

Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

# **Интерпретация диаграмм в онлайн-среде программирования роботов**

Курсовая работа студентки 371 группы

Гагиной Лады Владиславовны

Научный руководитель: ст. преп. Брыксин Тимофей Александрович

Санкт-Петербург

2015

# Оглавление

[Введение](#)

[Постановка задачи](#)

[Используемые инструменты](#)

[TypeScript](#)

[jQuery](#)

[JointJS](#)

[RaphaelJS](#)

[Разработка](#)

[Архитектура проекта](#)

[Архитектура интерпретатора](#)

[Алгоритм интерпретации](#)

[Разбор выражений](#)

[Модель интерпретации 2D-представления](#)

[Результаты](#)

[Список литературы](#)

## 1. Введение

Ни для кого не секрет, что в настоящее время технологии развиваются очень быстро и занимают всё большее место в повседневной жизни. В связи с этим, возрастает необходимость владения навыками алгоритмического мышления и программирования, которые стоит начинать развивать с детского возраста. Однако объекты, которыми оперирует программирование, абстрактны, что затрудняет их изучение. Тут на помощь приходит робототехника. Программируя роботов, дети сразу видят результаты своей работы. Кроме того, наглядность успехов мотивирует заниматься программированием и дальше. Таким образом, очевидна необходимость в робототехнических конструкторах, примером которого является ТРИК.

На кафедре системного программирования СПбГУ была создана визуальная среда программирования роботов TRIK Studio. Однако работа с ней подразумевает установку и настройку специального программного обеспечения, что может создать трудности. Решением может стать онлайн-среда визуального программирования роботов, которая снимет необходимость действий непосредственно перед программированием, кроме того, она может также содержать магазин готовых приложений для роботов (аналог App Store<sup>1</sup> для роботов).

Целью проекта QReal:Web стало создание онлайн-сервиса, реализующего основные функции TRIK Studio:

- создание диаграмм,
- интерпретация диаграмм для реальных моделей роботов и 2D-модели,
- генерация кода по диаграмме,
- управление роботами онлайн.

---

<sup>1</sup> <https://itunes.apple.com/us/genre/ios/id36?mt=8>

## 2. Постановка задачи

Целью моей курсовой работы в проекте является реализация онлайн-интерпретатора диаграмм для 2D-модели робота.

Для достижения цели сформулированы следующие задачи:

1. Реализовать обход диаграммы, построенной в редакторе.
2. Реализовать возможность разбора выражений Lua-подобного языка, используемых в блоках “Функция” и блоках “Условие”.
3. Создать механизм исполнения для 2D-модели робота.

### 3. Используемые инструменты

Проект, в рамках которого я делала курсовую работу - командный, часть функционала была реализована до того, как я присоединилась к команде, поэтому некоторые инструменты разработки были определены изначально.

Для разработки мной были использованы: язык TypeScript<sup>2</sup>, библиотеки jQuery<sup>3</sup>, JointJS<sup>4</sup> и RaphaelJS<sup>5</sup>

#### 3.1. TypeScript

TypeScript — язык программирования, разработанный компанией Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. В частности, язык даёт возможность использовать статическую типизацию и, как следствие, избежать допущения части ошибок, а также в нём есть поддержка ООП, модульности - что позволяет создавать гибкие и расширяемые приложения.

Кроме того, TypeScript позволяет использовать библиотеки на JavaScript в проекте, что существенно расширяет его возможности.

#### 3.2. jQuery

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML.

Функции библиотеки:

- получение доступа к элементам DOM;
- обращение к атрибутам и содержимому элементов DOM;
- управление элементами DOM.

---

<sup>2</sup> <http://www.typescriptlang.org>

<sup>3</sup> <http://jquery.com>

<sup>4</sup> <http://jointjs.com>

<sup>5</sup> [http://raphaeljs.com](http://dmitbaranov.com)

### 3.3. JointJS

Библиотека JointJS была выбрана в ходе исследования моим коллегой, оказавшись лучшей среди аналогов: одновременно бесплатной, хорошо документированной, простой в использовании и предоставляющей поддержку всего необходимого функционала.

