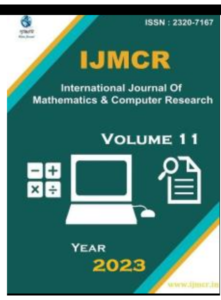
International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 11 Issue 04 April 2023, Page no. – 3371-3379

Index Copernicus ICV: 57.55, Impact Factor: 7.362

DOI: 10.47191/ijmcr/v11i4.10



Analysis of Interactive Voice Servers and Paraphrases: Application to the Consultation of Results

KABAMBA LUBANGI Nico
Faculty of Computer Science / Our Lady of Lomami University

ARTICLE INFO	ABSTRACT
Published Online: **27April 2023** Corresponding Author: **KABAMBA LUBANGI Nico**	The introduction of voice as an additional form of communication to deploy applications has certainly revolutionized the world of computing and this constitutes a means of accessing information without a computer. This article deals with the problem of the oral modality as a means of interaction for a priori increased usability of the interactive system. We will present a typology of systems where the use of speech technologies could represent a real contribution for users in the communication between man and machine, and we will discuss the issue of voice interaction. Then, we will describe an application case based on the study of the implementation of an IVS for the consultation of the results of the examinations.

KEYWORDS: Interactive voice servers, paraphrases, pattern recognition, voice synthesizer, unsupervised learning, neural network, hebb's law, perceptron algorithm, voice telematics, exam results.

1. INTRODUCTION

Human-machine voice systems are increasingly present around us in the world of cellular telephony, whether to inform a subscriber of the credit he has or to give him an urgent communication. To exchange information with a machine, they offer a natural interface for humans, not occupying the hands permanently and requiring no training. They also make it possible to reduce the cost of systematic tasks and to associate a voice identity with a service.

The main objective of this article is to demonstrate how it is possible to make computer applications of this kind use by anyone with any type of telephone to interactively access remote information systems (in our case, access to test results). Indeed, using another classic approach (internet or other computer program) requires a minimum of knowledge and the question of how to allow access to all information resources has been raised and several resolution initiatives have been developed. These generally revolve around information and communication technologies which, despite their high-speed propagation, establishes a digital divide between landlocked areas which still remain disadvantaged.

Thus the solution proposed here makes it possible to improve communication between the information system and its users through the analysis and design of the human-machine interface.

In this work, it is therefore a question on the one hand of establishing a state of the art on interactive voice servers and of studying some solutions that can allow us to set up a voice server. On the other hand, designing and producing initially a voice application allowing the interactive consultation of the results of the examinations before proceeding in a second step to the integration of a voice synthesizer in order to improve that proposed by our solution.

2. PATTERN RECOGNITION

Pattern recognition (or sometimes pattern recognition) is a set of techniques and methods aimed at identifying computer patterns from raw data in order to make a decision depending on the category assigned to this pattern[1]. It is considered to be a branch of artificial intelligence that makes extensive use of machine learning[2] techniques and statistics.

[1] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern classification*, Wiley-interscience, 2001 (ISBN 0-471-05669-3)

[2] L'**apprentissage automatique** (*machine learning* en anglais), champ d'étude de l'intelligence artificielle, concerne la conception,

The word shape is in the very general sense, it is not just about geometric shape.

The shapes or patterns to be recognized can be of very varied natures. It can be visual content (bar code, face, fingerprint...) or sound (speech recognition), medical images (X-ray, EEG, MRI...) or multispectral (satellite images) and many others.

a. Methods

Pattern recognition can be performed using various machine learning algorithms such as: neural[3] network, statistical analysis, use of hidden[4] Markov models, search for isomorphism of graphs or sub-graphs.

The shapes sought can be geometric shapes, describable by a mathematical formula, such as: the circle or the ellipse, the Bézier curves, the splines, the line.

They can also be of a more complex nature: letter, number or fingerprint.

Recognition algorithms can work on black and white images, with white outlines of objects in the image. These images are the result of edge detection algorithms. They can also work on predefined areas of the image resulting from the segmentation of the image.

A well-known algorithm for pattern detection is the Hough[5] transform, which is an algorithm based on the parametric estimation method.

b. Some pattern recognition algorithms

In learning, the neural network uses examples to adapt to the problem. He can modify:

- The weights of its connections;
- Its architecture is – in other words – creating or eliminating neurons, connections.

