

КАЧЕСТВО И ИННОВАЦИИ ОБРАЗОВАНИЕ

№ 4
2015

ISSN 1999-513X

9 771999 513000



Качество • Инновации • Образование • №4-2015

КАЧЕСТВО и ИПИ (CALS)-технологии

журнал в журнале

А.Ю. Ролич

ПРОГРАММНЫЙ КОНСТРУКТОР ДИНАМИЧЕСКИХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ДЛЯ ИНТЕРНЕТ ВЕЩЕЙ

Интернет вещей или Internet of Things – концепция развития современного Интернета, подразумевающая полную интеграцию повседневных устройств и объектов в единую информационную сеть. В работе рассматривается проблема эффективного конструирования динамических пользовательских интерфейсов в рамках этой концепции. Проводится обзор и анализ систем конструирования пользовательских интерфейсов, даются определения новым типам физических объектов – «Интернет вещам» и «Web вещам». Формулируются требования к разрабатываемой системе конструирования динамических пользовательских интерфейсов для Интернета вещей. Предлагается метод (фреймворк) построения системы с наилучшей эффективностью для конструирования динамических пользовательских интерфейсов для Интернета вещей. Проводится моделирование программного конструктора динамических пользовательских интерфейсов для Интернета вещей, предлагаются модели системы. Приводятся результаты экспериментального исследования и практического применения разработанной системы.

Ключевые слова: *Интернет вещей, САПР, умные вещи, интерфейс, динамический интерфейс, пользовательский интерфейс, визуальное проектирование*

Введение

Интернет вещей (Internet of Things) – концепция развития современного Интернета, подразумевающая полную интеграцию повседневных устройств и объектов в единую информационную сеть, которая позволит им взаимодействовать и общаться между собой, обмениваясь информацией об окружающей среде, реагируя и влияя на процессы, происходящие в окружающем мире, без вмешательства человека [1-3].

Сегодня большинство существующих приборов, имеющих выход в глобальную сеть, представляют собой изолированные острова, которые не могут взаимодействовать, обмениваться информацией и использоваться совместно для достижения максимальной пользы. Причиной этого является отсутствие единых стандартов и наличие множества закрытых протоколов и технологий. Переход на Web-ориентированную платформу с применением открытых стандартов позволит использовать единые форматы представления данных и протоколы для организации взаимодействия и совместного использования устройств. В результате появляется возможность создавать мобильные, Web- или Desktop-приложения без необходимости изучения протоколов каждого отдельного устройства.

Актуальность темы заключается в том, что быстрые темпы развития Интернета вещей и появление новых видов «Интернет вещей» приводят к потребности создания новых пользовательских интерфейсов и поиску путей интеграции новых устройств с уже созданными. По данным J'son&Partners [4] к 2015 году количество «Интернет вещей» достигнет 16 млрд. устройств. Конструирование персональных интерфейсов для каждой «Интернет вещи» вручную программистами становится весьма трудоемкой задачей. Рядовой пользователь вряд ли сможет самостоятельно создавать пользовательские интерфейсы, позволяющие публиковать данные, полученные от личных «Интернет вещей», в социальных сетях или на стороннем сайте, а тем более, интегрировать вещи различных производителей.

Возникает потребность в создании визуального программного конструктора динамических пользовательских интерфейсов, способного создавать пользовательские виджеты как для новых, так и для уже имеющихся «Интернет вещей» и позволяющего существенно сократить время на создание персональных интерфейсов для «Интернет вещей».

Основная цель конструирования пользовательских интерфейсов для Интернета вещей – визуализация данных, полученных от «Интернет вещей», в реальном времени, а также представление этих данных в необходимом пользователю формате. Не менее важной задачей является предоставление возможности управления «Интернет вещами» с помощью пользовательских интерфейсов, а также конфигурировать сценарии взаимодействия «Интернет вещей» друг с другом.

Целью настоящей научно-исследовательской работы является разработка конструктора динамических пользовательских интерфейсов для Интернета вещей [5], который даст пользователям возможность самостоятельно создавать веб-виджеты для своих «Интернет вещей» и позволит повысить эффективность конструирования персональных интерфейсов за счет снижения затрат и уменьшения требований к пользователям (дружественное конструирование персональных интерфейсов). Для повышения эффективности конструирования динамических персональных интерфейсов конструктор должен базироваться на фреймворке с наибольшей производительностью во всех ведущих веб-браузерах.

Для достижения поставленной цели необходимо решение следующих задач:

- обзор и анализ существующих решений;
- исследование методов и алгоритмов решения поставленной задачи;
- разработка моделей программного конструктора динамических пользовательских интерфейсов для Интернета вещей;
- проведение эксперимента.

