложных схем и диаграмм с иерархическими взаимосвязями.
* Графовые модели отличаются универсальностью и гибкостью и предоставляют возможности для использования широкого спектра алгоритмов, таких как раскраска графов и поиск кратчайшего пути, которые оптимизируют структурирование и маршрутизацию элементов схем.
* Иерархические модели позволяют структурировать данные и процессы в соответствии с уровнями вложенности. Эти модели упрощают восприятие сложных взаимосвязей и способствуют минимизации пересечений связей в диаграммах с большим количеством элементов. Применение методов, основанных на иерархической структуре, таких как алгоритмы Сугиямы, улучшает читаемость и упрощает дальнейшую обработку схем.
* Включение алгоритмов линейного программирования и эвристических методов, таких как жадный алгоритм и метод ближайшего соседа, обеспечивает баланс между точностью решений и скоростью их получения.
* Внедрение современных методов визуализации на основе графов, включая динамическую обработку данных и раскраску, позволяет значительно сократить время проектирования и повысить читаемость схем.
* Тенденции развития включают интеграцию алгоритмов с облачными технологиями, а также использование веб-редакторов, которые упрощают совместную работу и доступность инструментов.
* В зарубежной практике активно используются методы искусственного интеллекта, включая глубокое обучение, для автоматической оптимизации схем и диаграмм. Эти подходы перспективны и требуют адаптации для использования в российских условиях.
* Российские исследования в первую очередь сосредоточены на разработке решений для конкретных задач, таких как автоматизация проектирования сетей и оптимизация маршрутов, что подтверждает важность дальнейших работ в этой области.
* Автоматизация процессов проектирования и визуализации схем и диаграмм – это активно развивающаяся область. Дальнейшие исследования в этом направлении будут способствовать созданию более мощных, гибких редакторов схем и диаграмм.

**Библиографический список**

1. Шатлов К.Г. Редактор графов и блок-схем // [Доклады Томского государственного университета систем управления и радиоэлектроники](https://cyberleninka.ru/journal/n/doklady-tomskogo-gosudarstvennogo-universiteta-sistem-upravleniya-i-radioelektroniki), - 2007. - № 1(15). - c. 57-60.
2. Якимов И.М., Абзалова Л.Р., Кирпичников А.П., Мокшин В.В. Краткий обзор графических редакторов структурных моделей сложных систем // Вестник Казанского технологического университета. - 2014. - № 3 (58). - c. 213-221.
3. Ткаченко А. Ю. Визуализация графа с использованием парадигмы объектно-ориентированного программирования // Вестник Бурятского государственного университета. Математика, информатика. - 2015. - № 1. - c. 33-41.
4. Капустин И.В. Разработка графического редактора сетей средствами языка JavaScript // Теория и практика современной науки. - 2016. - № 9 (15). - c. 176-181.
5. Васильев Ю.М., Фридман Г.М. Визуализация кооперативных схем: гибридный эвристический алгоритм для минимизации количества пересечений ребер при укладке графа // [Известия Санкт-Петербургского государственного экономического университета](https://cyberleninka.ru/journal/n/izvestiya-sankt-peterburgskogo-gosudarstvennogo-ekonomicheskogo-universiteta). - 2017. - №1-2 (103). - c. 87-93.
6. Якимов И.М., Кирпичников А.П., Мокшин В.В., Костюхина Г.В., Шигаева Т.А. Комплексный подход к моделированию сложных систем в системе BPwin-Arena // [Вестник Казанского технологического университета](https://cyberleninka.ru/journal/n/vestnik-kazanskogo-tehnologicheskogo-universiteta), - 2014. - №6. T. 17. - c. 287-292.
7. Григорьева Т.Е., Панов С.А. Разработка графической нотации для многоуровневого компьютерного моделирования бизнес-процессов // Объектные системы, - 2016. - №13. - с. 15-21.
8. Кознов Д.В. Предметно-ориентированное визуальное решение для сбора и упорядочивания информации при разработке информационной Web-системы // Компьютерные инструменты в образовании, - 2013. - № 5. - с. 3-16.
9. Шаров О.Г. Исследование и разработка методов и средств реализации диаграммных графических языков САПР: автореф. дис. … канд. тех. наук / О.Г. Шаров. – Ульяновск, 2007. - 19 с.
10. Ларчик Е.В. Реализация поддержки навигационных сервисов в Eclipse GMF: автореф. дис. … канд. техн. наук. – Санкт-Петербург, - 2012. - 41 с.
11. Степанов П.А. Модели, алгоритмы и программные средства определения визуальных языков на основе вычислительных моделей: автореф. дис. …канд. техн. наук/ П.А. Степанов – Санкт-Петербург, - 2019. - 20 с.
12. Yang S. Towards Automatically Extracting UML Class Diagrams from Natural Language Specifications, [MODELS '22: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings](https://dl.acm.org/doi/proceedings/10.1145/3550356). Montreal University, Canada, 2022, pp. 396-403.
13. Y. Ye, J. Hao, Y. Hou, Z. Zhan, W. Shishi, X. Yuyu, L. W. Zeng Generative AI for Visualization: State of the Art and Future Directions, journal: Visual Informatics 8(5), 2024.
14. Chen Q., Cao S., Wang J., Cao N. How does automation shape the process of narrative visualization: A survey of tools, Intelligent Big Data Visualization Lab, Tongji University, 2024, pp. 4429-4448, vol. 30.
15. Hua J., Huang M. L., Nguyen Q. V., Drawing large weighted graphs using clustered force-directed algorithm, Proceedings of the 18th International Conference on Information Visualisation, The Hong Kong University of Science and Technology, 2014, pp. 13-17.

Поступила в редакцию

Сдана в редакцию

Запланирована в номер

**Сведения об авторе(-ах)**

|  |  |
| --- | --- |
| Ф. И. О. *(полностью)* | Лапардин Максим Леонидович |
| Должность | Магистрант |
| Место работы, учебы | Донской государственный технический университет (ДГТУ) |
| Адрес места работы, учебы | РФ, г. Ростов-на-Дону, пл. Гагарина,1 |
| Контактный телефон | 8(928)906 37 07 |
| Адрес электронной почты | makleo2002@mail.ru |
| Адрес, на который следует выслать авторский экземпляр журнала  *(с указанием почтового индекса)* | РФ 344010 г. Ростов-на-Дону,  ул. Мечникова, 154а, кв.1010 |

**Authors**

|  |  |
| --- | --- |
| Surname, first name, patronymic *(in full)* | Lapardin Maxim Leonidovich |
| Position | Graduate student |
| Place of work, studies | Don State Technical University (DSTU) |
| Address of the place of work, studies  *(including the name of the country)* | Gagarin sq., 1, Rostov-on-Don, Russian Federation |
| Contact telephone number | 8(928)906 37 07 |
| Email | makleo2002@mail.ru |
| Address where the author’s copy should be sent *(including the post code)* | Mechnikova Str., 154a, ap. 1010, Rostov-on-Don, Russia, 344014 |

1. \* Работа выполнена в рамках инициативной НИР  
   \*\* Автор: e-mail: [makleo2002@mail.ru](mailto:makleo2002@mail.ru)

   \*\*\* The research is done within the frame of independent R&D [↑](#footnote-ref-1)