Context Aware Virtual Assistant with Case-Based Conflict Resolution in Multi-User Smart Home Environment

Bauyrzhan Ospan[1,4], Nawaz Khan[2], Juan Augusto[2], Mario Quinde[2,3], Kenzhegali Nurgaliyev[1,4]

[1]Department of Information Technology
Kazakh University of Technology and Business
Astana, Kazakhstan
bospan@nu.edu.kz, knurgaliev@nu.edu.kz

[2]Research group on Intelligent Environments
Middlesex University
London, UK
n.x.khan@mdx.ac.uk, j.augusto@mdx.ac.uk, mario.quinde@udep.pe

[3] Department of Industrial and Systems Engineering
Universidad de Piura
Piura, Peru

[4] Cleverest Technologies LLP
Astana, Kazakhstan
bospan@cleverest.tech. kennurgali@cleverest.tech

***Abstract*—In this work we will examine and develop a Virtual Assistant that is used as a control interface of the Smart Home environment in the Middlesex University laboratory. Specifically, the system is constructed to give multiple users an ability to control appliances by voice or text commands. The main purpose of the research is to create an user friendly interface with adaptive context aware Case-Based conflict resolution system. In addition, Artificial Neural Networks were used to classify user inputs to create a natural dialogue. Further, a set of random double-blinded evaluation tests were established with general positive results in terms of interface justification.**

***Index Terms*—case-based reasoning, context awareness, multiple user systems, smart home, virtual assistant**

I. Introduction

Modern technologies may help to manage Activities of Daily Life (ADL) more comfortably, especially for people with special needs and younger population [8]. For this purpose, Smart Home environments are developed to help users control devices that are often called Virtual Assistants [8]. They monitor behaviors of people with special needs [3] to classify changes in ADL and alarm or notify relatives, health-care professionals and organizations [14]. For example, there are Ontology-based platforms that allow analyzing user's actions and perform context recognition [1]. Likewise, there are visual tools that create personal preferences table of each user based on semantic metadata in Smart Home systems [7]. Thus, the majority of researches is aimed at developing systems helping alone living users. On the other hand, conflicting orders by different users inside the system can lead to some risks [4]. Despite a number of researches in Smart Home solutions that are automated, adaptive, multi-functional and interactive, there is a lack of investigation and development in systems that operate in conflict resolution between multiple users with different needs and priorities. Some researchers, in order to solve this dilemma, developed context-aware automation platforms [4], [5] that resolve conflicts between automated devices. Another group of research is focused on preference Rule-Based Reasoning (RBR) [8]. Even though these systems introduce an efficient set of rules, they are not able to handle the adaptation problem.

Smart home is a network of actuators and sensors which should be non-disruptive regardless the conflicts. Consequently, our gadget is designed to be aware about the context within the house. For instance, a person turns on a room light. A mild actuator changes the illumination level within the room, so the context is changed. The person can also change a context in terms of temperature value, protection (fire alarm), safety necessities and energy saving purposes within the house. According to Roy at al., there is a system known as MavHome that permits to create a context-conscious framework. It mechanically modifies status of devices updating air-conditioning and light features, by setting an impartial entity for each tool primarily based on the anticipated function of the user [11]. It is helpful for automatic switching devices to be so-called smart appliances without human command. However, human orders rely upon his behavior, motivation and needs. As a result, our device must react directly to person's requests.

Another method to solve various multi-person conflicts is a Reactive Behavioral System (ReBA). That is a context- conscious tool which operates with devices through assigning priority to customers [13]. However, since it separates appliances in order to operate effectively, other customers with lower significance cannot have influence the choice making technique afterwards. Also, we should consider a method that requires consumer intervention in conflict decision. Any user-centric application can make recommendations of feasible answers, specifically, for one-of-a-kind media devices such as television, radio or smart table [14]. In this example, users are able to pick out unique media carrier from suggestions primarily based on their preferences. In a nutshell, the system context is gathering context notions from actuators and sensors. This implies new entity detection for the smart home context using modern-day scenarios or probability on future actions. Considering context notions, the device can act consistently with previously defined regulations, preset policies and instructions. In different spots, these characteristics may be exceptional, however there is a shared space – a smart house. We assumed that each person must have conflicting phrases in terms of identical context (a device). Those functions cannot be activated concurrently. Our motivation is to create a gadget that might be capable of solving predefined conflicts and fulfill newly arisen troubles between context notions in an Ambient Assisted environment.

In this research, we propose a Virtual Assistant (VA) system that controls Smart Home environment. It based on the user inputs via voice and text. It consists of the dialogue graphical web interface for multiple users and resolves conflicts in changing environment. As a result, we developed number of subsystems to satisfy infinite number of requests. First of all, an interaction with the VA has to be user-friendly and natural, as human-to-human interaction. Secondly, the system has to be able to control and monitor the state of the Smart Home devices. Finally, the VA has to know how to resolve conflicts between multiple users, and adapt to the new user cases.

The rest of this work is organized as follows. The architecture of the system is described in Section II. The system algorithms and sequences are described in Section III. Section IV demonstrates our results and validation. Finally, Section V draws the conclusion and predictions for the future work.

