

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Системное программирование

Ивашева Валерия Михайловна

Реализация веб-редактора правил системы
«Умный дом»

Курсовая работа

Научный руководитель:
доцент кафедры системного программирования, к.т.н. Ю. В. Литвинов

Санкт-Петербург
2020

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Существующие решения	6
2.1.1. NodeRED	6
2.1.2. Homey Flow	6
2.1.3. HomeKit	8
2.2. Используемые технологии	9
2.2.1. REAL.NET	9
2.2.2. Vue.js	10
2.2.3. Vuex	11
3. Описание решения	12
3.1. Метамодел ь	12
3.2. Веб-редактор, пользовательский интерфейс	12
3.3. Веб-редактор, реализация	14
4. Апробация	15
Заключение	17
Список литературы	18

Введение

В современном мире все большую популярность набирает концепция «Интернета вещей» [2]. Она заключается в возможности объединения различных устройств при помощи различных технологий передачи данных для совместной работы. Умные вещи — это устройства «Интернета вещей», которые могут работать интерактивно и автономно. Примером применения данной концепции является система технологии «Умный дом».

«Умный дом» — это система, позволяющая объединить и автоматизировать различные устройства и бытовые приборы в доме, а также настраивать их взаимодействие без участия человека.

Функции, которые могут быть включены в систему «Умного дома»:

- контроль и управление электроснабжением;
- контроль и управление освещением;
- управление микроклиматом (например, кондиционирование воздуха, отопление, вентиляция);
- безопасность (контроль утечки газа, воды);
- управление бытовыми приборами.

Использование системы «Умного дома» позволяет сэкономить на расходах за электроэнергию и теплоснабжение. Оно приносит удобство для пользователей за счет создания различных сценариев управления умных вещей или приборов. На данный момент российский рынок «умных домов» развивается и растет. В 2017 году общий объем российского рынка «умный дом» достиг 176 млн. евро или 7,9 млрд. рублей. Это более чем в три раза больше по отношению к уровню за 2012 год. Прогнозируется, что в ближайшие 5 лет рынок будет расти средними темпами 24% в год [11].

Современные системы состоят из датчиков и актуаторов (исполнительные устройства, которые выполняют команды, заданные сценари-

ем). Пользователь должен иметь возможность задавать различные сценарии управления домом. Примерами таких сценариев являются отправка на телефон сообщения при срабатывании датчиков утечки воды или включение света и поддержание температуры помещений в различное время суток.

Для создания сценариев пользователь может воспользоваться специальным редактором. Основная задача системы редактора: предоставить пользователю возможность графически определять правила совместной работы датчиков и исполнительных устройств дома. Таким образом, необходимо специальное программное обеспечение, которое предоставляет возможность легко создавать желаемые сценарии пользователям, необязательно знакомым с программированием.

Для задания сценариев можно использовать визуальный язык, в котором программы создаются при помощи манипулирования графическими элементами, а не текстовыми командами. На кафедре системного программирования математико-механического факультета СПбГУ разрабатывается REAL.NET [16] — среда для визуального моделирования. Мы можем воспользоваться ей для наших целей. На данный момент на кафедре разрабатывается веб-версия редактора — REAL.NET Web. Тестированием веб-редактора REAL.NET Web может послужить создание на его основе веб-редактора правил системы «Умный дом».

1. Постановка задачи

Целью работы является создание на основе платформы REAL.NET веб-редактора правил системы «Умный дом», включающего в себя предметно-ориентированный визуальный язык программирования для задания пользователем сценариев работы устройств.

Для достижения цели нужно решить следующие задачи.

1. Спроектировать пользовательский интерфейс веб-редактора.
2. Разработать предметно-ориентированный визуальный язык.
3. Реализовать веб-редактор для моделирования пользователями сценариев работы «умного дома».
4. Провести апробацию прототипа.

2. Обзор

При обзоре существующих решений рассматривались редакторы, которые позволяют создавать и автоматизировать различные сценарии для «умного дома». Цель обзора — изучение способов задания сценариев.

