

Санкт-Петербургский государственный университет

Программная инженерия

Белков Роман Владимирович

Интеграция программирования на языке Python в образовательные решения TRIK

Выпускная квалификационная работа

Научный руководитель:
ст. преп. Я. А. Кириленко

Рецензент:
ООО "ИнтеллиДжей Лабс"
программист Д. А. Мордвинов

Санкт-Петербург
2017

SAINT PETERSBURG UNIVERSITY

Software engineering

Roman Belkov

Integration of Python programming into TRIK educational solutions

Graduation Thesis

Scientific supervisor:
senior lecturer I. A. Kirilenko

Reviewer:
IntelliJ Labs Co. Ltd
Software Engineer D. A. Mordvinov

Saint Petersburg
2017

Оглавление

Введение	4
Постановка задачи	6
1. Обзор технологий	7
1.1. TRIK	7
1.1.1. trikRuntime	8
1.1.2. TRIK Studio	9
1.1.3. TRIK Stepik checker	11
1.2. Фреймворки взаимодействия Python и C++	11
1.2.1. PyQt	12
1.2.2. PySide	12
1.2.3. PythonQt	13
1.2.4. Boost.Python	13
1.2.5. SWIG	14
2. Разработка требований	15
3. Архитектура	17
3.1. Робот TRIK	17
3.2. TRIK Studio	18
3.3. TRIK Stepik Checker	18
4. Апробация	19
5. Документация	20
Заключение	21
Список литературы	22

Введение

По своим вычислительным ресурсам робототехнические контроллеры, доступные широкому кругу пользователей, приближаются к показателям персональных компьютеров десятилетней давности. Такая тенденция позволяет постепенно применять в разработке современные методологии и технологии наравне с классическими для контроллеров низкоуровневыми языками и технологиями. Поскольку робототехника активно используется для STEM [5, 4] образования, внедрение популярных технологий, используемых в промышленном программировании, позволит методистам разрабатывать программы обучения, направленные на более широкий круг пользователей. Одной из популярной технологий, получившей широкое распространение в образовательной сфере в последнее время, является язык Python.

Python на данный момент является четвертым по популярности языком согласно индексу языков программирования TIOBE [2] на май 2017 года и вторым по популярности языком после Java согласно списку PyPL [6] на май 2017 года. В последние несколько лет Python стал активно внедряться в образовательные программы. Например, в Массачусетском технологическом институте, одном из ведущих [12, 13] университетов в области инженерного и технического образования, студентам первого года обучения читают курс "Introduction to Electrical Engineering and Computer Science I" [14], который представляет собой программирование роботов на языке Python. Другие зарубежные университеты тоже достаточно быстро перешли на использование Python в вводных курсах по программированию [7], тем самым обеспечив Python первенство среди языков программирования, используемых в университетах США.

За трендом, установленным университетами, практически сразу же последовали школы [11, 9], переходя на Python и внедряя новые курсы обучения программированию на языке Python. В России Python – один из 5 предложенных в ЕГЭ языков программирования на протяжении уже нескольких лет.

TRIK¹ – это многоцелевой кибернетический контроллер и одноимённый металлический конструктив для прототипирования роботов. Одним из многих применений контроллера является обучение программированию студентов и школьников. Данный контроллер примечателен тем, что обладает достаточными вычислительными мощностями для решения сложных робототехнических задач и реализации ресурсоёмких алгоритмов, а также отсутствием необходимости навыков пайки и знания электротехники. Для контроллера TRIK существует среда TRIK Studio [18, 15], позволяющая облегчить знакомство с робототехникой школьникам младших и средних классов с использованием визуального программирования. Одними из наиболее значимых достоинств среды являются генераторы кода на текстовых языках программирования и интерпретатор текстового кода для 2D модели робота.

Среда TRIK Studio позволяет преподавателям произвести более плавный переход от визуального программирования к текстовому и впоследствии обучать сложным синтаксическим конструкциям текстовых языков программирования, используя наглядность уже созданных визуальных диаграмм. ПО контроллера TRIK и TRIK Studio образуют программное обеспечение образовательных решений TRIK, которое на данный момент поддерживает следующие языки программирования: языки платформ Java и .NET, JavaScript, C++, Pascal. Если добавить к перечисленным языкам Python, то TRIK по праву может считаться идеальной робототехнической платформой для обучения школьников и студентов.

¹www.trikset.com

Постановка задачи

Целью данной выпускной квалификационной работы является разработка и внедрение программного решения, позволяющего использовать язык Python для обучения программированию роботов на базе контроллера TRIK. Для достижения поставленной цели необходимо выполнить следующие задачи.

