

Автоматизация с человеческим лицом

Дмитрий Нестерук

✂ dnesteruk

DotNext Moscow 2026

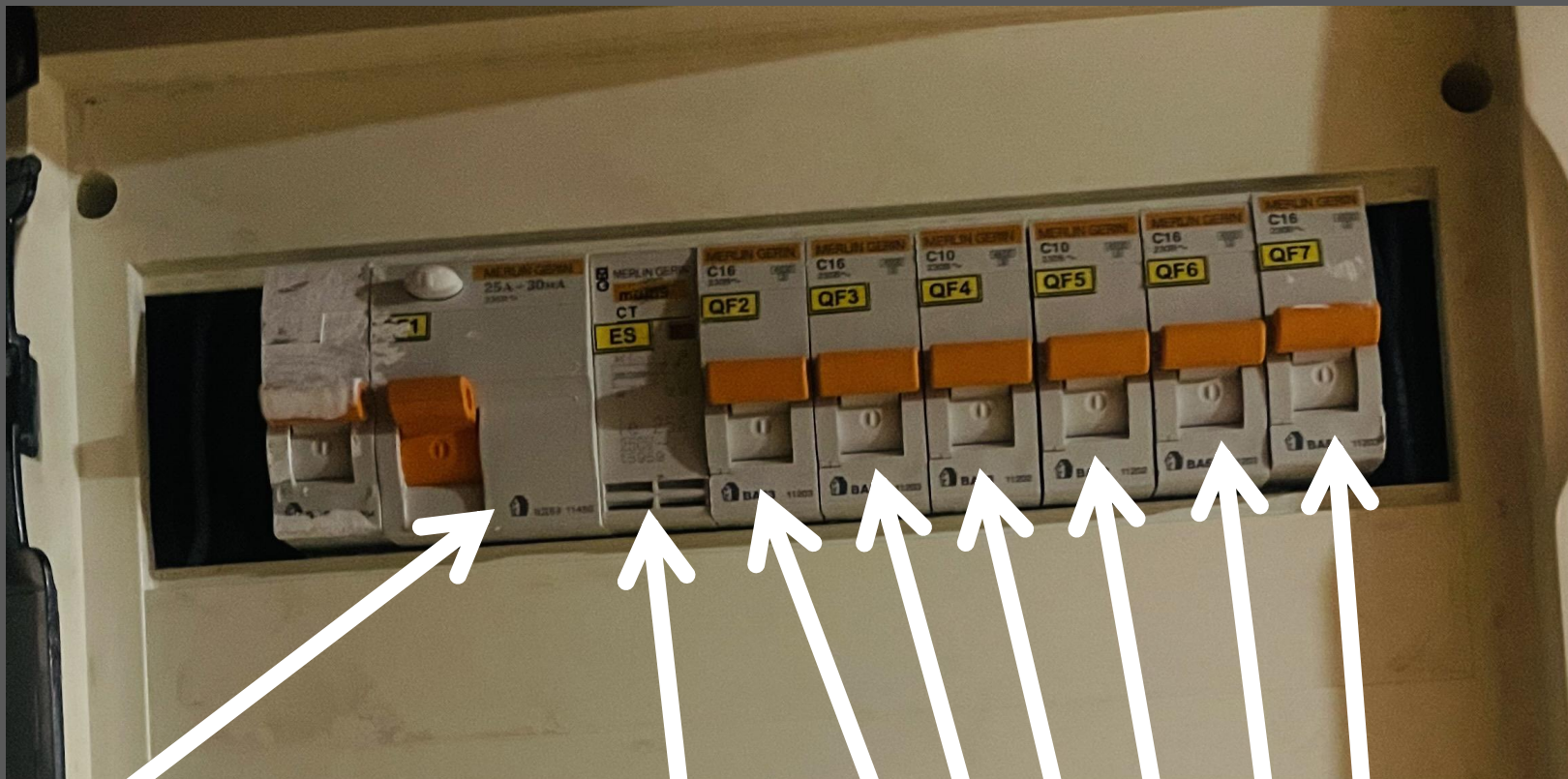
О чем рассказ?

- О примитивности инженерных систем (пром автоматизация, умный дом и т.д.)
- Как ИИ может вклиниться и улучшить положение дел
- Парадигма для создания систем автоматизации
- Больше коллекция мыслей/наблюдений а не готовый рецепт

A scene from Star Trek: The Next Generation showing the bridge of the USS Enterprise-D. Captain Picard, T'Pol, and others are at their stations. A blue speech bubble is overlaid on the left side of the image.