II. System Architecture

To cover all the objectives, we have introduced and developed the VA with specific modules in terms of functional units. Firstly, we have developed a user-friendly graphical interface based on web and cloud technologies to give the user an intuitive way of interacting with the VA. For this objective, we have used a dynamic web framework, cloud voice recognition and generation technologies, and animated emotional icons to mimicry natural face emotions in the dialogues. In addition to user-friendly interface, we have developed and implemented the adaptive natural language classification algorithms based on Artificial Neural Network (ANN). It helped to make the VA understand different variates of user commands and argumentation sentences, because these are effective algorithms for sentence classification [13]. Therefore, users were able to speak to VA in the way they speak with other people. Secondly, we used Farm Side Middlesex University Smart Home laboratory and Smart Home devices control unit to manage and monitor Smart Home devices. In order to do that, we have developed and integrated software API to implement communication between Smart Home control unit and the VA. Finally, we have introduced and developed a conflict resolution system that uses Case-Based Reasoning (CBR) to make a decision about conflict between multiple users. In order to make such decision, the VA firstly, identifies the argumentation of the user to perform an action in the dialogue form; secondly, analyzes the similarities between this case and the previous cases using the k-nearest neighbor (kNN) algorithm; and finally, chooses the best-fit case [4,7,15]. In other words, the VA asks the user to explain why he or she wants to change the device status if there is a conflicting instruction from another user. Then, it makes a decision about what to do with the device based on the users' arguments and the previous cases. We have determined different kinds of users categorized by a profile. The system should provide support for each of them in a different way. We have used updated user types and preference system proposed by [8] (Table 1). Moreover, the argument priorities were established based on personal user preferences [10] because a number of researches show that the preference-based argumentation increases the ability of the systems to meet user expectations [9]. The preferences are the reasons why the user wants to change the device status. For an adult person: Security Preference is the highest priority (position 0). For Elderly: Health issues are in the most significant position. For Young people: Energy saving problems (position 5) will be the least important than for other profiles.

TABLE I
List of users and preferences

User types	Preferences (positions 0,1,2,3,4,5)
Adult (26-69)	Security, Health, Work, Food, Energy, Entertainment
Elderly (older 70)	Health, Security, Food, Energy, Entertainment, Work
Young (up to 25)	Security, Work, Entertainment, Health, Food, Energy

Introducing the reason is the way to increase usability of the system through the implicit interpretation of user needs. Con- text awareness of the system is the key for creating high usability by making the interaction between user and system more natural [6]. We have used "Ceteris paribus" preferences binary tree representation. At the moment, CP-nets are the most effective qualitative approach for presenting preferences in the strict known environment such as Smart Home [9].

The architecture of the system is layered and modular. Each functional part (module) of the system is designed to execute a complete task. The component is a group of modules that perform a complex task that consists of a number of simple tasks. Additionally, in order to give better understanding of the system, the architecture is divided into layers. Layer is a functional level of the system that sequentially divides it into hierarchical groups that communicate only with layer above or layer below except Database layer. Layer is a white rectangle with a grey icon and a name in it. The web page of the web interface is demonstrated as a white-colored box with a blue drawing and text in it. A blue hexagonal icon with a white image inside and text below is the module of the system. A white quadrilateral with a grey illustration of the device and its name underneath is the icon of electronic device driven by Smart Home environment. A colored rectangle with text in the bottom right corner is the illustration of components. All communications inside the system are demonstrated as arrows in Fig.1.

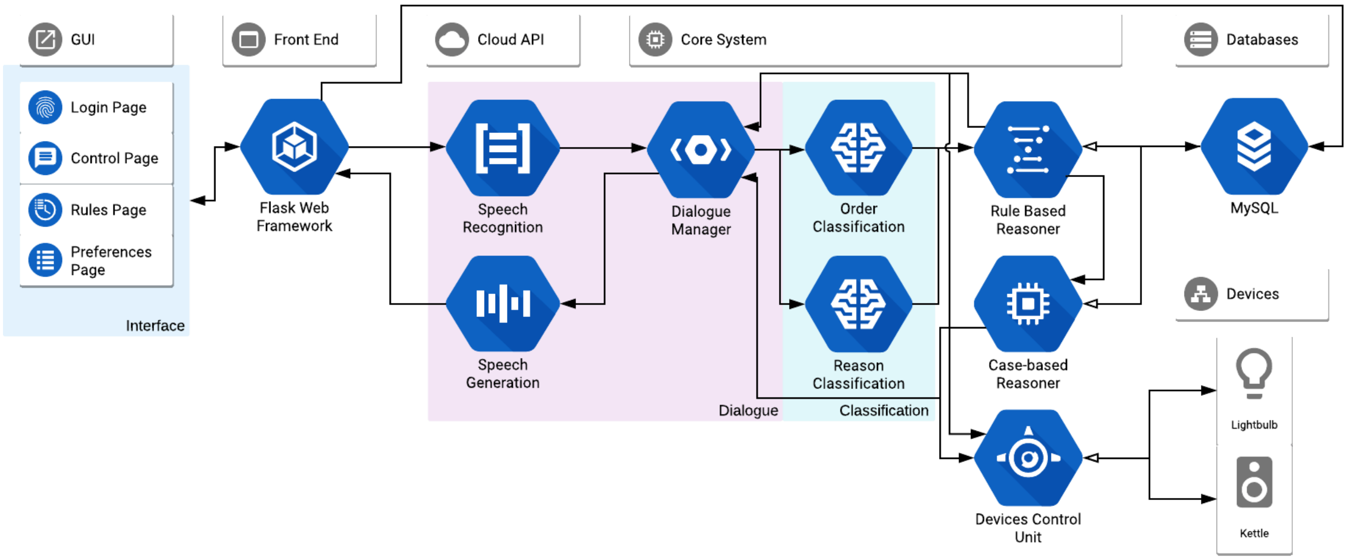

Fig. 1. Architecture diagram of the system

A. Graphical User Interface, Front End and Cloud API

Communication with users is one of the principal elements of our project. Having user profiles (Table 1), it is crucial for the interface to deliver prompts for each of the users in the best manner [19]. The interaction will be managed by a little "smilie" [20]. The aspect of the related GUI (Graphical User Interface) should be based on the profiles suggested by [21], [22]. Moreover, a general perception of the Virtual Assistant depends on non-verbal aspects during interaction [23].

