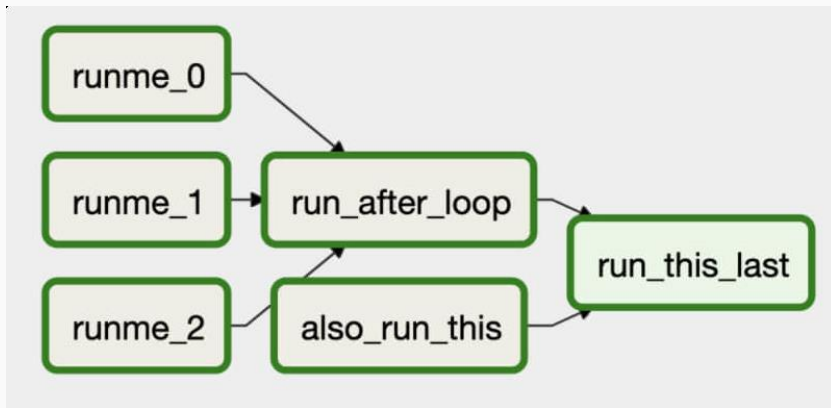


# Редактор графов рабочих процессов для Airflow

Пяйве Олег

# Актуальность

- Автоматизация помогает выполнять рутинные задачи быстро и безошибочно
- На практике в компании Huawei для автоматизации применяется Airflow
- Граф зависимостей между составными частями процесса в Airflow описывается с помощью кода на Python.



# Существующие решения

Luigi и Oozie — менее популярные аналоги Airflow: аналогично ему, нет графического интерфейса, план выполнения процесса создается с помощью кода.

Azkaban — имеет графический интерфейс, но не позволяет работать со сложными процессами.

Использование графического интерфейса удобнее, чем написание кода. Однако, аналогов Airflow по функциональности с графическим интерфейсом нет.

# Цель и задачи

Цель — создание сервиса, позволяющего в графическом интерфейсе создавать графы рабочих процессов и конвертировать их для использования с Airflow.

Задачи:

- Провести обзор предметной области и изучить существующие аналоги
- Разработать архитектуру приложения и интерфейс для взаимодействия клиентской и серверной части (API), а также нотацию для записи графов
- Разработать клиентскую часть приложения
- Разработать серверную часть приложения
- Протестировать полученный продукт на реальных пользователях.

# Серверная часть

Flask — создание веб-приложений

SQLAlchemy — работа с базой данных

Jinja — генерация кода по шаблону

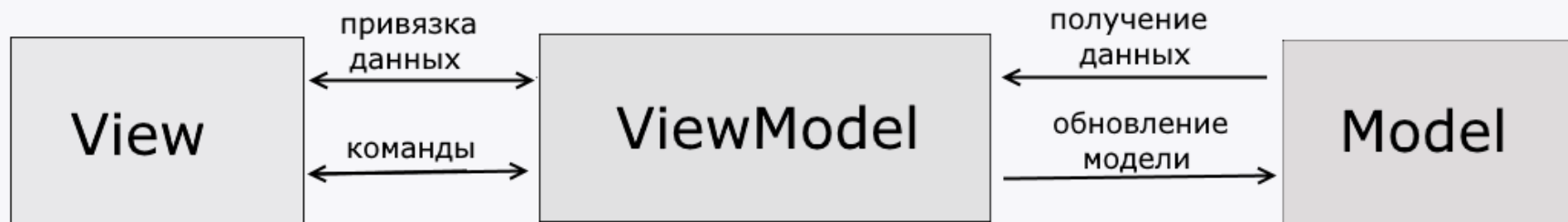
# Клиентская часть

MXGraph — библиотека для работы с графами: добавление и редактирование операторов и связей между ними

