SEAMLESS HUMAN- DEVICE INTERACTION

[1]Dr.M.S.SATHISH BABU, [2]Dr. M ROBERTS MASILLAMANI, [3]Ms.S.DHIVYA BHARATHI
Associate Professor[1], Professor[2], Assistant Professor[3]
satish_babu5@yahoo.com, rmasillamani@gmail.co.in, dhivya04bharathi@gmail.com
Department of CSE, Kathir College of Engineering, Coimbatore, Tamil Nadu, India.

ABSTRACT

Automation play an vital important role in our daily lives.Reducing human labor, effort, time and errors due to human negligence are the main goal of every automated system.The development of modern technology, smart phones have made a huge necessity for every person in the universe.Many Applications are being developed on Android systems which are used to human in various ways.Natural language processing is the another upcoming technology.It enables us to command and control things with our voice.Hence our paper presents on a basis of micro controller based on voice controlled home automation system.It is implemented through the smart phones. This upcoming device makes every human to have control over every appliance in his/her home with their own voice.All that we need is that the user needs an Android Smartphone, which is present on everybody's hand.Adriano Uno microcontroller is a control circuit, which processes the user commands and controls the switching of devices.

1.INTRODUCTION

The main aim of the technology is to increase efficiency. The area of 'Internet of Things' implement the requirements like Security, Automation and Remote Control. Once a home starts to adopt this system,the Security will works on the respective sensors. The used sensors detect the fault and the flaws if any. For example the level of water in the house rises due to a tap right on, the sensors get activated and through the sensors tpushing for ubiquitous computing in all spheres of life. Automation is the area that aims to achieve the features of simplicity by increasing efficiency. Voice controlled House willl further increse the efficiency.. When the first computers came around,it is seen that its just a machine to achieve the level of sophistication.But automation helps to find the idea can arise to narrate commands using voice to a machine.It was only realized by science fiction.

1. To make everything in the house automatically controlled using technology and perform the jobs that we do normally manually

2. To connect all of the systems and devices to a central system so that they can be controlled from anywhere and respond to one another.

3. Through SMS it can control the appliances

2.PROPOSED SYSTEM

2.1Raspberry Pi

Raspberry pi board is a packing considerable computing power into a footprint no larger than a credit card.

It is capable of some good things, but there are a few things to learn before the implemention

2.2.ARM vs. x86

The processor at the heart of the Raspberry Pi system is a Broadcom BCM2837 system-on-chip (SoC) multimedia processor. This means that the vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the centre of the board (see Figure 1-1).It's not just this SoC design that makes the BCM2837 different to the processor found in your desktop or laptop, however. It also uses a different instruction set architecture (ISA), known as ARM.

2.3.Getting Started with the Raspberry Pi

It shows the process of how the Pi differs from another computing devices. Take it out of its protective anti-static bag and place it on a flat once you received your Pi

2.4.Connecting a Display

Before starting the Raspberry Pi you need to connect the device on your respective display. The Pi supports three types of different video outputs,the outputs are follwed :composite video, High definition multimedia interface video and DSI video. Composite video and High definition multimedia interface video are readily accessible to the end-user,some specialized hardware are needed by the DSI.

2.5.Composite Video

This video, available via the yellow-and-silver port at the top of the RaspberryPi known as an RCA phonon connector is designed for connecting the Raspberry Pi to earlier display devices. As the name suggests, the connector creates composite of the colors found within an image—red, green and blue—and sends it down a single wire to the display device, typically an old cathode-ray tube (CRT) TV. The yellow RCA photo connector, for composite video output When no other display device is available, a composite video connection will get you started with the Pi. The quality, however, isn't great. Composite video connections are significantly more prone to interference, lack clarity and run at a limited resolution meaning that you can fit fewer icons and lines of text on the screen at once.

2.6.HDMI Video

A better-quality picture can be obtained using the HDMI (High Definition Multimedia Interface) connector, the only port found on the bottom of the Pi . Unlike the analogue composite connection, the HDMI port provides a high-speed digital connection for pixel-perfect pictures on both computer monitors and high-definition TV sets. Using the HDMI port, a Picas display images at the Full HD 1920x1080 resolution of most modern HDTV sets. At this resolution, significantly more details available on the screen.

2.7.Connecting a Keyboard and Mouse

Now that you've got your Raspberry Pi's out put devices saved, it's time to think about input. As a bare small amount, you're going to need a keyboard, and for the majority of users, hardware is a necessity too. then, some bad news: if you've got a hardware with a PS/2 connector—a circle pin with a horseshoe-shaped packs of pins—then you're going to have to go out and buy a rearrangement. The old PS/2 communication has been superseded, and the Piexpects your physical to be connected over the Universal Serial Bus (USB).