The interface of the system is a graphical web interface developed as a web site shown at Fig. 2 and Fig. 3.

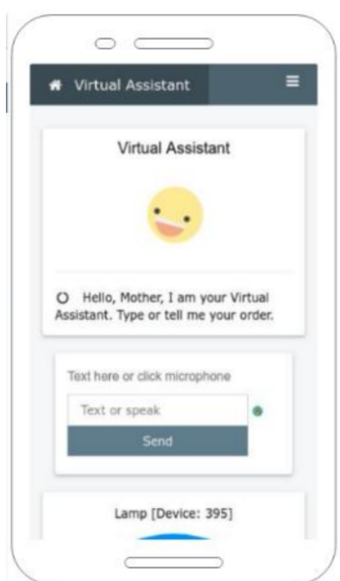

Fig. 2. Virtual Assistant Page

GUI is expressed by three functional and one login pages:

- Login Page - Welcome interface - user chooses his or her account
- Control Page - Dialogue interface - user communicates with system in form of dialogue by voice or text chat, gives orders and reasons
- Rules[1] Page - Rules editing interface - user edits list of instructions[2] given to the system
- Preferences[3] Page - Rules editing interface - user edits list of instructions given to the system

Flask Micro Web Framework is used as a Front-End controlling technology or a Back-End of the interface, because most of the project is coded by Python 3 language. Firstly, a header menu with hyperlinks to each page is located on the top of the interface. In the upper left corner of the Control Page there is an emotional icon representing facial expressions of the Virtual Assistant and the text of the VA answers below it. In addition, a usage manual is provided below the VA that can be easily closed by clicking "x" button. In the middle of the page, one can find an input box with microphone activating and send buttons. In order to explain the system, there is an instruction block under the input box. Finally, right hand side of the interface is filled by devices status indicators that show current statuses.

The function units of the Speech Generation and Speech Recognition were established via W3C Web Speech API4. These are JavaScript Application Programming Interfaces (API) that are used by most of the modern browsers. As a result, the system sends the record of the user input to W3C API and it returns text that is the recognition of the record.

[1]Rule is set of instructions given for system to do automatically. For example, system is obligated keep kettle turned off from 11:00 till 13:00 everyday because of energy saving issue

[2]See Section III

[3]Preference is set of issues that justifies orders of the user to the system. Every type of user has its own priority level for preferences.

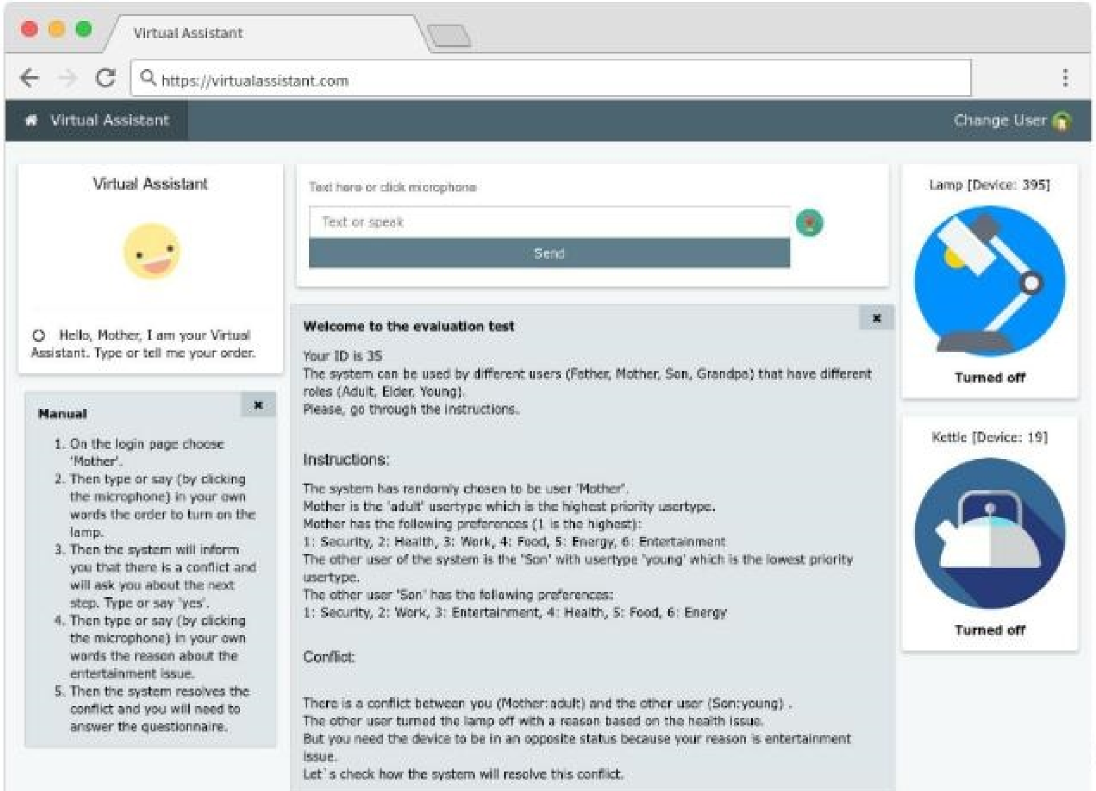

Fig. 3. Control Page of the web interface

Moreover, the system sends text to the browser of the user in such a way that the browser automatically converts it into voice. The modules of the Speech Generation and Speech Recognition are inserted in the Dialogue component with the Dialogue Manager module.

