

Remote Control Architecture of Actuators Based on Voice Recognition Using Object Virtualization in CoT Network

Sehrish Malik¹, Do-Hyeun Kim^{*2}

^{1,2}Department of Computer Engineering, Jeju National University, Jeju-do, 63243, Republic of Korea

*Corresponding author E-mail: kimdh@jeju.ac.kr

Abstract

Background/Objectives: The concept of virtualization in cloud computing is gaining significant importance now a days; in smart IoT environments. The IoT environments consist of two main networks as sensors network and actuators networks.

Methods/Statistical analysis: In this work, we propose architecture for remote control of actuators networks with the help of a virtual assistant (VA) using voice recognition and cloud based virtualization of smart environment.

Findings: We create custom virtual assistant skills for our developed IoT platform and demonstrate how user can remotely control the actuators via system services such as querying the system, querying a web service, and getting response in both visual and spoken form.

Improvements/Applications: Developed system provides quick virtual assistant to the smart home users; keeping the interactions within context. It improves the context learning of the VA in smart spaces.

Keywords: CoT(Cloud of Things); IoT(Internet of Things); virtualization; actuator platform; actuator middleware; actuator control;

1. Introduction

The number of connected devices is expected to reach 24 billion by 2020 [1]. With this increase in connected devices; rapid increase in data is also seen, highlighting the right data storage as one of the vital issues. The appropriate processing power and understanding of data into meaningful information to the user is required; which is hard to achieve at IoT level with such large numbers of data. Hence, the solution lies in the integration of cloud computing resources into IoT named as CoT (Cloud of Things) [2].

Virtualization is identical representation of physical objects into digital world, also referred as digitalization. It is an abstraction of the physical system into the software system to look and behave in a similar manner but with added benefits such as connectivity, exposure, flexibility, scalability, reliability, minimum cost and application ease [3]. IoT devices in the smart network are represented as virtual objects. These IoT devices are either sensor objects or actuator objects. Data is sensed from the sensor objects and control decisions are made based on sensed readings. These objects are translated into virtual objects in virtualized system. In order to emulate different control scenarios; multiple virtual objects are combined into single logic known as CVO (Composite

Virtual Objects). The composite virtual objects expose the services of the smart IoT objects along with independent services. Multiple virtual objects can also be integrated into VB (Virtual Blocks) in order to combine offered services logic [4].

An IoT platform represents the sensor and actuator network system for a smart environment. An ideal IoT platform consists of five layers as business layer, application layer, middleware layer, network layer and perception layer. The perception layer perceives the data from the environment. Network layer collects the data from the lower layer and send it to the network. Middleware layer provides the service management and data storage after receiving the data from lower layer. Application layer presents the data into possible applications as smart home, smart city etc. Business layer models the data into meaning service to sell it.

Intelligent Virtual Assistants (IVAs) are becoming vital part of smart environments. The smart environments users prefer voice commands over writing the text command, due to ease and flexibility. The IVAs working involves the user asking a voice query to the IVA, on other end the IVA receives the voice query and converts it into text and parses the message for understanding the question and replies accordingly. In such systems, the accurate voice recognition at IVA's end is mandatory as many queries can be misinterpreted.

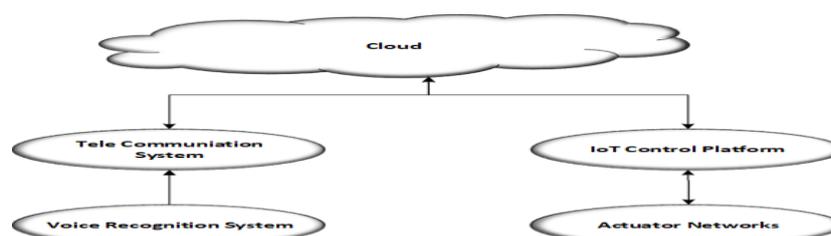


Figure 1: Remote Control of Actuators in CoT Networks

This work is an extended version of the previous work published in ICCT 2018 [5]. Figure 1 shows the conceptual diagram of the presented work. We have developed a virtualized IoT indoor system with sensor and actuator networks. In this work we primarily focus on smart control on actuator networks.

