**Общая характеристика работы**

**Тема.** Автоматизация процессов проектирования и визуализации компьютерных схем и диаграмм.

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

**Предметом исследования** являются математические модели, методы, алгоритмы и программные средства, ориентированные на создание, редактирование и поддержку диаграмм и схем, отвечающие специфическим требованиям адаптивности и гибкости.

**Целью диссертационной работы** повышение эффективности проектирования и сопровождения схем и диаграмм за счет разработки отечественного программного решения, адаптируемого под нужды российских пользователей и обеспечивающего возможность гибкой настройки под конкретные задачи и требования.

Для достижения поставленной цели были сформулированы следующие **задачи**:

1. Провести анализ существующих редакторов схем и диаграмм, выявить их функциональные возможности и ограничения.
2. Исследовать потребности пользователей и требования к современным редакторам.
3. Разработать математическую модель, обеспечивающую адаптивность редактора схем к различным прикладным задачам.
4. Создать алгоритмы и методы, позволяющие автоматизировать процесс визуализации и взаимодействия с крупномасштабными диаграммами.
5. Реализовать прототип редактора, включая возможность редактирования интерфейса и гибкой настройки параметров, и провести его тестирование.

**Методология и методы исследования**. Для решения поставленных в работе задач использованы: методы системного и функционального анализа, объектно-ориентированного и модульного проектирования, математического моделирования, включая графовые структуры и эвристические алгоритмы, методы оптимизации схем, методы разработки пользовательских интерфейсов, теория обработки и визуализации данных, методы тестирования программного обеспечения.

**1. Архитектура системы**

Клиент-серверное приложение (SPA) для векторного редактора схем и диаграмм

**Основные компоненты:**

1. **Клиентская часть (Frontend)**

* Написана на React + TypeScript
* Использует Fabric.js или Konva.js для работы с векторной графикой
* Интерактивный UI с поддержкой drag-and-drop
* Визуальное редактирование схем и диаграмм
* Настройка элементов и сохранение шаблонов
* Реализация undo/redo

1. **Серверная часть (Backend)**

* Node.js (NestJS) + PostgreSQL
* REST API для CRUD-операций (создание, чтение, обновление и удаление) с диаграммами
* WebSockets для синхронизации изменений в реальном времени

1. **База данных**

* PostgreSQL для хранения схем и диаграмм

**2. Основные структуры данных**

Редактор будет работать с графами, деревьями и геометрическими структурами.

1. **Графовые структуры (Graph)**

* Диаграмма = ориентированный граф (Graph G = (V, E))
* Вершины (Nodes) – элементы (объекты) диаграммы
* Рёбра (Edges) – связи между элементами (объектами)
* Взвешенные рёбра (Weighted Graph) – используются для оптимизации расположения
* Вес рёбер – расстояние и приоритет связи

1. **Иерархическое дерево (Tree Structure)**

* Используется для группировки элементов
* Поддерживает вложенные схемы и модульную структуру

1. **Пространственные структуры**

* КД-дерево (k-d tree) – для быстрого поиска пересечений
* R-дерево – для эффективного хранения и поиска объектов на холсте

1. **Модели данных**

* JSON-структура для хранения диаграмм
* SVG-формат для экспорта

**3. Алгоритмы**

Требуется расставить элементы так, чтобы минимизировать пересечения и улучшить читаемость диаграммы.

1. **Алгоритм Форсированной Симуляции (Force-Directed Layout)**

Это метод для визуализации графов, который использует силы притяжения и отталкивания между вершинами, чтобы расположить их на плоскости. Цель заключается в минимизации пересечений рёбер и создании эстетически приятной структуры. Этот алгоритм подходит для визуализации социальных сетей и других типов графов.

1. **Алгоритм Сугиямы (Sugiyama Algorithm)**

Это метод для расположения графов, преимущественно ориентированных, с целью минимизации перекрещиваний рёбер и улучшения читаемости. Он часто используется в визуализации иерархических структур, таких как организационные схемы. Алгоритм выполняется в несколько этапов, включая цикл разбиения на уровни и перестановку рёбер.

