



УДК 004.822:514

«УМНЫЙ ДОМ», УПРАВЛЯЕМЫЙ ЗНАНИЯМИ

Корончик Д. Н.

** Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь
denis.koronchik@gmail.com*

В работе приводится описание подхода к реализации «Умного дома» на основе технологии OSTIS, который дает целый ряд преимуществ, в сравнении с существующими подходами. Суть описываемого подхода заключается в том, что устройства в доме управляются знаниями.

Ключевые слова: умный дом, интернет вещей, виртуальная модель, управление знаниями

Введение

Количество устройств подключенных к интернету растет большими темпами. По некоторым оценкам количество таких устройств в 2014 году составило около 8 миллиардов, а к 2018 году их количество превысит 18 миллиардов [forecastjoy, 2014]. В связи с этим все чаще стала упоминаться концепция «интернет вещей» (ИВ) [wp, 2015]. Суть данной концепции заключается в том, что все устройства (физические или виртуальные объекты) связаны между собой в одну большую вычислительную сеть, в которой они решают различные задачи, взаимодействуя друг с другом. Работу в этом направлении ведут некоторые компании. К примеру, компания Google инициировала открытый проект «Physical Web» [Physical Web, 2014].

Одним из способов реализации «Интернета вещей» является подход на основе агентов и базы знаний [hb, 2015]. Технология OSTIS изначально ориентирована на обработку знаний в базе знаний с помощью агентов, что в свою очередь позволяет реализовать описанный в [hb, 2015] подход к построению «Интернета вещей».

Частным видом интернета вещей может выступать «Умный дом». В данной статье будут рассмотрены наработки проекта OSTIS, которые могут быть использованы для реализации «Интернета вещей» на примере реализации виртуальной модели «Умного дома».

1. Существующие подходы

Существующие подходы к построению «Умного дома» имеют целый ряд недостатков:

- сложность встраивания новых устройств в инфраструктуру «Умного дома». Проблема заключается в том, что нет единого стандарта, а также подключение устройств может отнимать много времени и средств (так как зачастую это требует специализированных навыков);
- зачастую технологии, созданные для создания «Умных домов», позволяют решать узкий спектр задач (заложенных в них разработчиками). Основную роль при решении задач выполняет человек;
- расширение функционала «Умного дома», сводится к программированию необходимых паттернов поведения, что требует достаточного уровня знаний (хотя есть проекты в которых это сведено к минимуму, за счет использования визуальных языков программирования [majordomo, 2015]).

2. Архитектура

Как и любая система, построенная с использованием технологии OSTIS, «Умный дом» состоит из 3-х основных подсистем: базы знаний, машины обработки знаний и пользовательского интерфейса.

В базе знаний состоит из следующих разделов:

- описание моделей устройств (предметная область / онтология устройств). В данном разделе описываются различные устройства и их модели (указывается производитель, класс устройства, потребляемая мощность и другие его характеристики);
- структура дома. В данном разделе указываются устройства, которые относятся к конкретному дому, их модели, положение в доме и

т. д. Благодаря тому, что системе известны модели устройств и их полное описание, у нее появляется возможность отвечать на различные вопросы. Примеры возможных вопросов:

- каким образом можно подключить одно устройство к другому;
- как пользоваться теми или иными функциями устройства;
- когда необходимо провести техническое обслуживание устройства;
- и т. д.

Показатели жизнедеятельности дома. В данном разделе присутствуют показания с различных датчиков дома и состояние устройств.

Кроме перечисленных выше в базе знаний имеются разделы описывающие машину обработки знаний и пользовательский интерфейс.

Машина обработки знаний «Умного дома» состоит из множества агентов, которые в основном являются частью устройств. Схема взаимодействия устройств с базой знаний показана на рисунке 1.



Рисунок 1 - Схема взаимодействия устройств с базой знаний

Чтобы показать каким образом происходит управление устройствами через базу знаний, рассмотрим простой пример: включение (выключение) устройства. Каждое устройство имеет два агента. Один из этих агентов отвечает за включение устройства и условием его инициирования является добавление устройства во множество включенных устройств (рисунок 2).



Рисунок 2 - Пример включенного устройства

Второй агент отвечает за выключение устройства. Его условие инициирования - это удаление устройства из множества включенных устройств.

Разные устройства содержат разный набор таких агентов. К примеру, телевизор может содержать агентов, которые отслеживают уровень звука,

текущую программу, текущий источник сигнала и т. д. Агенты могут быть реализованы как аппаратным так программным способами.

Помимо агентов, которые отвечают за управление устройствами, машина обработки знаний содержит в себе целый набор агентов, которые отвечают за решение задач и ответа на вопросы пользователя. Примерами таких агентов могут быть:

- агент поиска рецепта блюда по ингредиентам, которые имеются в холодильнике;
- определения способа подключения одного устройства к другому;
- получение прогноза погоды из интернета;
- и т. д.