JointJS позволяет:

- создавать диаграммы;
- манипулировать элементами диаграмм на сцене;
- получать доступ к элементам диаграмм.

### 3.4. RaphaelJS

RaphaelJS - библиотека, написанная на JavaScript, позволяющая использовать векторную графику в браузере и манипулировать векторными объектами.

## 4. Разработка

### 4.1. Архитектура проекта

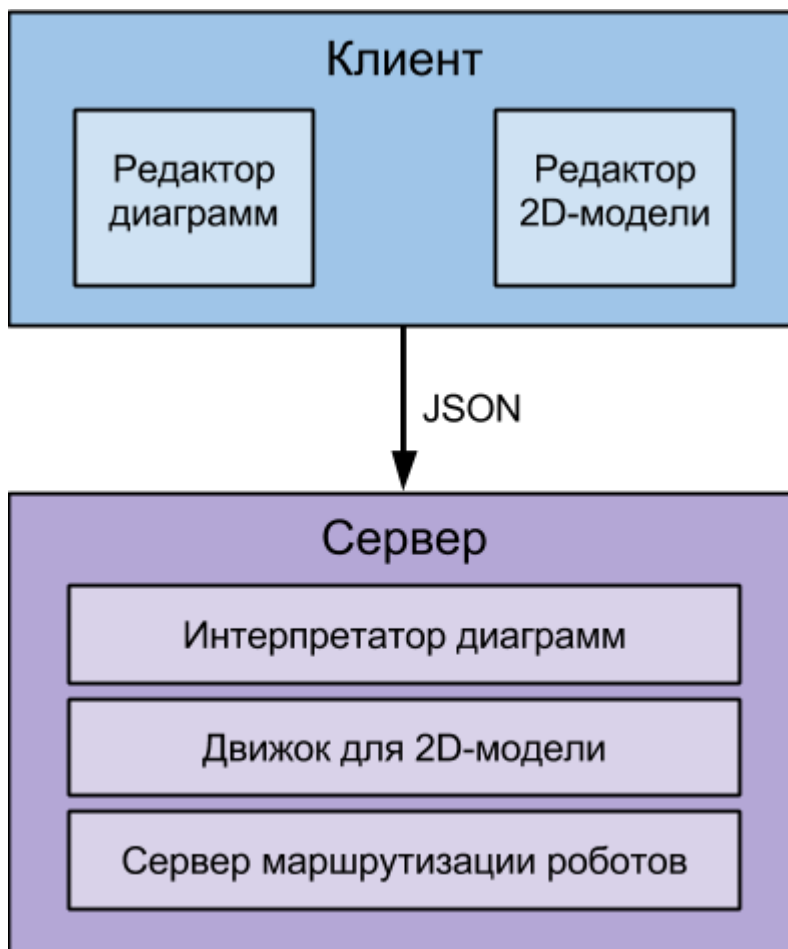


Рис. 1 Взаимодействие компонент сервиса

Пользователь в браузере составляет диаграмму поведения робота, после чего переключается на вкладку 2D-модели, где нажимает на кнопку “Play”. Текущее состояние диаграммы в редакторе в формате JSON<sup>5</sup> отправляется на сервер, где поступает в обработку интерпретатору диаграммы.

<sup>5</sup> JavaScript Object Notation, <http://json.org/>

При появлении в диаграмме блоков, отвечающих за части робота (моторы и сенсоры), интерпретатор вызывает движок, передавая ему необходимые значения, которые преобразуются в направление для 2D-модели. 2D-модель робота, используя библиотеку RaphaelJS, вызывает перемещение робота в редакторе 2D-модели клиента.