1. Сделать обзор архитектуры существующих образовательных решений TRIK.
2. Выявить требования к программному решению.
3. Разработать архитектуру программного решения.
4. Реализовать программное решение.
5. Внедрить разработанное решение в образовательные продукты TRIK.

1. Обзор технологий

Как правило, образовательное робототехническое решение состоит из следующих компонентов: методические материалы и техническое решение. К методическим материалам относятся презентации, онлайн-курсы, семинары, программы школ, создаваемые и проводимые преподавателями или командой, разрабатывающей образовательное решение. Техническое решение состоит из следующих компонентов: аппаратной платформы, конструктора для прототипирования моделей роботов и программного обеспечения, предназначенного для взаимодействия пользователя с аппаратной платформой.

1.1. TRIK

TRIK — это кибернетический контроллер под управлением операционной системы на основе ядра Linux. Он предназначен для управления роботами, беспилотными летательными аппаратами, средствами передвижения и встраиваемыми устройствами. Несмотря на то, что контроллер TRIK в первую очередь предназначен для обучения, он может быть использован для решения широкого круга задач — например, построения ”умного дома”. Это возможно благодаря мощной аппаратной части контроллера: процессорам ARM², DSP³, MSP⁴.

В образовательных решениях TRIK программное обеспечение представлено следующими компонентами:

- Библиотека времени исполнения trikRuntime на работе
- Среда программирования роботов TRIK Studio
- Интерпретатор 2D модели робота для проверки задач на Stepik

²[ru.wikipedia.org/wiki/ARM_\(архитектура\)](http://ru.wikipedia.org/wiki/ARM_(архитектура))

³ru.wikipedia.org/wiki/Цифровой_сигнальный_процессор

⁴ru.wikipedia.org/wiki/MSP430

1.1.1. trikRuntime

trikRuntime – это набор библиотек, реализующий часть технической составляющей TRIK. Несмотря на то, что для TRIK созданы другие библиотеки времени исполнения, например, trik-sharp [10, 20] или TrikKotlin [19], trikRuntime остаётся наиболее популярным и широко используемым набором библиотек. Полная архитектура trikRuntime представлена на рисунке 1 (страница 8).