Execute attack
pattern Gamma

(c) Star Trek



УЗО (защита от
поражения током)

Контактор (карточка номера)

Автомат (КЗ и сверхтоки)

Автомат(ический выключатель)

- Может только вырубиться (возврат вручную)
- Не может сказать почему он вырубился (пользователь, перегрев, КЗ)
- Никого вообще не информирует что он вырубился (щёлк в ночи)
- Управление только “на месте”



Взаимодействие двух реальностей

- Проблема: успешность проекта зависит от квалификаций и заказчика и исполнителя
 - У обычного заказчика (не эксперта) нет квалификаций 😊
- Заказчик хочет быстро, дешево и качественно
 - Не обладает инженерными знаниями/подходом
 - Не знает мало/ничего про возможности автоматизации
 - Не пишет спецификации
- Быстро + дешево = один исполнитель
 - Нет детального проекта, есть “хочу свет вот тут”
 - Исполнитель = электрик, а не инженер

Автоматизация в реальном здании

- Вариант 1: ее нет вообще
 - Можно, но зачем?
- Вариант 2: минимально/избирательно
 - Карточный выключатель у входа, мастер-выключатель у кровати
- Вариант 3: гулять так гулять
 - Сделайте мне все красиво, бюджет резиновый

Все варианты подразумевают минимальную или очень ограниченную коммуникацию заказчика и исполнителя

Проблема реконфигурации

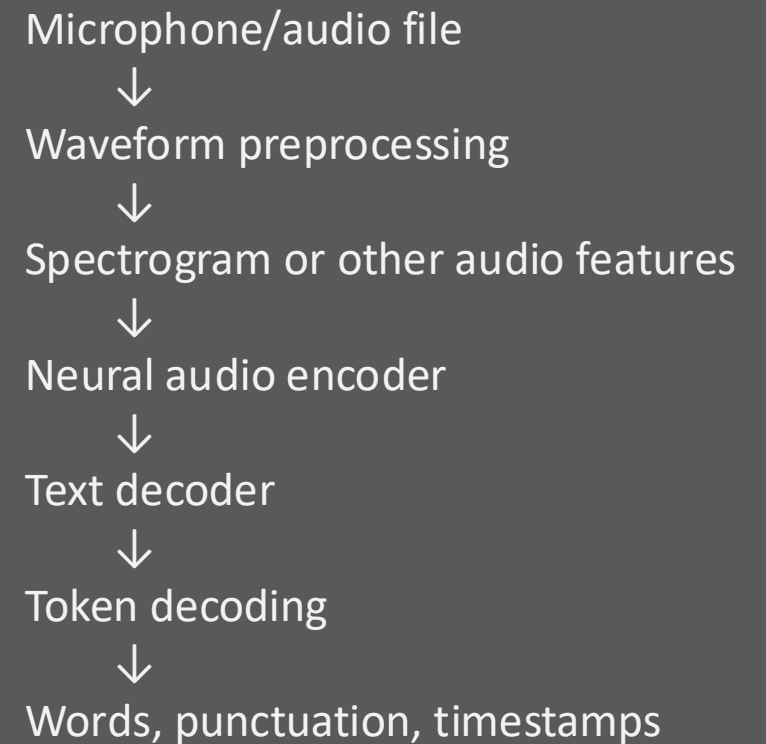
- Допустим заказчик смог сформулировать свой набор начальных требований
- Система внедрена
- Требования меняются
- Каким образом вносить изменения:
 - Вернуть инженеров, пусть меняют систему
 - Поменять систему самому... но ведь требуются навыки (например разработки, workflow editing etc.)

AI (LLM, GPT) + ASR

Новые технологии

Automatic Speech Recognition (ASR)

- Speech recognition в базовом виде встроен во многие ОСи и работает терпимо
- AI-based ASR (Whisper и т.п.) – работает очень хорошо
- Специализированные задачи
 - Транскрипция видео
 - STT (голосовой ввод)
 - AI voice chat (ChatGPT Advanced Voice)
- Перевод
 - Не только с одного языка на другой, но и...
 - Из одной предметной области в другую



ASR/AI как универсальный переводчик

- Вам нужно подготовить спеку системы
- Заказчик и исполнитель общаются на разных языках
- Что делает ИИ
 - Вытягивает из заказчика инфу “что тебе надо, скажи”
 - Доносит ее до исполнителя
 - Является переводчиком/медиатором в обе стороны
- В результате можно подготовить спеку в 2 реализациях
 - Plain-text описание для заказчика (делаем вот это)
 - Проекты (напр. ЭОМ, слаботочка) для реализации

LLMы поверх ASR

- ASR – механизм удобного ввода для
 - Объяснения того что нужно (спецификация системы)
 - Управления системой на высоком уровне
 - Внесения изменений в работающую систему
- LLM – обработка этого ввода
 - Понять что хочет пользователь, а потом
 - Вызвать нужный API; или
 - Написать новую конфигурацию системы

Как выглядит готовая система

- В качестве примера возьмем систему “умный дом”
- Примитивные беспроводные реализации
- Проводная реализация – шина данных
 - KNX, DALI, Modbus
- Много устройств которые умеют
 - Класть данные на шину (напр. показания датчиков)
 - Получать данные с шины (напр. сигнал с выключателя)
- Можно дергать через API
 - Например, Falcon.NET for KNX