Объектом исследования являются системы, позволяющие конструировать пользовательские интерфейсы, а также системы для визуализации данных, поступающих от «Интернет вещей». При этом, предмет исследования состоит в анализе структуры объекта исследования, требований, методов, алгоритмов и средств его реализации.

Новизна заключается в разработке системы, которая позволит пользователям создавать и конфигурировать персональные динамические интерфейсы для новых типов физических объектов («Интернет вещей» и «Web вещей»), связывать их между собой и использовать встроенные методы визуализации данных, поступающих от новых типов физических объектов («Интернет вещей» и «Web вещей»), не имея навыков программирования.

Практическая значимость исследования заключается в применении результатов исследования на практике. Приводятся результаты практического использования методики конструирования динамических пользовательских интерфейсов для новых типов физических объектов («Интернет вещей» и «Web вещей») и рекомендации по ее использованию.

1. Обзор и анализ систем конструирования пользовательских интерфейсов

1.1. Объекты для конструирования пользовательских интерфейсов

Определение 1. «Интернет вещь» – это «Умная вещь», имеющая встроенные технологии взаимодействия с внешней (интернет) средой и уникальный идентификатор в этой среде.

Определение 2. «Web вещь» – это «Интернет вещь», имеющая возможность самостоятельно публиковать данные в социальных сетях или на других порталах.

Виджет – это небольшой независимый программный модуль, работающий в некоторой среде (сайте, браузере, мобильном телефоне) и исполняющий, как правило, одну определенную функцию.

Виджеты можно разделить на:

1. Веб-виджеты.
2. Виджеты для рабочего стола.
3. Виджеты для мобильных устройств.

Основные функции виджетов:

1. Информирование.
2. Управление и взаимодействие.
3. Реклама и маркетинг.

1.2. Средства конструирования пользовательских интерфейсов. Обзор существующих решений

В настоящее время существует довольно широкий спектр программных конструкторов пользовательских интерфейсов:

- **Конструктор сайтов Wix** [6] предоставляет качественный сервис для создания сайтов, не имеет возможность создавать отдельные интерфейсы и виджеты, в том числе и для Web вещей.

- **Платформа виджетов Яндекс.** Яндекс разработал собственную виджетную платформу: Яндекс.Виджеты [7]. Яндекс не имеет графического конструктора виджетов как такового и предоставляет специализированные библиотеки (API) для разработчиков Яндекс.Виджетов.

- Конструктор виджетов **Widgetbox Flite** [8] может создавать виджеты на HTML5 или Flash, или выбрать уже готовые. По нашему мнению, Widgetbox является одним из лучших конструкторов веб-виджетов на сегодняшний день. При этом данный инструментарий не предоставляет никакого функционала по взаимодействию с Web вещами.
- Конструктор социальных кнопок **AddThis** [9] включает в себя работу более, чем с 300 социальными сетями и закладками. Этот сервис позволяет получить виджет с социальными кнопками.
- Платформа **ThingWorx** [10] предоставляет полный дизайн приложения и интеллектуальную среду, позволяющую организациям быстро создавать приложения M2M и инновационные решения, определить взаимодействие людей, систем и интеллектуальных вещей.
- Язык **ClickScript** [11] – язык визуального программирования в браузере, с помощью которого можно писать программы для управления «Умными вещами».
- Конструктор интерфейсов для «Интернет вещей» **WoTKit** [12] – это веб-платформа для создания широкого спектра сенсорных и актуаторных сетей. На сегодняшний день WoTKit является одним из наиболее совершенных сервисов для online взаимодействия с «Интернет вещами».

Таблица 1. Сравнительный анализ аналогов

	WoTKit	ClickScript	Яндекс. Виджеты	Wix	Widgetbox	AddThis	Разрабатываемая система
Генерация кода и его выдача	-	+	+	+	+	+	+
Графический интерфейс конструктора	±	+	±	+	±	±	+
Drag and Drop технология	-	+	-	+	-	-	+
Сохранение результатов работы конструктора на сайте	+	+	+	+	+	-	+
Интеграция с социальными сервисами	-	-	+	-	+	+	+
Визуализация данных, полученных от «Интернет вещей»	+	-	-	-	-	-	+
Система управления «Интернет вещами»	±	±	-	-	-	-	+
Пользователю необходимы знания в области программирования	-	±	+	-	-	-	-

2. Выбор методов построения системы

Систему можно разрабатывать с нуля или использовать готовый инструментарий (фреймворк) с возможностью расширения ее функциональности. Для повышения эффективности построения системы необходимо разработать модели, методы и алгоритмы построения систем с расширенной функциональностью. Ниже рассмотрим имеющиеся инструментарии и проведем анализ методов, эффективности применяемых алгоритмов и возможности расширения их функциональности.

