

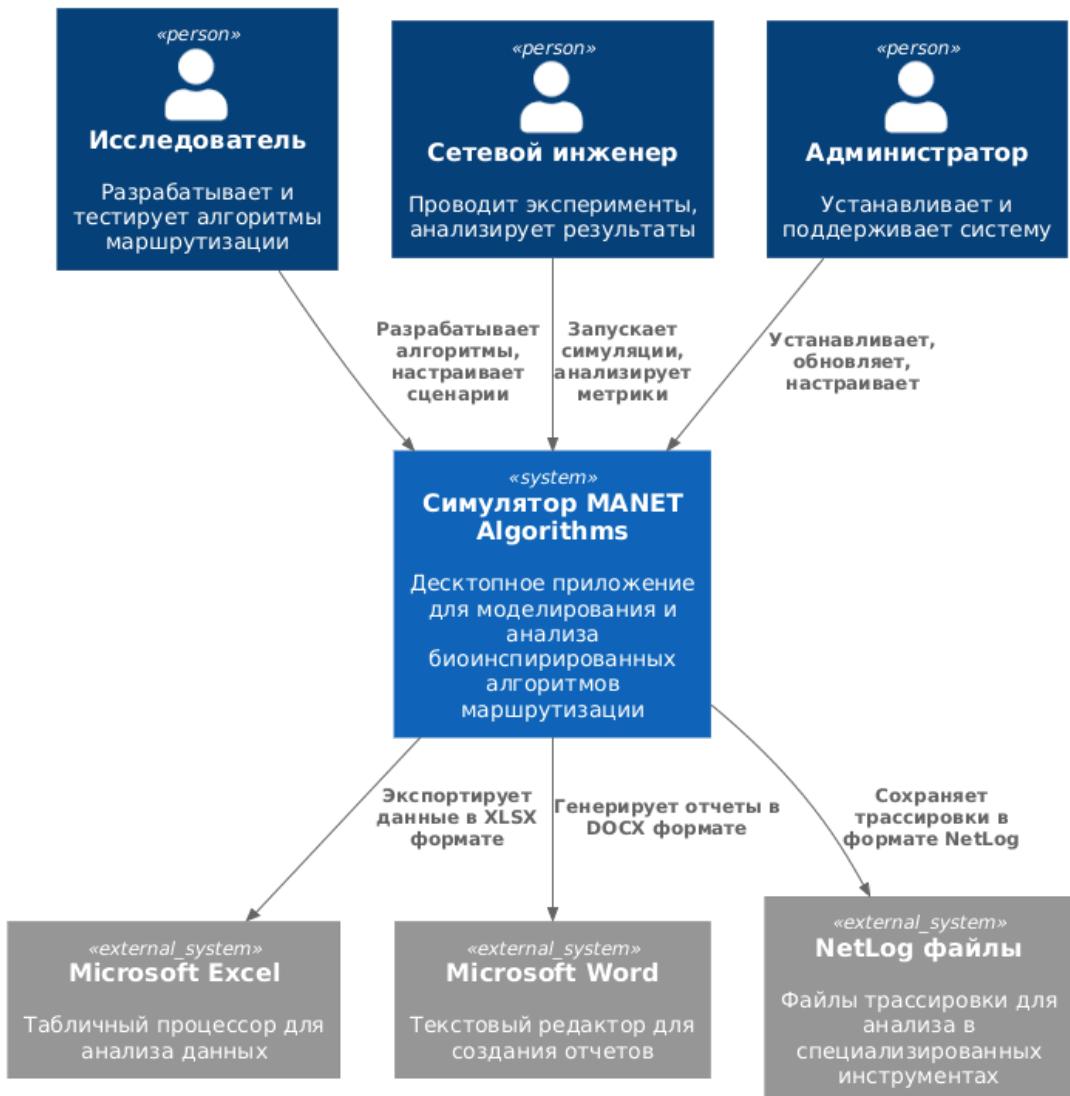
Лабораторная работа №2

Тема: Использование нотации C4 model для проектирования архитектуры программной системы

Цель работы: Получить опыт использования графической нотации для фиксации архитектурных решений.

