

# Диаграммный исполнитель

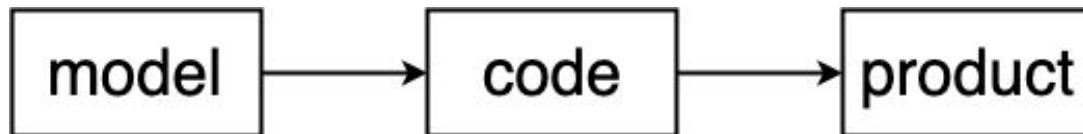
Артемий Безгузиков, 646

Научный руководитель: к.т.н., доц. Т.А. Брыксин

Рецензент: А.Н. Барташев

# DSM подход

**DSM** (Domain Specific Modeling) – предметно-ориентированное моделирование

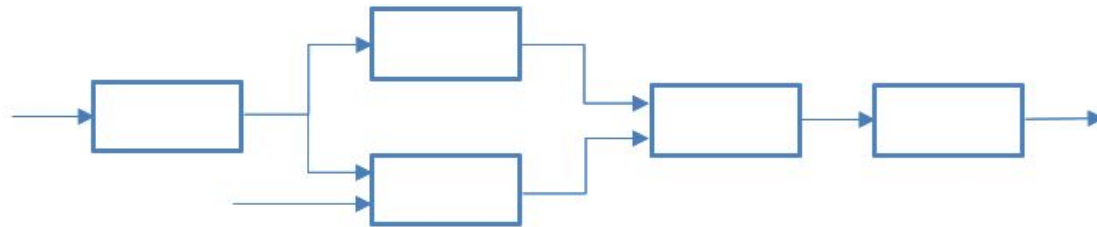


**DSM-решение** – инструменты для разработки на созданном графическом языке

**DSM-платформа** – система для создания DSM-решений

# Паттерн “Pipes and Filters”

Решение представимо в виде нескольких независимых шагов, выполняемых отдельными обработчиками



Пример: командная оболочка UNIX

В основе парадигмы **Dataflow**-программирования:  
программа есть граф вычислений

# Цели работы

*Цель: создание DSM-платформы для диаграммных языков, программы которых реализуют паттерн “Каналы и фильтры” на языке Python*

Были поставлены следующие задачи

1. Спроектировать основные компоненты DSM-платформы
2. Разработать язык метамоделирования
3. Реализовать описанную DSM-платформу
4. Провести апробацию работы

