# Cloud Based Wizard of Oz as a Service

## *(Theoretical, experimental analysis of WoOz on Azure Cloud Environment for Intelligent Robotics)*

Peter Sinčák, Eva Novotna, Tomáš Cadrik,
Gergely Magyar, Marian Mach, Maria Vircikova
Department of Cybernetics and Artificial Intelligence
Technical University of Kosice
Slovakia
peter.sincak@tuke.sk

Filippo Cavallo, Manuelle Bonaccorsi
Scuola Superiore Santana, Pisa Italy
Institute of Biorobotics
Ambient Assistant Living Laboratory
Italy
f.cavallo@sssup.it

***Abstract*—The paper deals with theoretical and experimental issues of an idea towards computer Cloud Based Wizard of Oz on the Microsoft Azure Cloud environment. Wizard of Oz is a common tool in Social Robotics and especially in special applications like mental illness treatment, Ambient Assisted Living, and many other applications. The final goal is to create a system with the ability to learn and replace a human wizard by intelligent software Agent, which simulate a behavior or the human.**

***Keywords*—*Wizard of Oz, Cloud Computing, Remote Brain concept, Mother and Agent, Cloud Robotics, Software as a Service, intelligent agent, human behavior***

## I. INTRODUCTION

Cloud Computing is not new technology but under rapid development. The overall influence of this technology is very high. A very nice and interesting study related to various applications e.g. to managing intelligent buildings [1]. Also, Cloud robotics has been introduced in 2010 [2] but the original idea of the remote brain introduced in early 1997 by prof. Inaba [3] in Japan. The concept of the remote brain was taking into consideration the virtual environment, virtual Body, and preview procedure (see fig 1). This idea completely fits current technological possibilities of Cloud computing towards Cloud Robotics. This concept from 1997 even thinks about Mother and Agent and inheritance of form knowledge of the systems. By this approach, the "Mother" a Cloud, which, in fact, can "think over" and simulate various aspects of real action over the real body and real environment. The Evaluation is a key component that considered as decision resulting in teaching and knowledge acquisition.

The overall Cloud robotics concept rely on the latency of communication between a computer and Cloud. The main intention of this paper was to investigate it for purposes to apply it in Wizard of Oz application. This fact should lead towards to possibility to create this as a Service so the user can connect to the Web page and connect a robot and does not have to install some things on his computer. The testing for this idea was and NAO robotic platform. The response time has been calculated. The similar research for the direct application in Ambient Assisted living in [4]. The results also confirm that there is a technological potential for this solution.

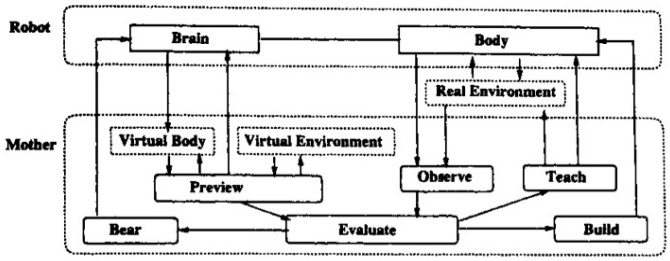


Fig. 1. The basic concept of Remote Brain Robot from 1997 from prof. Ibana [3]

The basic concept of Wizard of Oz comes from social robotics, but it is just another notion for teleoperation of the robot by a human. The origin of this name come from 1900 when American writer L.F. Baum wrote some the book for children about the fantasy Land of Oz. The fact is that social robotic domains specified a Wizard of Oz based on book story on a person Tinman on fig 2. from Wizard of Oz story.

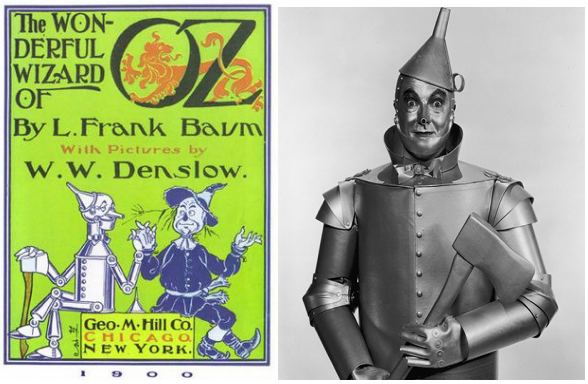


Fig. 2. The book which Inspired Wizard of Oz Notion in Social Robotics Tinman is Half Man Half robot, so the major concept of Human operating a Robots is behind this assumption.