2.1. Обзор фреймворков

По сути, виджет – это локальное веб-приложение, использующее HTML, CSS и JavaScript. Поэтому рассмотрим Javascript фреймворки, которые позволят повысить эффективность конструирования динамических пользовательских виджетов.

jQuery UI [13] – это библиотека на основе jQuery, реализующая более 20 плагинов, среди которых плагины, организующие различное поведение (например, перетаскивание или растягивание элементов), восемь видов виджетов (такие как календарь, диалоговые окна, систему вкладок и т.д.) и анимационные эффекты. Лицензия MIT License и GNU GPL.

ExtJS [14] – фреймворк для построения веб-интерфейсов. Библиотека ExtJS написана на JavaScript и работает во всех популярных сейчас браузерах (хотя и с некоторыми различиями), предназначена для создания сложных и насыщенных интерфейсов, которые очень похожи на desktop-виджеты. Лицензии GPLv3 и коммерческая лицензия (для одного разработчика – 600\$).

Dojo Toolkit [15] – это AJAX фреймворк, развиваемый группой Dojo Fundation и нацеленный на нишу серьезных проектов, где могут потребоваться как обычные функции манипуляций с DOM, так и работа с Jabber-сервером. Лицензия: BSD и AFL.

Yahoo! UI Library (YUI) [16] – библиотека JavaScript для создания богатых интерактивными возможностями приложений или/и пользовательского интерфейса. Использует AJAX, анимацию,

надстройки над XMLHttpRequest и DOM, Drag&Drop, слайдеры, слайды, календари, деревья и другие компоненты, составляющие понятие «Web 2.0». Имеется собственное API для работы с JavaScript и подробная документация с примерами. Лицензия BSD.

2.2. Тестирование производительности фреймворков

Для тестирования производительности фреймворков был выбран тест SlickSpeed по выбору CSS селекторов. Количество тестов для каждого браузера – 30.

Тестирование производилось в пяти браузерах:

- Google Chrome 27.0.1453.110
- Internet Explorer 9
- Opera 12.15
- Mozilla Firefox 14.0.1
- Safari 5.1.7

Тестирование показало, что наилучшей производительностью в большинстве браузеров (Google Chrome, Mozilla Firefox, Opera, Safari) обладают фреймворки ExtJS 4, YUI 3.10 и jQuery 1.9.1. При работе с данными браузерами Dojo 1.9 показал себя несколько слабее. Однако тестирование, проведенное в браузере Internet Explorer 9, показало, что фреймворки YUI 3.10 и jQuery 1.9.1 показывают достаточно малые показатели производительности (время выбора селектора достаточно велико). Количественные данные можно видеть в таблице 2. Это среднее значение полученных при тестировании данных на протяжении 30 тестов.

Судя по этим данным можно сделать вывод, что фреймворки YUI 3.10 и jQuery 1.9.1 не обладают поддержкой IE9, что, в свою очередь, поднимает вопрос об их кроссбраузерности, т.к. доля пользователей IE9 еще достаточно велика. Фреймворки ExtJS 4 и Dojo 1.9 показали себя намного лучше, причем Dojo 1.9 оказался самым быстрым среди всех фреймворков для IE9.

Таблица 2. Среднее время выбора селекторов в Internet Explorer 9

ExtJS 4	YUI 3.10	Dojo 1.9	jQuery 1.9.1
69,92 мс	278,75 мс	61,75 мс	184,08 мс

Было принято решение усреднить полученные данные для всех браузеров.

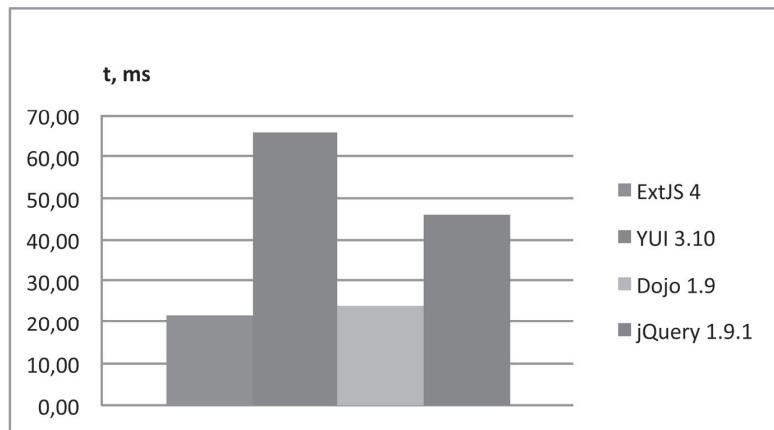


Рис. 1. Усредненное время работы селекторов для браузеров

Таким образом, наибольшей производительностью в сочетании с кроссбраузерностью обладают фреймворки ExtJS 4 и Dojo 1.9. Использование данных фреймворков позволяет разработать систему, обладающую максимальной эффективностью конструирования динамических пользовательских интерфейсов для «Интернет вещей». Главным отличием этих фреймворков является лицензирование. Поскольку Dojo 1.9 является полностью свободным программным продуктом и его функционал не зависит от типа лицензии, Dojo 1.9 является наилучшим методом (фреймворком) для построения системы.

