XXI IMEKO World Congress “Measurement in Research and Industry”
August 30 – September 4, 2015, Prague, Czech Republic

TESTING OF THE VOICE COMMUNICATION IN SMART HOME WITH KNX TECHNOLOGY

Jan Vanus[1], Marek Smolon[1], Radek Martinek[1], Michal Kelnar[1], Jiri Koziorek[1], Petr Bilik[1], Jan Zidek[1]

[1] VSB TU Ostrava, Department of Cybernetics and Biomedical Engineering, 17. listopadu 15, 708 33, Ostrava Poruba, Czech Republic (jan.vanus@vsb.cz, smolon.marek@gmail.com, radek.martinek@vsb.cz, michal.kelnar@vsb.cz, jiri.koziorek@vsb.cz, petr.bilik@vsb.cz, jan.zidek@vsb.cz).

Abstract – This paper focuses on the proposed method used to test the success or accuracy of voice command recognition under different conditions, which is used to control technical functions in the Smart Home system through the application of KNX technology. The work describes the development of an application used to program voice recognition, to program and start the simulation model and the actual implementation of the communication interface between different parts of the created system. The actual recognition of voice commands was done on the .NET Framework 4.0 platform using the C# programming language. To transfer information between detected voice commands and the KNX technology, machine code commands were sent over the UDP server using a KNX / IP router from the computer control to the simulation Smart Home module. At the end, 10 speakers used a microphone to test the functionality and quality of the voice command recognition system under different surrounding conditions.

Keywords: voice recognition; .NET Framework; microprocessor; synthesis; fieldbus; KNX; ETS; C#

1. INTRODUCTION

Voice commands can be used to provide comfortable control of operational technical functions in intelligent buildings [1]. In real-life implementations of the designed systems for voice communication [2], [3], [4], [5] with the control system it is necessary to resolve an issue with additive noises for the particular environment [6], [7]. Voice control in smart apartments is preferred mainly by older or handicapped fellow-citizens [8]. This paper focuses on the design and implementation of a comfortable voice control operational and technical functions in intelligent buildings with the KNX technology system. The whole work could be notionally divided into three parts, namely the development of applications for voice recognition, programming and animation has finished school model and implementation of a communication interface between these parts. The actual recognition of voice commands will be implemented on the platform. NET Framework 4.0 using C# programming language. Recognition results in machine code will be sent over UDP server using KNX / IP router from the host to the school model. After completing work on a model of intelligent building and a successful implementation of voice control is carried out functional and quality voice recognition testing. The task of the designed software G.H.O.S.T for voice control of operational technical functions of a Smart House is to provide users with easy and simple access to control of their house. This application can serve as a complement to other technologies, supporting Tele Care in Smart Home Care. For example, for determining the position of the senior citizen´s in Smart Home Care.

2. DESCRIPTION OF DEVELOPMENT ENVIRONMENT

The application has been created in “Sharp Develop” environment, which is an open source integrated development environment. After launching G.H.O.S.T application, an introductory screen with basic menu is presented to the user. In the upper part of the screen there is a button in form of an icon with a microphone, which serves for turning off and on of voice recognition. The basic menu consists of six buttons (Visualization, “Statistics”, Voice commands, Settings, Help and Close) providing switching to other screens. Each of the submenus provides the user with the possibility to control functions of smart home or the G.H.O.S.T program itself. Visualization screen (Fig. 1) - contains floor plan of the apartment controlled by KNX system displaying system status. Icon indicating status of lighting and status of drawing down or up of the blinds has been inserted to every room. Indication of blinds movement relates to bedroom, kitchen and living room, where these components are used.

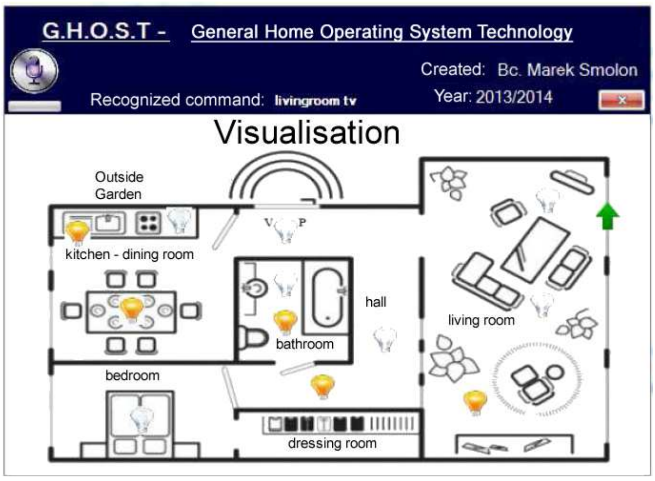
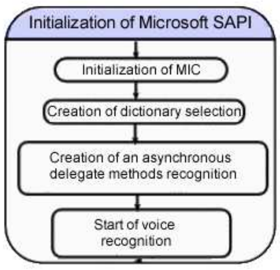

G.H.O.S.T - General Home Operating System Technology
Created: Bc. Marek Smolon
Year: 2013/2014
Recognized command: livingroom tv
Visualisation
Outside Garden
V
P
kitchen - dining room
hall
living room
bathroom
bedroom
dressing room

Fig. 1 Application appearance - Visualization Screen.