2.1. Существующие решения

2.1.1. NodeRED

Часто применяемой системой для настройки «умного дома» является NodeRED. NodeRED [14] — это инструмент визуального программирования для интернета вещей. Редактор для создания сценариев имеет графовый вид, то есть сценарий задается графом, где вершины — это элементы, а ребра — это связи между ними. Пользователь может подключать друг к другу устройства, строя сценарий в виде графической схемы, и генерировать по нему код. Интерфейс поделен на три основных части — это список элементов, место для размещения элементов и свойства объектов, которые иногда включают в себя элемент для написания кода на JavaScript. На рисунке 1 изображен сценарий, созданный в системе NodeRED.

У системы достаточно сложный механизм задания правил, что является ее недостатком, но при этом она предоставляет возможность подключения различных модулей для расширения своих возможностей.

2.1.2. Homey Flow

Homey Flow [6] — редактор для создания и управления сценариями, который входит в состав программного обеспечения, предоставляемого в комплекте с контроллером «Homey Athom». Редактор имеет табличный вид, то есть структура сценария задана редактором, а пользователь только указывает нужные элементы и условия в таблице. В редакторе системы сцена разделена на зоны: события, условия, действия. Такой способ размещения упрощает создание правил, так как исчезает

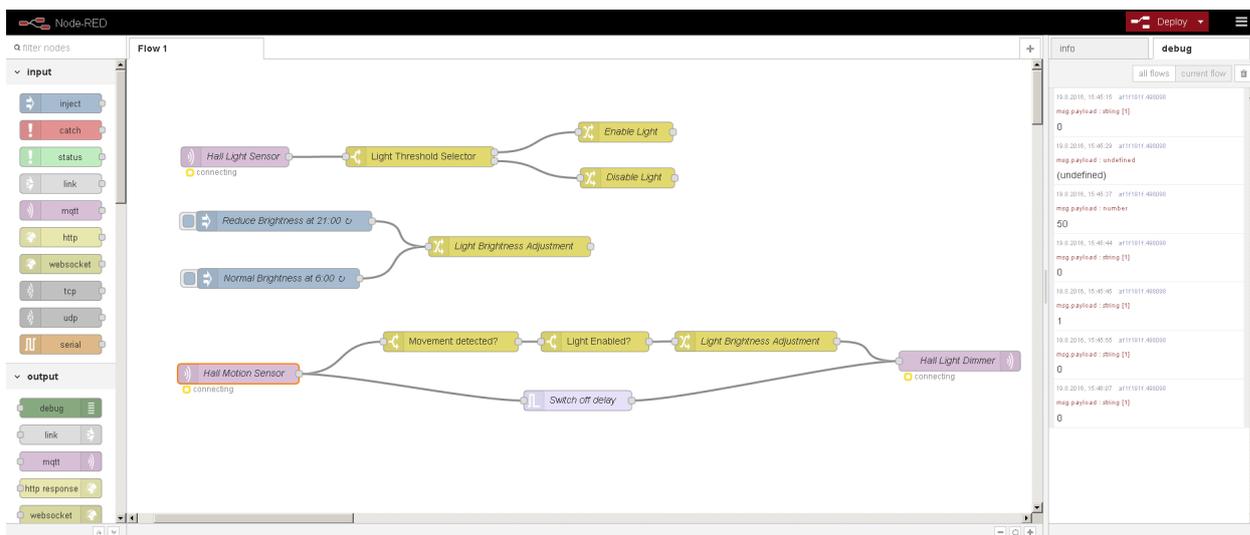


Рис. 1: Сценарий, созданный в NodeRED (из [15])

вероятность появления паутины связей. Блоки условий и действий дополнительно разделены на основные и альтернативные. На рисунке 2 изображено создание сценария в редакторе Homey Flow.

Homey Flow предоставляет удобный и понятный интерфейс, но для доступа к нему необходимо наличие контроллера «Homey Athom».