The rest of the paper is organized as follows, section 2 presents related work. Section 3 illustrates the proposed system design and architecture. In section 4, we present the environment and implementation details along with the results while section 5 concludes the paper.

2. Related Work

The presented work in [4], highlights the importance of virtual objects and their services and proposes a design and architecture for the composite virtual objects (CVO) in Web of Objects (WoO) for IoT environments. The proposed design intends to compose services for virtual objects and to create cooperation among the real world objects in order to build then intelligent service features.

Senaas is an event driven sensor virtualization approach proposed for IoT using cloud computing. It supports the IoT objects by providing a semantic overlay on IoT cloud logic. It addresses the issues of connectivity, security and reliability. It also provides the dynamic sensing and responses for resolving the monitoring issues too [6].

The research works proposed in [7] and [8] aim to develop a virtualized IoT environment system. They propose creation of new services and applications using virtual representation of real world objects. The research and results show that these projects have

resolved issues such as heterogeneity, flexibility, reliability, scalability and capability.

The work presented in [9] explores the applications of the Internet of Things and cloud computing. It also proposed a cloud computing and IoT based cloud manufacturing service system for on-demand use, efficient sharing of resources, full sharing, optimal allocation of resources and possible applications of the technologies.

A CoT based smart city IoT system's survey is presented in [10]. The work discusses IoT platforms, services involved in cloud of things, smart city requirements and integration of IoT and cloud for developing a CoT-based smart city platform. The research works carries in [11] and [12] review the literature regarding the integration of IoT and cloud along with analyzing the benefits out of it and possible dimensions along with issues and challenges faced. These works discuss the implementation of the cloud into smart environments e.g. smart city and also highlight the issues involved[15,16].

3. Proposed Remote Control Architecture of Actuators based on Voice Recognition

In this section, we present the design of our proposed approach for virtualization of IoT platform. Figure 2 shows the architecture diagram of the system. The basic architecture consists of a voice recognition system, telecommunication system, actuator middleware, actuator control platform, and virtualization toolbox and cloud platform.