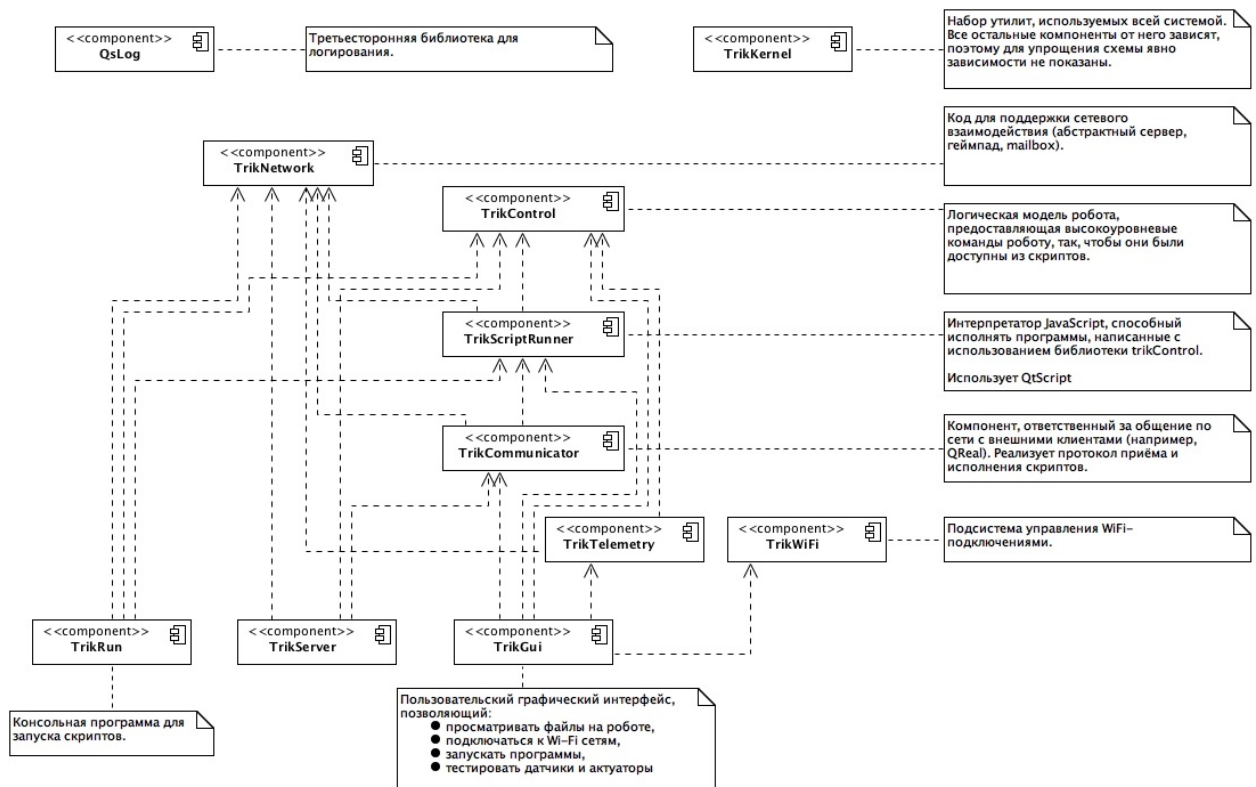


Рис. 1: Архитектура trikRuntime

Основой набора библиотек trikRuntime является библиотека trikControl, предоставляющая API для взаимодействия с сенсорами, актуаторами и другими периферийными частями робота. Ниже перечислены некоторые классы trikControl: Brick — класс, отвечающий за контроллер, инициализирующий периферию робота и дающий к ней доступ, Sensor, ServoMotor и PowerMotor — классы, отвечающие за работу с сенсорами, сервомоторами и силовыми моторами соответственно.

trikRuntime можно использовать двумя способами: запускать написанный на C++/Qt код, использующий библиотеку trikControl или за-

пускать JavaScript-код. JavaScript является основным языком для программирования роботов TRIK. Для удобства пользователя скрипты на языке JavaScript можно запускать из консоли с помощью компонента trikRun, из графического меню робота (компонент trikGui) и из TRIK Studio. Каждый из перечисленных способов использует компонент trikScriptRunner.

Подробно рассмотрим, как исполняется код на языке JavaScript с помощью компонента trikScriptRunner. Архитектура компонента trikScriptRunner находится на рисунке 2 (страница 10). trikScriptRunner использует технологию QtScript⁵ (с версии Qt 5.0 – QJSEngine⁶). QtScript – это язык программирования, который реализует стандарт ECMAScript⁷ с расширением в виде поддержки Qt. QtScript практически не отличается от JavaScript с точки зрения пользователя и в то же время позволяет без труда переиспользовать существующий C++ код, написанный с использованием Qt. Далее для простоты изложения подразумевается, что JavaScript в контексте данной работы – это QtScript или JavaScript, использующий QJSEngine.

1.1.2. TRIK Studio

TRIK Studio – это среда визуального и текстового программирования, поддерживающая наиболее популярные в России и Европе образовательные робототехнические платформы: Lego NXT, Lego EV3, TRIK, некоторые версии STM32. В активной стадии разработки – поддержка большего количества платформ. Поддержка аппаратных платформ осуществляется специальными плагинами.

Плагин представляет собой генератор или интерпретатор текстового языка программирования. Модульная архитектура плагинов TRIK Studio позволяет легко переиспользовать готовые компоненты, расширяя функциональность при необходимости. Среда предоставляет каркас, необходимый для генераторов: преобразование графа потока управ-