Диаграмма системного контекста

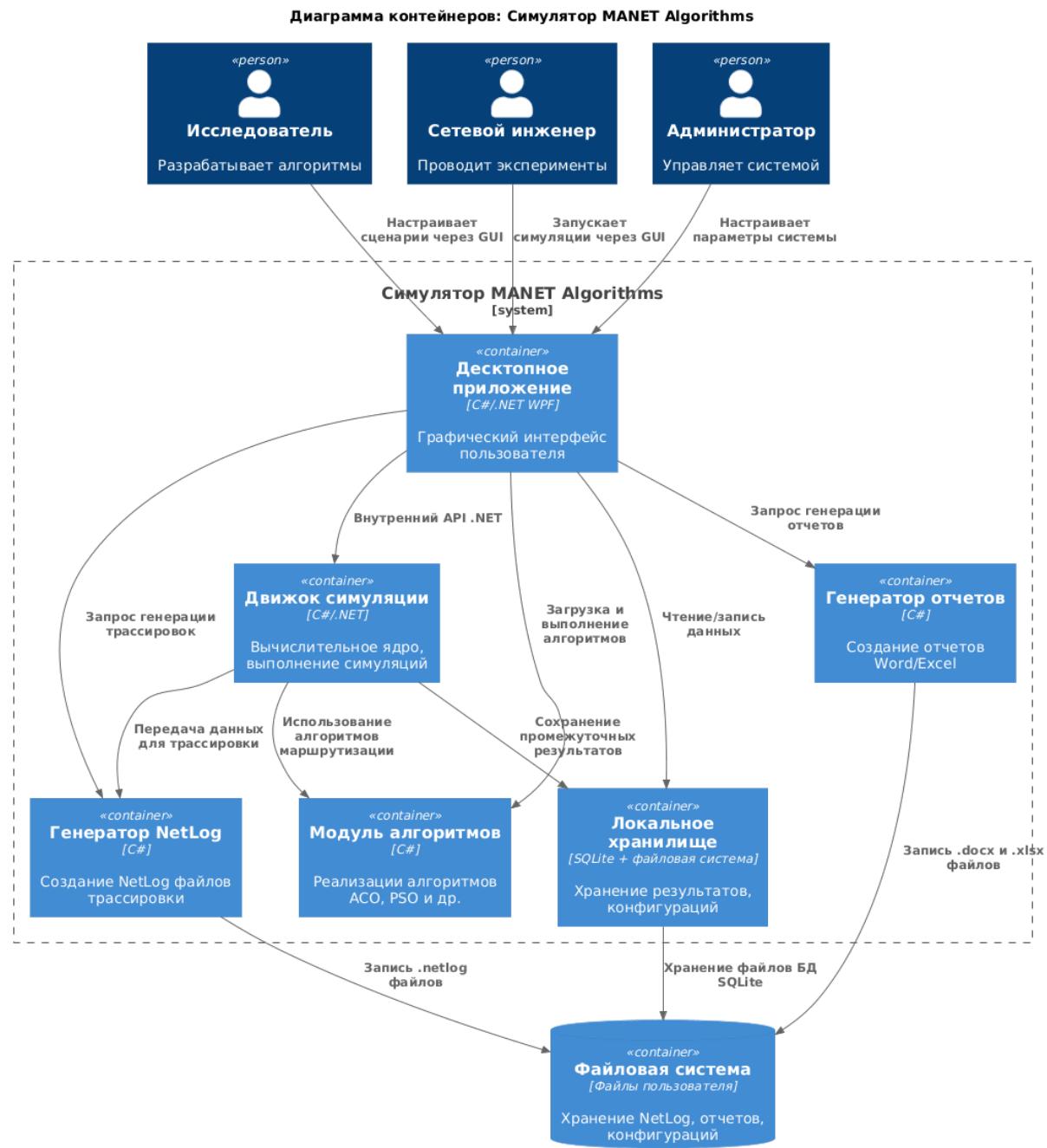
Системный контекст: Симулятор биоинспирированных алгоритмов маршрутизации в MANET



1. Персонажи:
 - 1.1. Исследователь – разработчик алгоритмов маршрутизации
 - 1.2. Сетевой инженер – пользователь системы для проведения экспериментов
 - 1.3. Администратор – техническая поддержка системы

2. Целевая система:
 - 2.1. Симулятор MANET Algorithms – десктопное приложение для Windows
3. Внешние системы:
 - 3.1. Microsoft Excel – для детального анализа числовых данных
 - 3.2. Microsoft Word – для оформления итоговых отчетов
 - 3.3. NetLog файлы – специализированный формат для анализа сетевых трассировок

Диаграмма контейнеров



Описание элементов диаграммы:

1. Десктопное приложение (C#/NET WPF): Основной графический интерфейс
2. Двигок симуляции: Ядро системы, выполняющее вычисления
3. Модуль алгоритмов: Библиотека с реализациями биоинспирированных алгоритмов

4. Генератор NetLog: Компонент для создания файлов сетевой трассировки
5. Генератор отчетов: Создание документов Word и Excel с результатами
6. Локальное хранилище: SQLite база данных для хранения истории экспериментов

Архитектурный стиль и обоснование выбора:

Выбранный стиль: Монолитная (Layered) архитектура с четким разделением ответственности