2.8Connecting External Storage

While the Pi uses an SanDisk card for its main storage device—known as a boot device—you may find that you run into space limited faster. Although large SanDisk cards holding 32 GB, 64 GB or more are available, they are often hey are very costly. Thankfully, there are devices that provide an additional hard disk to any PC when connected through a USB cable. Known as USB Mass Storage (UMS) devices, these can be physical hard disk, solid-state drives (SSDs) or even portable pocket-sized flash drives.

3. SYSTEM ARCHITECTURE

The objective of the system is to interpret the human Natural Language orders, extracting useful information from the context. In this way, the system can find candidate devices and fire simple or complex actions that may require orchestration. The proposed architecture considers two distinct entities beyond the user. These two entities are the Controller and the set of IoT devices. The Controller mediates between the user and the devices. It interprets the intention of the user from Natural Language expressions, deducing what devices should be involved to attain the desire outcome and overseeing the entire process. Each IoT device is expected to offer one or several independent actions that can be described according to common format. Each action can be described independently and tied to a binding that implements the underlying protocol messages and controls the execution. This abstraction is compatible with state of the art IoT frameworks.

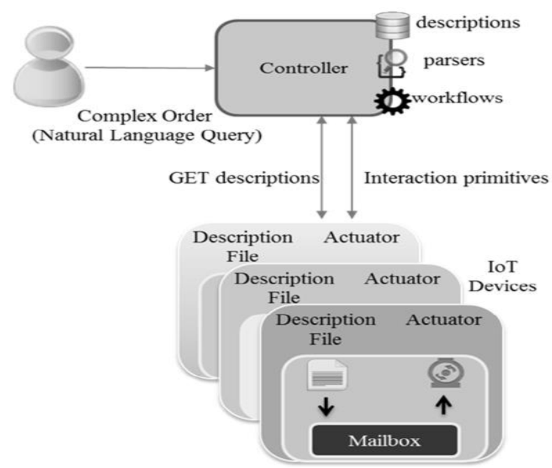

Fig. 1. System Architecture. Interaction schema between entities: user, controller, and set of devices.

A. Describing IoT Devices

As discussed in section II.B, there are significant efforts around describing resources on the Internet that influence IoT

Environments. Resources can be described by means of an ontology from which their properties can be inferred. This is especially worrying for describing IoT device actions since these devices may be numerous and responders may run on constrained devices. Alternatively, resources can also be tagged with several keywords that, incorrectly selected, can describe device actions accurately and that can be handled by limited devices. For that reason, the proposed system uses tags for describing devices. In addition, adding new devices and their ontologisms in a system can be computationally expensive since it may require ontology matching whereas tags allow new devices to be incorporated at a reasonable computing cost. Thus, the proposed system requires every device willing to accept Natural Language orders to be described using metadata to the Controller once discovered.

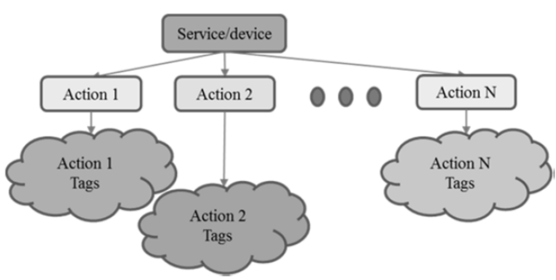

Fig. 2. Service/device description file schema. The definition is held in the actions that describe it, i.e., what the object can do. In addition, each action includes a set of tags in natural language used to describe it

B. The Controller System

This section describes the modules of the Controller and their relations The Detector allows the user to make queries to the system. Any order in the form of voice, text or gestures can be captured by the Order Detector. The Order Detector uses several input devices as microphones, keyboards, or messaging interfaces. The way input devices are distributed and their use are out of the scope of this article. Natural Language expressions captured by the Order Detector are converted into text and fed to the Order Interpreter. The Order Interpreter is the module that parses user-orders from the Order Detector. It processes the entire text and produces an order primitive that is a set of selected words from the user input and a collection of clusters of related words. The Workflow Orchestrator processes the resulting order primitive to obtain the desired outcome. It produces a sequence of device actions whose execution fulfills the user intentions as expressed in the order primitive. The Workflow Orchestrator relies on the Cluster Manager to find devices and thus, device actions, whose single or compound execution leads to the desired outcome.