Таблица 3. Усредненное время работы селекторов для браузеров

ExtJS 4	YUI 3.10	Dojo 1.9	jQuery 1.9.1
21,88 мс	66,04 мс	24,16 мс	46,27 мс

2.3. Выбор фреймворка

Таблица 4. Таблица альтернатив и критериев при выборе фреймворка

	jQuery UI	ExtJS	Dojo	Yahoo! UI
Лицензия	MIT License и GNU GPL	Лицензии GPLv3 и коммерческая лицензия (для одного разработчика 600\$)	BSD и AFL	BSD
Полный размер библиотеки	1.7MB	67.7MB	7.9MB	33.6 MB
Drag and Drop технология	+	+	+	+
Документация и примеры	Много	Много	Много	Средне
Широкий выбор компонент (оценка)	Удовлетв.	Отлично	Отлично	Отлично
Кросбраузерность	±	+	+	±
Производительность (рейтинг)	3	1	2	4

3. Моделирование программного конструктора динамических пользовательских интерфейсов для Интернета вещей

3.1. Концептуальная модель

Основные функции программы «Программный конструктор динамических пользовательских интерфейсов для Интернета вещей» определены на диаграмме вариантов использования (рис. 2).

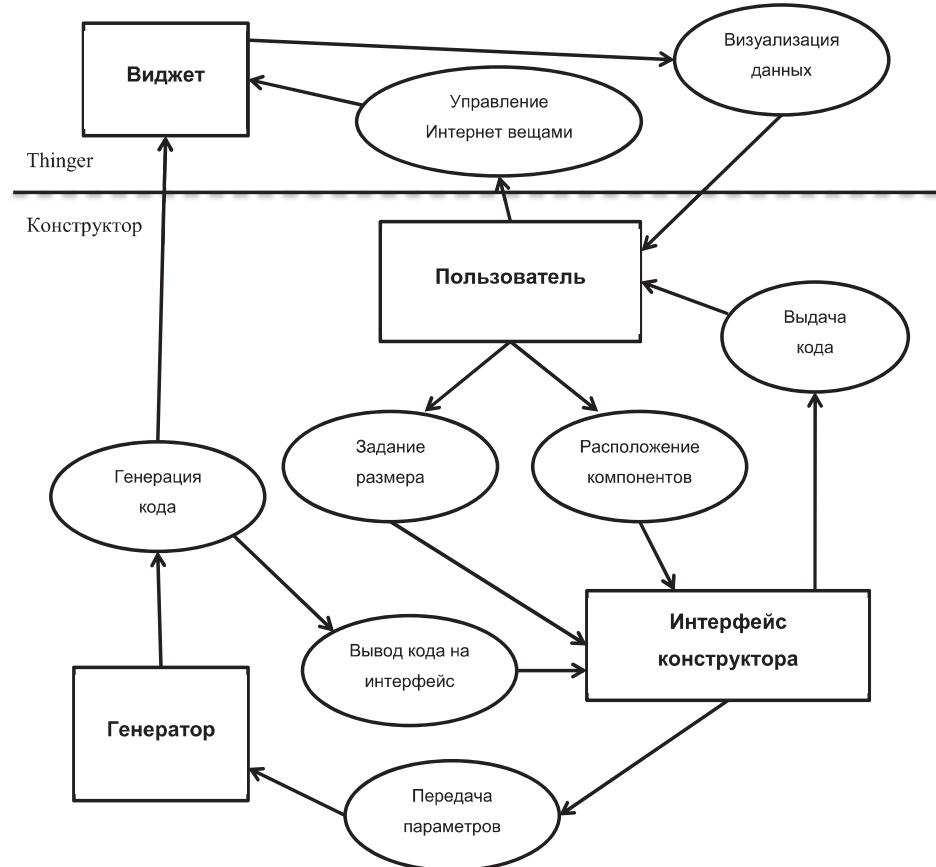


Рис. 2. Диаграмма вариантов использования программы «Программный конструктор динамических пользовательских интерфейсов для Интернета вещей»

3.2. Логическая модель

На основании данной информации была сформирована диаграмма классов концептуальной модели (рис. 3), демонстрирующая основные связи между составляющими программы.