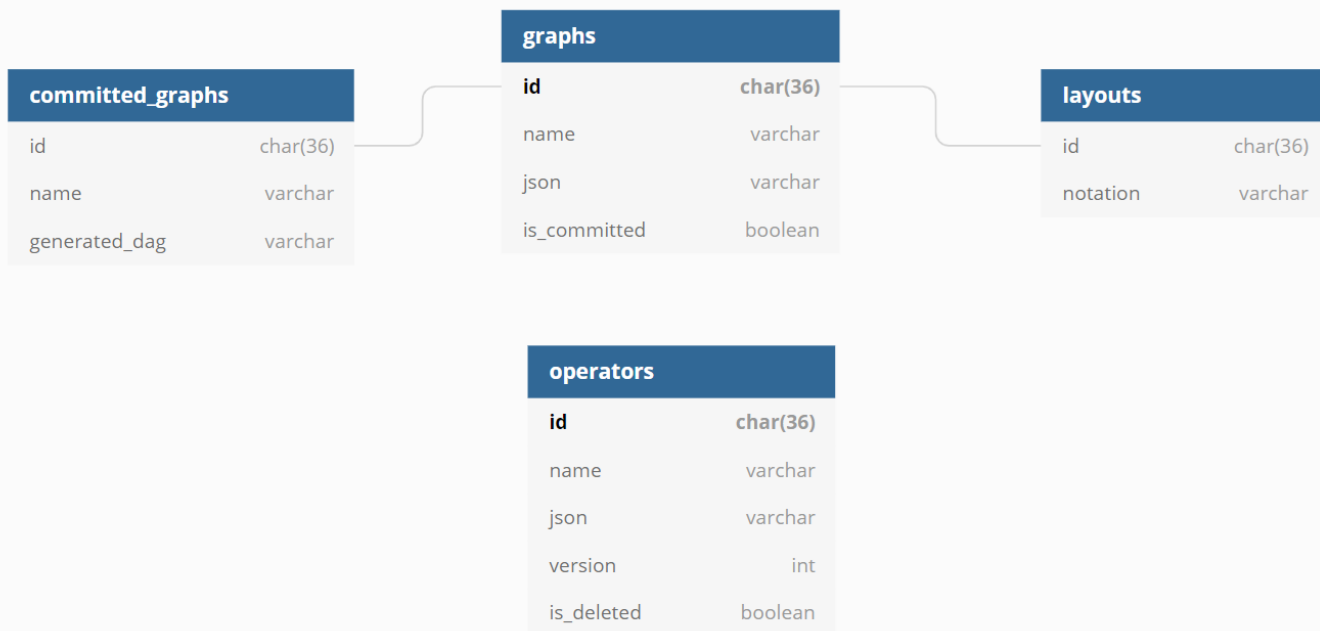
KnockoutJS — инструмент для создания авто-обновляемых списков

# KnockoutJS

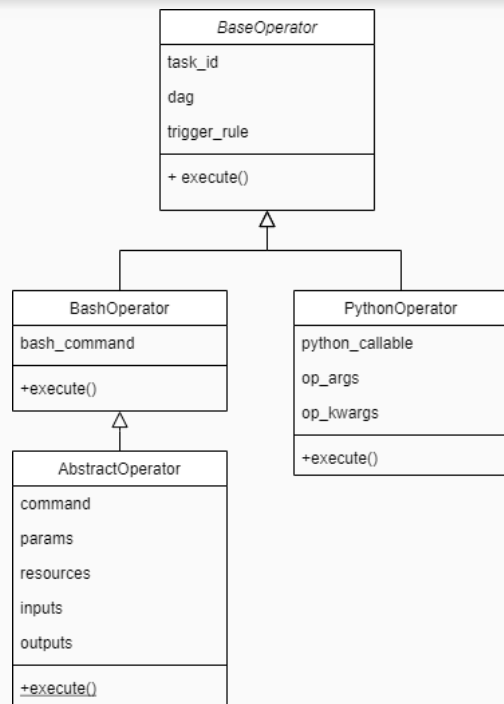
- Используется паттерн Model-View-ViewModel
- При изменении ObservableArray KnockoutJS меняет соответствующие HTML-элементы страницы



# Сохранение графов в базе и валидация



# AbstractOperator: архитектура



Перед выполнением кода AbstractOperator скачивает все входные файлы на ту машину, на которой он выполняется. Затем выполняет команду и выгружает выходные файлы в общее хранилище.