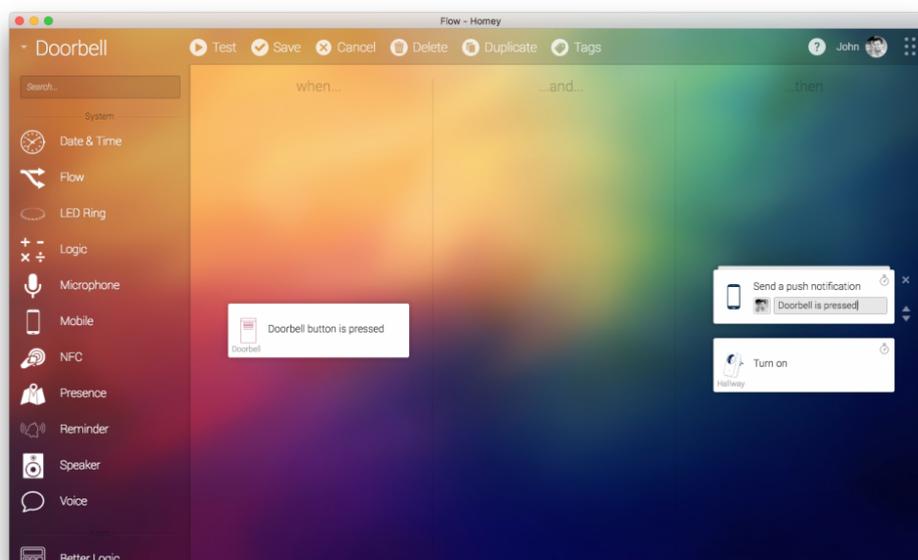


Рис. 2: Создание сценария в Homey Flow (из [8])

2.1.3. HomeKit

HomeKit [17] — это программный фреймворк, который позволяет использовать iOS устройство для настройки и управления экосистемы домашних умных устройств. Устройства различных производителей могут быть добавлены в приложение «Дом» (официальное приложение от iOS для управления системой). Также были опубликованы принципы взаимодействия HomeKit с различными сенсорами, поэтому любой может собрать и подключить новый гаджет к «умному дому». На рисунке 3 изображено создание сценария при помощи приложения «Дом» в HomeKit.

Приложение имеет простой и понятный интерфейс для настройки и управления системой. Но существенным минусом является необходимость наличия как минимум одного устройства iOS и высокая стоимость оборудования.

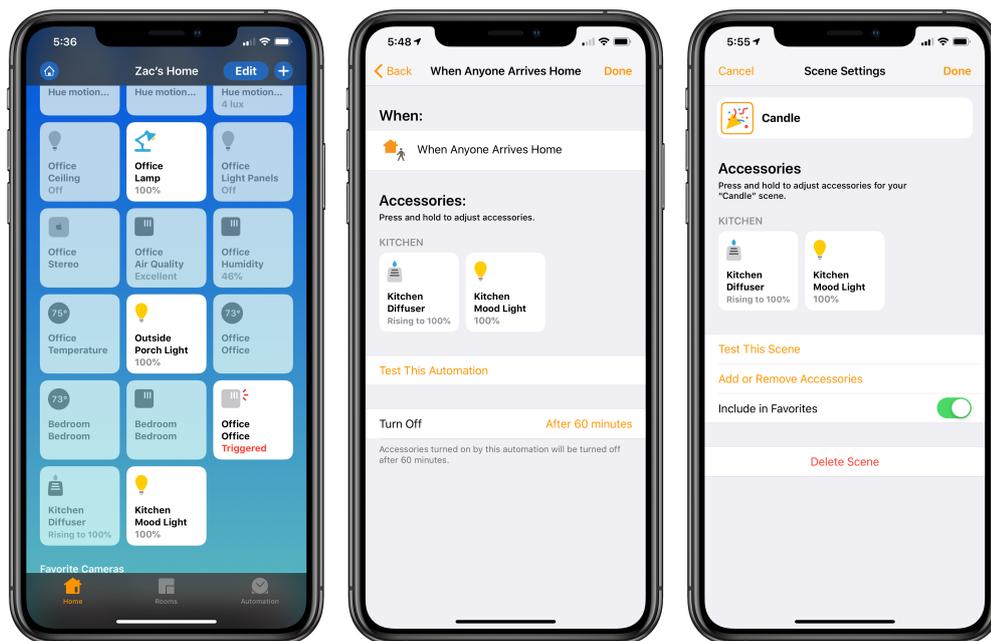


Рис. 3: Создание сценария в HomeKit (из [5])

Вывод

В ходе обзора было выявлено, что большинство систем требует проприетарного аппаратного обеспечения. Однако тут выделяется Node-

RED, но он достаточно сложный и предполагает навыки программирования на JavaScript. Редакторы систем для создания сценариев, рассмотренные в обзоре, имеют табличный и графовый вид. Графовый редактор гибче табличного за счет того, что структуру сценария создает сам пользователь, но в нем есть риск появления паутины связей.