⁵doc.qt.io/qt-4.8/qtscript-module.html

⁶doc.qt.io/qt-5/qjsengine.html

⁷<https://ru.wikipedia.org/wiki/ECMAScript>

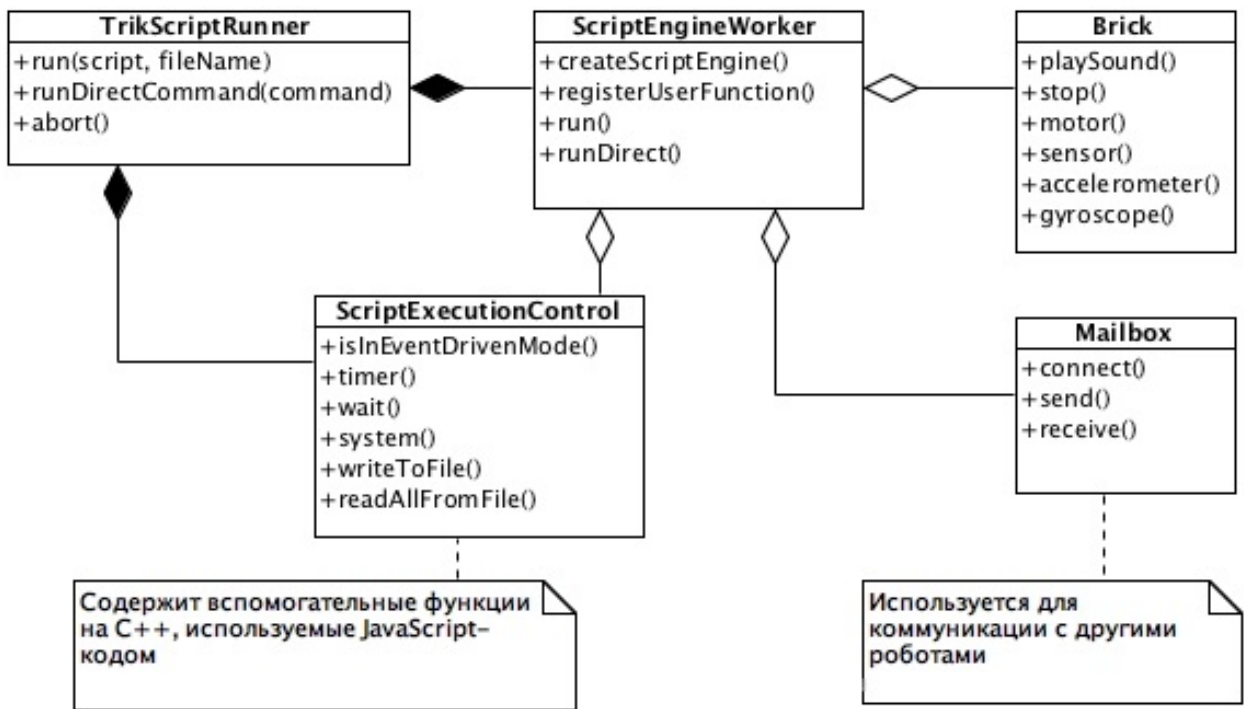


Рис. 2: Архитектура компонента trikScriptRunner

ления к представлению, пригодному для генерации структурного кода, подсветку синтаксиса на базе библиотеки QScintilla⁸. Фактически, для генераторов необходимо реализовать вручную только текстовые представления конструкций языка программирования и поддержку компилятора (для компилируемых языков). Разнообразие уже созданных генераторов и интерпретаторов языков позволяет без труда понять, как именно создать новый плагин для целевого языка.

Помимо плагинов, поддерживающих генерацию или интерпретацию языков программирования, TRIK Studio обладает инструментами автоматизации и отладки: например, пользователь может из среды загрузить и запустить программу на роботе, а после запуска программы получать вывод программы в консоль TRIK Studio.

⁸www.riverbankcomputing.com/software/qscintilla/intro

1.1.3. TRIK Stepik checker

Для популярной платформы онлайн-образования Stepik⁹ существует вводный курс в робототехнику ”Первый шаг в робототехнику”¹⁰ от методистов TRIK. Для этого курса потребовалось создать специализированную систему, способную проверять решения пользователей на корректность. Так появился TRIK Stepik checker, являющийся выделенной в самодостаточную систему частью TRIK Studio. TRIK Stepik checker способен проверить визуальную диаграмму или код на языке JavaScript.

1.2. Фреймворки взаимодействия Python и C++

Существует большое количество наборов библиотек, позволяющих решить задачу использования существующего кода на C++ из языка Python. Поскольку ПО контроллера TRIK и среда TRIK Studio являются свободно распространяемым программным обеспечением, в данной работе рассматривались только следующие фреймворки: PyQt, PySide, PythonQt, Boost.Python, SWIG, которые также являются свободно распространяемым ПО.

Прежде, чем приступить к рассмотрению фреймворков взаимодействия, были выделены следующие требования, которым должен удовлетворять выбранный фреймворк.

- Возможность поддержать исполнение Python из C++ приложения.
- Поддержка интеграции с Qt.
- Активная поддержка фреймворка компанией или сообществом.
- Доступная и полная документация.
- Лицензия со свободными условиями.

⁹stepik.org

¹⁰stepik.org/462

1.2.1. PyQt

PyQt [16] – это набор обёрток для фреймворка Qt, который позволяет использовать большую часть функциональности Qt в Python.