We can state that this approach is a bit over hyped, and the rest of IT community calls Wizard of Oz as a Teleoperation and it is not very known outside of Social Robotics or Human-Robot Interaction field. The major focus is how to create a flexible WoOz with general purpose and under current technological situation influenced mainly with Cloud Computing. The following parts of the paper describe the technological background of such an idea using Cloud for Wizard of Oz application as it is on fig. 3.

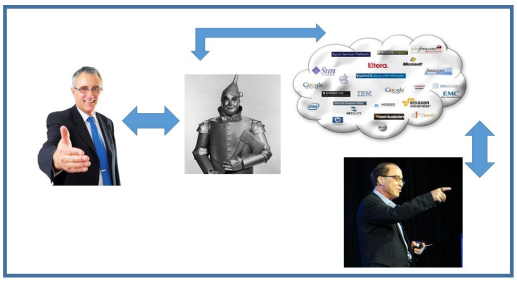

Fig. 3. The basic technological concept of Cloud Based Wizard of Oz on Human-robot Interaction scenario.

## II. Selected Communication Methods

For the purpose of testing whether it is possible to create the cloud-based Wizard of Oz were three technologies investigated. All of those methods are utilizing the WebSocket technologies. The methods are Windows Communication Foundation (WCF), SignalR and standard usage of the WebSocket technology.

### A. *Windows Communication Foundation*

WCF [5] is a programming model created by Microsoft Corporation. This model can be used for creating the service-oriented applications (SOA). It supports many types of transports, messaging patterns, encodings, network topologies and hosting models. WCF allows to send data in the form of asynchronous messages from one service endpoint to another. The endpoint may be a client that requests data from the service endpoint. WCF uses standard Http ports 80 and 443.

WCF includes two bindings: NetHttpBinding and NetHttpsBinding, which support communication via Websockets. These connections use Websocket services, and they can recognize whether they are request-answer or duplex and according to the information it can change its function. WCF service can simply react to incoming messages, which it receives and acts based on predefined actions. A goof example is reading from the database, database recording, responding to client message or sending a message to another service.It offers one or more endpoints, and it can communicate with one or more clients at the same time

.

### B. *SignalR technology*

SignalR [6] is a library designed for ASP.NET developers, which allows to create interactive, multiuser real-time applications with asynchronous techniques. Real-time, in this case, means that the server can send data to the client without refreshing the website.

SignalR separates the programmer from the low-level details and brings the experience of continual connection between client and server. It ensures one of the best accessible technology for server, as well as a client for the continual duplex communication. This technology offers two levels of abstraction over transport protocols that are used for keeping a connection with the server. These levels are characterized by two different kinds of API for virtual connections. We used for experiments the Persistent connection, which is a strong connection with low latency using full-duplex communication.

Creation of the connection between client and server consists of the following steps:

- Client sends the request, server answers the request,
- client uses the answer in order to choose the best connection,
- client sends the connection request with chosen connection type and immediately when server receives connection request, the connection is set up.
- The client is allowed to open and close the connection as well as send and receive any data.

### C. *Standard Websocket Method*

Websocket connection [7] is a bi-directional and fully duplex communication type that uses simple TCP protocol. It begins as a standard Http connection, which upgrade itself to a Websocket connection. It is necessary to negotiate a connection between client and server. After the successful handshake, it is possible to use communication between the client and server. Communication can close thanks to controlling frame named close. The connection created by Websockets uses the same ports like protocols Http and Https: 40 and 433. Frames from client to the server are hidden because of security reasons, and it does not matter whether the Websocket protocol runs on TSL, or not. If the server receives unmasked frame, it must immediately close the connection. The server cannot mask any frame. When the client receives a masked frame, he must close the connection. When using Websockets, we can decrease the latency because the server can send data right away as soon are they possible Experiments

There were experiments on the Microsoft Azure [8] cloud platform. On this platform were two types of services created: A chat service and a wizard of Ozz service.

The chat function is used for calculating latency when a user or a robot send a string and when he or it receives the answer. A service simply received a string and modified this string. The string was then sent back to the client.

The Wizard of Ozz service contained a web page with four buttons. When a user clicks on one of those buttons, the chosen movement is saved to Azure storage concretely queue storage. When a queue has a movement, this is sent to the client and deleted from the storage. Then the client (robot) will execute this movement.

Each of the three methods was tested. The testing robotic platform was the humanoid robot Nao. For each communication method were two cloud services created in the C# language. Also, two clients in the C# language for each communication method were created. The standard Websocket method client was also created in the Python programming language. The Python client was created only for this method because the SignalR library for python is not created yet. The Library for WCF was able to use only the standard SOAP protocol and not communication using Websockets.