Falcon.NET (KNX)

```
var groupAddress = new GroupAddress("1/1/10");
```

```
// Send ON  
bus.WriteValue(groupAddress,  
    new GroupValue(true),  
    Priority.Low);
```

Проблемы идентификации

- Устройство = широкий I/O набор
 - Без спецификации не разобраться даже интегратору
- Кто-то должен решить как соединить входы и выходы
 - И описать их назначение
- Две ключевые парадигмы:
 - “Соединение проводов” – создается адрес, в который одни устройства пишут телеграммы, а другие получают
 - Кастомная логика – сторонний контроллер или компьютер

AI-driven конфигурация

- “Хочу чтобы температура плавно снижалась вечером”
- LLM интерпретирует это как запрос для изменения настроек
- Создает скрипт для системы
- Система начинает работать по новым правилам
- Какой-то фидбэк пользователю

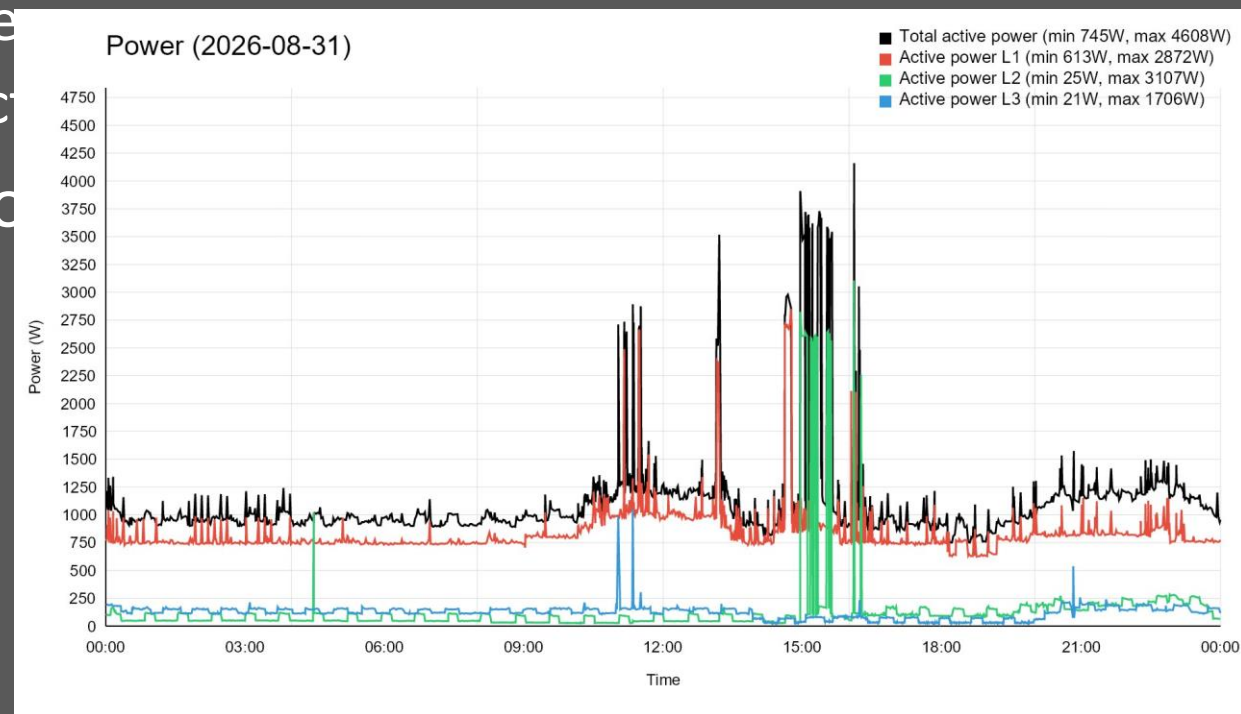
AI as Intent Compiler

МСР, API и иже с ним

- У нас есть обобщенный API который ходит в систему
- Этому API не нужно ничего особенного
 - Современные LLMы понимают концепцию API без оберток
- Обертки полезны для “жесткого” контроля выполнения
 - “Температура теплого пола не может превышать 40°” vs
 - `throw ArgumentException("Temp too high");`
- Можно ли доверять LLM с ограничениями для системы?

Человеческое лицо

- Лучший интерфейс – отсутствие интерфейса
 - Датчик включает свет в кладовке
- Голосовой ввод-вывод как средство №1
 - “Алекса”, “Алиса” и т.п. - мало, ме
 - Визуализация системы (лицо, абс
- Тактильные интерфейсы взаимо
 - Панели, крутки и т.п.
- Мессенжеры, СМС и т.д.



Выводы?

- Взаимодействие уровня Star Trek гипотетически возможно
- Пользователям не нужно разбираться в том как все работает на уровне техники
- ИИ может компилировать намерения пользователей для
 - Первоначального создания системы
 - Управления системой
 - Внесения изменений в уже работающую систему
- Потенциально нивелируем все friction points в создании более сложных систем

Спасибо за внимание!

- Вопросы?
- X dnesteruk