# Основная страница



# Добавление оператора

Graph editor

Toloka

AMT

Search\_eng

Relevance\_

Formula\_ca

Evaluation

Collect\_ol

Inputs Add Remove

|      |     |
|------|-----|
| name | txt |
|------|-----|

Outputs Add Remove

|                     |      |
|---------------------|------|
| features            | json |
| model_config        | json |
| fixed_model_config  | json |
| production_features | tsv  |
| extractor           | zip  |

Parameters Add Remove

|             |     |           |
|-------------|-----|-----------|
| pipeline_id | str | test_oleg |
| markup_path | str | /apps     |
| language    | str | ru        |
| country     | str | ru        |

Resources Add Remove

Path should start with "/" and be without prefix "hdfs://". For example:  
"/user/oleg/database.csv"

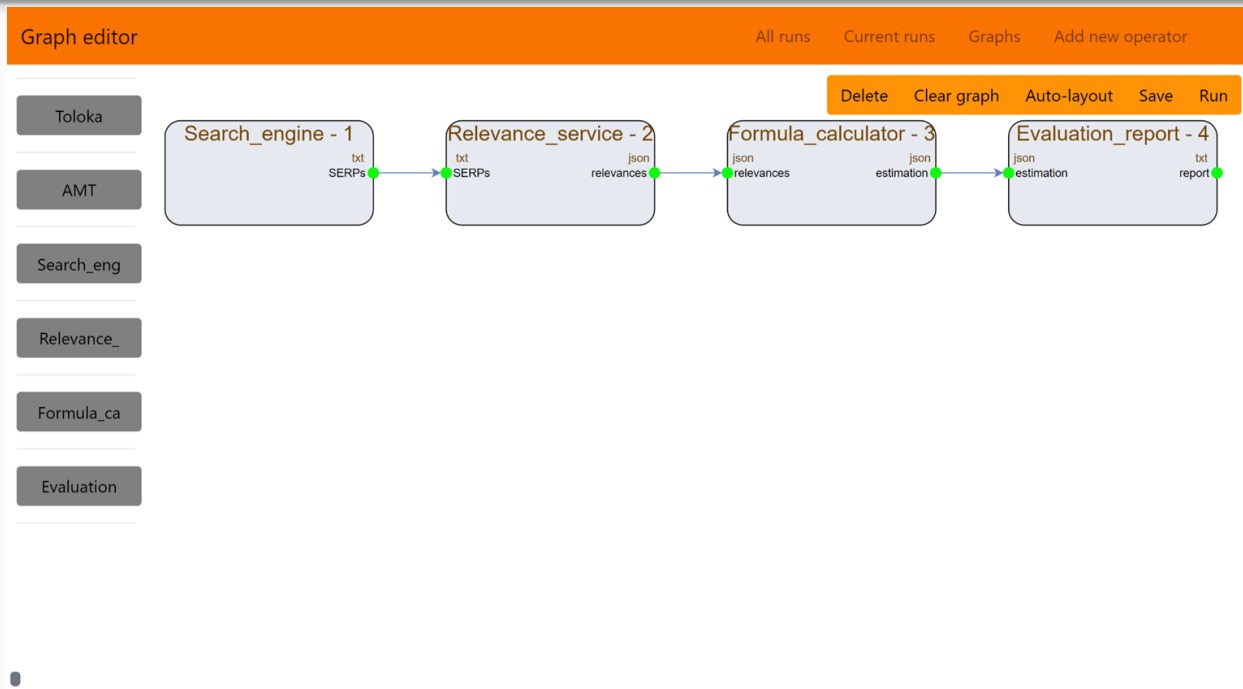
|                         |                               |
|-------------------------|-------------------------------|
| collect_old_features.py | /user/admin/collect_old_featu |
|-------------------------|-------------------------------|