2.2. Используемые технологии

2.2.1. REAL.NET

Процесс создания визуального языка может осуществляться с использованием метамоделирования. Метамоделирование включает в себя создание метамоделей — модели, описывающей множество всех корректных конструкций языка. Сама метамодель может быть создана с помощью визуального языка, называемого метаязыком, который определяется некоторой метамоделью (она называется мета-метамоделью).

REAL.NET [16] — инструмент для быстрого создания визуальных предметно-ориентированных языков. В его основе лежит принцип «глубокого метамоделирования», идея которого заключается в том, что сущность в модели рассматривается одновременно и как тип, и как экземпляр некоторого типа.

Ядром платформы REAL.NET является репозиторий. В репозитории хранятся все модели и метамоделей, а еще предоставляется интерфейс для редактирования этих самых моделей.

На основе REAL.NET была реализована графическая технология управления «умной теплицей» [12], как десктопное приложение. Она позволяет создавать сценарии совместной работы датчиков и актуаторов, связывая их вместе на сцене при помощи логических операций И и ИЛИ, а также используя интервалы для задания различных условий. Таким образом, сценарий представляется как граф, где узлы — это датчики, актуаторы, условия и интервалы, а ребра — это связи между ними. На рисунке 4 изображено создание сценария в REAL.NET «Умная теплица».

Следующим этапом стала разработка веб-редактора для «Умно-

го дома», моя коллега сделала первичный анализ, выполнила наброски экранов, а также начала реализацию самого веб-редактора на TypeScript, используя библиотеку jQuery UI [13] и для визуализации элементы набора инструментов bootstrap. Было решено использовать ее наработки, но сам веб-редактор переписать на Vue, используя компонентный подход.

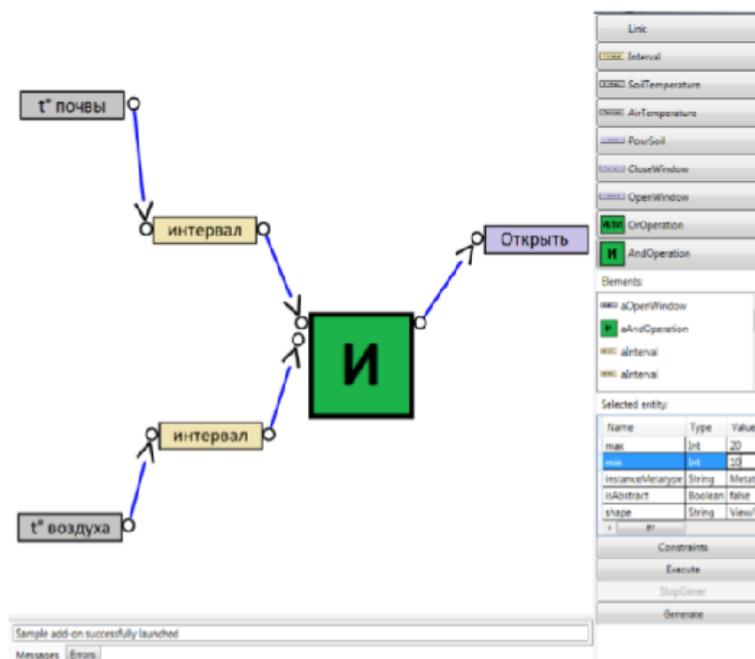


Рис. 4: Создание сценария в REAL.NET «Умная теплица» (из [12])

2.2.2. Vue.js

Для создания пользовательского интерфейса было решено использовать уже готовый фреймворк или библиотеку. Рассматривались следующие варианты:

- React [7] — это библиотека, а не полноценный фреймворк, поэтому для расширения функций понадобятся сторонние библиотеки. Использует Virtual DOM (Document Object Model), что позволяет увеличить скорость работы приложения на стороне клиента, за счет работы с легким JavaScript-объектом вместо прямой работы с DOM.

- Vue.js [9] — JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Прост для изучения, для начала работы с этим фреймворком достаточно базовых знаний языка JavaScript, а также HTML и CSS. В Vue тоже используется виртуальная DOM, но, в сравнении с React, Vue отличается более высокой производительностью и стабильностью. Одна из концепций Vue — это использование компонент. Компоненты — это переиспользуемые экземпляры Vue со своим именем. За счет использования компонент увеличивается переиспользование кода.
- Angular [1] — это фреймворк для создания клиентских приложений, разрабатываемый Google. Основным языком для разработки является Typescript. Используется также компонентный подход. Angular подходит для разработки крупных проектов.