The Cluster Manager collects tags from the device descriptions and indexes them, along with other related words as described in section IV.A. The set of related words, word type and tags extracted from device action description has an Identification String (IdS).The Workflow Orchestrator queries the Cluster Manager to find the sets of tags matching the terms in the order primitive provided by the interpreter and identifies those sets by theories. The Workflow Orchestrator using a scoring algorithm can deduce which device actions should be triggered to get the desired outcome. The algorithm is described in section IV .Finally, the Message Builder/Interpreter instantiates any protocol binding needed to interact with devices. It also monitors device status and device action execution, updating the Workflow Orchestrator accordingly.

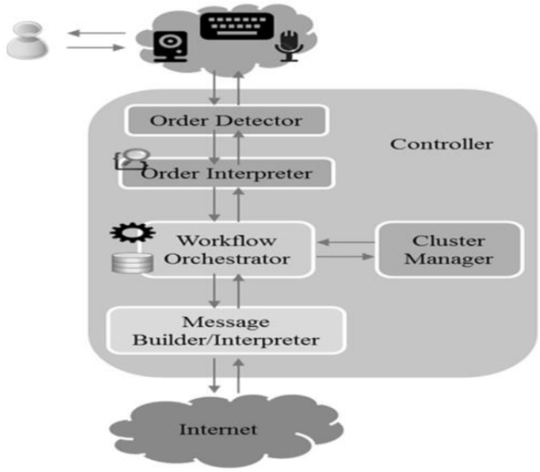

Fig. 3. Controller Architecture. The Controller System contains the distinct

4. NATURAL LANGUAGE PROCESSING

The system processes Natural Language in two stages. The first stage requires searching synonyms and related words to create a database of terms that can expand the keywords provided by the device to describe an action. These data feed the Cluster Manager. The second process is the analysis of the user's queries to create the Order Primitives by the Order Interpreter.

A. Device Metadata Processing

Natural language tags. This set of tags is likely to be provided by the manufacturer or a developer but it also may be provided by the end-user. As it was discussed in Section II, the main problem of selecting the appropriate words is that they must fit into the terms that any potential user would use. This is a double-edges word.

On the one side, a device description containing many related tags can target a large number of potential users by494 IEEE Transactions on Consumer Electronics, Vol. 63, No. 4, November 2017default. The reason is that habits, culture or age-segmentation affects the way users use Natural Language so it can be thought that adding more words to a description would increase the number of potential users. However, this can dramatically increase the number of false positives as device actions would be candidate for more orders than they should be.On the other side, a succinct device description that precisely describes an action may decrease the number of false positives improving user experience but may target a smaller number of potential users. To tackle these problems, the system performs a controlled expansion of keywords by fetching synonyms and related words from online dictionaries. This process can be controlled attending to the evolution of Natural Language expressions and cultural traits.

How the Natural Language evolves is studied by the system analyzing the context in terms of which words or terms are most used by the user regulating the dataset of synonyms or related words handled by the Controller. Thus, manufacturers or developers are expected to add a small set of terms in natural language that describe the action. The subsequent word expansion process will handle user diversity. A simple example is shown in Table I. This table contains a working example of a television that can bewitched on and off or tuned to a different station.

TABLE 1
EXAMPLE: TELEVISION ACTIONS

Action name	Action Description	Action Tags
switch off	switches off the television	"television", "switch off"
switch on	switches on the television	"television", "switch on"
change channel	changes the current television channel	changes the current television channel "television", "channel", "change"

The first time the system processes a tag for a new device action, it sends a query to an online dictionary of synonyms. After that, it explores the related words to each tag.

The system prototype uses a lexical database for English named WorldNet from Princeton University [29],but other language sand databases can be used. The system considers as related words those that, based on cultural traits, are frequently used in combination to the original word within the same sentence. Moreover, terms thecae the same meaning in some contexts, are related words too. The set of related words obtained in this way is saved in local cache as they do not vary very frequently.

The Cluster Manager uses the sets of related words to cluster tags based on their synonyms and related words. Using the example in Table I, an action tag expressed as "television"(noun) will be spanned to "TV" and "tally" so these terms will be in the same group. Hence, the Cluster Manager manages groups of these related words.

The system considers the following as the most important relations:
1) "dobjt." Direct Object The object of the verb.
2) "nmod." Noun modifier A term that modifies or specifies the meaning of other word.
3) "compound." A term that modifies or specifies the meaning of a term.
4) "dep." Depends A term that depends from other.
5) "nummod." Numeric modifier Specifies the meaning of a noun with a quantity.
6) "case." Provides more information of nominal elements.
7) "nsubj." Nominal subject A noun phrase which is the syntactic subject of a clause.
8) "advmod."Adverb modifier.