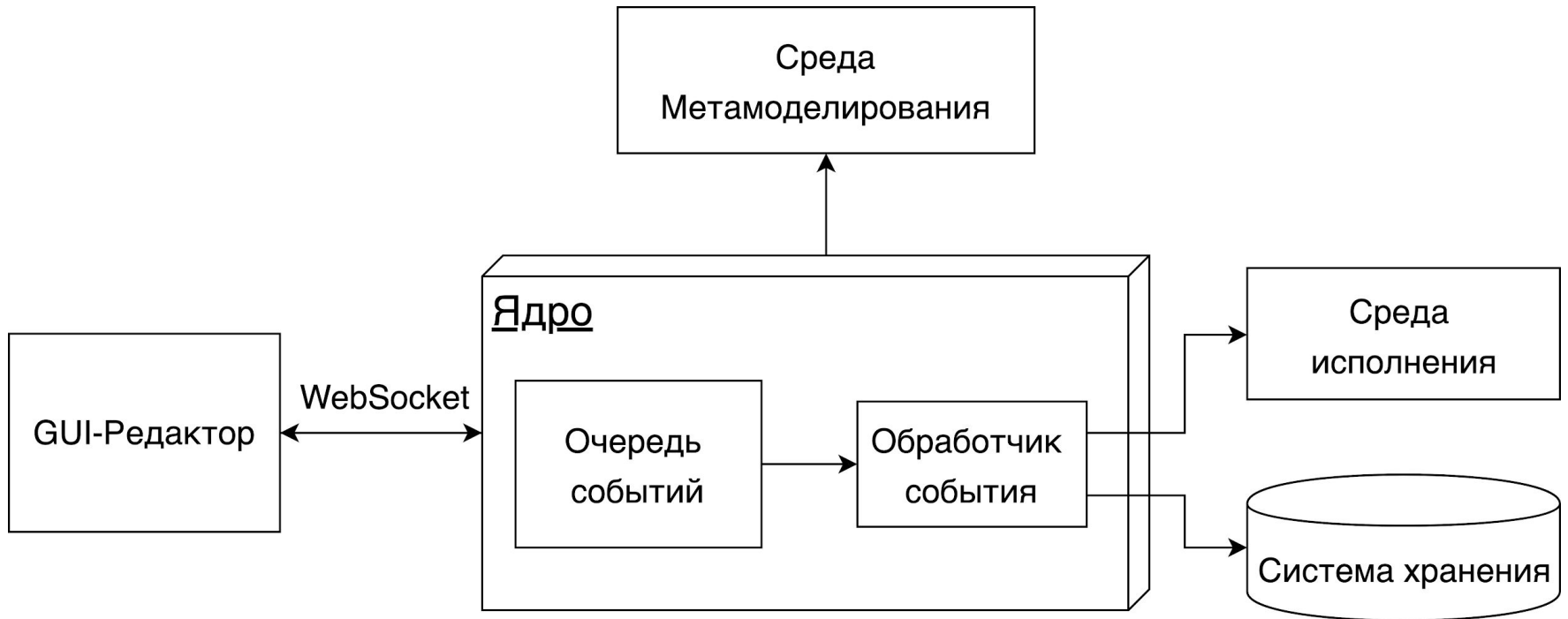
# Критерии для DSM-платформы

1. Динамическое расширение языка
2. Поддержка функций языка Python
3. Наличие интегрированной среды исполнения
4. Переиспользование вычислений
5. Консольный вывод и ошибки фактически исполняемого исходного кода
6. Вывод графиков и произвольных изображений
7. Работа в браузере

# Существующие DSM-платформы

- **MetaEdit+**. Отсутствие среды исполнения
- **Microsoft Modeling SDK**. Для Windows, код на C#, неотделим от Visual Studio
- **Eclipse Modeling Project**. Сложность вхождения, отсутствие решений для Python
- **QReal**. Сложность встраивания, отсутствие важных графических компонент, приложение для ПК