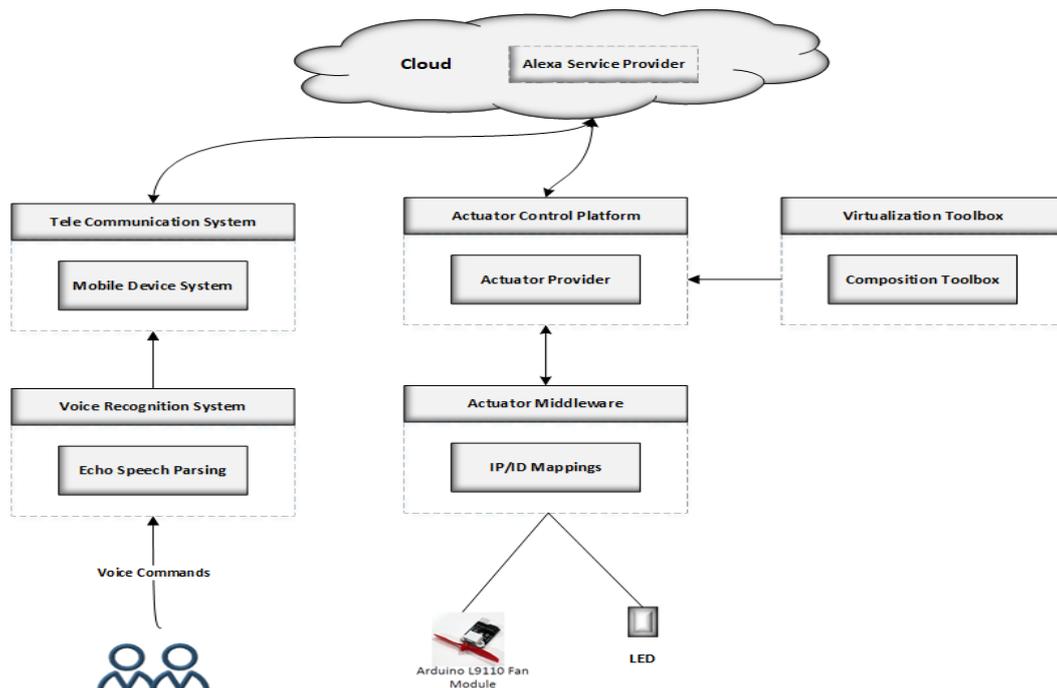


Figure 1: Remote Control of Actuators based on Voice Recognition using IoT platform and smart phone in CoT Networks

At one end, user speech is passed onto a voice recognition device. The voice commands are then forwarded to the voice recognition system which has the speech parsing logic implemented. The Voice recognition system parses the speech and takes out the meaningful information from the speech to be processed further. At telecommunication system we have our system installed for the user for easy access and visualization, so that the user can see the parsed sentence intent and its understood action to be taken by the system. After understanding and parsing the user command it sends the systems understood command to the Telecommunication

system module. In telecommunication system module, the parsed meaningful text is displayed at the user level and passed onto the cloud too. At cloud we have the service provider; logic implemented along with cloud based virtualization of our IoT platform.

The actuator provider provides the business logic for the actuators involved in the actuator platform. Actuator middleware performs the ID/IP mapping of the actuators from network layer in order to create virtual objects at upper level. The virtualization toolbox provides the business logic for the virtualization of the actuators

network from physical world into virtual world. The actuator platform checks the current status of the targeted actuator, checks whether control command can be performed and passes the control command to the targeted actuator via actuator middleware which is an intermediary between devices network and IoT control platform.

The speech recognizer module has three basic sub modules as NLU engine, text parser and speech synthesizer. The NLU engine helps in intelligent voice recognition and understanding of natural language. The text parser parses and matches the intent and the

utterances of the user while the speech synthesizer is used to return the responses.

The four states of virtual assistant are idle, busy, expecting speech and recognizing. Figure 3 below shows the state diagram of virtual assistant module in IoT platform. When no speech is expected then the VA is in idle state, when the VA is invoked by the user then it is in expecting speech state; when the user is speaking then the VA is in recognizing state and when its processing the speech then it is in busy state.

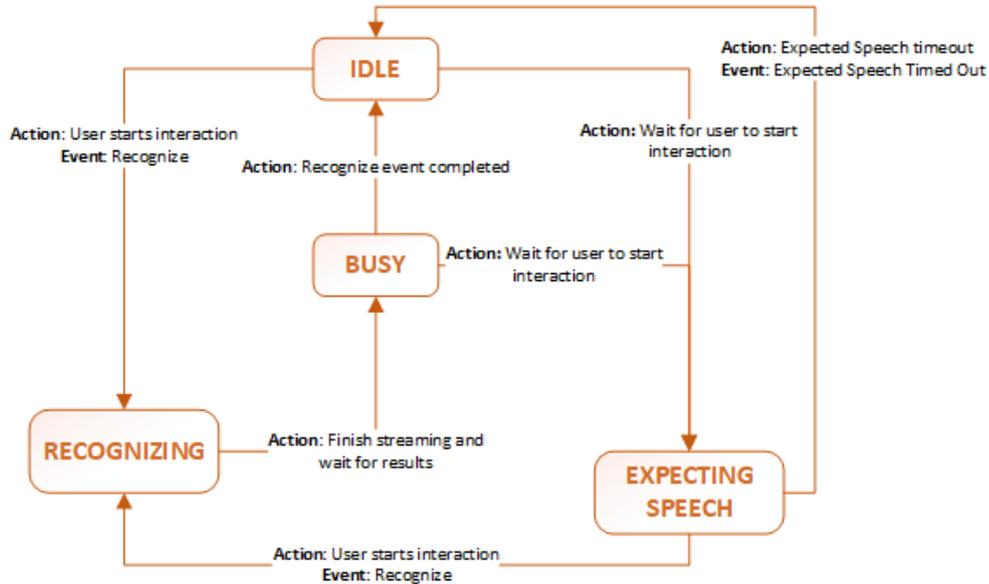


Figure 2: Virtual Assistant Module State Diagram

4. Implementation Results and Discussion

The implementation environment for our system is described in table 1.