5.Hardware Settings

The Raspberry Pi's hardware is controlled by settings which in a file called config.txt, which islocated in the /boot directory This file tells the Pi how to set up its various inputs and outputs, and at what speed the BCM2837 chip and its connected memory module should run. The contents of directory, with highlighted If you're having problems with graphics output, such as the image not filling the screen or spilling over the edge, config.txt is where you'll be able to fix it. Normally, the file is empty or—on some distributions—simply not present; this just means that the Pi will operate using its preset defaults.

If you want to make changes and the file isn't there, just create a new text file called config.txt and fill in the settings you want to change. The file can control almost all aspects of the Pi's hardware, with the exception of the way the (CPU) and graphics processing unit (GPU) sections of the BCM2837 apportion the memory. You'll learn how to alter that split in the "Memory Partitioning—start.elf" section, later in this chapter.

The config.txt file is only read when the system first starts up. Any changes made while the Pi is running won't take effect until the system is restarted, or switched off and back on again. In the event that the changes are unwanted, simply deleting the file from the /boot directory should be enough to restore the defaults once more. If the Pi won't boot with your new settings, just remove the SanDisk card and delete config.txt from the boot partition on another PC, and then reinsert it into the Pi and try again.

6.Software Settings

In addition to config file which controls various features of the raspberry Pi's hard devices , there's are another important text file in the boot: codling. This file contains what is known as the kernel mode line—options passed to the Linux kernel as the Pi boots up.

continuous line:
dwc_otg.lpm_enable=0
Console=ttyAMA0,115200
Kgdboc=ttyAMA0,115200
Console=tty1 root=/dev/mmcblk0p2
Rootfstype=ext4 root wait

The first option, dwg_otc.lpm_enable, tells the Pi to disable the On-The- Go (OTG) mode of its USB controller, to prevent problems that can occur when the functionality is enabled without proper support in the operating system. The majority of Linux distributions for the Pi disable this mode. The console option tells Linux that it should create a serial console—device ttyAMA0—and at what speed it should operate. In most cases, the speed should be left at the default of 115,200 Kb/s (kilobytes per second). If the Pi is being used to communicate with older devices, this can be reduced accordingly.

The kgdboc kernel option enables debugging of the Linux kernel over the serial console created using the console parameter. For most users, this is unnecessary. For developers, having access to kernel debugging over a serial connection is most useful. Many distributions leave this enabled just in case. The serial console entry creates the device tty1, which is the text-filled screen you see when you first boot the Pi. Without this entry, you would not be able to use the Pi without communicating something to the serial console created by the first console option.

7. IMPLEMENTATION AND RESULTS

We carried out a proof-of-concept implementation to validate the algorithms defined in this article using the actor model. The actor model is a well-known mathematical model in which computation primitives are actors.
Actors can create other actors and communicate with others via message passing. Actors, being small single-threaded applications, can be used to model constrained devices. Actors modelling devices evolve and change its state according to incoming messages. We refer to these actors as virtual devices. In our experiment the Controller is an actor which is virtual-device-aware. The experiment was a simulation of a small IoT domain (a Smart Home) with 12 devices with 43 different actions in total.

To build a database of test vectors we created an online form to gather natural language orders from a big population. It was advertised to students and academics from our home institution and through social networks. Subjects received a form to fill where different situations where explained, e.g., "You are in your bed and you want to switch off the lamp in the desk" or "you are listening the radio and you want to change the dial to 100.5 F.M." A total of 2170 sentences (orders) were collected.

We used every order as an input to the system and the results of processing all the orders were stored in a database and later inspected by people. The system returns the candidate actions for user orders with an associated rank and places them in decreasing order, the first one being the most likely one according to the algorithm. It is possible that several candidate actions have the same rank leading to a draw between different actions. We have used the set of ranked actions returned by the Workflow Orchestrator as a measure to evaluate the performance of the system. The criterion for assigning a mark to the results was the following:

1) If the correct action matches the first candidate action in the result, that result is evaluated with 5 (perfect fit); if it was in second place it is evaluated with 4, and so on until 1.

If the correct action is in sixth or higher position within the results, the result is evaluated with 1.

2) If there is a draw, the result evaluation is decreased by one point, but in any case, the evaluation cannot be lower than one.

92% of the results of processing an order contain a correct action in the first or the second position without training the system. Thus, it proves that 92% of the orders were correctly understood by the system. The prototype implementation has been carefully designed for a small memory footprint and low CPU consumption pursuing its use in small CE commodity devices. Our target device during the simulation and testing stage was a board equipped with a quad-core RIS running

900MHz with 1GB of memory.