Из-за низкого порога вхождения, обширного набора стандартных инструментов и наличия хорошей документации для разработки был выбран Vue.js.

2.2.3. Vuex

Vuex [10] — это официальный плагин для Vue.js, который предлагает централизованное хранилище данных для использования в приложении. Данное хранилище содержит состояние приложения. Оно реактивно, то есть если состояние приложения изменилось, то и компоненты обновились. Состояние изменяется мутациями, которые выполняются в ответ на действие, вызывающее событие мутации через диспетчера.

Библиотека vuex-module-decorators [3] позволяет работать с Vuex, используя Typescript.

3. Описание решения

3.1. Метамодел

На рисунке 5 представлена метамодел «умного дома».

Корнем иерархии наследования узлов созданной нами метамодел языка является `abstractNode`, остальные члены метамодел должны быть его наследниками. И также существует единственное отношение ассоциации, которое связывает между собой наследников абстрактного узла. Так как «умные дома» обычно состоят из датчиков и актуаторов, было решено вынести их в метамодел. Таким образом, у нас еще добавляются следующие элементы: `abstractRoom`, `abstractSensor`, `abstractActuator`. Реальные датчики, актуаторы и комнаты у каждого «умного дома» свои, они могут быть дописаны в метамодел при подготовке решения уже для конкретного «умного дома». На данный момент для тестирования было решено добавить комнаты (`Kitchen`, `bedroom`), актуаторы (`lamp`, `conditioner`) и сенсоры (`timer`, `temperature`). У `abstractRoom` есть свойства `sensors` и `actuators`, которые хранят находящиеся в этой комнате сенсоры и актуаторы соответственно. Для задания условий нам нужен также элемент `interval`, который имеет свойства `minValue` и `maxValue` — это левый и правый интервал соответственно. Для связи различных условий и элементов у нас также есть логическая операция И. Элемент `switch` отвечает за элементы, которые можно задать с помощью булевской переменной, это может быть условие включен прибор или нет.

3.2. Веб-редактор, пользовательский интерфейс

Редактор разделен на две части.

1. Боковая панель, которая содержит в себе элементы из метамодел.
2. Сцена, на которой пользователь задает сценарии.

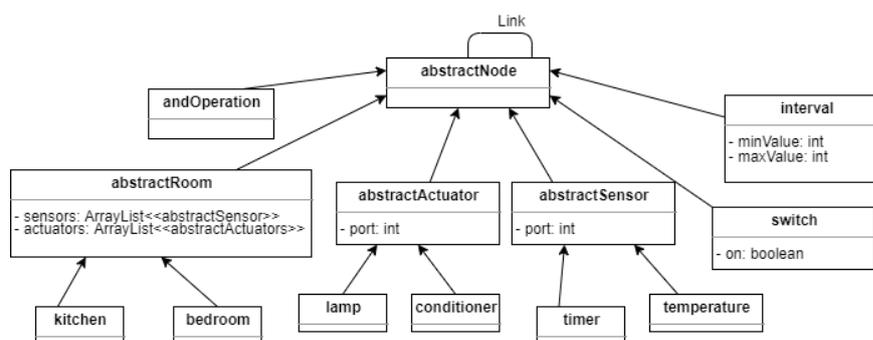


Рис. 5: Метамодел ь «умного дома»

Для реализации рассматривались два вида редакторов: графовый и табличный. В основе системы REAL.NET графовая модель языков, есть узлы и есть рёбра. На данный момент уже есть десктопный графовый редактор, и также разрабатывается его веб-версия. Графовый вид избыточен и сложнее, чем нужно, также возникает риск запутаться в связях между элементами. Приоритетом для данной задачи является удобство использования так, как редактор разрабатывается для конечных пользователей. Поэтому было решено выбрать табличный вид.

Редактор новой системы разделен на следующие области: помещение, состояние, условие. Пользователь оперирует сгруппированными по комнатам датчиками и мы не заставляем его задавать конкретные устройства. Вдобавок мы избавляемся от не вполне очевидной конструкции «событие — условие», при использовании которой может возникать путаница. Например, при создании правила «включить свет, если на часах 7 утра и за окном темно», логично предположить, что срабатывание таймера — это событие, а освещенность — условие, но и в противной ситуации сценарий будет работать аналогично.

