

*Real/MV*

Реализация кроссплатформенной архитектуры  
CASE-пакета.

Никандров Георгий  
545 гр.

---

---

# *Постановка проблемы*

Для создания каждого редактора надо писать код на языке программирования, отсюда много ошибок и сложностей. UML 2.0 содержит 13 редакторов, иногда нужны еще редакторы – много кода.

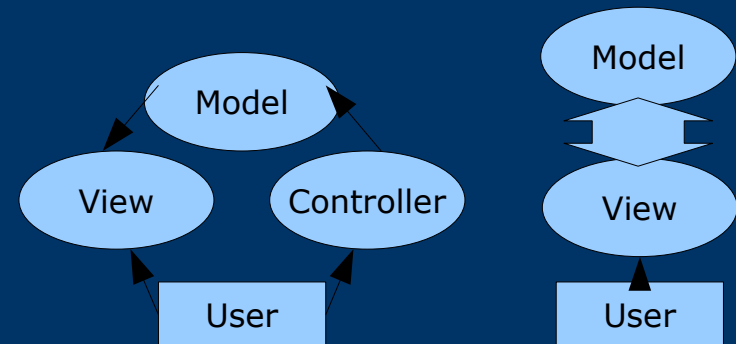
Решение проблемы – реализация методологии генерации новых редакторов по их описанию на простейшем языке разметки.

---

---

# Архитектура

- Использование фреймворка Qt
  - Простота и мощьность
  - Кроссплатформенность
  - Free Software
- Model-View
  - Контроллер и представление объединены
  - Более тесное взаимодействие позволяет упростить разработку
  - Прозрачная интеграция в Qt



# Представление (View)

- В качестве основы для написания графово-графической библиотеки используется Qt Graphics View Framework, предоставляющая набор базовых операций
    - Управление элементами и состоянием
    - Распределение событий
    - Группировка, контейнеры
  - Базовые элементы, составляющие ядро UML, реализованы в коде
- 
-

## Представление (View) - 2

- Для описания графических объектов используется SVG+XHTML, потом транслятором переводится в простейший C++
  - Средний размер кода C++ на 1 элемент – 37 строк.
  - Всё остальное берет на себя либо базовые классы, либо графово-графическая библиотека.
- 
-

# Модель данных (Model)

- Представление графовой структуры в самом репозитории в виде дерева.
- Используется SQL база данных
  - Диаграмма 10000 элементов – 10 секунд на P3-800
  - Можно использовать любые БД – от простейшей локальной SQLite до Oracle.

# Модель данных (Model) - 2

- Ленивое чтение – сначала читается минимум данных, необходимых для отображения, потом подгружается остальное
- Кэширование данных на стороне клиента для ускорения работы

# Структура интерфейса

The screenshot displays the unREAL software interface with several key components and callouts:

- Object Explorer:** A tree view on the left showing the repository hierarchy of objects, including 'система TerStore' and various requirement types like 'reqnFeatured' and 'reqnConceptAlternative'.
- Diagram Explorer:** A tree view below the Object Explorer showing the hierarchy of objects on the diagram, such as 'система TerStore', 'учёт операций', and 'используемая БД'.
- Property Editor:** A table at the bottom left showing properties for the selected object 'система TerStore', including name, description, priority, source, type, and state.
- Diagram:** A central requirements diagram with nodes and relationships. Nodes include 'система TerStore', 'учёт операций', 'используемая БД', 'простые', 'объекты учета', and 'печать в Word'. A dashed line labeled '<< Trace >>' connects 'объекты учета' to 'печать в Word'. A thick black box highlights a 'Diagramma случаев использования: работа с TerStore'.
- Mini Map:** A small overview map on the top right showing the overall structure of the diagram.
- Palette:** A vertical palette on the right side containing various diagram elements and their symbols, such as 'Featured Element', 'Concept Alternative', 'Leaf', 'Parent Node', etc.
- Callouts:** Five blue speech bubbles with white text pointing to specific interface elements:
  - 'Иерархия объектов в репозитории' (Hierarchy of objects in the repository) points to the Object Explorer.
  - 'Иерархия объектов на диаграмме' (Hierarchy of objects on the diagram) points to the Diagram Explorer.
  - 'Навигация' (Navigation) points to the Mini Map.
  - 'Редактор свойств' (Property Editor) points to the Property Editor table.
  - 'Палитра элементов' (Element palette) points to the Palette.



# Результат

- Расширяемая, модульная архитектура с изолированными модулями
    - Новая графово-графическая библиотека
    - Репозиторий с использованием СУБД
    - Генератор кода редакторов по их описаниям на XML
  - Подход апробирован на наборе редакторов UML 2.0
- 
-