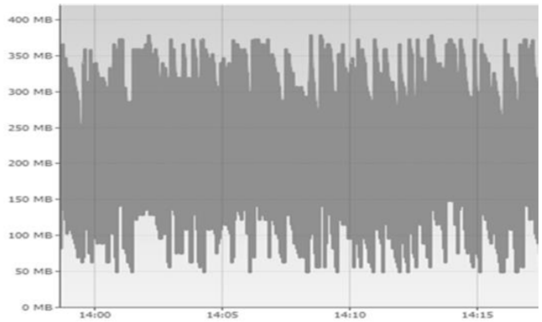

Fig. 5. Test results for the system using 1Gbyte of memory.

8. CONCLUSIONS

The IoT paradigm is gradually complementing existing consumer technology with a lot of new connected devices offering a huge new range of possibilities. Usability is among the most critical problems in IoT. New devices should be easy to use in order to be adopted. However, as has been discussed in this article, interfacing, controlling and coordinating a set of devices simultaneously can be tedious even when every device in the set is simple to use.

Nowadays, the use of Natural Language as a user interface has become a trending topic in research. We have shown a system that uses Natural Language as a common interface for simple and complex orders diminishing the complexity of the Human-Device Interaction, and minimizing the gap between developers and final users.

As discussed, different users with apparently similar knowledge and expertise may use different voice commands to trigger the same actions. This work has adopted tags and related words instead of ontologies (since their processing consumes many resources to be efficient) to expand the reference space for every action demonstrating that without training it is possible to reach up to the 92% of success.

This prototype presents a modular architecture that can be easily instantiated in constrained commodity consumer technology devices as shown in the performance tests. Its resiliency (asynchronous message driven) and flexibility (every module can be instantiated separately) facilitates its deployment in existing and forthcoming IoT domains. Our future work will continue the exploration of the potential of our architecture, improving the controller adding the skill of learning from ambiguous queries and including context-aware recognition system in order to be more effective in the recognition of user's intentions.

REFERENCES

[1] E. M. Maximilien, and M. P. Singh, "A framework and ontology for dynamic web services selection," *Internet Computing*, IEEE, vol. 8, no. 5, pp. 84-93, Sep. 2004.

[2] M. R. Huq, N. T. T. Tuyen, Y. K. Lee, B. S. Jeong, and S. Lee, "Modeling an Ontology for Managing Contexts in Smart Meeting Space," in *SWWS 2007: Proceedings of the 2007 International Conference on Semantic Web and Web Services*, Las Vegas, Nevada, USA, Jun. 25-28, 2007, pp. 96-102.

[3] L. Sommaruga, A. Perri, and F. Furfari, "DomoML-env: an ontology for Human Home Interaction," in *SWAP 2005 – Semantic Web Applications and Perspectives*, Proceedings of the 2nd Italian Semantic WebWorkshop,University of Trento, Trento, Italy, Dec. 14-16, 2005, vol.166.

[4] H. Pigot, B. Lefebvre, J. G. Meunier, B. Kerherv, A. Mayers, and S. Giroux, "The role of intelligent habitats in upholding elders in residence," in *5th International Conference on Simulations in Biomedicine*, Slovenia, Apr. 2003, pp. 497-506.

[5] P. Bailis, J. Yang, V. J. Reddi, and Y. Zhu, "Research for practice: seb security and mobile web computing," in *Communications of the ACM*, vol. 60, no. 1, pp. 50-53, Jan. 2017.

[6] C. Treude, and M. A. Storey, "How tagging helps bridge the gap between social and technical aspects in software development," in *Software Engineering (ICSE) 2009*. IEEE 31st International Conference on, pp. 12-22, IEEE., May. 2009

[7] B. J. Jansen, and A. Spink, "How are we searching the World Wide Web? A comparison of nine search engine transaction logs," *Information Processing & Management*, vol. 42, no. 1, pp. 248-263, Jan. 2006.

[8] W. Jones, A. J. Phuwanartnurak, R. Gill, and H. Bruce, "Don´t take my folders away!: organizing personal information to get things done," in *CHI´05 Extended Abstracts on Human Factors in Computing Systems*, ACM, Portland, OR, USA, Apr. 02-07, 2005, pp. 1505-1508.

[9] J. Hirschberg, and C.D. Manning, "Advances in natural language processing," in *Science*, vol. 349, no. 6245, pp. 261-266, Jul. 2015.

[10] C. Fellbaum, "WordNet: An Electronic Lexical Database," in *Language*, vol. 76, no. 3, pp. 706-708, Sep. 2000.