Также для большего упрощения было решено ограничиться только одним типом логических операций для определения условий срабатывания правила: «И» (так как активация различных правил и так соответствует логической операции «ИЛИ» и для пользователя будет удобно применять только один вариант группировки условий).

На рисунке 6 изображен следующий сценарий: «На кухне включить лампу 1 на 100% и лампу 2 на 60%, когда на часах 6:00 — 7:00». Для

задания сценария пользователю нужно всего лишь выбрать необходимые элементы из боковой панели и перетащить их в нужное место, то есть помещение, где происходит действие, в Space, состояние в Comfort и условие в Condition, а затем задать необходимые параметры.

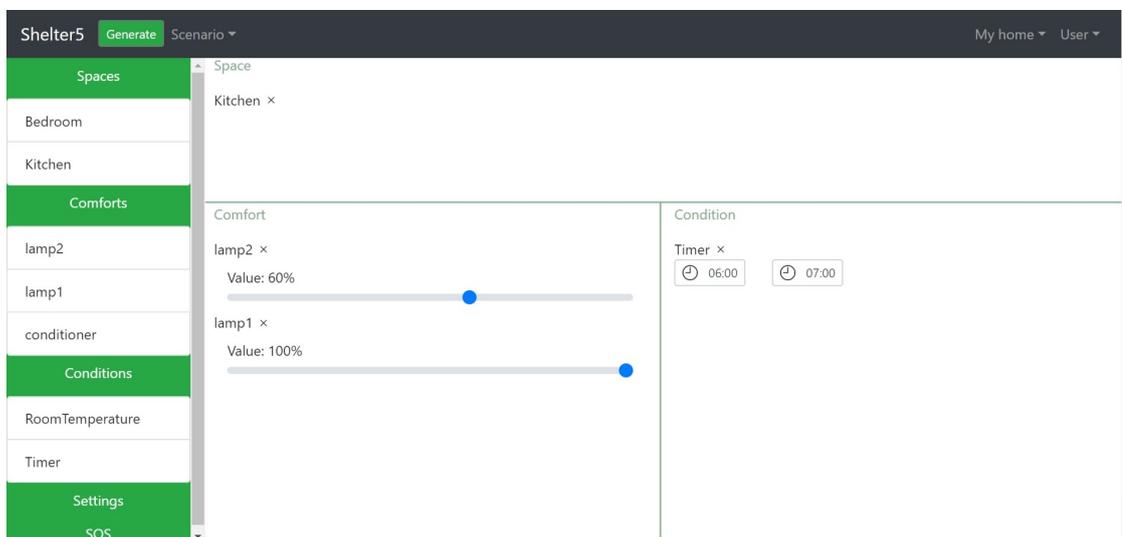


Рис. 6: Сценарий, созданный в веб-редакторе

3.3. Веб-редактор, реализация

На рисунке 7 изображена диаграмма классов редактора, отвечающих за хранение данных. Для каждого элемента панели, которыми оперирует пользователь, у нас есть свой класс, в котором содержатся их свойства, на рисунке — это Comfort, Space, Condition. VuexModule — это класс из библиотеки vuex-module-decorators, которая отвечает за корректную работу Vuex и Typescript. Все элементы для хранилища являются наследниками этого класса. Классы ComfortPanel, SpacePanel, ConditionPanel отвечают за соответствующие элементы сценария на сцене. Боковая панель отображает элементы из классов Comforts, Spaces, Conditions, которые содержат в себе элементы из метамодели.

Для того чтобы получить данные из метамодели, редактор обращается к серверу REAL.NET Web, используя Repo API, который он нам предоставляет. За работу с сервером отвечает модуль Service, который посылает запросы на REAL.NET Web и обрабатывает их.