Пользовательский интерфейс для системы «Умного дома» является достаточно сложной частью с точки зрения реализации. Основным способом взаимодействия пользователя с системой является речевой интерфейс. После распознавания команды, в базе знаний формируется команда или вопрос с аргументами, на что реагирует соответствующий агент, которые выполняет данную команду. Пример команды включения устройства представлен на рисунке 3.

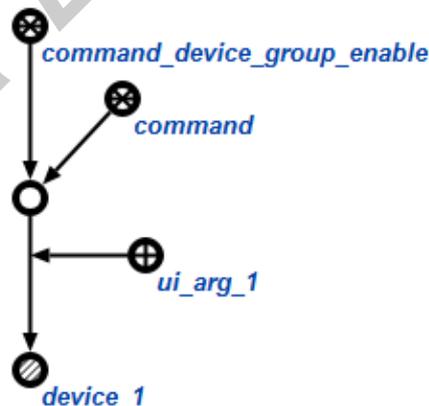


Рисунок 3 - Пример команды включения устройства

В результате инициирования команды (путем добавления её во множество инициированных команд), устройство `device_1` будет добавлено во множество включенных устройств (`device_enabled`). Это приведет в действие агента, который отвечает за включение устройства и оно включится. На этом работа пользовательского интерфейса не завершена. Чтобы дать обратную связь пользователю, для каждой команды заданы текстовые шаблоны ответов, которые используются для синтеза речи (рисунок 4).

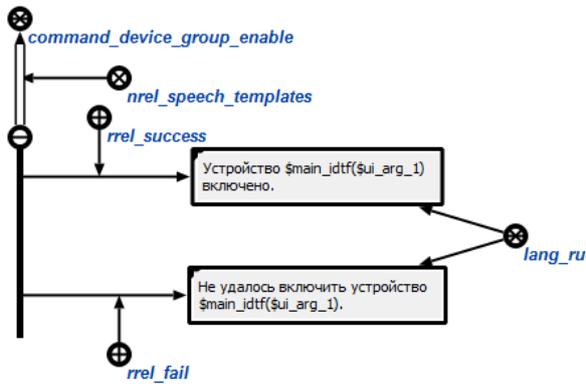


Рисунок 4 - Пример текстовых шаблонов

Для каждой завершенной команды (вопроса) инициируется команда генерации текстового ответа (рисунок 5). Где первым аргументом является команда, на результат которой необходимо сгенерировать текст. Второй аргумент указывает на язык, который необходимо использовать для генерации текста. Третий аргумент указывает на атрибут, который необходимо использовать для выбора шаблона (если команда завершена успешно, то *rrel_success*, иначе *rrel_fail*).

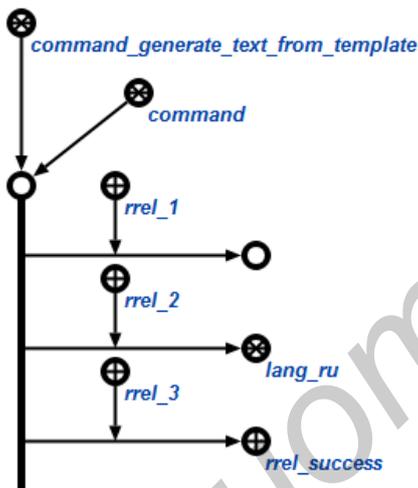


Рисунок 5 - Пример команды генерации текста по шаблону

В результате успешного выполнения команды, генерируется sc-ссылка, содержимым которого является сгенерированный текст. Этот текст подается на вход генератору речи, которая проигрывается пользователю.

Помимо речевого интерфейса могут быть реализованы и другие интерфейсы, например графический или какие-то другие.

3. Реализация

Реализация прототипа данной системы была осуществлена на языке программирования C++, с использованием Unreal Engine 4 [ue, 2015] в качестве программной платформы, в которой была реализована виртуальная модель дома с устройствами (рисунок 6).



Рисунок 6 - Реализованная виртуальная модель дома

Для взаимодействия виртуальных устройств с базой знаний был реализован сетевой клиент на C++. Помимо виртуальной модели дома было реализовано приложение для Android, на Java. Оно реализует речевой ввод и вывод. Общая схема взаимодействия между реализованными компонентами и приложениями представлена на рисунке 7.