Редакторы диаграмм и 2D-модели описаны в рамках дипломной работы Захарова Владимира 444 группы. Сервер маршрутизации роботов описан в рамках дипломной работы Агеева Дениса 444 группы.

## 4.2. Архитектура интерпретатора

При интерпретации используется паттерн “Абстрактная фабрика”. Каждому узлу на сцене соответствует блок из множества, реализующий выполнение каждой конкретной задачи. При обработке узлы диаграммы передаются фабрике, которая в зависимости от значения поля “Тип” узла (Рис. 2) вызывает интерпретацию нужного блока.

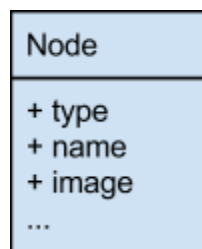


Рис. 2 Узлы диаграммы

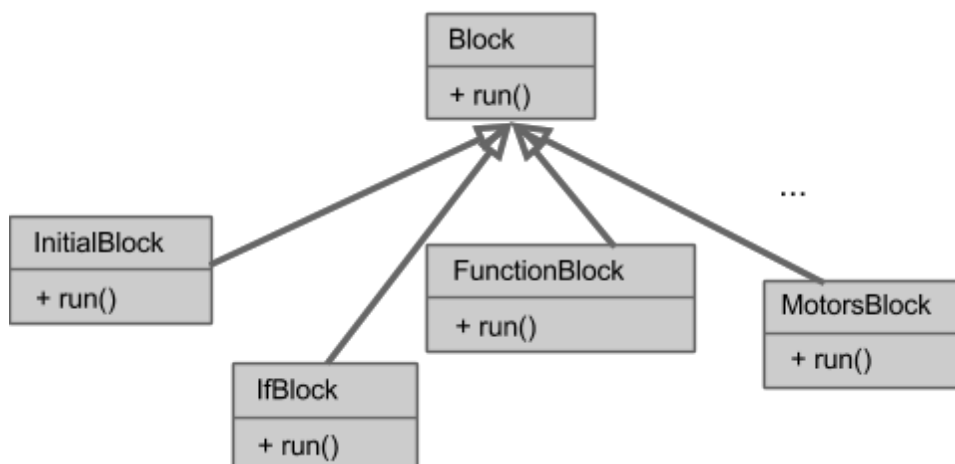


Рис. 3 Используемые блоки



### 4.3. Алгоритм интерпретации

1. Нахождение начального узла диаграммы (библиотека JointJS не даёт возможности как-то выделить начальный узел диаграммы, поэтому его поиск осуществляется с использованием поля “Тип” узла).
2. Вызов интерпретации нужного блока с помощью фабрики.
3. Нахождение выходящих связей и узлов по ним средствами библиотеки JointJS.
4. Проверка корректности данной части диаграммы (прим. имеет ли блок “Условие” ровно две выходящие связи, помеченные как “Истина” и “Ложь”) и сообщение об ошибках.
5. Вычисление выражений, если они есть и выбор следующего узла для интерпретации.
6. См. пункт 2.

### 4.4. Разбор выражений

Для интерпретации и выбора следующего узла необходим разбор выражений. В TRIK Studio используется Lua<sup>7</sup>-подобный язык для выражений в блоках “Условие” и присваиваний в переменные в блоках “Функция”.

Для их разбора и вычислений был реализован парсер, использующий метод рекурсивного спуска.

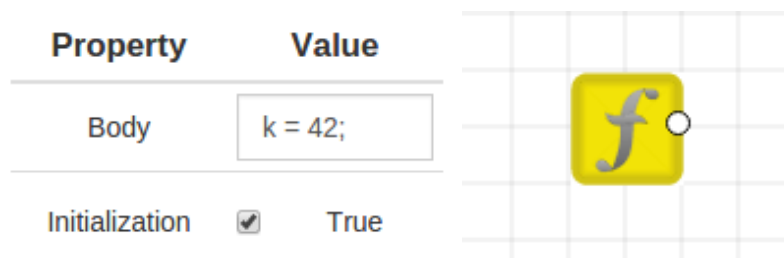


Рис. 4 Пример выражения

<sup>6</sup> <http://www.lua.org/about.html>

#### 4.5. Модель интерпретации 2D-представления

Для наглядного представления результатов интерпретации был реализован движок для 2D-модели робота (Рис. 5).