B. Core System

Core System is the main functional layer. Consequently, functions such as decision-making, dialogue compilation, text classification and device controlling are all executed in the Core System. It consists of Dialogue Manager, Classification Component, Rule-Based Reasoner, Case-Based Reasoner and Device Control Unit.

1) Dialogue Manager: One of the central modules of the system. Dialogue Manager receives phrases from the user and classifies it by Classification component and sends it to the Rule- Based Reasoner. Based on data from the Rule-Based Reasoner and actions done by the system it sends human readable report or response to the user via interface. In addition to Cloud API, Dialogue Manager is a part of the Dialogue component that is the subsystem to communicate with user in natural way. Lastly, Dialogue Manager is a set of if-else rules and dialogues database developed to pretend human-like speech.

2) Classification component: is a group of two Classification modules that are Order and Reason Classification. Their function is to classify orders and reasons from the user input raw data. As a result, they need a classification algorithm. Text classification is a non-trivial complex task that cannot be solved sequentially. Artificial neural network systems (ANN) provide better solution [13]. Backpropagation as ANN is very useful in recognizing complex patterns and performing nontrivial mapping functions. This method is very useful for artificial neural network weights of the synapses optimization. This learning rule is based on supervised learning [13]. In order to train the neural network, a set of training documents and a specification of the pre-defined categories the documents belong to are required. Training and testing sets were created specifically for this experiment. Common English phrases and their synonyms were used to create a set of sentences and their class. During the training, the connection weights of the neural network are initialized to random values. The training examples in the training set are then presented to the neural network classifier in random order, and the connection weights are adjusted according to the Backpropagation learning rule. This process is repeated until the learning error falls below a predefined tolerance level (99.95%).

Fig.4 represents a Feed Forward Neural Network (FNN) diagram used in the project. The rounded objects represent the artificial neurons. The straight lines that are connecting the neurons are called weights that are the multiplication coefficients for input signal. Topology of the Artificial Neural Network consists of Input, Output and Hidden layers (Fig.4).

Input layer consist of input neurons that are binomial

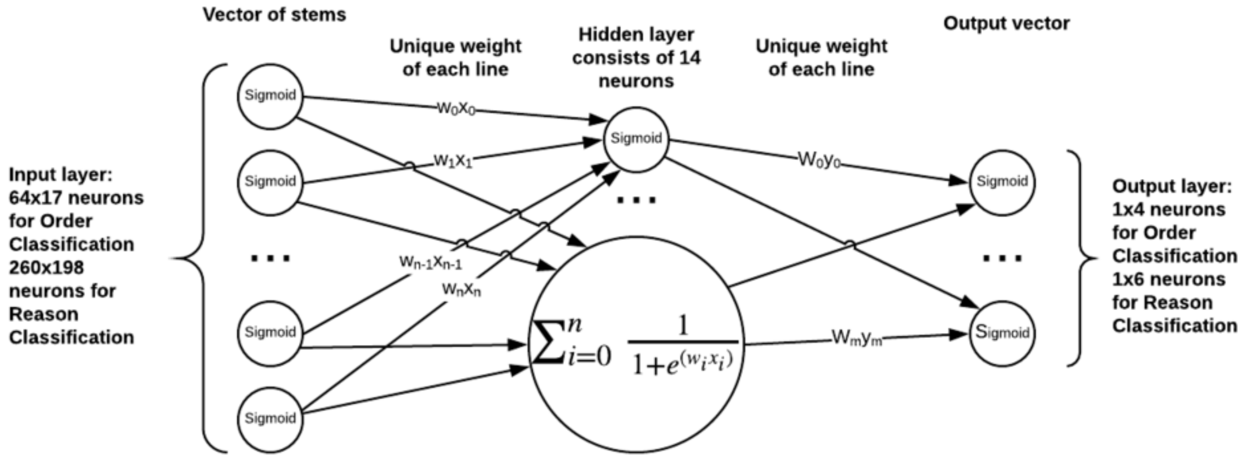

Fig. 4. Classification Neural Network Topology

representation of the unique stemmed words inside the text that is already cleaned from punctuation and meaningless words like articles. While morphological variants are different word forms, they represent same concept.

Output layer gives vector of possibilities that the text is inside of one of the classes[4]. The most possible class is the classification answer of the neural network.

Hidden layer is the regression function of FNN, or layer between input and output layers. FNN applies a sequence of linear functions to the data. These functions are result of sigmoid linear transformation followed by a squashing nonlinearity. While training the ANN, it was concluded that the accuracy of the neural network does not significantly change between one hidden layer neural network. The most efficient number of hidden layer neurons (14) was found through *delta*. It was calculated by 1000 iterations of Backpropogation training algorithm [13] with standard training set especially created for this experiment. In-text categorization and other similar tasks it is desirable to combine these morphological variants of the same word into one canonical form and represent it like vector that is called stemmed version.