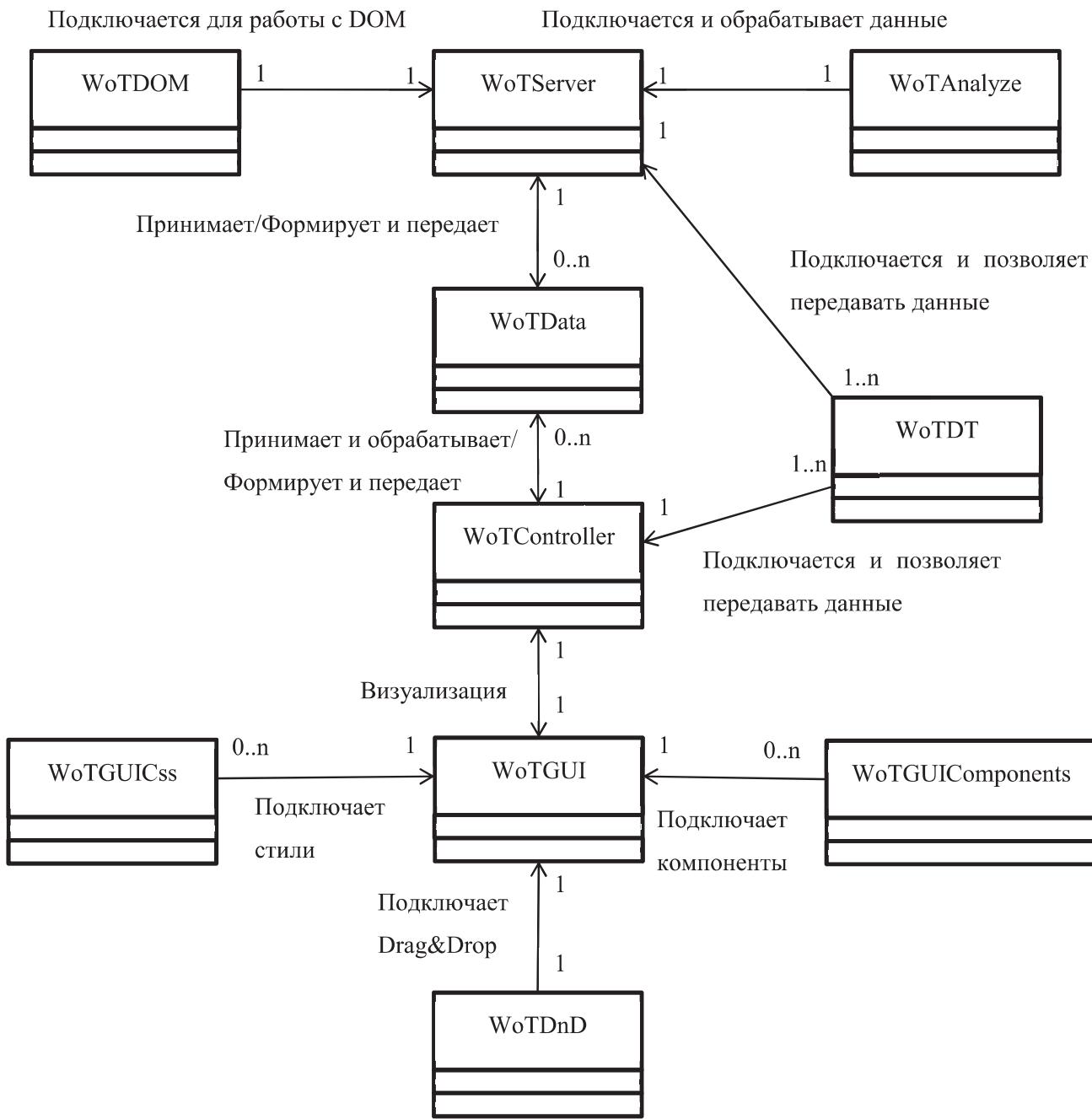


Рис. 3. Диаграмма классов концептуальной модели программы «Программный конструктор динамических пользовательских интерфейсов для Интернета вещей»

4. Проведение эксперимента

4.1. Схема работы системы

Схема работы системы представлена на рисунке 4, а схема взаимодействия программ – на рисунке 5.