# Архитектура



Языки программирования: Kotlin, KotlinJs, Python

Протоколы взаимодействия: REST, WebSocket, GRPC

# GUI-редактор

Diagram executor New Save Save As Open

## Configurer

Cached Run Remove  Full

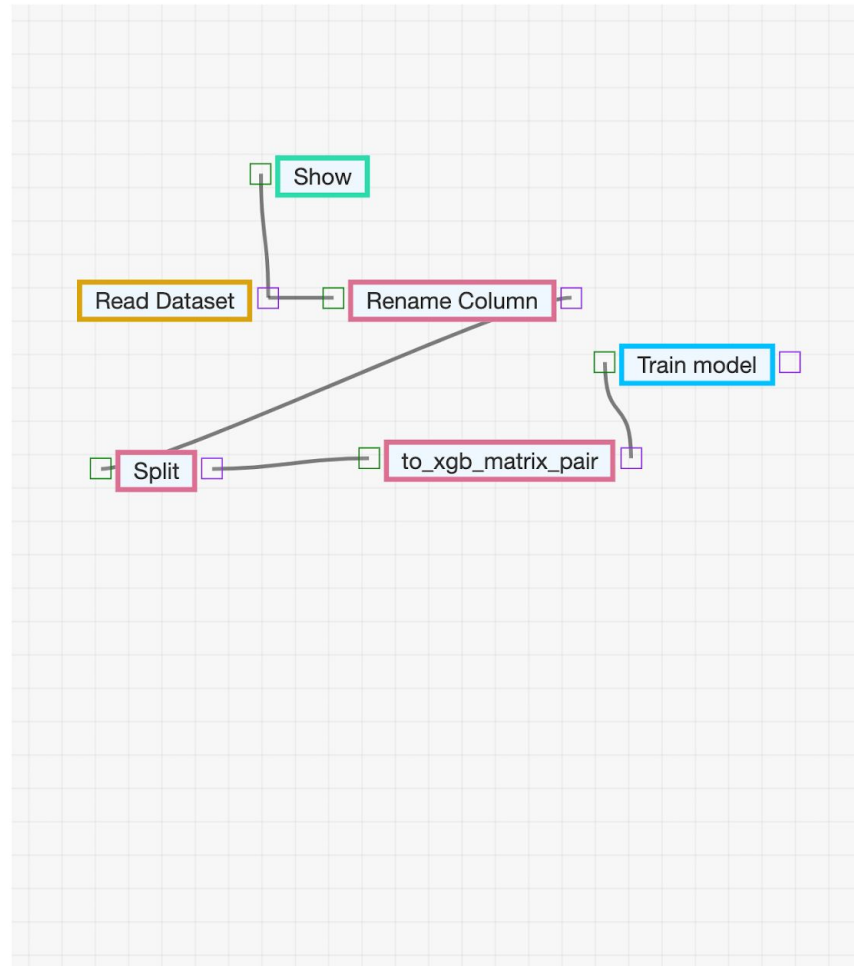
**Function:** model(pair, params)

**Signature:** (XGBM\_Pair) -> XG\_Boost

**Params:** max\_depth (int) 0  
num\_round (int) 0

**Description:** Train XGBoost models.

**Code:** 



Pure Render Resource File System

## Model

Search

### XgBoost

model : (XGBM\_Pair) -> XG\_Boost

xgboost\_train : (XGBoost\_Matrix) -> XG\_Boost

validate : (XGBoost\_Matrix, DataFrame) -> XG\_Boost

### Sklearn

randomForest : (DataFrame) -> Model

## Data

Search

### DataFrame

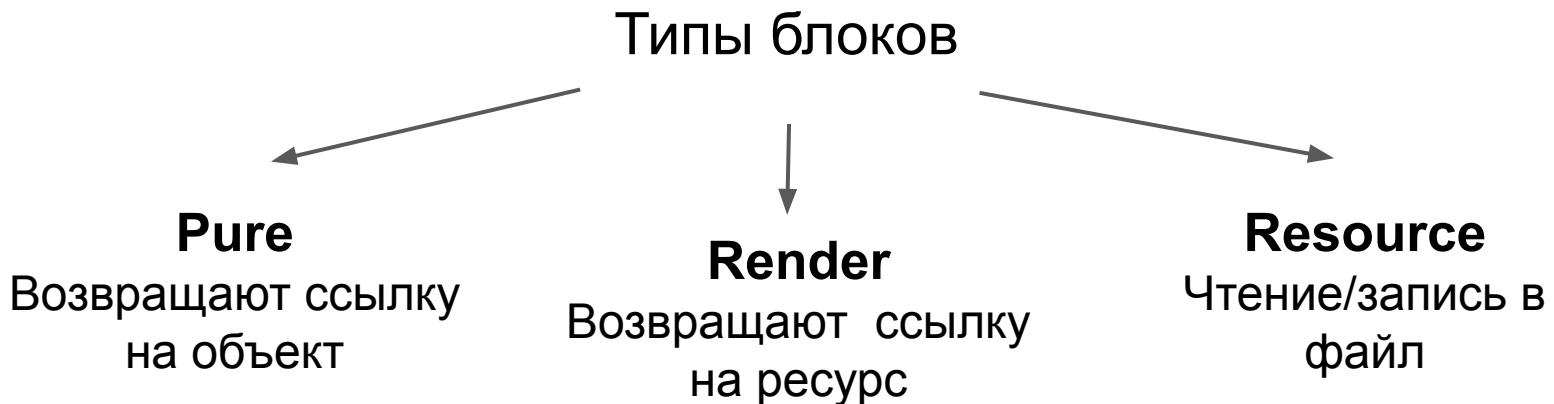
rename\_column : (DataFrame) -> DataFrame