Modifying neuron transition functions

There are two major classes of learning algorithms: supervised learning and unsupervised learning.

i. Unsupervised learning

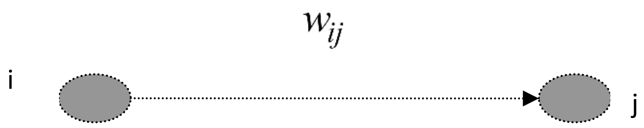
We consider examples $X^1, X^2, \ldots, X^n$; an architecture A and W_0 initial weights.

The problem is to find weights W such that:

- the examples are correctly grouped;
- the generalization is correct: the new examples are assigned to the cluster correctly.

Example of Unsupervised Learning: Hebb's Law

Hebb's law (1949) applies to connections between neurons, as shown in the following figure:



i is the upstream neuron, j the downstream neuron and w_{ij} the weight of the connection.

Hebb's law is expressed as follows: "If 2 cells are activated at the same time then the strength of the connection increases".

The change in weight depends on the co-activation of pre-synaptic and post-synaptic neurons, as shown in the following table. and being respectively the activation values of neurons i and j, and ∂w_{ij} (partial derivative of the weight) corresponding to the change in weight made.

Table 1: Hebb's Law

x_i	x_j	∂w_{ij}
0	0	0
0	1	0
1	0	0
1	1	+

Hebb's law can be modeled by the following equations, assuming that w_{ij} (t+1) denotes the new weight and w_{ij} (t), old weight:

$$w_{ij}\ (t+1) = w_{ij}\ (t) + \partial w_{ij}\ (t)$$

∂w_{ij} (t) = $x_i \ . \ x_j$ (Co-activity is modeled as the product of the two activation values).

The learning algorithm iteratively (little by little) modifies the weights to adapt the response obtained to the desired response. It is in fact a question of modifying the weights when there is an error only.

Step 1

Initialization of the weights and the threshold S to randomly chosen (small) values.

Step 2

Presentation of an entry $E_1 = (e_1 \ , \ldots, e_n)$ of the learning base.

Step 3

l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou impossibles à remplir par des moyens algorithmiques plus classiques.

[3] Un **réseau de neurones artificiels** est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques.

[4] Un **modèle de Markov caché** (MMC) — en anglais *Hidden Markov Models* (HMM) (ou plus correctement, mais non employé **automate de Markov à états cachés**) est un modèle statistique dans lequel le système modélisé est supposé être un processus markovien de paramètres inconnus.

[5] La **transformée de Hough** est une technique de reconnaissance de formes inventée en 1962 par Paul Hough, utilisée dans le traitement d'images numériques. L'application la plus simple permet de détecter les lignes présentes dans une image, mais des modifications peuvent être apportées à cette technique pour détecter d'autres formes géométriques : c'est la *transformée généralisée de Hough* développée par Richard Duda et Peter Hart en 1972

Calculation of the obtained output x for this input: $a = \sum (w_i - e_i) - S$ (the threshold value is introduced here in the calculation of the weighted sum).

$X = Sign(a)$ (If a > 0 then x = +1 else a ≤ 0 then x = -1).

Step 4

If the output x is different from the desired output for this input example then modification of the weights (μ is a positive constant, which specifies the step of modification of the weights):

$$W_{ij}\ (t+1) = W_{ij}\ (t) + \mu(\ x_i\ .\ x_j\ )$$

As long as all the examples of the learning base are not processed correctly (i.e. modification of the weights), return to step 2.

Example of application of Hebb's learning algorithm:

We choose for the neurons, a binary behavior. The entrees e_1 and e_2 are considered neurons.

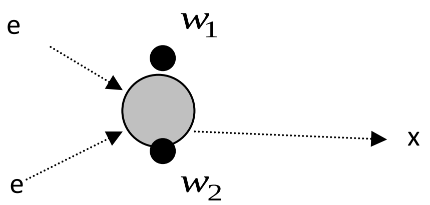


Figure 1. Example application of Hebb's algorithm

We thus obtain three neurons.

We consider as a learning base, the data of the following **table:**

e_1	e_2	x	
1	1	1	E_1
1	-1	1	E_2
-1	1	-1	E_3
-1	-1	-1	E_4

Step 1

$\mu = +1$, the weights and the threshold S are zero

Step 2

We consider the entry $E_1 = (1,1)$

of the learning base: example 1 corresponding to the first line.

Step 3

Calculation of the obtained output x for this input: a = $w_1.e_1 + w_2.e_2 - S$ = $0.0.1+0.0.1\text{-}0.0 = 0$

D'où $a \leq 0$. *Donc* $x = -1$

Step 4

The output x = -1 is different from the desired output $d_1 = 1$ for this example entry E_1 then we must modify the weights by applying.