User types that he wants to change device status because he needs to finish a project about flu (Ex.1).

Example 1. Input is: I need to finish my project about flu. System classified it as: 'Work': 0.999, 'Health': 0.0566

Classifier says that the request is about work on 99.9%, and a little bit (0.05%) about heath issue, because this is all about flu. The system normally does not accept any classification lower that 30% possibility, and if user says something not correlated to the predefined preferences, the system will skip classification and ask again.

3) Rule-Based Reasoner: RBR receives data about the user's order and compares it with rules in the Database and actual status of the devices. If there is a conflict it executes Case-Based Reasoner (CBR). It was proposed to use Mahalanobis distance that is the global linear transformation tool of the input spaces of the cases from different classes separated by a large margins. Using of Mahalanobis distance significantly increases kNN accuracy [17].

4) Case-Based Reasoner: CBR is an artificial intelligence approach to determine a similarity amongst a set of cases. CBR is the foundation of the many intelligent systems. This method indexes cases from history of the system usage and retrieves best similar case by measurement of distance between the input case and past cases. The result of finding best match is stored inside the cases database to update the cases database and adapt old solution to the new variables and environment. CBR is based on a suggestion that the similar cases have similar solution [6].

The decisions in our system is binary (0 for not change device status, 1 for change it). Therefore, it is accessible that CBR assumption works in our system. As most of CBR systems uses k-nearest neighbour algorithm, the accuracy of this classification algorithm seriously depends on the metrics or system of measurement used to compute distances between different cases. The nearest neighbor algorithm determines the similarity or dissimilarity of a new case with the case base and follows a cyclical process of Retrieve-Reuse-Revise- Retain. In our case inputs to the system are in different system of measurements. Cases in the system are build on preferences and priorities of users of the system. Both two variables have different basis. As a result, standard Euclidean Distance cannot be used as a distance measurement algorithm.

Firstly, it was proposed to use linear representation of the input. As a result, the input to the CBR system was in the form of 14 variable arrays, where two of them - user types[5] and others were linear binomial representation of reasons[6]. Secondly, it was proposed to use preference based weights for each of the reason inputs based on user preferences. Lastly, it was proposed to use modular input type in the form of four by

[4]Meaning that the class is a group of reasons for Reason Classification and a group of orders for Order Classification

[5]User who creates rules and who wants to change the device status.

[6]Where 1 is a reason of existence and 0 is null.