After entering a particular voice command, the application is able to evaluate the command and visualize the changes performed on this screen. Statistics screen - enables the user to test quality of voice recognition. Voice Commands screen - serves for checking the required commands. The individual recognized commands are listed in Rich Text Box component.

2.1. Application Structure and Description of Source Codes

The internal function of G.H.O.S.T application is illustrated below in form of a UML schemes. The application has been developed using object-oriented language C# on .NET 4.0 platform. The resulting application is designed in “Sharp Develop” environment so that at first the respective graphical part of the program is designed (the required components are arranged in the application window, their style and proper-ties are created) and then these graphical elements are assigned certain functions in text editor. Because the developed program uses a greater amount of graphical elements, it logically contains quite a large amount of service methods. Listing all methods would not be very synoptic in a single UML scheme. The full extent of the graphical part roughly consists of a hundred graphical components. The text part, which implements main functionality of the whole application, spreads out in about seven hundred lines of suitably commented code. Two basic classes have been set in the application. The first one is „Main Form“ class, which inherits its basic properties from „Form“ class. This class can be labelled as main, as it contains majority of methods providing program functionality. The other class named “Voice commands” has been generate only for the purpose of increasing lucidity of the written program. After the program is started, initial configuration of the program is performed using method „Main Form Load ()“. In the initialization method, import of settings from an external file is performed first and then the voice recognizer and synthesizer are set. „Main Form Load ()“ initialization method also includes a block marked as Microsoft SAPI initialization in the scheme (Fig. 2).

a) 
Initialization of Microsoft SAPI
Initialization of MIC
Creation of dictionary selection
Creation of an asynchronous delegate methods recognition
Start of voice recognition

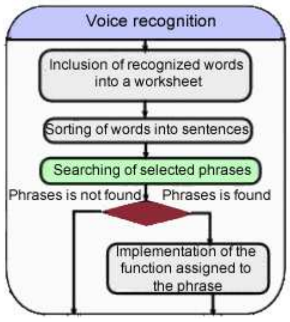
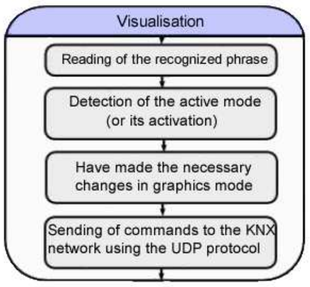
b) 
Voice recognition
Inclusion of recognized words into a worksheet
Sorting of words into sentences
Searching of selected phrases
Phrases is not found
Phrases is found
Implementation of the function assigned to the phrase

Fig. 2 a) UML scheme – Public Partial Class MainForm - Initialization of Microsoft SAPI, b) UML scheme – Public Partial Class MainForm - Voice Recognition.

Automatic selection of the used micro-phone, creation of a simple dictionary of options containing only an option for waking the system up from sleep and starting voice recognition takes place here. In the end of the method, asynchronous voice recognition is started. During recognition of the wake up phrase, the program automatically performs voice recognition method called „_speech_recognizer_recognized ()“. This method contains the key algorithm for voice recognition. Here it is possible to work with the recognized words stored in memory. A "foreach" type loop has been incorporated in the method, whose task is to store all recognized words into buffer.

The UML scheme – Public Partial Class MainForm - Visualization is described in Figure 3.

a) 
Visualisation
Reading of the recognized phrase
Detection of the active mode (or its activation)
Have made the necessary changes in graphics mode
Sending of commands to the KNX network using the UDP protocol

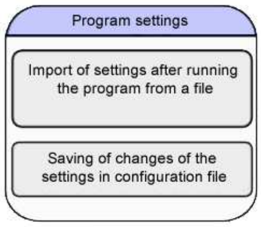
b) 
Program settings
Import of settings after running the program from a file
Saving of changes of the settings in configuration file

Fig. 3. a) UML scheme – Public Partial Class MainForm – Visualization, b) shows how the UML scheme – Public Partial Class MainForm - Program settings has to be performed.