Each experiment consist of a graph with four types of data: the minimal value how long the client receives information from the cloud, the maximal value, the average value, and the modus. All the times that are showed in the next graphs are measured in milliseconds. Each connection process was done 50 times, and the results are the statistics from those connections. Those and more experiments were firstly done and described in [9].

### D. *First Experiment*

The purpose of the first experiment was to compare three communication methods from the view of speed using the chat client. Each client had a measuring mechanism, which stored the time when the client send the request to the cloud and when it receives the modified string.

The results of the first experiment are showed in Fig. 1. As we can see from this graph, the average of the first three methods is very similar. The problem is with the Python client, which is worse than other methods. Also, statistically for this client is much worse than the other clients. What is interesting, the python client has the best minimal time (only 25 milliseconds), and according to the maximal time it was not the worst.

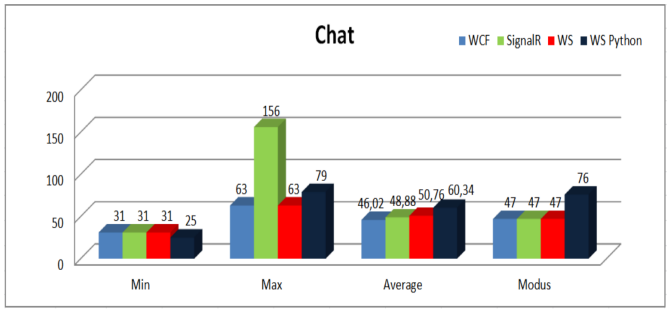


Fig. 4 Comparison of three communication methods using the chat service. All values are measured in milliseconds.

### E. *Second Experiment*

The purpose of the second experiment was to test the Wizard of Ozz when using one robot connected to the cloud service. To measure the response time, the mechanism that saves the time when the robot sends a request and receives data. If no movement is selected, the client receives the string with value "Empty". The results of the experiments are shown in the next figure.

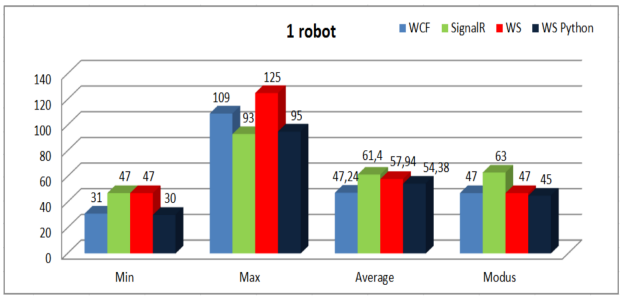


Fig. 5 Comparison of three communication methods while communicating with one robot. All values are in milliseconds.

The WCF service seems to be the best from all the tested methods. Now the python client was not the worst one. According to the modus and minimal value, it was the fastest method. According to another statistic, the Python client is the second best. The times are worse than in the chat service because there is a time needed to access the cloud storage. However, what is interesting, the time when using WCF and the standard Websocket method was not affected by this need. Python client was even better as with the chat service. It can be because of the Linux operating system in the Nao robot, which is more native for the usage of the Python language.

### F. *Third Experiment*

The third experiment was similar to the second one. The difference was that we used two Nao robots instead of one. The clients are communicating with the cloud service at the same time. The results represented on the outcomes from all of the clients. The results of the experiments are in Fig. 3.

As we can see from this figure, the results are very similar as they were in the case when we used only one robot. What is interesting, the minimal value when using the Python client is only ten milliseconds. It can be a coincident because it seems to be too low. The maximal value is also best for this Python client. However, the most important information such as average is the standard Websocket client in C# the best.

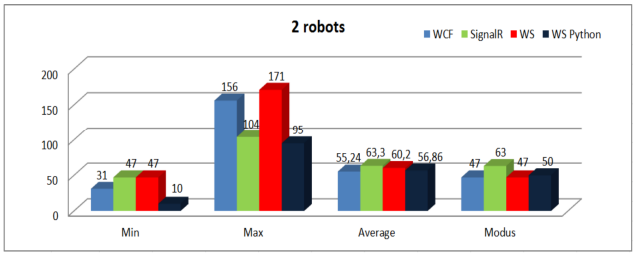


Fig. 6 Comparison of three communication methods while communicating with two robots. All values are in milliseconds.

When testing the combination of two robots connected to one cloud service, there is one interesting feature. Each robot has its communication channel. So when we are using only one queue in the Azure storage, the fact that robot will connect in the concrete time is asynchronous. That means that we do not know which robot will download the selected action in which time. It is necessary to solve this problem, for instance where each robot will have its storage queue, or where each robot can have a configuration number, and the selected actions will be related to this number.