Для регулирования скорости и централизованного управления движением на сцене используется класс `Timeline`, который выдаёт событие каждый раз по прошествии некоторого промежутка времени (прим. 10мс) вызывает класс `Engine` для пересчета координат и направления робота. Когда пересчет произошёл `speedFactor` раз (прим. 1000мс / 28 кадров в секунду), вызывается модель робота для отрисовки нового положения робота.

Для изменения положения робота использованы средства библиотеки `RaphaelJS`.

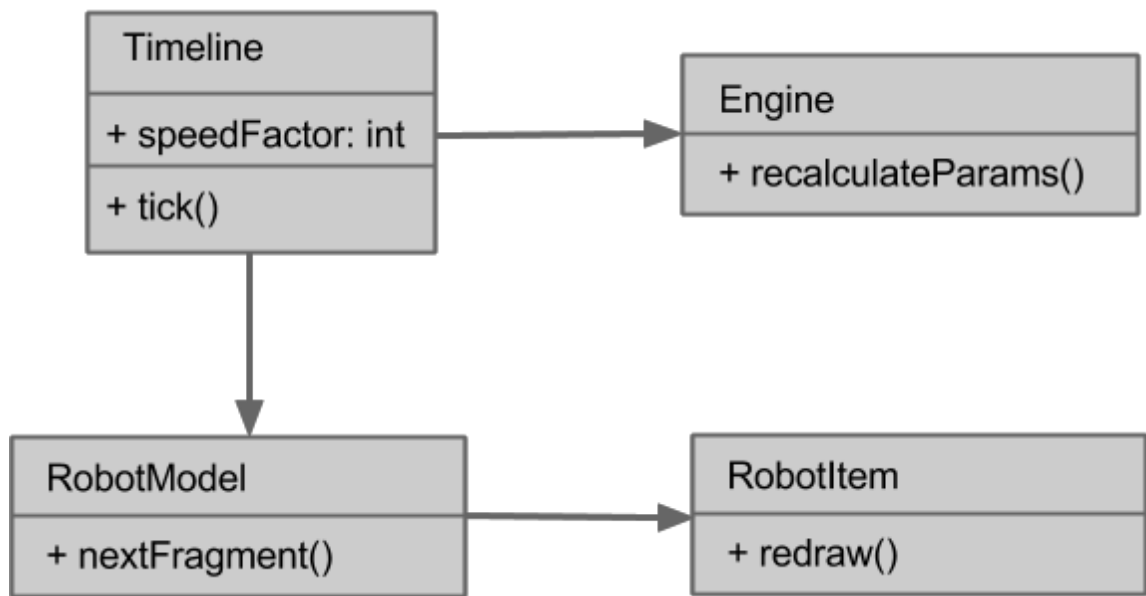


Рис. 5 Схема модели интерпретации 2D-представления

## 7. Результаты

В качестве итога проделанной работы были получены следующие результаты:

1. Реализован обход диаграммы.
2. Реализован разбор выражений внутри блоков “Условие” и “Функция”.
3. Реализован механизм интерпретации для 2D-модели.

## 8. Список литературы

[1] Eloquent JavaScript: A Modern Introduction to Programming — Marijn Haverbek, 2014

[2] Приемы объектно-ориентированного программирования. Паттерны проектирования — Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 2010

[3] JointJS, URL: <http://jointjs.com>

[4] RaphaelJS, URL: <http://dmitrybaranovskiy.github.io/raphael/>

[5] TypeScript, URL: <http://www.typescriptlang.org>