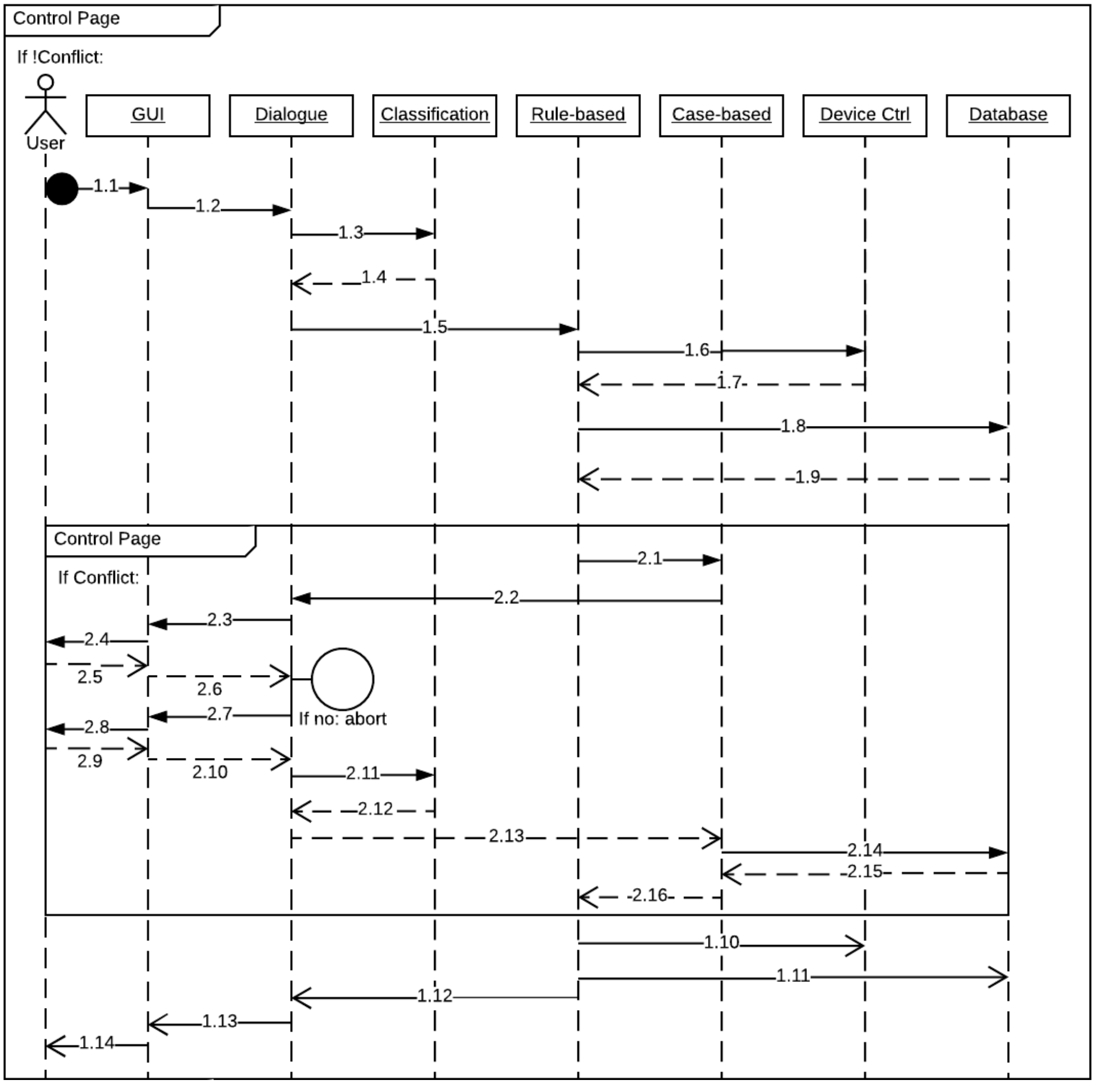

Fig. 5. UML Sequence diagram of the system

one matrix, where first two arguments are user types weights and others are numerical representation of reason weights. As a result, modular form of input in cooperation with Mahalanobis Distance Case-Based Reasoner showed the best result in 200 cases tests with training and test correlation by 90/10.

5) Device Control Unit: The Smart Home environment of the Middlesex University is driven by the Vera Secure system. Consequently, we have developed API to communicate with it. Vera Secure system has own web interface. In order to communicate with it, we were sending and receiving GET and POST requests. Special POST requests change the status of the devices, which are kettle and light in our experiment. Next, updating information about device statuses is done by parsing response from special GET request sent to the Vera Secure server. We have used MySQL database to store data of the project. For example, it was used to store the cases of the Case-Based Reasoner in a form of modular input.

III. Scenarios

Action scenarios inside the system are discussed in this section. We have taken into consideration consumer profile categories in terms of cases and required actions. Users needed to decide what kind of the provided help and comments Virtual Assistant expects. Usage of unique predefined eventualities – pre-programmed behavior of Smart House appliances to a concrete event - machine can define a command from PC or Avatar or set a timer for work deadline. Most of the features and occasions could be controlled manually or through voice control. Sequence diagram of the system can be found in Fig.5. It shows two main scenarios inside the system - Regular and Conflict ones.

A. Regular Case

According to Fig.5, firstly, the user sends command to the system (1.1) that is analyzed by Dialogue Component (1.2) via Classification ANN (1.3, 1.4). Then, the system sends a classified order to the RBR (1.5) and checks its context (1.6 - 1.9). If there is no conflict, it changes device status (1.10, 1.11) and answers to the user (1.12 - 1.14). Else, it executes Conflict Resolution Scenario by Case-Based Reasoner (2.1).

B. Conflict Case

Lifelines of the system inside this scenario are indicated by number 2 in the Sequence diagram (Fig. 5). Conflicting Case is the extension of the Regular Case. As a result, it inherits all other sequences of the Regular Case except the unique case of the conflict resolution done by CBR. Thus, Conflict Resolution scenario starts with informing the user about a conflict (2.2 - 2.4). If the user cancels command (2.5, 2.6), it aborts Conflict Resolution loop. Else, the system asks for argumentation (2.7, 2.8) from user (2.9, 2.10) and classifies it (2.11, 2.12). After the collection of data (2.13) CBR reads old cases (2.14, 2.15) and makes a decision (2.16).

Elderly, Adult or Young profile presence generates automatic solution to conflicting dialogues. Nevertheless, there are some cases when users' commands can be insufficient to the system. Once one of the Users asks Virtual Assistant a question with an identical matter - "light on!" or "light off!", system needs a concrete clarification - ""in which room?" or "which light?". Nonetheless, Virtual Assistant stores last accessed device to predict utterance from the user.

Additionally, the device automatically receives facts about states of the gadgets inside the room, and in case of its comparable entities with consumer request, it is able to stop an execution, sending the prompt ("your light is already on"). It is crucial that consumer must utter precisely which appliance in what room he desires to work with, so that not to conflict with different context domains. We can examine a circumstance when the consumer asks to turn a kettle off, when it has a clash with different person's time schedule for a breakfast. Another instance of conflicts between context notations can be a case when a user would love to decrease a heating level inside the residence because of energy saving issues. Nevertheless, the system states that it is too cold in rooms. It could have a conflict with environment and active user profiles. System can suggest to remain the heating level unchanged, but a final choice has to be made by the User whose Energy preference is higher.

Similarly, it is possible to have a situation when a particular User request overrides a parallel utterance from another user. For example, Young individual asks to operate a device, while Adult or Elderly profile give a prompt to shut down this gadget. It refers to the case when system relies on Profile information to give precedence to the request and begin to carry out Adult's or Elderly's command, stopping the Young prompt. We have proposed conflict resolution system, where two or more users have tried get control over the same device.

IV. Results and Validation

To evaluate the system, we have constructed a randomized double-blind experiment. In order to perform the randomized double-blind experiment, we developed a scenario creating sub-module. Firstly, it created a random scenario for the tester by choosing conflicting device and reasons for tester by using crypto-secure random function based on operation system's inner clocks. Secondly, it provided the testers with instructions to provide command and arguments in their own words. The example of such scenario is given in the figure below. As a result, the scenarios were hidden both from the researchers and from the testers till the beginning of the experiment.

Thirteen testers were asked to fill out the evaluation form. In the evaluation form the testers were asked to answer questions about Speech recognition, classification, conflict resolution, general idea, interface design, interaction with the VA and user friendliness, using a quantitative scale, where 0 is very bad and 4 is very good. Furthermore, they were provided with the additional text box under each question to justify their answers if they thought that the answers needed to be explained. Moreover, one more question was about what they would include or change in the system.