PyQt очень популярен для переиспользования существующих библиотек, написанных на C++ с использованием Qt из Python. PyQt не предоставляет возможности для встраивания частей на языке Python в Qt-приложение. Для удобного переиспользования кода на C++ существует инструмент SIP¹¹, который по специальным sip-файлам порождает код обёрток на языке C++. Однако sip-файлы весьма неудобны тем, что являются почти точными копиями заголовочных файлов. Это приводит к тому, что разработчикам приходится изменять sip-файлы при каждом изменении C++ кода библиотеки. Более того, SIP обладает весьма слабым парсером языка C++, что не позволяет поддерживать некоторые конструкции динамически. Таким образом, приходится прибегать к инъекциям в уже сгенерированный код, что является плохой практикой в силу высокими затрат такого подхода на поддержку и высокой вероятности допустить ошибку.

1.2.2. PySide

PySide – это набор обёрток для фреймворка Qt. Проект PySide стартовал как аналог PyQt и практически не отличается от PyQt. Единственное значимое отличие – PyQt распространяется под коммерческой или GPLv3 лицензией, в то время как PySide распространяется под LGPL лицензией, что позволяет использовать его с большей свободой.

Плюсом фреймворка является то, что Qt Company¹² официально поддерживает PySide, что обеспечивает активную разработку набора библиотек.

Однако, данный продукт не обладает актуальной документацией, что затрудняет его использование. Также стоит отметить, что PySide на данный момент поддерживает меньшую часть набора библиотек Qt

¹¹www.riverbankcomputing.com/software/sip/intro

¹²en.wikipedia.org/wiki/The_Qt_Company

по сравнению с PyQt.

1.2.3. PythonQt

PythonQt – это библиотека для Qt, нацеленная на встраивание Python в C++/Qt приложение. PythonQt разрабатывался для платформы MeVis-Lab¹³ с целью предоставления возможности использования Python-кода для расширения возможностей платформы или специальной обработки информации [8].

Для использования PythonQt необходимо в C++ коде описать обёртку необходимого объекта и зарегистрировать её в желаемом модуле Python. Для упрощения данной работы PythonQt имеет генератор, способный по заголовочным файлам C++ и системе типов сгенерировать все необходимые обёртки. Хотя генератор находится в разработке и не содержит некоторых необходимых функций, анализ кода генератора показал, что временные затраты на доработку его до состояния, необходимого для данной работы, приемлемы.

1.2.4. Boost.Python

Boost.Python [1] – это библиотека для взаимодействия C++ и Python, позволяющая использовать скрипты на языке Python как часть программы, написанной на C++.

Основная цель Boost.Python – предоставить возможность переиспользовать код, написанный на C++, в коде, написанном на Python, средствами лишь компилятора C++. Это отличает его от, например, SIP, который использует специальный язык для создания промежуточной коммуникации. При этом в Boost.Python учитываются основные различия между двумя языками и предоставлены интерфейсы для удобного взаимодействия.

Для автоматизации процесса создания обёрток над кодом существуют следующие инструменты: Pyste и Py++. Pyste¹⁴ – генератор обёр-

¹³www.mevislab.de

¹⁴www.boost.org/doc/libs/1_60_0/libs/python/pyste/doc/running_pyste.html

	PyQt	PySide	PythonQt	SWIG	Boost.Python
встраивание Python в C++	-	-	+	-	+
интеграция с Qt	+	+	+	-	-
поддерживается	+	-	-	+	+
документация	±	±	±	±	±
обёртка C++ для Python	+	+	-	+	+
лицензия	GPL	LGPL	LGPL	GPL	Boost.License

Таблица 1: Сравнение фреймворков

ток для Boost.Python, который был разработан, как часть фреймворка Boost.Python. Разработка была прекращена в 2011 году. Py++¹⁵ – генератор обёрток для Boost.Python, используемый в популярной библиотеке планирования движения OMPF [17]. В отличие от Pyste, Py++ поддерживается разработчиками OMPF и, следовательно, может быть использован при разработке новых проектов с использованием Boost.Python.

Однако, поскольку Boost.Python разрабатывался для работы с чистым C++ кодом, то в нём отсутствует поддержка фреймворка Qt. Это не позволяет использовать сигналы и слоты Qt из Python, что является недостатком.

1.2.5. SWIG

SWIG [3] – инструмент для связывания программ и библиотек, написанных на языках C и C++, с интерпретируемыми или компилируемыми языками. Отличием SWIG от вышперечисленных наборов библиотек является возможность работы не только с Python, но и многими другими языками программирования.

Связывание происходит посредством интерфейсных файлов с расширением `.i`, в которых описываются экспортируемые функции. Сам процесс связывания достаточно удобен, SWIG является генератором связующего кода и не требует отдельного генератора.