Close Add operator

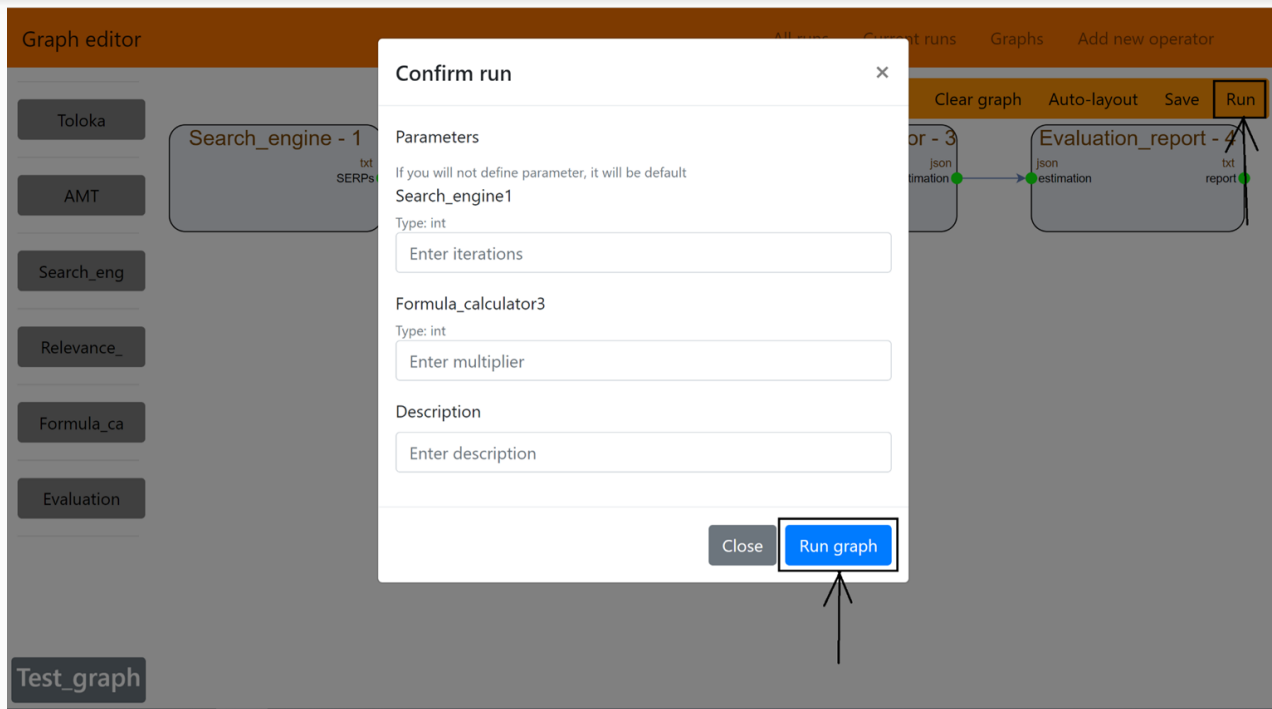
it runs Graphs Add new operator

Clear graph Auto-layout Save Run

# Создание графа



# Сохранение и запуск графа



# Список запусков

Graph editor

Toloka

AMT

Search\_eng

Relevance\_

Formula\_ca

Evaluation

Test\_graph

to-layout Save Run

valuation\_report - 4  
txt report

List of all runs of DAGs

Name Airflow page

Test\_graph Airflow page

| Name       | Start date                       | Status  |
|------------|----------------------------------|---------|
| Test_graph | 2020-09-09T15:40:10.223964+00:00 | success |

Close

# Тестирование

Юнит-тестирование валидации графа.

End-to-end тестирование реальными пользователями:

- Добавление операторов и работа с ними: редактирование и удаление
- Составление графа из операторов
- Сохранение и редактирование сохранённого графа
- Запуск и просмотр статуса выполнения

# Итог

- Проведен обзор предметной области
- Разработана архитектура приложения и интерфейс для взаимодействия клиентской и серверной части (API), а также нотация для записи графов
- Разработана клиентская часть приложения
- Разработана серверная часть приложения
- Продукт протестирован на реальных пользователях и используется для автоматизации