The majority of the responses were positive[7]. The average of the all answers are higher than "Good" level (three and more) except "Conflict Resolution" question, which has 2.9 average value with 1.2 standard deviation amount. The basic quantitative analysis of the answers are shown in Fig. 6.

The speech recognition question has no negative responses. Despite of some mistakes in the recognition, majority of the users were positive about speech recognition. Moreover, one user justified a high score with a comment that the system recognized voice despite a difficult accent. The questions about classification have only positive responses. The system managed to classify the reasons in a correct but different manner than users. For example, one of the testers was asked to say something about entertainment issue, and the user said about having a cup of tea with friends. The system classified it as "Food" issue, which, certainly, is also a correct classification. Moreover, the system could handle very creative and non-trivial cases from users. For example, a user was asked to give a prompt about safety and said "I heard a suspicious noise inside the house", which was the unique and tricky phrase. However, the system managed to classify it as a security issue.

The question about conflict resolution quality was the hardest for the system, because it has two negative answers. Firstly, there was disagreement about priorities policy of the system and the general view of the user. The conflict was that Younger user wanted to turn on the lamp to make his home- work, but Adult user turned it off because of the entertainment issue. As a result, the system decided that the lamp would be turned off because Adult has higher priority while work issue is not the exceptional case. The tester had a different point of view on this case. The second case was unique and new to the CBR, and it did not manage to give an appropriate output. That was the case when the system failed because of different reasons. Analysis of the system logs and simulation of the case gave us better understanding of the system fault. As a result, there is a list of possible cause effects and ways to solve them:

[7]83 of 91 quantitative answers are positive, 5 neutral and 3 negative.

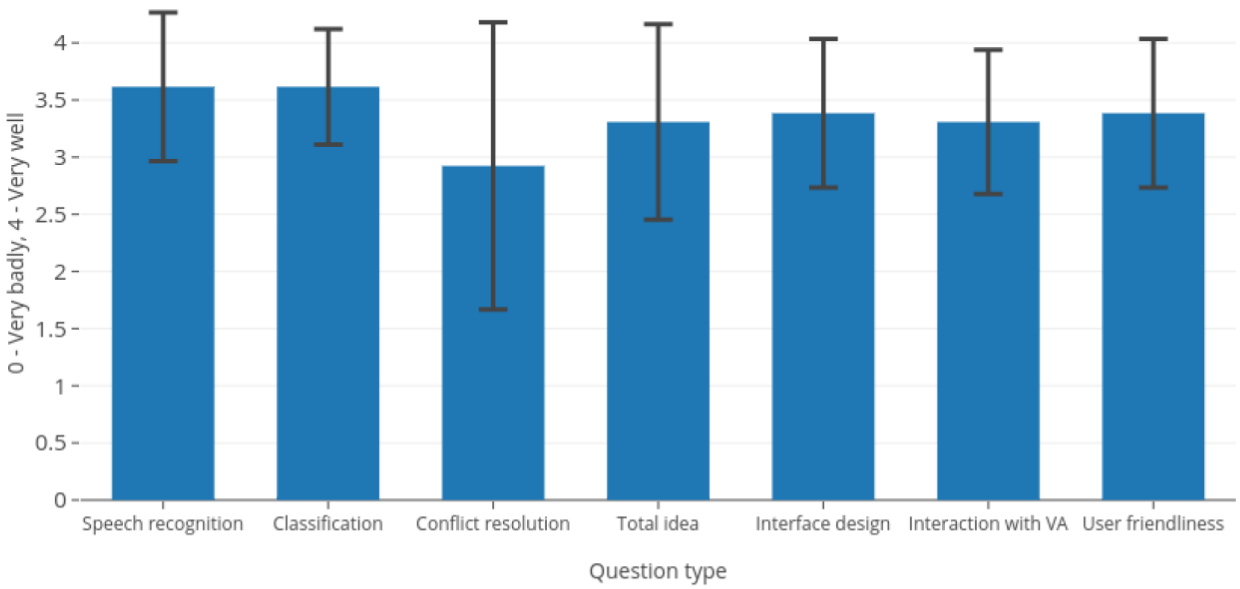

Fig. 6. Quantitative analysis of the answers

- Lack of training data: create more training cases and store them into Databases.
- Ineffective distance measurement algorithm: replace k-Nearest Neighbor algorithm with Artificial Neural Network or other machine learning algorithm. Combination of heuristic algorithms gives higher results than separately [11].
- Ineffective input matrix: replace input variables by new system that will take into account difference between reasons.

The evaluation of the preferences based conflict resolution was mostly positive with one negative answer. However, tester did not provide justification of his answer.

All three questions about design and user friendliness of the interface have no negative responses. On the other hand, there are some issues about design in the additional question box. For example, four of six responses in the additional questionnaire were about changing the voice recognition control from button to automated voice capture.

V. Conclusion and Future Work

This paper presents Virtual Assistant in the multi-user Smart Home environment. The Virtual Assistant was created to manage and monitor Smart Home environment by dialogue based interaction with users. Different machine learning algorithms and cloud technologies were implemented to meet the project objectives. The main focus of the research was to develop an adaptive context aware conflict resolution system based on CBR. The validation tests were positive.