$w_1 = w_1 + e_1.x = 0.0 + 1.1 = 1$

$w_2 = w_2 + e_2.x = 0.0 + 1.1 = 1$

Step 5

Go back to step 2

Step 2

We go to example 2, the input $E_2 = (1,\text{-}1)$

Step 3

Calculation of the obtained output x for this input:

$a = w_1.e_1 + w_2.e_2 - S = 1.1\text{+}1.\text{-}1\text{-}\ 0.0 = 0$

From where a ≤ 0. So x = -1.

Step 4

the output x = -1 is different from the desired output $d_1 = 1$ for this example entry E_2 then we must modify the weights by applying.

$w_1 = w_1 + e_1$.x = 1 + 1.1 = 2

$w_2 = w_2 + e_2$.x = 1 + 1.-1 = 0

Step 5

Go back to step 2

…/ The entrance or the entrance E_3 (third line) is correctly processed: a = -2 et x = -1 (the output is good).

We pass directly, without modification of the weights at the entrance E_4 (4th line). This one is also handled correctly. We then return to the beginning of the learning database: the entry E_1.

It is correctly treated, as well as the second E_2. The learning algorithm is then complete: the entire learning base has been reviewed without modifying the weights.

ii. Supervised learning

We consider examples $(X^1, Y^1), (X^2, Y^2)$, …, (X^n, Y^n); an architecture A and W_0 initial weights. initial weights. The problem is to find weights W such that $Y^k = F_{A,W}(x^k)$

We use multilayer networks, because the result may not correspond to Y^k.

As a reminder, a multilayer network includes external layers (input and output), internal or hidden layers (for intermediate processing). The problem is therefore to associate X^k à Y^k, k = 1, …, n.

Inputs are presented on the first layer and are propagated according to the law $X_i = f(A_i)$ avec $A_i = \sum W_{ij} X_j$

and S^k is the computed output on the last layer.

Gradient backpropagation minimizes the error between the desired output (Y^k) and the calculated output (S^k). This error can be calculated as follows:

$$e(W) = \frac{1}{m}\sum_{k=1}^{m}\left\|S^k - Y^k\right\|^2$$

$$= \frac{1}{m}\sum_{k=1}^{m}\left\|S_s^k - Y_s^k\right\|^2 \quad \text{with s} \in \text{output layer.}$$

Supervised Learning Example: The Perceptron Rule

Hebb's rule presented above does not apply in some cases, although a solution exists. Another learning algorithm has therefore been proposed, which takes into account the error observed at the output.

The Perceptron algorithm is similar to the one used for Hebb's law. The differences are in the modification of the weights.

Perceptron Algorithm

Step 1

Initialization of the weights and the threshold S to randomly chosen (small) values.

Step 2

Presentation of an entry $E_1 = (e_1, \ldots, e_n)$ of the learning base.

Step 3

Calculation of the obtained output x for this input: $a = \sum(w_i - e_i) - S$ [∑(wi*ei)-S] (the threshold value is introduced here in the calculation of the weighted sum). $x = sign(a)$ (If a > 0 then x = +1 else a ≤ 0 then x = -1)

Step 4

If the Perceptron output x is different from the desired output d_1 for this example entry E_1 then modification of the weights (μ is a positive constant, which specifies the step of modification of the weights) :

W_i (t+1) = W_i (t) + μ((d_1 - x) e_i) [correction d_i not d_1, the index i is linked to the entry concerned and not to the index of W]

Reminder: d_1 = +1 if E is class 1, d_1 = -1 1 if E is class 2 and (d_1 - x) is an estimate of the error.

Step 5

As long as all the examples of the learning base are not processed correctly (i.e. modification of the weights), return to step 2

Example of how the Perceptron algorithm works

We take as a basis for learning

e_1	e_2	d	E
1	1	1	E_1
-1	1	-1	E_2
-1	-1	-1	E_3
1	-1	-1	E_4

Table 3: Learning base

Step 1: Initial conditions: w_1 = -0,2, w_2 = +0,1, μ = 0,1, S = 0.2

Step 2: We consider the input E_1 = (1,1) of the learning base: example 1 corresponding to the first line. a(1) = -0.2 -0.1 = -0.3<0 ;

Step 3: Calculation of the output obtained x for this input: x = -1≠1 (the desired output d(1) = +1, hence the modification of the weights).