Table 1: Implementation Environment

Components	Version
Operating System	Microsoft Windows 10 pro (x64)
Microsoft Visual Studio	2015
Microsoft SQL Server Management Studio	2016
Intel System Studio IoT Edition	2017
Echo Dot	Generation 2
Yocto Linux	2014
Intel Edison with Kit for Arduino	2014

The implementation environment for our system is described in table 1. For actuator platform and middleware implementation, we have used Windows 10 pro (x64) as the operating system, Visual Studio 2015 as development tool and Microsoft SQL Server as database management system. For actuator network connection testing, actuator emulators are built on Microsoft visual studio and the testing connection of actuator with Edison board is also done using fan and LED modules on Intel Edison with Kit for Arduino. Edison board will have micro CoAP controllers, Lib CoAP of C version, and Yocto and Linux environment. Alexa Echo generation 2 is used as virtual assistant.

4.1 Virtual Device Manager

Virtual device manager allows managing devices virtually as adding a device, deleting a device or updating a device. Each device has its related information such as URI, type, location and properties. URI is unique resource identifier for each device,

type tells kind of device, location shows where the device is located, properties show what basic function device can perform (Figure 4).

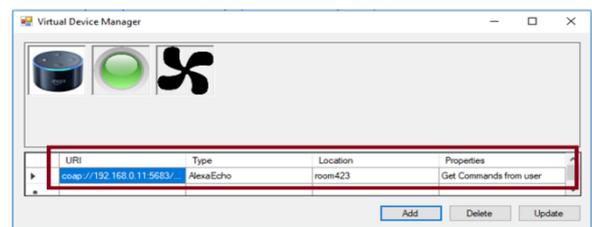


Figure 4: Virtual Device Manager

4.2 Visual Service Composer

Visual Service Composer allows composing the services virtually and defines the service flow to and from the virtual assistant to the actuator platform (Figure 5). In the service flow to the actuator platform from the virtual assistant the Alexa's recognized voice is passed onto NLU logic unit. Then text converted is given input to parser. The parser takes out the meaning from the data and forwards the command to destined actuator units.

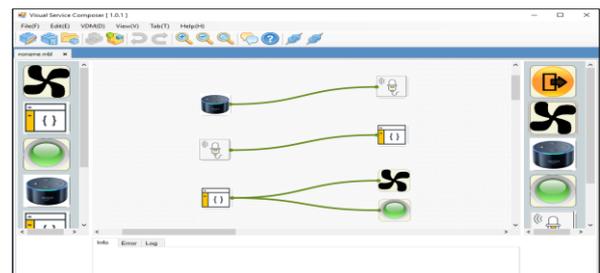


Figure 5: Virtual Device Composer

In the service response flow from the actuator platform to the virtual assistant; the status query result is returned from the actuators to the text parser and managing unit. The unit converts the result into spoken sentence format and passes to NLU. NLU converts it to the voice and passes to Alexa echo.

4.3 Custom Skill Implementation

When designing and building a custom skill, you create a set of intents, a set of sample utterances, an invocation name, a cloud based service and configurations. Virtual Assistant's Cloud based APIs implementation uses following main controller functions as shown in table 2. Using these controller functions we have defined new skill sets and deploy them for Alexa.