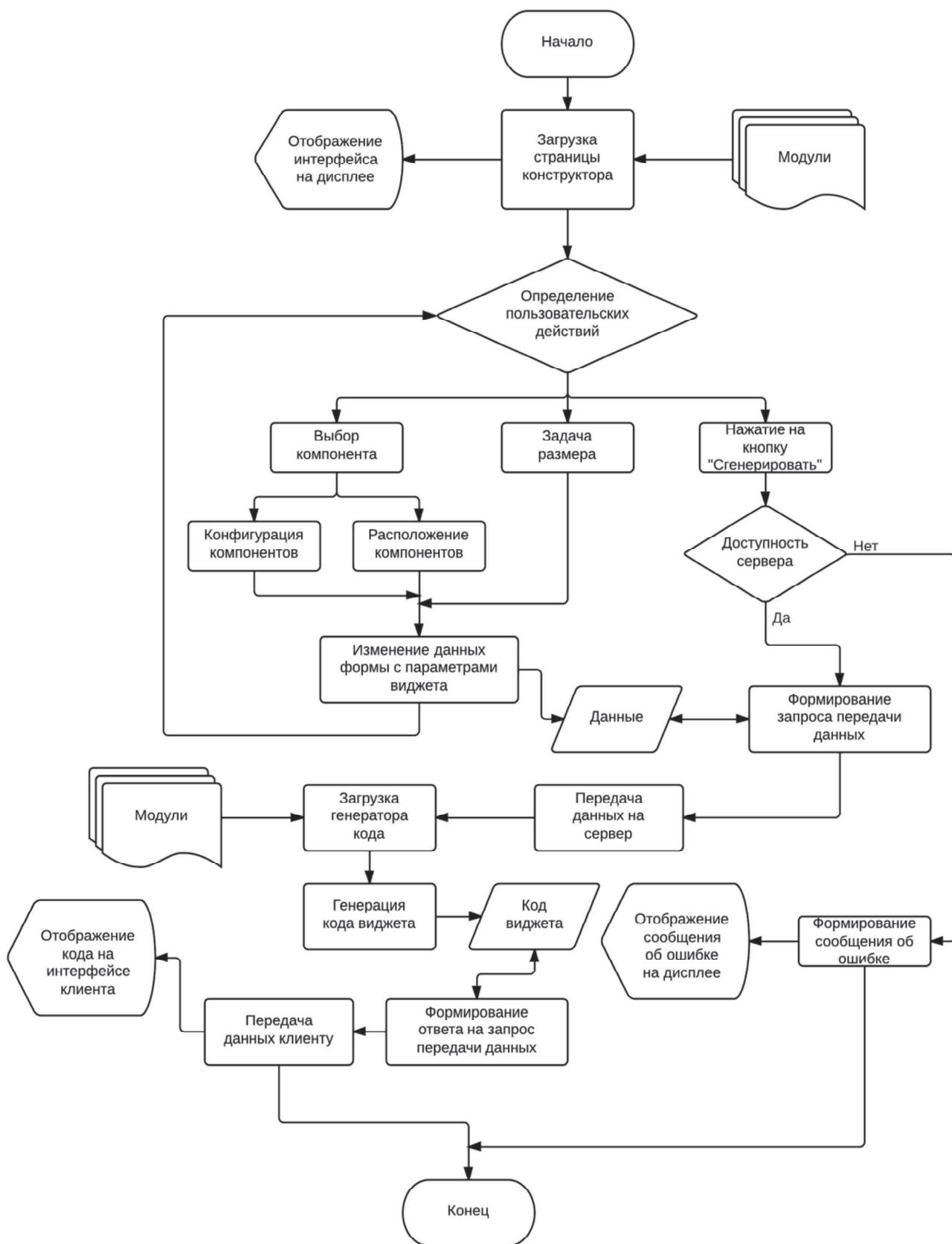


Рис. 4. Схема работы системы

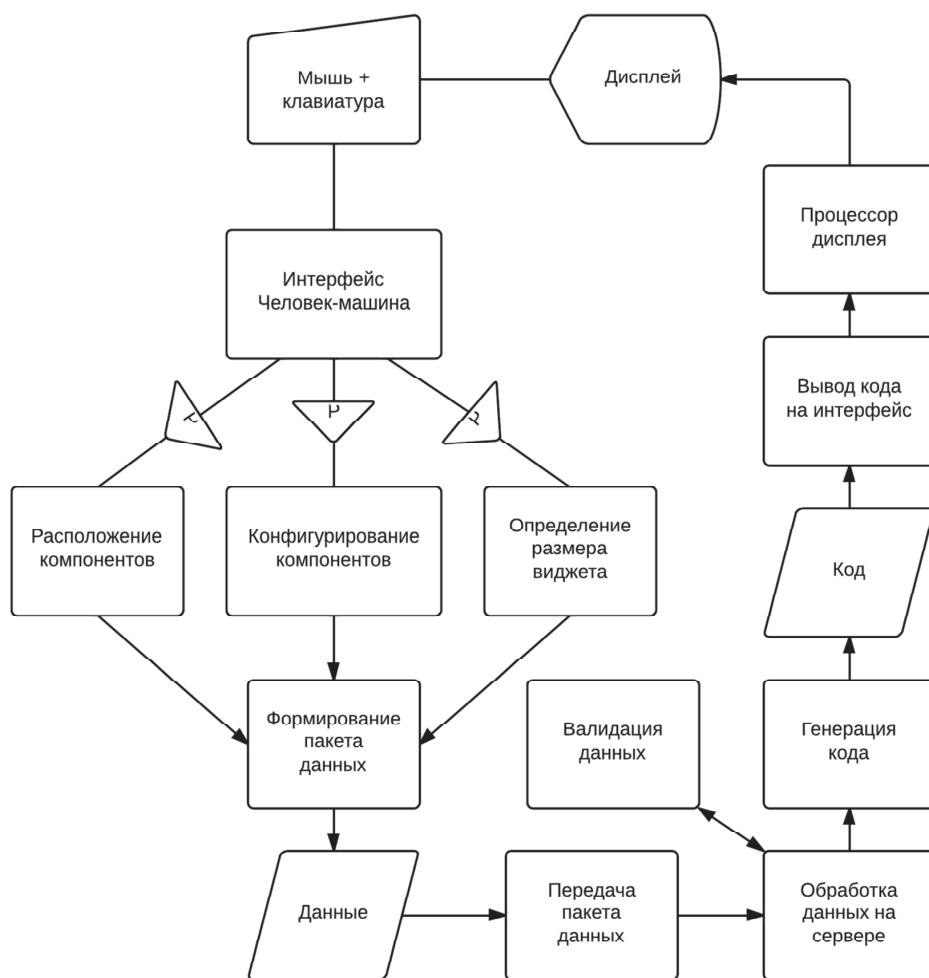


Рис. 5 Схема взаимодействия программ (P – передача управления (постоянная))

4.2. Результаты эксперимента

Одним из примеров результата работы системы служит виджет для системы персонального энергоаудита, сконструированный с помощью разработанной системы. Он представлен на рисунке 6.

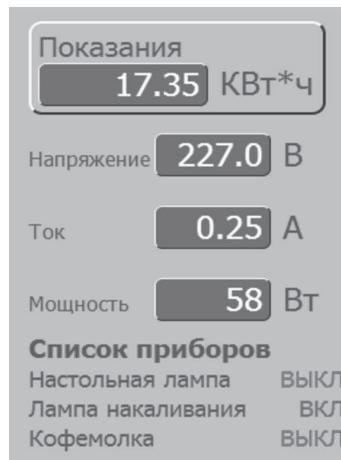


Рис. 6. Виджет для «Умного электросчетчика»

Виджет позволяет обеспечить динамическую визуализацию данных, получаемых от беспроводной системы персонального энергоаудита.

Данный виджет показывает:

- показания электросчетчика;
 - напряжение;
 - ток;
 - потребляемую мощность;
 - сведения об идентификации приборов.

Данный виджет является информером, но может быть динамически преобразован в управляющий виджет по запросу пользователя.

Заключение

Показано, что виджеты являются эффективным средством для создания динамических пользовательских интерфейсов. Они позволяют конструировать полноценные информационные порталы со свойством социальной направленности. При появлении большего количества «Интернет вещей» [2] пользователям потребуется большое количество пользовательских интерфейсов, позволяющих публиковать данные, полученные от личных «Интернет вещей», в социальных сетях или на стороннем сайте и обеспечивать взаимодействие «Интернет вещей» различных производителей.