A string type variable has been created for the buffer. Words are stored in this variable with spaces in form of sentences. This perhaps kind of unusual type of buffer has been used because in the next step only certain phrases may be searched for in sentences using contains() command. Method for phrase search is implemented in secondary class "Voice commands". If a control phrase is recognized in the buffer, the whole buffer is cleared and an operational action is performed. Choice of the operational action is created using switch-type command. After choosing and performing a certain action, the program responds to the user using the synthesizer. The described operational action is implemented using method "lights blinds change of the state ()". This method forms the basis for the visualization screen. Its basis is again a "switch" type command. Using the condition-al command „switch“, this method chooses a block of code performing the respective changes both in the visualization part and in the smart building model by sending the respective commands KNX fieldbus through UDB server.

3. INTERCONNECTION OF THE APPLICATION AND KNX COMMUNICATION FIELDBUS

Interconnection between the computer and the communication bus was achieved using a Siemens KNX IP Router N146 (Fig. 4). The described module has a KNX interface on one side and an Ethernet connector on the other side. However, when using this router, UDP protocol must be used to communicate via a network. A modem was added between the computer and the KNX IP Router in order accomplish automated assignment of IP addresses.

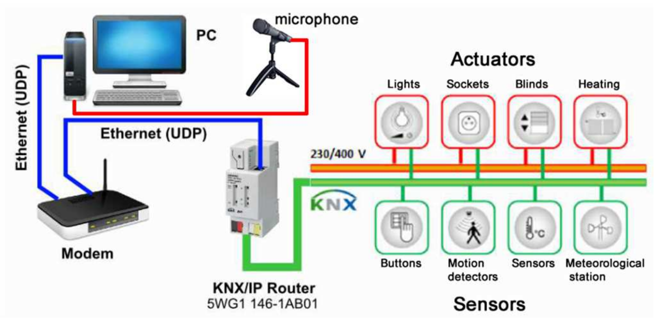


Fig. 4 Connection of the whole communication network.

Connection is very simple if is consider the wiring connection alone. Much more difficult is the need to adapt software to individual protocols (KNX and UDP). A slight disadvantage is also the fact that the used KNX/IP router module uses rather the simple UDP communication protocol, which does not guarantee transfer reliability. However, for this developed application this solution is more than sufficient. Difference between TCP and UDP protocol is described follow. TCP is a connection-oriented protocol which means that to establish "end-to-end" communication it requires so-called "handshaking" to occur between the client and server. After a connection is established, data may be transferred in both directions. The KNX technology uses its own communication protocol for communication between the individual peripheral devices. The algorithm to generate codes sent over the fieldbus would be very complicated and the whole solution would be quite cumbersome. In-put data in form of component addresses and states of the individual modules are not fully static parameters and may be changed using ETS application. That is why a more straightforward way was chosen. The used way is based on mapping IP addresses and UDP messages being sent over the fieldbus on every change on the sensor part of the system. The Wire Shark application version 1.10.5 was very helpful in this application. Only a UDP client was programmed in the developed application, which is able to send these basic messages in hexadecimal format. Therefore the application can emulate pressing of any key through software (it is able to emulate any change both in the sensor part and in the operational part of the system). This way it was possible to get around the complicated generating of messages based on variable messages and to control the whole system in the most straightforward and in principle the most natural way for the fieldbus. A total of 31 similar operational actions were mapped into the application.

4. EXPERIMENTAL PART - TESTING AND COMMAND RECOGNITION SUCCESS RATE STATISTICS

Testing is a part of the created application. After the test is completed, a notification window is displayed with information about the particular test. Although the number of well recognized commands is one hundred of one hundred, the average recognition success rate is only 94.61%. This is because the number of recognized words is a real value, while average recognition success rate represents a program-wise estimated recognition success rate. Ambient noise, which influences the very recognition, is a major factor that has a great influence on aver-age recognition success rate. 10 persons of different sex and age have participated in testing of this system. Both men and women in age range of 22 – 50 years. An integrated microphone, which is a part of a PC, and a wireless microphone (Logitech Wireless Headset H600) have been used for testing. Every speaker tested a random voice command several times. Each of the ten speakers has tested the given command, in the first case (lights on) 100 times. In order to compare results of the individual speakers, testing was performed on the same command for each of them. The result is an average percentage success rate of the tested command and real recognition success rate of the success rate.

Microphone integrated in PC (without distance, without ambient noise) – test 1

The Figure 5 shows that the achievable success of voice commands represents 100% accuracy for three commands out of 10 and 99% accuracy for the remaining 7 commands. That means that in the second scenario, out of 100 spoken commands 99 were interpreted accurately.