Step 4

w_1 = -0,2 +0, 1.(1 + 1). (+1) = 0

w_2 = +0,1 + 0,1.(1 + 1). (+1) = 0

Step 5: We go back to step 2

Step 2: We consider the input E_2 = (-1,1) of the learning base: example 2 corresponding to the second line. a(2) = 0+0,3-0.2 = = +0.1>0

x(2) = +1 ≠-1 (the desired output d(2)= -1, hence modification of the weights)

FAKE

Step 4:

w_1 = 0 + 0,1.(-1 - 1). (-1) = +0,2

w_2 = +0,3 + 0,1.(-1 - 1). (+1) = +0,1

E3=(-1,-1)

a(3) = 0.2(-1)- 0,1(-1) -0,2 = = - 0.5<0 x=-1 OK

E4=(1,-1)

a(4) = +0.2+ 0,1(-1) -0,2 = = - 0.1<0 x=-1 OK

E1=(1,1)

a(1) = +0.2+ 0,1 -0,2 = = + 0.1>0 x=1 OK

E2=(-1,1)

a(2) = -0.2+ 0,1 -0,2 = = - 0.3. x=-1 OK

The Algo stops

Step 5: All the base examples have been correctly processed, learning is complete.

3. Interactive voice servers

An Interactive Voice Response (IVR) is a platform, a software and/or hardware package, which serves as a voice answering machine controlled by all the telephone keys or voice recognition technologies, allowing automatically exchange various information (messages, faxes, etc.). Access to this application is made by a simple telephone number or by a specialized number whose request is made with a telephone operator. More specifically, an IVR is responsible for delivering short and simple information.

The IVR is a vocal server allowing to query a database in the Information System (IS), it is interactive when a response to a question asked by the caller leads to a particular treatment on the part of the server. , hence its name[6].

IVRs allow users to interact with an information system using the spoken modality, often via a telephone.

For example, IVRs are used in industry for:

- the orientation of telephone traffic. They attempt to identify the most appropriate correspondent for the user;
- automatic appointment scheduling;
- automatic assistance;
- automatic ticket reservation;
- etc.

Basic types of use of an IVR

IVRs are grouped into three standard types of use: the information terminal, the telephone switchboard and the use of an IVR in conjunction with a database or CTI (Telephony-Computer Coupling).

- **The information terminal**

This function constitutes the basic function of an IVR. It allows the caller to be guided through a tree structure in order to find the information he is looking for. He goes from one menu to another by simple choice. The IVR broadcasts repetitive messages, more specifically practical information, without human[7] intervention . The IVR information terminals are sometimes used in the event of "overflow of calls", ie when the reception service provided by the staff is saturated[8].

- **The standard telephone**

The switchboard is responsible for welcoming the caller and offers him, through choices in voice menus, access to his correspondent. Choices are made by voice frequency on the caller's telephone keypad or sometimes by voice recognition. The switchboard makes it possible to transfer calls, guide them to the various departments and correspondents, filter them if necessary. The IVR must then forward the call, and for this it must control the PABX (Private Automatic Branch eXchange) which in turn can conduct calls without manual interposition based entirely on the number dialed, it is a very powerful and very flexible in terms of routing mechanism.

- **Computer Telephony Coupling**

The concept of CTI is the meeting and integration of the worlds of telephony and computing on a common platform[9]. The CTI makes it possible to query a database, to store information entered by the caller, but also to send him information specific to him. To achieve this, the database must be integrated into the SVI management software.

Indeed, computer telephony integration (CTI) notably allows the use of databases which contain the traces of calls (number, duration, quality, agent, etc.) and facilitate a statistical approach to help in decision-making. The functions of the CTI make it possible to load the requester's file on the business application when presenting the call to the operator. Thanks to the CTI, voice messages are computer files that can be placed in scenarios where the interlocutors navigate[10].

- ++

The most basic function for a voice server is the information terminal.

However, the SVI is able to play other roles apart from the three standard functionalities which have just been explained in the previous section. Among these many roles we have voicemail, unified messaging and fax-on-demand service which are the best known.

a. Operation of an IVR

An interactive system of the interactive voice server type is most often based on five modules: speech recognition, comprehension, the dialogue controller, speech generation and synthesis.

The first module of the system, speech recognition aims to transform the acoustic signal produced by the user into an equivalent textual chain.

The objective of comprehension is to identify the intentions of the user and to identify the speech acts used. It is a question of "giving meaning to the recognized sentence or sentences".

