
Научно-исследовательские работы 2026–2027 гг.

Михаил Пожидаев

7 августа 2026 г.

«Виртуальные джуны»

Агентские ТЗ на PlantUML

Больше ни строчки кода рутинного бойлерплейта! Правильнее делегировать написание кода ИИ-агентам. Современные LLM идеально справляются с ролью «виртуальных джунов», если с ними говорить на языке архитектуры.

Новый подход к Промптингу

- ТЗ — это **UML-диаграмма** классов;
- пишем в **PlantUML**: фиксируем поток мыслей прямо в тексте;
- идеально понимают даже недорогие модели (DeerSeek и др.).

Ренессанс Software Engineering

Возврат к золотому стандарту инженерии на новом уровне:

Сначала максимально проектируй → генерацию кода оставь напоследок.

Стань пионером AI-Driven Software Engineering!

Будем исследовать паттерны UML-промптинга, метрики качества сгенерированного кода и новые парадигмы проектирования ПО. **Залетай в проект!**

LLM-Агенты в BIM-проектировании

Интеллектуальный и нескучный CAD

AI-Driven Проектирование

Исследование и разработка **мультиагентных систем** для генерации, подсчёта и модификации BIM-проектов в стандарте ISO-10303/STEP напрямую через текстовые промпты пользователей.

Proof of Concept

От текста к 3D-модели: BIM-проект генерируется и комплексно модифицируется силами LLM по команде инженера на естественном языке.

Инженерная логика в традиционных компонентах: математическое обслуживание и верификация осуществляются протестированными открытыми библиотеками, подключенными в виде тулов.

Цель: Индустриальное переформатирование. Создание open-source аналога Bentley Copilot, где агент берет на себя рутину традиционного

Инженерные вызовы (Hardcore)

- **Context Window & Big Data:** Формат IFC — промышленный стандарт обмена, но он крайне неэффективен. Придётся решать проблему парсинга и обработки огромных объемов данных в условиях лимитов токенов.
- **Fine-tuning:** Базовые модели «галлюцинируют». Потребуется **дообучение моделей** для строгого соблюдения инженерной топологии.

Развитие SHARP-Math через Lean 4

Навыки математического вывода в LLM через символьные языки

Ищем замену текстовому пруфу

Даже с крутой span-level разметкой LLM нередко люто ошибаются. Обычный текстовый пруф — это не панацея, ведь нейронка может запутаться в собственных словах. Нам нужна железобетонная база, где математика проверяется не вероятностями, а строгой логикой.

Символьные вычисления — это надёжно

Предстоит реализовать поддержку **Lean 4** под капот пайплайна для математического ризонинга.

Вместо того чтобы фантазировать с промптами, мы заставим модель генерить формальные доказательства.

Если Lean компилирует код без ошибок, значит шаг 100% валидный.

Это сверхперспективный метод детекции галлюцинаций!

Как это будет?

- **Трансляция:** Переводим natural language шаги в Lean-код.
- **Auto-Proof:** задействуем тактики Lean для чека промежуточных стейтментов.
- **Абсолютный детект:** Если Lean находит ошибку, мы ловим точный span, где логика начала разваливаться.