Однако, SWIG не поддерживает встраивание Python в процесс приложения на C++ и не имеет поддержки Qt, что является значительными недостатками.

¹⁵bitbucket.org/ompl/pyplusplus

2. Разработка требований

Прежде, чем приступить к проектированию программного обеспечения, необходимо выявить требования заказчиков и будущих пользователей нового образовательного решения, а также оценить объёмы работ и риски, связанные с разработкой программного обеспечения, позволяющего использовать Python в образовательных решениях TRIK.

Сценарии использования основного образовательного решения TRIK представлены на рисунке 3 (страница 15).

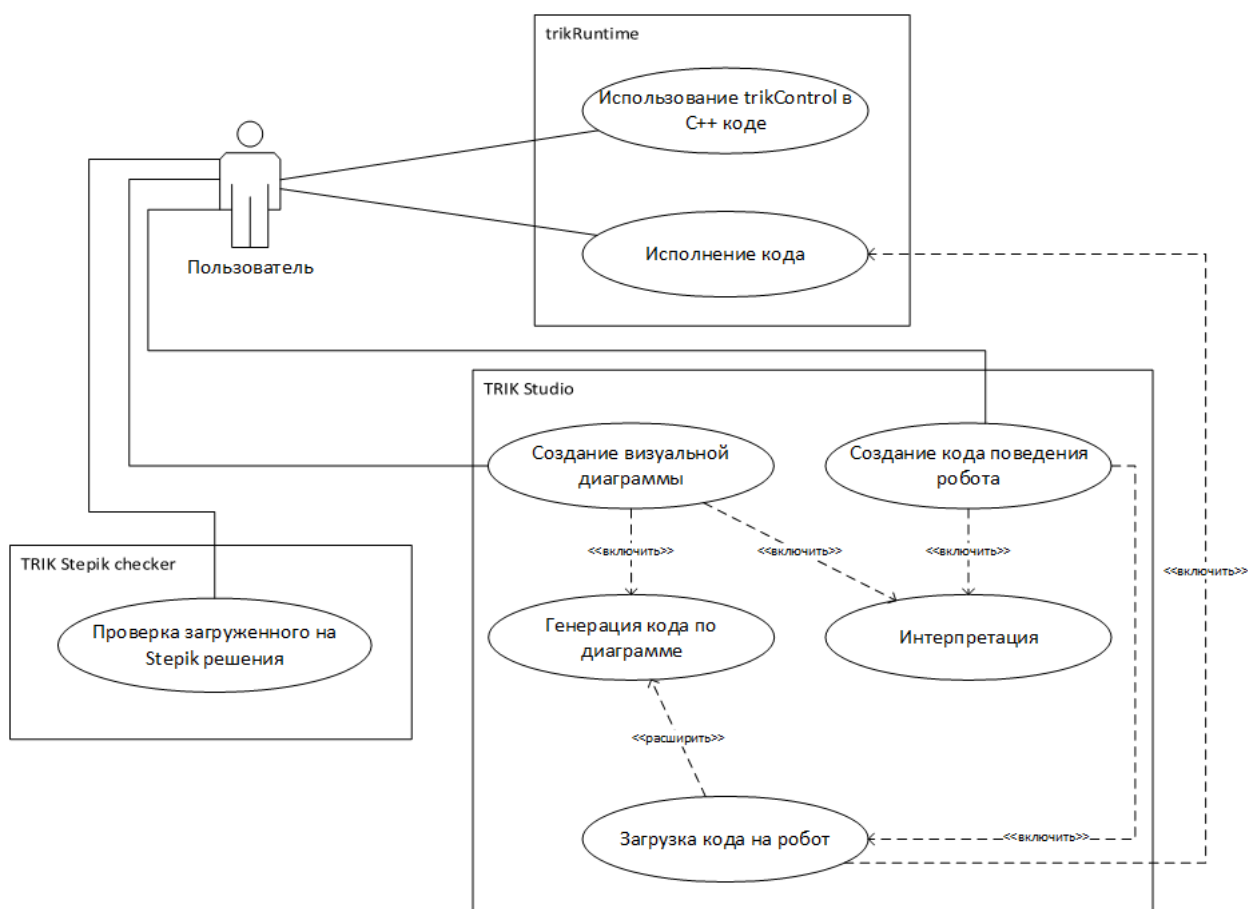


Рис. 3: Сценарии использования образовательного решения TRIK

Таким образом, были выделены следующие согласованные требования:

- В сжатые сроки (один месяц) предоставить версию продукта, обладающую минимальной необходимой функциональностью.
- Учитывать затраты на сопровождение решения.