A veritable conductor of the system, the dialogue controller is responsible for interacting with the information system, managing the interaction (choice of the response to be provided) and the history of the dialogue which may possibly be used by the comprehension or recognition module. Its role can extend from contextual interpretation to all or part of the generation. Two phases can be distinguished for the generation. The first is deep generation. It corresponds to the question "what to say?" », that is to say, it aims to determine the semantic content of the response to the user. The second phase, called surface generation, allows you to choose the words to express the response to the user. It corresponds to the question "how to say it?" ". The last component of the system, voice synthesis reproduces orally the message generated by the processing chain.

b. Architecture of an IVR

[6] VILLANEAU, J., RIDOUX, O., ANTOINE, J.-Y. (2004), "Compréhension de l'Oral Spontané : Présentation et Evaluation des Bases Formelles de LOGUS", *Revue d'Intelligence Artificielle (RSTI-RIA'04)*, Volume 18, N°5-6, pp. 709-742

[7] SENEFF, S. (1992), "TINA: a Natural Language System for Spoken Language Applications", *Computational Linguistics*, Volume 18, N°1, pp. 61-86.

[8] G, PUJOLLE, *Réseaux et Télécoms avec Exercices Corrigés*, Paris, 3e Edition Eyrolles, 2008, p. 289.

[9] Grossglauser, M., et D. Tse, 2002. Mobility increases the capacity of ad-hoc wireless networks. Dans les actes de IEEE/ACM Transactions on Networking, volume 10, pages 477-486.

[10] G, PUJOLLE, Les *Réseaux*, Paris, Edition Eyrolles, 2008, p. 953-970.

- Hardware architecture: A voice server is above all a computer. It can have the configuration of a PC.
- Software architecture: The IVR is equipped with software tools giving access to data or processing, capable of communicating, according to a given formalism, heterogeneous systems or applications carried by different systems. Each media card manufacturer provides with its electronic module a so-called "driver" software interface necessary to ensure the correct hardware and software operation of the latter. In fact the driver is a set of software allowing to manage and integrate the different peripherals (voice cards). It drives data destined for a peripheral or communication port for transfer.

d. Clients-IVRS interaction

IVRs have now become a strategic tool for customer-company relations.

Typically, for an automated switchboard, a welcome announcement welcomes the customer then offers him a choice of actions: putting him in contact with a switchboard operator, directing him to the distance selling service or the technical service. This customer interaction with IVR is mandatory in order to be able to orient oneself in the series of voice menus offered by the voice platform. This interaction is performed in 2 different ways: Using the DTMF keys on the phone[11] and using voice recognition.

We opt for the first form of interaction.

e. Choice of platform

There are several platforms on the market and in open source that allow us to implement an interactive voice server. The most common are: Websphere, Bayonne and Asterisk. Each of these three has its pros and cons. And in this study, our choice goes to Asterisk. We would not have enough time to detail all these platforms...

Indeed the development of voice servers in an Open Source environment gives them greater scope in the sense that with Open Source you can access the code, which means that voice applications can quickly and easily be improved.

Moreover, Open Source products have always given full satisfaction to developers by their simplicity and to customers by their efficiency. Asterisk also includes within it the various elements mentioned above, necessary for the design and implementation.

So with Asterisk, you don't need any additional resources.

An additional reason that guided our choice is that Asterisk offers advanced configuration and opening options, including an internal database and a communication system with external applications: AGI.

4. REALIZATION OF AN IVR FOR THE PUBLICATION OF THE RESULTS.

The realization of an interactive vocal server requires a certain number of tools.

In this part we will define the role of all the equipment that enters into the implementation of a voice server.

The telephone operator

The telephone operator occupies an important place in the realization of a voice server. It assigns a certain number of telephone lines through which clients can simultaneously access the voice server. However, the telephone operator can be bypassed for test sessions at the voice server level. In this case we use the simulator.

The mic

The MIC is a device that transmits an analog signal in digital form.

The MIC as its name suggests, Pulse Coded Modulation, uses a modulation technique to transform an analog signal into digital. When calling a voice server number, the telephone operator transmits the analog signals to the MIC which will take care of transmitting them in the form of digital signals to the voice card connected to it.

Some voice cards integrate the functions of the MIC, which bypasses the use of the MIC. In this case, the analog signals coming from the operator phone are directly received and converted into digital signals by the voice card. Pulse-coded modulation is a method of transmitting speech by sampling the signal (at a rate twice its maximum frequency) and digital coding: the amplitude of each sample is expressed by a number, which is represented by a train of binary pulses.