# Среда метамоделирования

Git-репозиторий с иерархией папок. В них находятся файлы-дескрипторы. Содержат функции на языке Python и конструкции метаязыка в комментариях

Доступно: *имя блока, сигнатура, описание, параметры*



# Генерация UI компоненты

```
# function=CalculateData
# signature=(Data, Data)->Data
# param@columns:string
# param@value:int
# param@coefficient:float
def simple_function(x, y, params):
    ...
    return x
```

## Configurer

---

**Function:** simple\_function(df, params)

**Signature:** (Data,Data) -> Data

**Params:** columns: string  
value : int  
coefficient: float

**Code:**



**Drag to Scene:**

CalculateData

# Среда исполнения

Сервер, написанный на языке Python

Поддерживает следующие команды

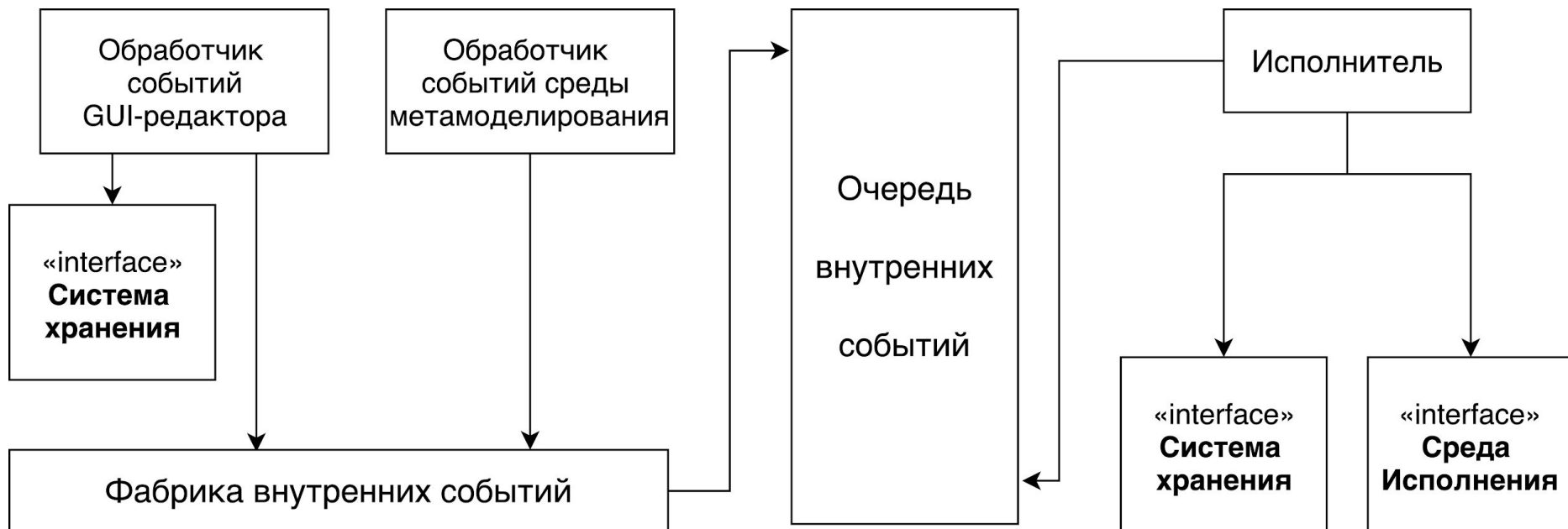
1. Обновить функции некоторого дескриптора
2. Выполнить некоторую функцию
3. Удалить значение по ссылке