Table 2: VA's Controller Functions

Function Name	Description
OnSessionStartedAsync	run when a user invokes your custom skill
OnSessionEndedAsync	run when a user ends their session with your custom skill
OnLaunchAsync	run when a user invokes your custom skill without an intent
OnIntentAsync	run when a user invokes your custom skill with a specific intent

```

Schema: <No Schema Selected>
1  {
2  "intents": [
3  {
4  "intent": "FanStatusIntent"
5  },
6  {
7  "intent": "LightStatusIntent"
8  }
9  ]
10 }
    
```

Figure 6: Alexa Intent Schema

Intent corresponds to a feature of the skill. In our case, our intent is to check the status of light and fan. So we have added two intents one for fan and one for light. (Figure 6)

```

ce.resx  AlexIntentSchema.json*  AlexaSampleUttera
1  LightStatusIntent about light status
2  LightStatusIntent is light switched on
3  LightStatusIntent what is the status of light power
4  LightStatusIntent what is the status of light
5  FanStatusIntent about fan status
6  FanStatusIntent is fan switched on
7  FanStatusIntent what is the status of fan power
8  FanStatusIntent what is the status of fan
    
```

Figure 7: Alexa Sample Utterances for Smart Actuator Control

Figure 7 shows the sample Alexa utterances. An utterance is a combination of intent and a phrase. We need to register a list of accepted utterances in the Developer Console so Alexa can match the user's spoken words to the skill's intents. Above is the screenshot of added utterances in the implementation; for the intents of light and fan. Multiple phrases can be provided for a single intent as to not enforce a rigid spoken syntax on users. LightStatusIntent is a skill that queries the actuator platform to determine the status of the light.

```

var httpClient = new HttpClient();

switch (intentName)
{
    case ("FanStatusIntent"):
        return await Tfl.Status.GetResults(session, httpClient);

    case ("LightStatusIntent"):
        return await Tfl.Status.GetResultsLight(session, httpClient);
}
    
```

Figure 8: Alexa Speechlet for Smart Actuator Control

Figure 8 shows the Alexa Speechlet function, which at first makes an Http client and the checks the intent name case to verify whether it is intent to control the fan status or it is intent to control the LED light. One the intent is matched; it then sends the status change query to targeted actuator module and waits for the response. Once the action at the IoT control platform is completed, it sends back the confirmation message which will be transformed into speech to be sent back to the user.

As can be seen in the figure 9, the Alexa intent response function generated the response to be sent back to the user. It confirms the actuator status from actuator module before composing the response. Then it structures a response sentence and sends the intent response consisting of a card text which will be shown as text on the user view and will read by Alexa for the user. The images size are large and small, based on the available resources.

```

2 references
private static AlexaUtils.SimpleIntentResponse ParseResults(string resultString, int opt)
{
    string stringToRead = String.Empty;
    string stringForCard = String.Empty;

    bool result = GetActuatorState("DS00006");
    stringForCard = "Light Status details to be displayed";
    if (result == false)
    {
        stringToRead = Alexa.AlexaUtils.AddSpeakTagsAndClean("The Light is switched off at the moment");
    }
    else
    {
        stringToRead = Alexa.AlexaUtils.AddSpeakTagsAndClean("The Light is switched on at the moment");
    }
    return new AlexaUtils.SimpleIntentResponse()
    {
        cardText = stringForCard,
        ssm1String = stringToRead,
        largeImage = "light.jpg",
        smallImage = "light.jpg",
    };
}
    
```

Figure 9: Alexa Simple Intent Response for Smart Actuator Control

A virtual assistant agent at back-end coordinates all the data, queries and actions taking place between the IoT control platform and Alexa service provider. A CoAP server is built to which actuator modules mounted at Edison board act as a client. It executes the control commands on the actuator modules and also gets their statuses and other information to be passed to the actuator middleware.