Выявлена потребность в создании визуального программного конструктора динамических пользовательских интерфейсов, позволяющего конструировать пользовательские виджеты как для новых, так и для уже имеющихся «Интернет вещей» и позволяющего существенно сократить время на создание персональных интерфейсов для «Интернет вещей».

Проведенный обзор и анализ систем конструирования пользовательских интерфейсов позволил дать определения новым типам физических объектов – «Интернет вещами» и «Web вещам» – и сформулировать требования к новой системе конструирования динамических пользовательских интерфейсов для «Интернета вещей».

Обзор и анализ методов построения системы с использованием инструментальных средств позволил на основе анализа альтернатив и критериев при выборе метода (фреймворка) построения системы сформулировать вывод о том, что фреймворк Dojo 1.9 обладает наибольшей эффективностью конструирования динамических пользовательских интерфейсов для «Интернет вещей», что подтверждено экспериментальными исследованиями.

В результате моделирования программного конструктора динамических пользовательских интерфейсов для «Интернета вещей» была разработана концептуальная модель построения динамических пользовательских интерфейсов для «Интернета вещей» и предложена логическая модель, позволяющая разработать алгоритмы построения динамических пользовательских интерфейсов для «Интернет вещей».

Результаты эксперимента подтвердили эффективность предложенных методов, моделей и алгоритмов. Приводятся результаты практического использования методики конструирования динамических пользовательских интерфейсов для «Интернета вещей» и рекомендации по ее использованию.