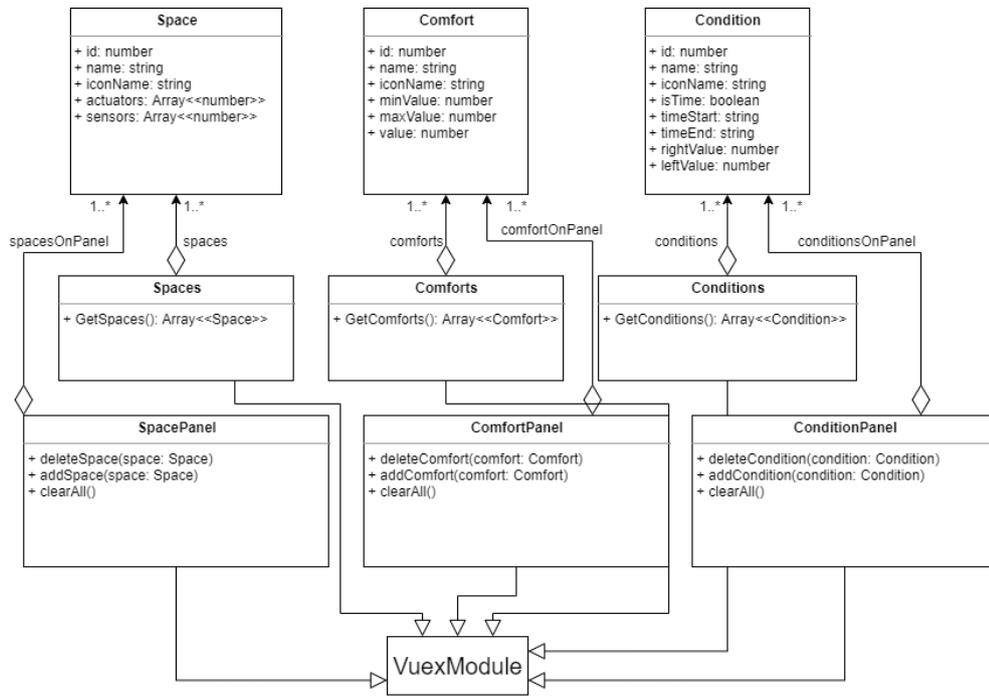


Рис. 7: Диаграмма классов хранилища редактора

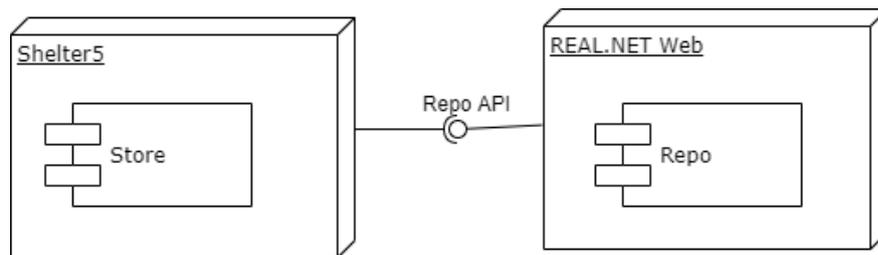


Рис. 8: Связь редактора с REAL.NET Web, диаграмма компонентов

4. Апробация

Апробация прототипа проходила следующим образом. Пользователь должен был реализовать 2 сценария.

1. «На кухне включить лампу 1 на 100% и лампу 2 на 60%, когда на часах 6:00 — 7:00».
2. «Включить кондиционер в спальне, если температура выше 20 градусов».

И затем ответить на вопросы.

1. Сколько времени заняло создание сценария, 1 и 2 соответственно?
2. Критика редактора, что понравилось, что нет?
3. Стали ли бы Вы пользоваться данным редактором?

Всего участвовало 14 человек. Все участники справились с созданием сценариев. На первый сценарий в среднем ушло 2 минуты 45 секунд, самое большое количество времени — 5 минут. Второй сценарий занял в среднем 1 минуту, это объясняется тем, что, во-первых, участники уже знакомы с системой, а, во-вторых, он немного легче, чем первый.

В целом, пользователи отмечают удобный и интуитивно понятный интерфейс, не перегруженный дизайн. 10 из 14 опрошенных стали бы пользоваться данным редактором.

Также благодаря апробации были выявлены следующие минусы.

- Использование технологии drag-and-drop для добавления элементов на сцену не очевидно. Решение: добавлять элементы двойным кликом мыши.
- Формы с выбором времени не очень удобны тем, что нужно слишком много кликать и при отсутствии ввода минут теряется все время. Решение: добавить ручной ввод и установить значение по умолчанию.
- Непонятно включен ли прибор. Решение: на кнопке добавить значение on/off в зависимости от нужного состояния.