1. **Эвристика Барицентра (Barycentric Heuristic)**

Это метод, использующий центр тяжести (барицентр) для расположения вершин графа таким образом, чтобы минимизировать пересечения рёбер и улучшить структуру графа. Вершины размещаются с учетом их соседей, что помогает сделать граф более симметричным. Эвристика часто применяется в задачах графового планирования и визуализации.

1. **Алгоритм Имитации Отжига (Simulated Annealing)**

Это стохастический метод оптимизации, вдохновленный процессом охлаждения металлов, который помогает найти глобальный минимум в сложных задачах. Он использует случайные изменения в решении и вероятность принятия худших решений на ранних этапах, чтобы избежать застревания в локальных оптимумах.

1. **Алгоритм раскраски графов**

Это метод, который назначает цвета вершинам графа так, чтобы смежные вершины имели разные цвета. Основная цель — найти минимальное количество цветов, необходимых для раскраски графа.

1. **Линейное программирование**

Это метод оптимизации, который заключается в максимизации или минимизации линейной функции, при этом на переменные накладываются линейные ограничения. Задача решается с помощью алгоритмов, таких как симплекс-метод.

1. **Эвристические алгоритмы**

Это методы поиска решения, которые не гарантируют оптимальность, но позволяют найти приемлемые решения за приемлемое время для сложных задач. Они используются, когда точные методы слишком медленные или невозможны. Эвристические алгоритмы включают в себя жадные алгоритмы.  
Жадные алгоритмы - это методы, которые принимают решения на каждом шаге, основываясь на локально оптимальном выборе, предполагая, что это приведет к глобально оптимальному решению. Жадные алгоритмы не всегда приводят к оптимальному решению. Примером жадного алгоритма является алгоритм поиска ближайшего соседа (Nearest Neighbor Algorithm). Это жадный алгоритм, используемый для задач кластеризации. Он начинает с произвольной точки и на каждом шаге переходит к ближайшей ещё не посещённой точке, пока все вершины не будут пройдены. Этот метод прост и быстр, но не всегда находит оптимальное решение, так как может застревать в локально неоптимальных маршрутах.

1. **Генетические алгоритмы**

Это эволюционные алгоритмы, которые имитируют процессы естественного отбора и мутаций для поиска оптимальных решений в сложных пространствах. Они работают с популяциями возможных решений, комбинируя их и эволюционируя в несколько поколений. Являются классом эвристических алгоритмов.

**4. Методы**

1. **Методы автоматического выравнивания**

Это набор техник для упорядочивания объектов на диаграмме. Например, элементы могут быть выровнены по сетке или центрированы для улучшения восприятия.

1. **Методы оптимизации размещения меток**

Это подходы, направленные на уменьшение наложений меток и улучшение их читаемости, что важно для диаграмм с большим количеством объектов и связей.

1. **Методы масштабирования и адаптации к экрану**

Поддержка динамического изменения масштаба диаграммы в зависимости от размера экрана или предпочтений пользователя.

1. **Методы минимизации пересечений**

Это подходы для улучшения визуального восприятия, включая упорядочивание слоёв в графах и перераспределение узлов, чтобы избежать пересечений рёбер.

1. **Методы кластеризации и группировки элементов**

Эти методы используются для объединения объектов в группы на основе их связности или характеристик. Алгоритм K-means помогает классифицировать элементы в зависимости от их свойств.

1. **Методы анализа графов**

Включает методы для анализа связности графа, например, поиск компонент связности, а также алгоритм маршрутизации (например, алгоритм Дейкстры) для поиска оптимальных связей.

**Поддержка шаблонов и кастомизации**

1. **JSON-шаблоны**

Использование JSON-шаблонов позволяет пользователю сохранять и загружать свои настройки для диаграмм. Это даёт возможность повторно использовать настройки.

1. **Модульная структура компонентов**

Позволяет пользователю добавлять новые фигуры и элементы в диаграмму, расширяя возможности редактора.

1. **Визуальное оформление**

Поддержка изменения цветов, толщины линий и типов соединений, добавление и редактирование текста, изменение размеров объектов, поворот объектов и т.д.

1. **Импорт и экспорт данных**

Сохранение и загрузка диаграмм в стандартном формате XML.