Рисунок 7 - Схема взаимодействия реализованных приложений

Распознавание голосовых команд реализовано с помощью стандартных средств Android. Распознанный текст обрабатывается с помощью сервиса api.ai [api.ai, 2015]. В результате обработки запроса, данный сервис возвращает JSON следующего вида:

```
{
  "id": "2f3bc87a-a875-423f-a5c9-aeb597a55ff9",
  "timestamp": "2016-01-28T21:09:52.053Z",
  "result": {
    "source": "agent",
    "resolvedQuery": "включи телевизор",
    "action": "ui_menu_iot_device_group_enable",
    "actionIncomplete": false,
    "parameters": {
      "ui_arg_1": "group_volume_tv"
    }
  },
  "contexts": [],
}
```

```

"metadata": {
  "intentId": "cff78e8d-dbf8-47a4-ba23-1a5e739d17f3",
  "intentName": "включить"
},
"fulfillment": {
  "speech": "Включаю"
}
},
"status": {
  "code": 200,
  "errorType": "success"
}
}

```

На основе полей *action* и *parameters*, генерируется команда в базе знаний и добавляется во множество инициированных команд. Далее происходит ожидание, пока инициированная команда будет завершена, после чего генерируется и иницируется команда генерации текстового ответа по шаблону. На основе результата этой команды происходит синтез речи (с помощью стандартных средств Android).

В настоящий момент в модели реализованы следующие виды устройств: лампочка, телевизор, холодильник. Каждое из этих устройств имеет агенты, которые отвечают за их включение и выключение.

Заключение

В результате работы был реализован прототип «Умного дома», который управляется базой знаний. Имеющаяся реализация позволяет расширять функционал виртуальной модели, за счет добавления новых агентов для решения задач, за счет добавление новых устройств. В дальнейшем, виртуальные устройства могут быть заменены на реальные, а имеющаяся система может работать на небольшом компьютере (Raspberry Pi) и управлять целым домом.

Реализованный подход дает некоторые преимущества:

- благодаря базе знаний, система имеет представление, о том какие возможности есть у дома в целом. Это позволяет решать задачи управления домом более эффективно. Помимо того, в будущем, онтология устройств может быть общей для всего мира и пополняться производителями как wikipedia;
- так как система построена на основе агентов, то это значительно упрощает интеграцию устройства в среду «Умного дома». В конечном итоге это сводится лишь к подключению питания к нему и автоматической регистрации устройства.

Библиографический список

[forecastjoy, 2014] Forecast joy [Электронный ресурс]. – 2014 – Режим доступа: <http://forecastjoy.com/tag/internet-of-things/> – Дата доступа: 02.12.2015

[Physical Web, 2014] Сайт проекта Physical Web [Электронный ресурс]. – 2014 – Режим доступа: <http://google.github.io/physical-web/> – Дата доступа: 30.11.2015

[wp, 2015] Статья wikipedia: Интернет вещей [Электронный ресурс]. – 2014 – Режим доступа: https://ru.wikipedia.org/wiki/Интернет_вещей – Дата доступа: 20.11.2015

[hb, 2015] Статья: Умный интернет вещей - кто он и с чем его едят [Электронный ресурс]. - 2015 - Режим доступа: <http://habrahabr.ru/post/259243/> - Дата доступа: 20.11.2015

[majordomo, 2015] MajorDoMo - умный дом и домашняя автоматика своими руками [Электронный ресурс]. - 2015 - Режим доступа: <http://majordomo.smartliving.ru/Main/HomePage> - Дата доступа: 10.11.2015

[ue, 2015] Unreal Engine Technology [Электронный ресурс]. - 2015 - Режим доступа: <https://www.unrealengine.com> - Дата доступа: 20.11.2015

[api.ai, 2015] Free Speech-to-text, Intent Recognition and NLP for IoT [Электронный ресурс]. -2015 - Режим доступа: <https://api.ai/> - Дата доступа: 22.11.2015

SMART HOME, THAT CONTROLS BY KNOWLEDGES

Koronchik D. N

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

denis.koronchik@gmail.com

Article proof of concept of smart home, based on knowledge base and implemented as agent-based. It contains information about simple agents workflow and principles of communication between them.

Introduction

Smart home is a small part of internet of things. There are no any technology that allows to create really intelligent systems to control home. Current state of existing technologies allows just to automate some home functionality. In this work made attempt to implement smart home model based on OSTIS technology. That allows to resolve some problems.

Main Part

Implemented model of smart home controlled by knowledge base. That means, that each device has set of agents, that start working by emitting events from knowledge base. Knowledge base contains information about each device, so system knows all available functionality and use it. This approach allows system to resolve more complex tasks, like an answer to question "What I can cook from food, that I have in my refrigerator?" or "Where I can to see news?" and etc.

Conclusion

Implemented model (with Unreal Engine 4, C++) shows, that described approach gives more opportunities, then existing technologies, to make smart home more intelligent. Implemented virtual devices can be replaced by real devices. Also speech interface was implemented by using Android and Java. It allows to run commands by speech.