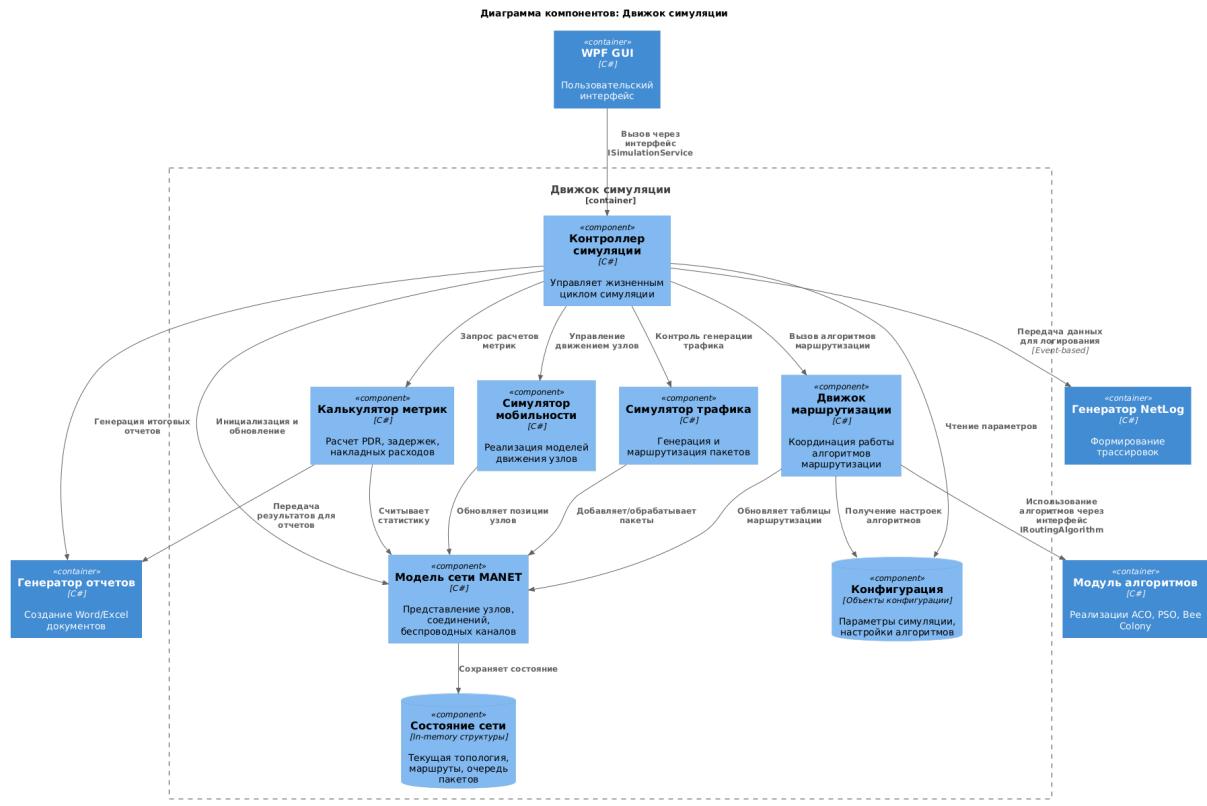
Причины выбора для десктопного приложения:

1. Простота развертывания:
 - Один исполняемый файл для пользователя
 - Минимальные зависимости (только .NET Framework/Runtime)
 - Легкая установка и обновление
2. Производительность:
 - Минимальные накладные расходы на межпроцессное взаимодействие
 - Прямой доступ к памяти между компонентами
 - Оптимизация для вычислительных задач
3. Интеграция с Windows экосистемой:
 - Нативная поддержка Office Interop для работы с Word/Excel
 - Доступ к файловой системе для генерации NetLog файлов
 - Использование WPF для современного GUI
4. Упрощенная разработка и отладка:
 - Единая кодовая база
 - Проще тестирование и отладка
 - Меньше точек отказа

Структура слоев:

1. Presentation Layer: WPF приложение (GUI)
2. Business Logic Layer: Двигок симуляции, генераторы отчетов
3. Data Access Layer: Локальное хранилище (SQLite + файлы)
4. Infrastructure Layer: Модули алгоритмов, NetLog генератор

Диаграмма компонентов



Описание компонентов движка:

1. Контроллер симуляции: Главный координатор, управляет временными циклами
2. Модель сети MANET: Объектное представление сети (Node, Link, Packet, Route)
3. Симулятор мобильности: Реализация моделей Random Waypoint, Gauss-Markov, Group Mobility
4. Симулятор трафика: CBR (Constant Bit Rate), FTP-like, HTTP-like потоки
5. Калькулятор метрик: Расчет Packet Delivery Ratio, End-to-End Delay, Routing Overhead
6. Двигок маршрутизации: Интеграционный слой для алгоритмов