Был разработан конструктор динамических пользовательских интерфейсов для «Интернета вещей», позволяющий пользователям самостоятельно создавать динамические пользовательские интерфейсы для «Интернет вещей» не имея навыков программирования, что позволяет повысить эффективность конструирования персональных интерфейсов за счет снижения затрат и уменьшения требований к пользователям (дружественное конструирование персональных интерфейсов).

Данное научное исследование (исследовательский проект № 14-05-0064) выполняется при поддержке Программы «Научный фонд НИУ ВШЭ» в 2015 г.

Список литературы

1. Восков Л.С., Пилипенко Н.А. Web вещей – новый этап развития интернета вещей // Качество. Инновации. Образование. 2013. № 2. С. 44-49.
 2. Efremov S., Pilipenko N., Voskov L. An integrated approach to common problems in the Internet of Things // in: Procedia Engineering, Vol. 100: 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014. PP. 1215-1223.

3. Ефремов С.Г., Пилипенко Н.А., Восков Л.С. Web Вещей: проблемы поиска и взаимодействия устройств // Качество. Инновации. Образование. 2014. № 12. С. 88-96.
4. J'son&Partners «Интернет вещей и межмашинные коммуникации. Обзор ситуации в России и мире» [Электронный ресурс]. URL: http://www.json.ru/poleznye_materialy/free_market_watches/analytics/internet_vewej_i_mezhmashinnye_kommunikacii_obzor_situacii_v_rossii_i_mire/ (дата обращения 01.03.15).
5. Ролич А.Ю. Конструктор пользовательских интерфейсов для Web'a Вещей // В кн.: «Новые информационные технологии». Тезисы докладов XXI Международной студенческой школы-семинара. – М.: МИЭМ НИУ ВШЭ, 2013. – 318 с. (С. 172-175).
6. Яндекс.Виджеты [Электронный ресурс]. URL: <http://wdgt.yandex.ru/widgets/> (дата обращения 01.03.15).
7. Конструктор сайтов Wix [Электронный ресурс]. URL: <http://ru.wix.com/> (дата обращения 01.03.15).
8. Конструктор виджетов WidgetBox [Электронный ресурс]. URL: <http://www.widgetbox.com/> (дата обращения 01.03.15).
9. Конструктор социальных кнопок AddThis [Электронный ресурс]. URL: <http://www.addthis.com/> (дата обращения 01.03.15).
10. ThingWorx – программная платформа, позволяющая объединять людей, системы и устройства [Электронный ресурс]. URL: <http://www.thingworx.com> (дата обращения 01.03.15).
11. ClickScript [Электронный ресурс]. URL: <http://clickscript.ch/site/home.php> (дата обращения 01.03.15).
12. WoTKit [Электронный ресурс]. URL: <http://wotkit.sensetecnic.com> (дата обращения 01.03.15).
13. jQuery UI [Электронный ресурс]. URL: <http://jqueryui.com/> (дата обращения 01.03.15).
14. ExtJS [Электронный ресурс]. URL: <http://www.sencha.com/products/extjs> (дата обращения 01.03.15).
15. Dojo [Электронный ресурс]. URL: <http://dojotoolkit.org/> (дата обращения 01.03.15).
16. YUI [Электронный ресурс]. URL: <http://yuilibrary.com/> (дата обращения 01.03.15).

Ролич Алексей Юрьевич,
магистрант 2 года обучения.

Национальный исследовательский университет «Высшая школа экономики».
123458, Москва, ул. Таллинская, д. 34.

A.Yu. Rolich

A SOFTWARE DYNAMIC USER INTERFACE TOOLKIT FOR SMART THINGS

Internet of Things – concept of the modern Internet, which implies the full integration of everyday objects and devices into a single information network. This paper considers the problem of designing an efficient dynamic user interfaces within this concept. A review and analysis of systems design user interfaces, provides definitions of new types of physical objects "Internet things" and "Web things." Formulated requirements for the system being developed design dynamic user interfaces for the Internet things. Proposed a method (framework) of construction of the system with the best performance for the construction of dynamic user interfaces for the Internet things. Conducted simulation software designer dynamic user interfaces for the Internet things, offers a model system. Proposed the results of experimental research and the practical application of the developed system.

Keywords: *Internet of Things, CAD, Smart things, interface, dynamic interface, user interface, visual design*

References

1. Voskov L.S., Pilipenko N.A. Web veshhej – novyj jetap razvitiya interneta veshhej // Kachestvo. Innovacii. Obrazovanie. 2013. № 2. P. 44-49.
2. Sergey Efremov, Nikolai Pilipenko, Leonid Voskov. An integrated approach to common problems in the Internet of Things, in: Procedia Engineering Vol. 100: 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014. , 2015. P. 1215-1223.