Валидность результатов эксперимента

Достоверность результатов данного эксперимента ставится под вопрос, так как на результат могли повлиять следующие факторы:

- непосредственное знакомство участников с автором;
- большинство участников знакомы с программированием.

Но даже несмотря на это, были выявлены, как минимум, недостатки системы, которые можно исправить.

Заключение

В рамках курсовой были решены следующие задачи.

- Проведено исследование существующих редакторов для создания сценариев «умного дома».
- Начата работа над пользовательским интерфейсом редактора.
- Спроектирован пользовательский интерфейс веб-редактора.
- Разработан предметно-ориентированный визуальный язык.
- Реализован веб-редактор для моделирования пользователями сценариев работы «умного дома».
- Проведена апробация прототипа.

Готовятся к публикации тезисы доклада «REAL.NET Web — web-based multilevel domain-specific modeling platform (Mikhail Kidiankin, Yurii Litvinov, Valeria Ivasheva, Elizaveta Kuzmina, Yuniya Kim and Angelina Chizhova)». Ознакомиться с редактором можно в GitHub-репозитории проекта [4].

Список литературы

- [1] Angular. — Access mode: <https://angular.io/> (online; accessed: 19.05.2020).
- [2] Atzori Luigi, Iera Antonio, Morabito Giacomo. The Internet of Things: A survey // Computer Networks, Volume 54, Issue 15. — 2010. — P. 2787–2805.
- [3] Github-репозиторий vuex-module-decorators. — Access mode: <https://github.com/championswimmer/vuex-module-decorators> (online; accessed: 19.05.2020).
- [4] Github-репозиторий проекта. — Access mode: <https://github.com/REAL-NET/Shelter5> (online; accessed: 20.05.2020).
- [5] HomeKit Weekly: VOCOLinc’s FlowerBud combines a Siri-controlled diffuser with a cool mood lamp. — Access mode: <https://9to5mac.com/2019/01/23/homekit-weekly-vocolinc-flowerbud-diffuser-lamp/> (online; accessed: 20.05.2020).
- [6] Homey Flow. — Access mode: <https://homey.app/en-us/explainer/flow/> (online; accessed: 12.12.2019).
- [7] React. — Access mode: <https://reactjs.org/> (online; accessed: 19.05.2020).
- [8] Things you can do with a doorbell. — Access mode: <https://homey.app/en-us/blog/homey-doorbell-highlighted-use-case/> (online; accessed: 15.12.2019).
- [9] Vue.js. — Режим доступа: <https://ru.vuejs.org/> (дата обращения: 19.05.2020).
- [10] Vuex. — Режим доступа: <https://vuex.vuejs.org/ru/> (дата обращения: 19.05.2020).

- [11] Короткевич Ю.П., Санталова Т.Н. Прогноз развития системы «Умный дом» на российском рынке // Сборник материалов X все-российской, научно-практической конференции молодых ученых с международным участием «Россия молодая». — 2018. — Р. 42902.1–42902.4.
- [12] Кузьмина Е.В. Графическая технология управления «умной теплицей» // сайт кафедры системного программирования СПбГУ. — Access mode: <http://se.math.spbu.ru/SE/diploma/2018/pi/Kuzmina-report.pdf> (online; accessed: 12.12.2019).
- [13] Официальная страница библиотеки jQuery UI. — 2019. — Access mode: <https://jqueryui.com/> (online; accessed: 20.05.2020).
- [14] Официальная страница проекта Node-RED. — 2019. — Режим доступа: <https://nodered.org> (дата обращения: 13.12.2019).
- [15] Програмуем управление освещением по датчикам движения и освещения на Node-RED. — Режим доступа: <https://habr.com/ru/post/396985/> (дата обращения: 15.12.2019).
- [16] Среда предметно-ориентированного визуального моделирования REAL.NET / Ю.В. Литвинов, Е.В. Кузьмина, И.Ю. Небогатиков, Д.А. Алымова // СПИСОК-2017. Материалы 7-й всероссийской научной конференции по проблемам информатики. — 2017. — Р. 80–89.
- [17] Умный дом, послушный дом. — Режим доступа: <https://www.apple.com/ru/ios/home/> (дата обращения: 12.12.2019).