# Система хранения

Состоит из трех модулей

- **Репозиторий диаграмм.** CRUD операции, MongoDB
- **Репозиторий пользовательский файлов.** Чтение, сохранение и удаление произвольных файлов. Используется блоками типа Resource
- **Хранилище временных файлов.** Временные картинки и графики для отображения пользователю. Используется блоками типа Render

# Схема работы



# Исполнение диаграммы

Status: Completed

Output (cached):

Signature: (DataFrame) -> DataFrame

Node: Drop factors

Status: Completed

Output (cached):

Signature:(DataFrame) -> DataFramePair

Node: Split

Status: Rejected

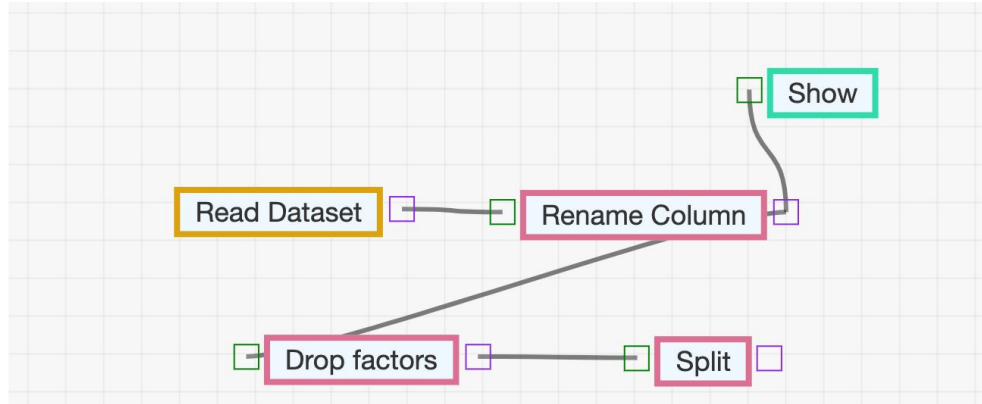
Output:

Signature: (DataFrame) -> html

Node: Show

Status: Completed

Output (cached):



Output

	Unnamed: 0	group	feature_0	feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7
0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	4	2	True	False	False	False	NaN	False	False	False
5	5	2	True	False	False	False	NaN	False	False	False
6	6	2	True	False	NaN	False	NaN	NaN	NaN	NaN
7	7	3	True	False	False	False	NaN	False	False	False
8	8	3	True	False	NaN	False	NaN	NaN	NaN	NaN
9	9	4	True	False	False	False	NaN	False	False	False
10	10	4	True	False	NaN	False	NaN	NaN	NaN	NaN
11	11	5	True	False	False	False	NaN	True	False	False
12	12	5	True	False	False	False	NaN	True	False	False
13	13	5	True	False	False	False	NaN	True	False	False

# Апробация. ML

*Задача: получить классификатор, который по некоторым признакам объекта предсказывает, к какому классу он относится*

Конкуренты: Microsoft ML Studio, Machine Flow

Преимущества полученного решения:

1. Добавление пользовательских функций с **произвольными** аргументами
2. Перенаправление на GUI-редактор **консольного вывода**

# Апробация. ML

Status: Completed

Output: 

Signature: (DataFrame) -> DataFrame

Node: **Fill gaps**

Status: Completed

Output: 