- Обеспечить одинаковое поведение при использовании Python в 2D модели или на реальном роботе.

3. Архитектура

Поскольку одним из требований к техническому решению была разработка версии продукта, обладающего минимальной функциональностью, было принято решение разрабатывать техническое решение в два этапа:

1. Разработка технического решения с минимальной функциональностью
 - Создание обёртки `trikRuntime` для использования с `PyQt`
 - Реализация генератора Python в `TRIK Studio`
2. Разработка функциональности, аналогичной JavaScript в `trikScriptRunner`
 - Использование `PythonQt` для встраивания в `trikRuntime`
 - Реализация интерпретации Python 2D моделью робота

3.1. Робот TRIK

Разработка программного обеспечения для робота представляла наибольшую трудность. Именно из-за сложности реализации данного компонента было принято решение разработки системы в два этапа. По результатам обзора 1 (страница 14) `PythonQt` или `Boost.Python` являлись наиболее подходящими фреймворками для решения данной задачи, но оба фреймворка обладали недостатками, для устранения которых потребовалось бы значительное количество времени: `Boost.Python` не хватало поддержки Qt и её пришлось реализовать в рамках данной работы, а `PythonQt` были необходимы патчи к генератору.

Поэтому для первого этапа был использован `PyQt`, который позволил в сроки, определённые требованиями, создать первую версию программного обеспечения для робота, хотя `PyQt` предполагал невозможность переноса

Переход на PythonQt позволил добиться более естественной интеграции наподобие той, что предлагает JavaScript. Более того, пользователям TRIK, привыкшим работать с JavaScript, не придётся привыкать к новым решениям, поскольку PythonQt позволил внедриться в trikScriptRunner и тем самым практически не отличаться от уже существующего решения, использующего JavaScript.

3.2. TRIK Studio

Разработка генератора Python не составляла большой трудности, поскольку для TRIK Studio уже написано множество генераторов. Единственной сложностью, связанной с генератором, было аккуратное оформление. Генератор требует достаточно большое количество внутренней технической информации, отличной друг от друга в различных плагинах, в которой легко допустить ошибку.

Интерпретация Python была более сложной задачей в сравнении с созданием генератора, поскольку возможность интерпретации текстовых языков программирования появилась в TRIK Studio сравнительно недавно. По сути, интерпретация JavaScript на момент создания данной работы была использована лишь на олимпиаде НТИ¹⁶. Однако, благодаря помощи разработчиков TRIK Studio трудности, возникавшие при создании интерпретатора, были преодолены.

3.3. TRIK Stepik Checker

Поскольку интерпретация текстовых языков программирования в TRIK Stepik Checker ничем не отличается от интерпретации в среде TRIK Studio, то вся работа, связанная с данным компонентом, заключалась в переносе изменений из TRIK Studio в Checker.

¹⁶nti-contest.ru/profiles/irs

4. Апробация

В качестве проверки применения представленного технического решения были проведены несколько экспериментов, тестирующих части системы отдельно и в комплексе. Был успешно проверен переписанный с JavaScript на Python код некоторых демонстрационных моделей TRIK: светового меча и дистанционно управляемой тележки. Также была проверена работоспособность кодогенератора Python на всех базовых примерах, поставляемых с TRIK Studio. Полученный порождённый код успешно запускался на работе TRIK.

Таким образом, в ходе апробации части системы и система в целом были протестированы, а полученный код моделей может быть использован для регрессионного тестирования представленного в данной работе технического решения.

5. Документация

Поскольку образовательные решения TRIK ориентированы в том числе и на школьников средних и старших классов, одним из важным моментов при создании описанной системы было написание базовой документации, описывающей техническое решение, и создание примеров программ для наглядности. Это упростило работу методистов и позволило первым пользователям быстрее разобраться с представленной системой. Документация разработчика состоит из комментариев для системы Doxygen¹⁷.

¹⁷<http://www.stack.nl/~dimitri/doxygen/>

Заключение

В рамках данной работы были получены следующие результаты.

1. Сделан обзор архитектуры существующего ПО образовательных решений TRIK.
2. Определены требования к программному решению.
3. Разработана архитектура программного решения.
4. Выполнена реализация программного решения.
5. Разработанное программное решение внедрено в образовательные решения TRIK.