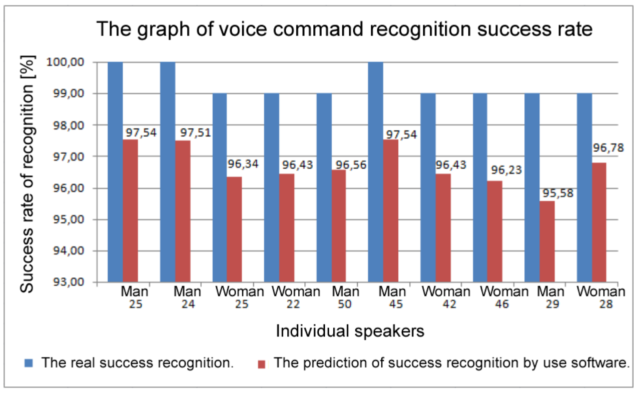


Fig. 5 Results of voice command recognition success rate - test 1.

Microphone integrated in PC (distance 3m, without ambient noise) – test 2

Similarly, further testing was done but the microphone distance from the PC was approximately 3 m. The level of success of voice recognition commands depends on the distance between the microphone and the speaker. The further the speaker is from the microphone, the lower the ability to recognize commands due to poor sensitivity of the microphone. All five speakers did the testing again using the same voice command, "living room TV." As demonstrated by figure 6, the success level of this command recognition is rather high.

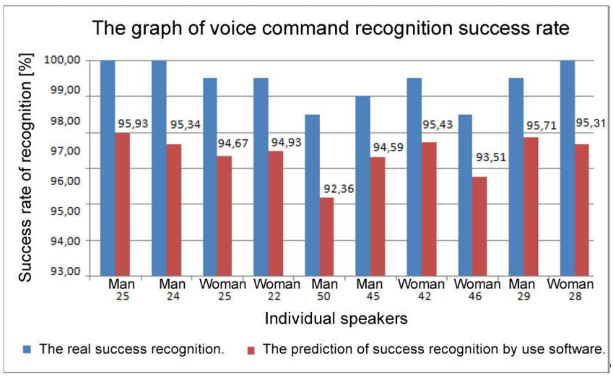


Fig. 6 Results of voice command recognition success rate - test 2.

Younger speakers achieved slightly worse results, which may be due to incorrect English pronunciation. Also, the programmable estimated recognition liability is lower, because of ambient noise detected by the microphone - the microphone is not able to catch the entire command.

Microphone integrated in PC (with ambient noise) – test 3

Because it is very likely that there will never be complete silence in the building, where the system will be installed, testing together with the presence of ambient noise was done. The TV set and radio were ON. The ability to recognize voice commands is shown in figure 7. In this test the programmable estimated reliability of voice recognition was significantly lower, mainly due to the ambient noise that was used during this test (radio and TV). It did not, however, have a great impact on the actual recognition of the voice command.

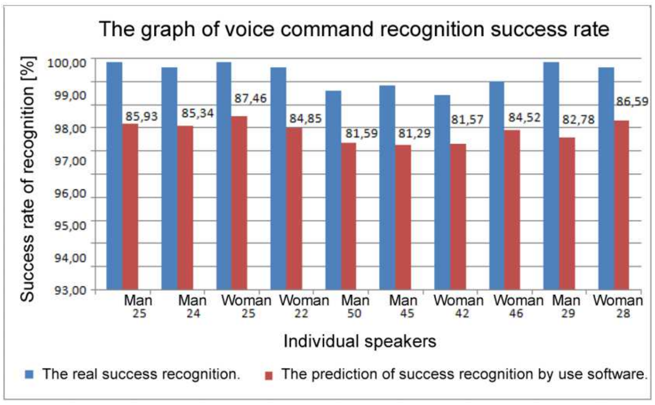


Fig. 7 Results of voice command recognition success rate - test 3.

Wireless microphone (without ambient noise) – test 4

Testing was also done with a wireless microphone (Fig.8). This solution offers advantages that an integrated PC microphone does not have. The main advantage is that our movement around the apartment or house is not limited. Using a suitable wireless microphone allows us to move freely and control operational and technical functions (lighting, blinds).