The future direction of the work is to implement context-aware system that analyzes user behavior and reports it as a case to the database of CBR. In order to change the paradigm from technology centered perspective to human centered perspective, it is crucial to introduce behavior adapting sub-system [2]. In addition, it is recommended to implement machine learning algorithms to measure distances between cases to increase accuracy.

All the materials and sources of the system can be found in the GitHub repository of the project[8].

References

[1] M. Alirezaie, J. Renoux, U. Köckemann, A. Kristoffersson, L. Karlsson, E. Blomqvist, N. Tsiftes, T. Voigt, and A. Loutfi, “An Ontology-based Context-aware System for Smart Homes: E-care@home,” Sensors, vol. 17, no. 7, p. 1586, Jul. 2017

[2] A. Aztiria, J. Augusto, R. Basagoiti, A. Izaguirre and D. Cook, "Learning Frequent Behaviors of the Users in Intelligent Environments", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 6, pp. 1265-1278, 2013.

[3] A. Fleury, M. Vacher and N. Noury, "SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results", IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 2, pp. 274-283, 2010.

[4] R. B. Hadj, C. Hamon, S. Chollet, G. Vega, and P. Lalanda, “Context-based conflict management in pervasive platforms”, 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017.

[5] M. Kabir, M. Hoque, H. Seo and S. Yang, "Machine Learning Based Adaptive Context-Aware System for Smart Home Environment", International Journal of Smart Home, vol. 9, no. 11, pp. 55-62, 2015.

[6] N. Khan, U. Alegre, D. Kramer and J. C. Augusto, "Is ‘Context-Aware Reasoning= Case-Based Reasoning’?", International and Interdisciplinary Conference on Modeling and Using Context, Springer Cham, pp. 418-431, 2017.

[7] S. Mayer, N. Inhelder, R. Verborgh, R.Van de Walle and F. Mattern, "Configuration of smart environments made simple: Combining visual modeling with semantic metadata and reasoning", 2014 International Conference on the Internet of Things (IOT), pp. 61-66, 2014.

[8] K. Nurgaliyev, D.D. Mauro, N. Khan and J.C Augusto, "Improved Multi-user Interaction in a Smart Environment Through a Preference-Based Conflict Resolution Virtual Assistant", 2017 International Conference on Intelligent Environments (IE), pp. 100-107, 2017.

[9] C. Oguego, J. Augusto, A. Muñoz and M. Springett, "A survey on managing users’ preferences in ambient intelligence", Universal Access in the Information Society, vol. 17, no. 1, pp. 97-114, 2017.

[8] https://github.com/BiggyBaron/VirtualAssistant

[10] C. Oguego, J. Augusto, A. Muñoz and M. Springett, "Using argumentation to manage users' preferences", Future Generation Computer Systems, vol. 81, pp. 235-243, 2018.

[11] B. Ospan, "Simulation of a Simple Bio-Mimetic Robot with Neuromorphic Control System and Optimization Based on the Genetic Algorithm", International Journal of Innovations in Engineering and Technology, vol. 8, no. 4, 2017.

[12] F. Portet, M. Vacher, C. Golanski, C. Roux and B. Meillon, "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects", Personal and Ubiquitous Computing, vol. 17, no. 1, pp. 127-144, 2011.

[13] S.Ramasundaram and S.P. Victor, "Text Categorization by Backpropagation Network", International Journal of Computer Applications, vol. 6, pp. 1-5, 2010.

[14] T. Reichherzer, S. Satterfield, J. Belitsos, J. Chudzynski and L. Watson, "An Agent-Based Architecture for Sensor Data Collection and Reasoning in Smart Home Environments for Independent Living",Springer International Publishing Cham, pp. 15–20, 2016.

[15] K.Q. Weinberger and L.K. Saul, "Distance metric learning for large margin nearest neighbor classification", Journal of Machine Learning Research, pp. 207–244, 2009.

[16] S. K. Das, N. Roy, A. Roy, "Context-aware resource management in multi-inhabitant smart homes: A framework based on Nash H-learning", Pervasive and Mobile Computing, vol. 2(4), pp. 372-404, 2006.

[17] A. Kulkarni, "A Reactive Behavioral System for the Intelligent Room", Doctoral dissertation, MIT, Massachusetts, USA, 2002.

[18] C. Shin, H. Yoon, H. Woo, "User-centric conflict management for media services using personal companions", ETRI journal, vol. 29(3), pp. 311-321, 2007.

[19] R. Sharma, F. F. H. Nah, K. Sharma, T. S. S. S. Katta, N. Pang, A. Yong, "Smart Living for Elderly: Design and Human-Computer Interaction Considerations", International Conference on Human Aspects of IT for the Aged Population, Springer International Publishing, pp. 112-122, 2016.

[20] A. Ortiz, M. del Puy Carretero, D. Oyarzun, J. J. Yanguas, C. Buiza, M. F. Gonzalez, I. Etxeberria, "Elderly users in ambient intelligence: Does an avatar improve the interaction?", Universal access in ambient intelligence environments, Springer Berlin Heidelberg, pp. 99-114, 2007.

[21] D. Hawthorn, "Possible implications of aging for interface designers" Interacting with computers, vol. 12(5), pp. 507-528, 2000.

[22] P. Jia, Y. Lu, B. Wajda, "Designing for Technology Acceptance in an Ageing Society through Multi-stakeholder Collaboration", Procedia Manufacturing, vol. 3, pp. 3535-3542, 2015.

[23] S. K. S. E. A. James, I. Vales, "Personal healthcare assistant/companion in virtual world", AAAI Fall Symposium Series, 2009.