В ходе работы промежуточные результаты представлялись докладом на VII Всероссийской конференции "Современное технологическое обучение: От компьютера к роботу", а также докладом на всероссийской конференции "СПИСОК-2017".

Одним из направлений развития полученных результатов является создание полноценного аналога QJSEngine для языка Python.

Список литературы

- [1] Abrahams David, Grosse-Kunstleve Ralf W. Building hybrid systems with Boost. Python // CC Plus Plus Users Journal. — 2003. — Vol. 21, no. 7. — P. 29–36.
- [2] BV TIOBE Software. TIOBE Index. — URL: <https://www.tiobe.com/tiobe-index/>.
- [3] Beazley David M et al. SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++. // Tcl/Tk Workshop. — 1996.
- [4] Benitti Fabiane Barreto Vavassori. Exploring the educational potential of robotics in schools: A systematic review // Computers & Education. — 2012. — Vol. 58, no. 3. — P. 978–988.
- [5] Bybee Rodger W. What Is STEM Education? // Science. — 2010. — Vol. 329, no. 5995. — P. 996–996. — <http://science.sciencemag.org/content/329/5995/996.full.pdf>.
- [6] Carbonnelle Pierre. PYPL PopularitY of Programming Language. — URL: <http://pypl.github.io/PYPL.html>.
- [7] Guo Philip. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. — 2014. — URL: <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext> (online; accessed: 20.04.2017).
- [8] Heckel Frank, Schwier Michael, Peitgen Heinz-Otto. Object-oriented application development with MeVisLab and Python. // GI Jahrestagung. — 2009. — Vol. 154. — P. 1338–51.
- [9] Javidi Giti, Sheybani Ehsan. Making Youth Excited About STEM Education // J. Comput. Sci. Coll. — 2010. — Vol. 26, no. 1. — P. 140–147. — URL: <http://dl.acm.org/citation.cfm?id=1858449.1858474>.

- [10] Kirsanov Alexander, Kirilenko Iakov, Melentyev Kirill. Robotics Reactive Programming with F#/Mono // Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia. — CEE-SECR '14. — New York, NY, USA : ACM, 2014. — P. 16:1–16:5. — URL: <http://doi.acm.org/10.1145/2687233.2687249>.
- [11] Learning STEM. Python. — 2010. — URL: <https://www.stem.org.uk/rx34zw> (online; accessed: 20.04.2017).
- [12] Limited QS Quacquarelli Symonds. QS Top Universities in Engineering and Technology 2015/2016. — URL: <https://www.topuniversities.com/university-rankings/faculty-rankings/engineering-and-technology/2015>.
- [13] Limited QS Quacquarelli Symonds. QS Top Universities in Engineering and Technology 2017. — URL: <https://www.topuniversities.com/university-rankings/university-subject-rankings/2017/engineering-technology>.
- [14] MIT. Introduction to Electrical Engineering and Computer Science I. — 2011. — URL: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/> (online; accessed: 1.05.2017).
- [15] Mordvinov Dmitry, Litvinov Yurii, Bryksin Timofey. TRIK Studio: Technical Introduction // Proceedings of the FRUCT'20. — 2017. — P. 296–308. — URL: <https://fruct.org/publications/fruct20/files/Mor.pdf>.
- [16] Riverbank. PyQt Whitepaper. — 2013. — URL: <https://www.riverbankcomputing.com/static/Docs/PyQt4/pyqt-whitepaper-a4.pdf> (online; accessed: 16.04.2017).
- [17] Şucan Ioan A., Moll Mark, Kavraki Lydia E. The Open Motion Planning Library // IEEE Robotics & Automation Magazine. —

2012. — December. — Vol. 19, no. 4. — P. 72–82. — URL: <http://ompl.kavrakilab.org>.

- [18] Terekhov Andrey, Litvinov Yurii, Bryksin Timofey. QReal:Robots an Environment for Teaching Computer Science and Robotics in Schools // Proceedings of the 9th Central Eastern European Software Engineering Conference in Russia. — CEE-SECR '13. — New York, NY, USA : ACM, 2013. — P. 10:1–10:5. — URL: <http://doi.acm.org/10.1145/2556610.2591506>.
- [19] Белков Роман. Программирование роботов ТРИК с помощью JVM. — 2016. — URL: http://se.math.spbu.ru/SE/YearlyProjects/spring-2016/371/Belkov_report.pdf.
- [20] Кирсанов Александр. Программирование роботов с помощью .NET/Mono. — 2015. — URL: <http://se.math.spbu.ru/SE/diploma/2015/bmo/444-Kirsanov-report.pdf>.