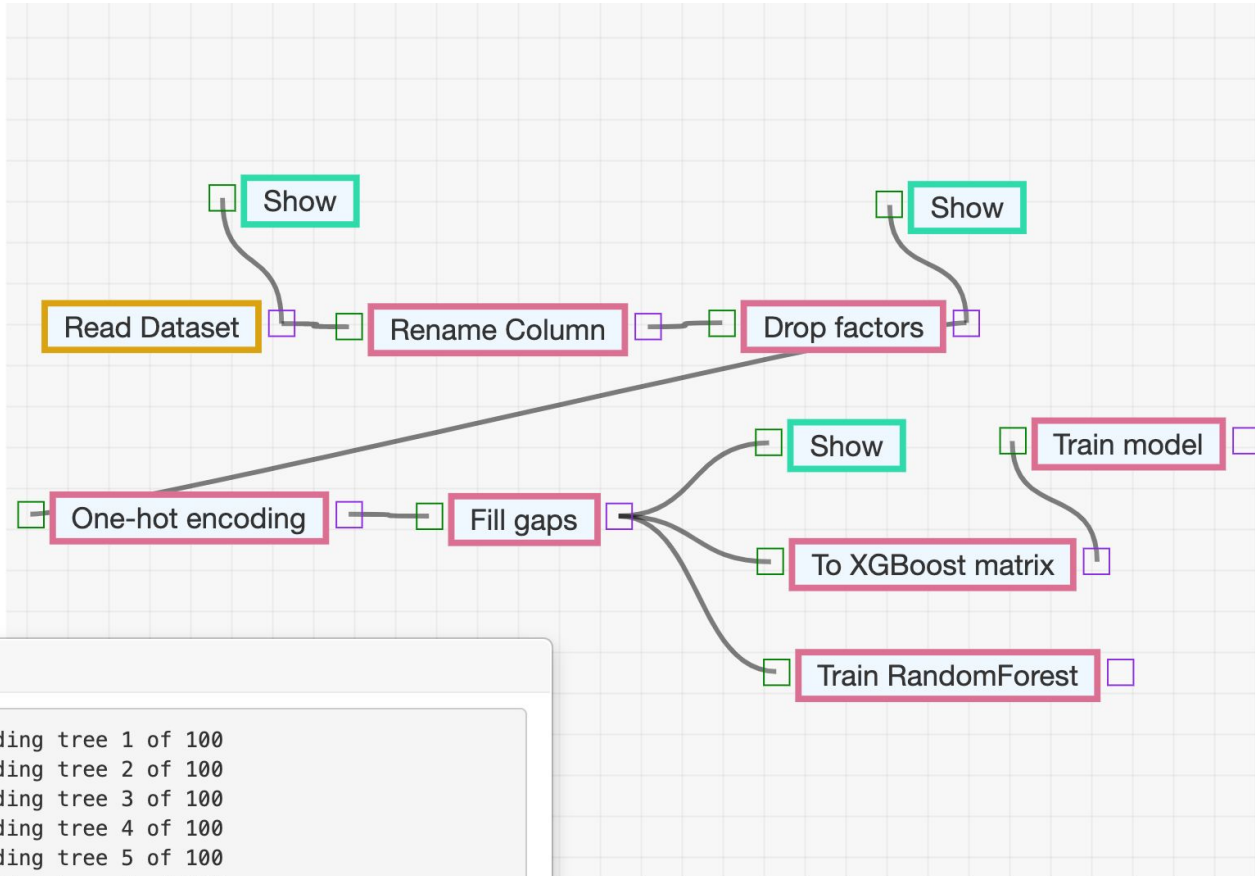
Signature: (DataFrame) -> Model

Node: **Train RandomForest**

Status: Completed

Output: 

Ok



Output

```
building tree 1 of 100  
building tree 2 of 100  
building tree 3 of 100  
building tree 4 of 100  
building tree 5 of 100  
building tree 6 of 100  
building tree 7 of 100  
building tree 8 of 100  
building tree 9 of 100  
building tree 10 of 100  
building tree 11 of 100
```



# Апробация. Томография

Исследование: “Технология восстановления особых областей на основе данных акустической томографии”

Автор: Гонта К.А.

