

# Архитектура и прототипирование metaCASE-системы

Таран Кирилл, 345гр.  
СПбГУ, Математико-Механический факультет  
научный руководитель: ст.пр. Т.А.Брыксин

- Основная функциональность QReal
- Альтернативная архитектура
- Улучшение многопользовательской работы
- Запланированные, но не реализованные функции
- Реализация прототипа

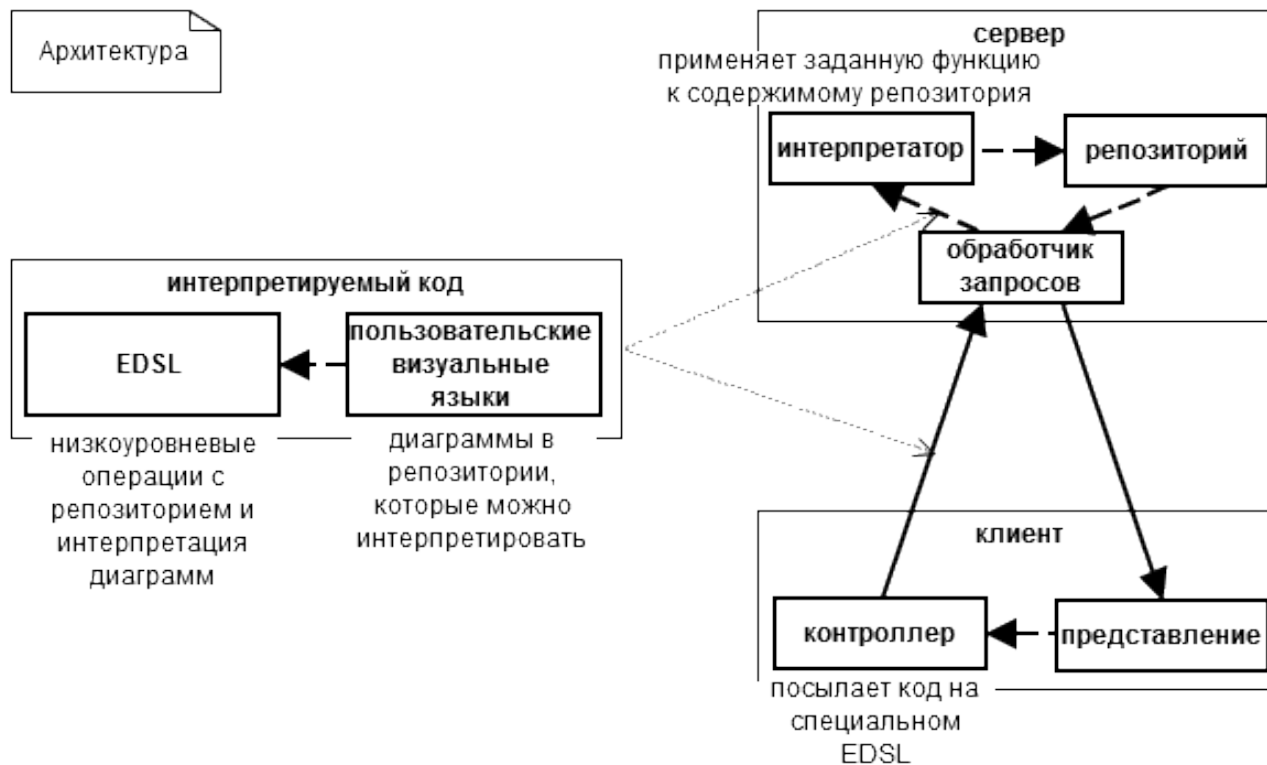
- Ключевая функция - многопользовательская работа
  - в QReal - синхронизация и удаленный репозиторий
- Отказоустойчивость
- Снижение требований к ресурсам
- Запланированные функции
  - откат действий, история событий

# Какой должна быть архитектура?



- Формат для сообщений от клиента
- Короткие сообщения могут производить большие изменения
- Расширение пользователями или сторонними разработчиками
  - изменение системы без изменения программы
- CLI интерфейс (удобно для разработки)
- Недостаток - сложность
  - устраняется при реализации как EDSL на базе интерпретируемого языка

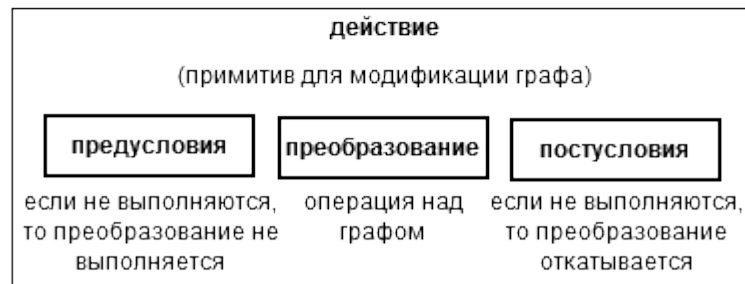
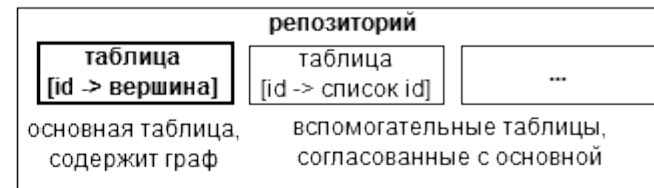
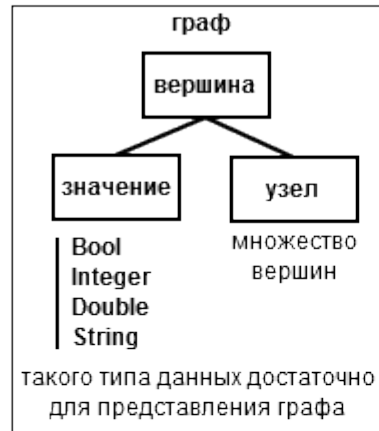
- Клиент-серверная архитектура



- Интерпретация команд на сервере

- Основная и вспомогательные таблицы, хранящие диаграмму

Примитивы для  
внутреннего  
представления



- Атомарные действия с проверками

- Язык программирования - Haskell
- HInt - библиотека для интерпретации Haskell "из кода"
- Gtk2Hs и Diagrams - GUI клиента
  - событийно-ориентированное программирование
  - специальный язык описания графики



- Клиент-серверное взаимодействие - RPC
- Thrift - инструмент для описания сервисов
  - различные транспорты (общая память, сообщения)
  - генерация интерфейсов и кода на целевом языке
  - удалённый вызов процедур

EDSL для описания преобразований диаграммы:

- все преобразования являются значениями типа **Modification**
- **apply** пытается применить действие с проверками к диаграмме
- если преобразование не соответствует типу, или проверки не выполнены, то ошибка
- преобразования "знают" о предыдущих состояниях диаграммы
- команды-сокращения
  - **bind [1,2,3]** вместо **onHistory \$ insertKnot \$ Data.Set.fromList [1,2,3]**
  - при этом, любой корректный Haskell-код разрешён

- Спроектирована metaCASE-система
  - поддержка многопользовательской работы
  - удалённый репозиторий
  - возможность расширения встроенного языка
  - возможность реализации нескольких клиентов
- Реализован прототип
  - простая версия языка (граф, атомарные действия, базовые функции редактирования)
  - сервер
    - репозиторий для хранения диаграмм
    - интерпретатор для преобразования диаграмм
    - откат операций, историй действий
  - CLI клиент (загрузка, редактирование, отображение)
  - GUI клиент (загрузка и отображение)

- Полноценный графический редактор
- Дальнейшая разработка встроенного языка
- Базовые визуальные языки