## III. It is Possible to Create a Cloud Based Wizard of Ozz?

The experiments in the previous section showed that it is possible to create a Wizard of Ozz, which will have the main core build on the cloud technology. The average time between the request and the response is sufficient for this purpose. The real-time control of the robot is not possible because there is a delay, but for the teleoperation needs it is usable.

The teleoperator (Wizard) will have access to a website, which will contain the buttons for controlling robots. Which button will be pressed, that action will be executed. What is important, this teleoperation system can be from any place in the world.

When we are comparing the methods used in the section with experiments, we can say which method is the best for our purpose. The best choices are WCF and standard Websocket communication using the python client. WCF is the clear choice because this method is the fastest one. The choice why to use the standard Websocket method is that Python is very popular with the combination of robotics.

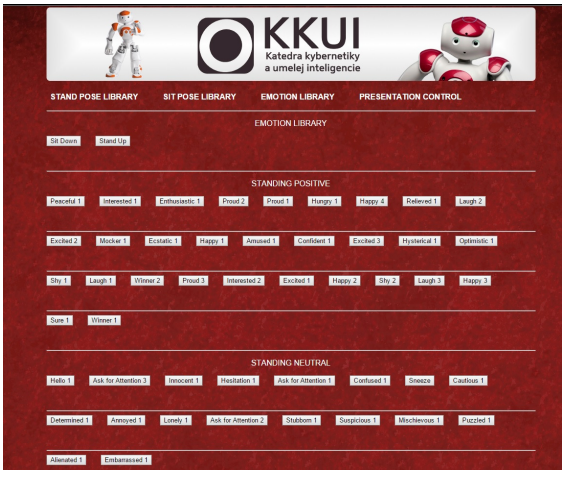


Fig. 7 An example of Cloud Based Wizard of Oz User Interface on Microsoft Azure Environment for Nao Robotic Platform.

.

The testing scenario is on Nao robot platform, and it is also expected to be compatible with Pepper robot and other platforms namely based on ROS. The testing scenario assumes that there are some different scenarios and it will also based on the acting robot if the robot has its predefined behavior on the board of the robot or the Cloud. We can state that both possibilities have been tested. There is a user interface for Human-Robot Interaction user in figure 7. That means, in the fact that the expert (Wizard) does not have to install any software just go to the web page and connect an acting robot. An example is in figure 7.

The major conclusion of this research is that Cloud Based Wizard of Oz is technologically possible and Cloud Technology. The study confirms (based on response times) that Cloud Robotics will be creating and enormous potential for social Robotics. Re-using and sharing knowledge toward learning based Wizard of Oz for some robots using Crowdsourcing and Internet knowledge is a trend in future robot like society

## Acknowledgment

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, ITMS: 26220220182, supported by the Research & Development Operational Program funded by the ERDF in Technical University of Kosice, Slovakia.

## References

[1] C. Talon , The Impact of Cloud Computing on the Development of Intelligent Buildings, CABA's Intelligent & Integrated Buildings Council (IIBC) White Paper Sub-Committee, IDG Energy Insights, 2013

[2] J. J. Kuffner, Cloud-Enabled Humanoid Robots, IEEE-RAS International Conference on Humanoid Robotics. 2010.

[3] M. Inaba, Remote-Brained Robots, International Joint Conference on Artificial Intelligence, 1997

[4] M. Aquilano at al.: Ambient Assisted Living and aging: preliminary results of RITA project, Conf Proc IEEE Eng Med Biol Soc. 2012;2012:5823-6. doi: 10.1109/EMBC.2012.6347318.

[5] M. Liu, "WCF 4.5 Multi-Layer Services Development with Entity Framework: Third Edition," Birmingham, 2012. 373 p. ISBN 978-1-84968-766-9

[6] J. M. Aguilar, "SignalR Programming in Microsoft ASP.NET," Washington, 2014. 243 p. ISBN 978-0-7356-8388-4A. Biere et al, Hanbook of Satisfiability, Fontiers in Artificial Intelligence and Applications, vol. 185, IOS Press, Amsterdam, 2009.

[7] V. Pterneas, "Getting Started with HTML5 WebSocket Programming: Develop and deploy your first secure and scalable real-time web application," Birmingham, 2013. 96 p. ISBN 978 -1-78216-696-2

[8] Microsoft Azure, Available online: "https://azure.microsoft.com/en-us/".

[9] E. Novotná, "Connection of robot to the cloud using websocket technology", Diploma thesis, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovak Republic, 2015, 67 pages.

[10] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.