4. Conclusion

The integration of IoT and cloud is of significant importance and opens doors to many new dimensions of smart environments applications. This work proposed architecture for the cloud-based virtualized IoT smart environment which uses speech recognition to take user commands. Virtualization is to represent the physical world into a virtualized world, which is described by the object's resources, attributes and features. In our proposed approach the services are composed by combinations of actuator virtual objects and sensor virtual objects based on user activities; with the use of virtual service manager and virtual device manager modules. The virtualization of actuators in IoT system is accessed by customized skills using Alexa echo dot at user level. In future works, we hope to extend the virtual assistant module and make it flexible and adaptable in understanding different speech styles and accents.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00756, Development of interoperability and management technology of IoT system with heterogeneous ID mechanism), and this work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT)), Any correspondence related to this paper should be addressed to DoHyeun Kim.

References

- [1] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*. 2013 Sep 1;29(7):1645-60.
- [2] Aazam M, Khan I, Alsaffar AA, Huh EN. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In *Applied Sciences and Technology (IBCAST)*, 2014 11th International Bhurban Conference on 2014 Jan 14 (pp. 414-419). IEEE.
- [3] What is Virtualization [Internet]; [updated 2017 Oct 26; cited 2018 Aug 29]. Available from: <https://www.networkworld.com/article/3234795/virtualization/what-is-virtualization-definition-virtual-machine-hypervisor.html>. (website)
- [4] Fattah SM, Kim HS, Chong I. Design of composite virtual objects for service entity creation in WoO based IoT environment. In *Information Networking (ICOIN)*, 2016 International Conference on 2016 Jan 13 (pp. 372-374). IEEE.
- [5] Sehrish M., D.H. Kim, "Intelligent Control Service Based on Voice Using Alexa Echo Dot in IoT Networks", *ICCT 2018*
- [6] Alam S, Chowdhury MM, Noll J. Senaas: An event-driven sensor virtualization approach for internet of things cloud. In *Networked Embedded Systems for Enterprise Applications (NESEA)*, 2010 IEEE International Conference on 2010 Nov 25 (pp. 1-6). IEEE.
- [7] Cast Project [Internet]; [cited 2018 Aug 29]. Available from: <http://www.ict-ccast.eu/>
- [8] CoBIs project [Internet]; [cited 2018 Aug 29]. Available from: <http://www.cobis-online.de>
- [9] Tao F, Cheng Y, Da Xu L, Zhang L, Li BH. CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. *IEEE Transactions on Industrial Informatics*. 2014 May;10(2):1435-42.
- [10] Petrolo R, Loscri V, Mitton N. Towards a smart city based on cloud of things. In *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities 2014 Aug 11 (pp. 61-66)*. ACM.
- [11] Botta A, De Donato W, Persico V, Pescapé A. On the integration of cloud computing and internet of things. In *Future internet of things and cloud (FiCloud)*, 2014 international conference on 2014 Aug 27 (pp. 23-30). IEEE.
- [12] Aazam M, Khan I, Alsaffar AA, Huh EN. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In *Applied Sciences and Technology (IBCAST)*, 2014 11th International Bhurban Conference on 2014 Jan 14 (pp. 414-419). IEEE.
- [13] Internet of things [Internet]; [updated 2018 Aug 16; cited 2018 Aug 30]. Available from: https://en.wikipedia.org/wiki/Internet_of_things (website)
- [14] Smart Environment [Internet]; [updated 2018 Jan 16; cited 2018 Aug 30]. Available from: https://en.wikipedia.org/wiki/Smart_environment (website)
- [15] Nallapaneni Manoj Kumar, Pradeep Kumar Mallick, "The Internet of Things: Insights into the building blocks, component interactions, and architecture layers", *Elsevier Procedia Computer Science Journal*, Volume 132, Pages 109-117, 2018, ISSN:1877-0509, UGC SI No: 46138 and 48229 DOI: <https://doi.org/10.1016/j.procs.2018.05.170>.
- [16] Nallapaneni Manoj Kumar, Pradeep Kumar Mallick, "Blockchain technology for security issues and challenges in IoT", *Elsevier Procedia Computer Science Journal*, Volume 132, Pages 1815-1823, 2018, ISSN:1877-0509, UGC SI No: 46138 and 48229, DOI: <https://doi.org/10.1016/j.procs.2018.05.140>.