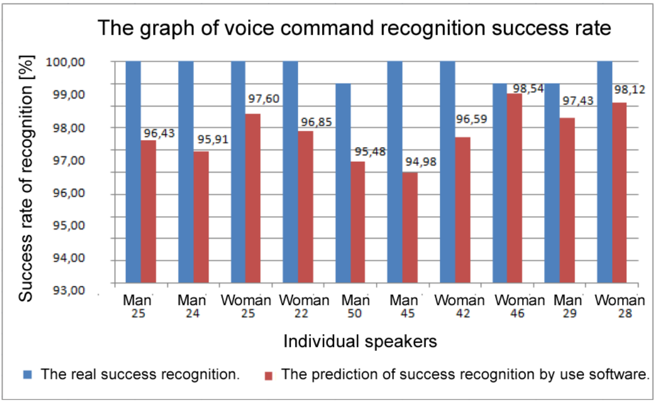


Fig. 8 Results of voice command recognition success rate - test 4.

The only restriction is the range of the selected wireless microphone. All five speakers did the testing again using the "shower" voice command. Using a portable microphone gave us almost a hundred percent success rate. This could be the result of several factors. One of the most important factors is the use of a high-quality microphone, mainly due to the fact that the user has the microphone very near to his mouth. The lower percentage for programmable estimated reliability may also be due to the incorrect pronunciation of certain speakers. Even in this case, the success rate of voice command recognition is very high despite the fact that testing was done with the presence of ambient noise. This is mainly due to the use of a high-quality microphone and due to the fact that the speakers were very close to the microphone.

4. CONCLUSIONS

The designed and implemented solution works according to the requirements. The indisputable advantage is zero cost, provided that the user is running Windows Vista (or higher) in the English version. As evident from the success achieved by the proposed voice recognition communication method, the voice recognition feature works very well. Ten people participated in the testing. Each person conducted five tests and uttered 100 voice commands. The average success rate of real voice recognition based on all tests is approximately 98.78% (note that interference, noise or greater distance between the speaker and the microphone were used during certain tests). Some minor errors occasionally occurred when the application was in idle mode (waiting for the activation phrase) and a conversation in Czech language was taking place in the room. This problem is eliminated by adding a restrictive conditions to the program, which accepts the initiation phrase if the application is at least 90% sure that the phrase was correctly received/said. Despite this, the program may sometimes recognize a Czech word as the actual initiation phrase. As a possible improvement - when used in the real world, it is advisable to use a microphone network covering the entire area of the building.

ACKNOWLEDGMENTS

This paper has been elaborated in the framework of the project Opportunity for young researchers, reg. no. CZ.1.07/2.3.00/30.0016, supported by Operational Programme Education for Competitiveness and co-financed by the European Social Fund and the state budget of the Czech Republic. This work is partially supported by the Science and Research Fund 2014 of the Moravia-Silesian Region, Czech Republic and projects SP2015/154, SP2015/181 of Student Grant System, VSB-TU Ostrava.

REFERENCES

[1] H. Merz, T. Hansenmann and Ch. Hubener, *Automated systems for buildings: Communication Systems KNX/EIB, LON a BACnet*, Grada Publishing, Prague, 2008.
H. Merz, T. Hansenmann and Ch. Hubener, *Automatizované systémy budov: Sdělovací systémy KNX/EIB, LON a BACnet*, Grada Publishing, Praha, 2008. [Czech]

[2] K. H. Park, Z. Bien, J. J. Lee, B. K. Kim, J. T. Lim, J. O. Kim, *et al.*, "Robotic smart house to assist people with movement disabilities," *Autonomous Robots*, vol. 22, pp. 183-198, Febuary 2007.

[3] C. L. Hsu and K. Y. Chen, "Practical design of intelligent remote-controller with speech-recognition and self-learning function", *Machine Learning and Cybernetics*, vol.6, pp. 3361-3368, July 2009.

[4] S. Soda, M. Nakamura, S. Matsumoto, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "Implementing Virtual Agent as an Interface for Smart Home Voice Control", *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*, vol. 1, pp. 342-345, December 2012.

[5] J. Vanus, M. Smolon, J. Koziorek and R. Martinek, "Voice control of technical functions in smart home with KNX technology", *Lecture Notes in Electrical Engineering*, vol. 330, pp. 455-462, January 2015.

[6] J. Vanus and V. Styskala, "Application of variations of the LMS adaptive filter for voice communications with control system", *Tehnicki Vjesnik-Technical Gazette*, vol. 18, pp. 553 560, December 2011.

[7] R. Martinek and J. Zidek, "Use of Adaptive Filtering for Noise Reduction in Communication systems", *The International Conference Applied Electronics (AE)*. pp. 215-220, Pilsen, Czech Republic, September 2010.

[8] J. Vanus, J. Koziorek, and R. Hercik, "The Design of the Voice Communication in Smart Home Care", *Telecommunications and Signal Processing (TSP)*, pp. 561-564, July 2013.