Цель работы: разработка и реализация эффективного алгоритма для выявления патологий организма человека с помощью ультразвука

Требование: *показать результаты проводимых экспериментов и процесс их получения*

# Апробация. Томография

Status: Completed

Output: 

Signature: (SensorImg) -> png

Node: **show\_sensor\_img**

Status: Completed

Output: 

Signature: (SensorData, WaterData, SensorImg) -> Img

Node: **image\_reconstruction**

Status: Completed

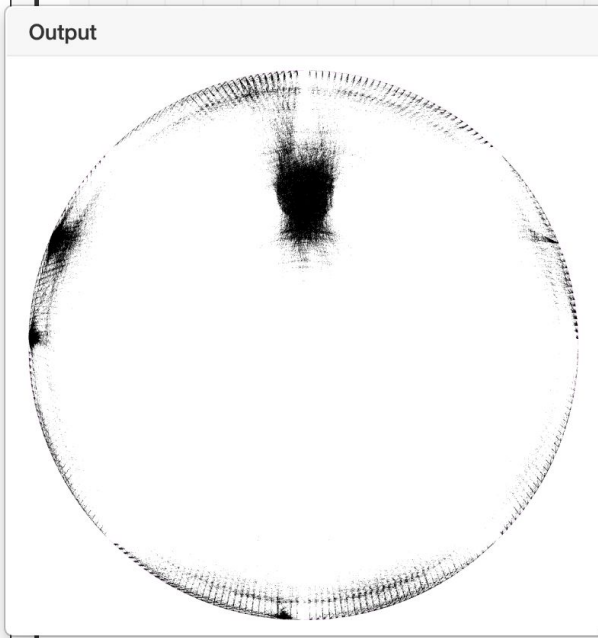
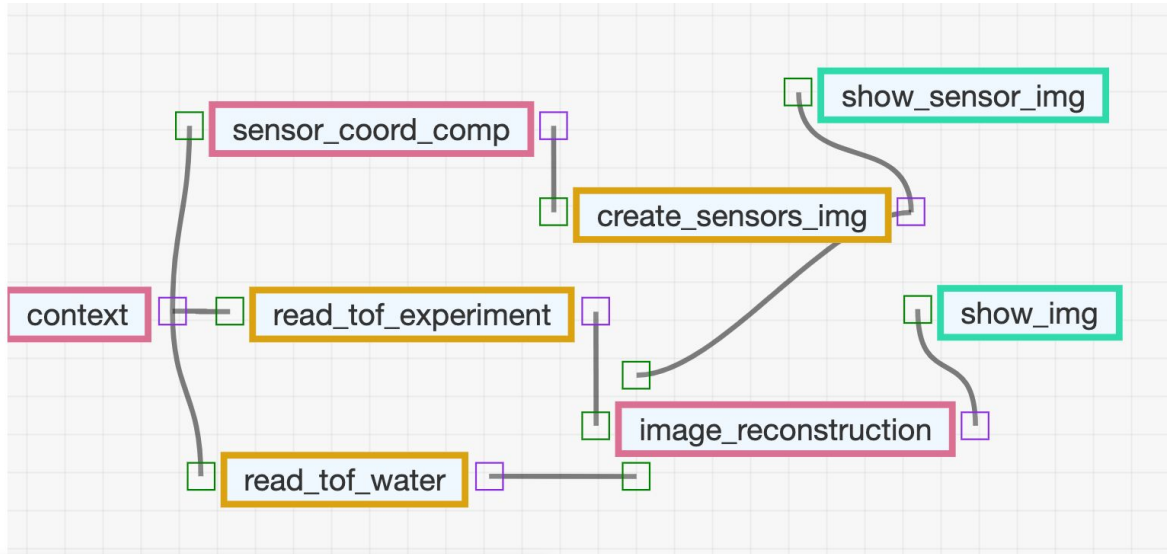
Output: 

Signature: (Img) -> png

Node: **show\_img**

Status: Completed

Output: 



# Результаты

1. Разработана архитектура решения, включающая среду метамоделирования, среду исполнения, GUI-редактор, систему хранения данных и связующее ядро
2. Созданы средства метамоделирования
3. Реализована DSM-платформа, позволяющая исполнять и динамически расширять созданные ею языки непосредственно в процессе исполнения программы
4. Произведена апробация работы с применением разработанной DSM-платформы
  - a. Для решения задачи классификации
  - b. Для томографического исследования
5. Работа представлена на конференции “СПИСОК-2019”