The simulator

The PSTN line simulator is a device that simulates an incoming call and can therefore play the same role as the internal telephone operator. Telephone extensions are directly connected to the ports of the simulator which is connected to the voice card.

The voice card

The voice card is an interface card that manages the connection between the voice input and the computer.

It comes in two types. There are digital voice cards and analog voice cards.

a. Solution implementation

Several installations and configurations are required before the development of the application.

Here is an exhaustive list of all the installations to be done:

[11] Un **code DTMF** (dual-tone multi-frequency) est une combinaison de fréquences utilisée pour la téléphonie fixe classique (sauf voix sur IP). Ces codes sont émis lors de la pression sur une touche du clavier téléphonique, et sont utilisés pour la composition des numéros de téléphones (en opposition aux anciens téléphones dits *à impulsions*, utilisant un cadran) ainsi que pour la communication avec les serveurs vocaux interactifs.

- **Installation of Linux (fedora 3)**[12]

Indeed Asterisk only works under linux, and we have opted for the distribution of fedora 3. We will perform a minimal installation, in this case the kernel, so as not to overload the server with packages that we will not need later. Thus all the packages necessary for the proper functioning of Asterisk will be installed manually.

- **Installation of Asterisk**

Asterisk is an open source and proprietary (released under GPL and proprietary license) private telephone exchange (PABX) for GNU/Linux systems. It allows, among other things, voice mail, queues, call agents, music on hold and call alerts, call distribution. It is also possible to add the use of conferences through the installation of additional modules and the recompilation of the binaries.
Asterisk implements the H.320, H.323 and SIP protocols, as well as a specific protocol called IAX (Inter-Asterisk eXchange). Asterisk can also play the role of registrar and gateway with public networks (RTC, GSM, etc.). Asterisk is extensible by scripts or modules in Perl, C, Python, PHP, and Ruby.
Installed version: Asterisk 1.2.9

- **Installation of Mysql**

The MySQL database server offers good performance. Indeed The versatility of the platforms is one of the strong points of MySQL, which functions on all the variations of Linux, UNIX or Windows[13]. And, of course, its open source nature allows full customization for users who want to add database server-specific functionality.
Installed version: mysql-5.0.27.tar.gz

- **Installing the Perl module**

P.E.R.L. stands for Practical Extraction and Report Language, which could be translated as "practical extraction and editing language".
Installed version: perl-5.8.5-24.tar.gz

- **Installation of the Asterisk module: AGI**

An AGI (Asterisk Gateway Interface) is a two-way communication channel between Asterisk and an external program, regardless of the language used for the latter.
The exchanges are carried out in text mode, certain parameters are passed when initializing the script from the numbering plan started with the AGI () command. Thus, from the application, you can give orders to Asterisk at the decision-making level, for example HANGUP to hang up or SET CALLERID to change the representation of the caller ID.
Agi can be developed in any language: Perl, Php, Pascal, etc. For the development of the application we will use Perl as language and Mysql as database management system.
One of the great interests of PERL is that it allows us to use the functions offered by AGIs such as: Answer (), hangup (), etc. And these functions are directly accessible thanks to this module.
Installed version: asterisk-perl-0.01.tar.tar

- **Installing the Perl:DBI module**

In our application we will need to connect to a Mysql database, under Perl we will need to install a module for the connection which will be a kind of driver.
Installed version: DBI-1.37.tar.gz
The application thus developed is in the form of a script. This script must be executable and placed in the /var/lib/asterisk/agi-bin/ directory.
To be accessible via a telephone, the voice application must be associated with a number. And must be upgraded to the extensions.conf file.
For example: exten =>4010, 1, Agi (server.agi)
Here the application is named server.agi, and corresponds to the number 4010.

b. Application tree

In an IVR, an application is made up of elementary components linked together. All the components making up the application of an IVR can be represented by a tree structure. The voice tree represents the different stages in the operation of the application.
This tree structure is made up of menus, themselves made up of sub-menus: from menu to menu, the caller's request becomes more and more precise. Choices are made using the telephone keypad (DTMF keys) or using voice recognition.
Generally the tree structure is produced graphically using a PC and software specific to the manufacturer of the IVR, then it is implemented in the memory of the vocal server via a serial link, in the case of a stand-alone IVR. With the tree structure generator you can quickly create the application, visualize it on screen and test it immediately, and also modify it if necessary.
This generator makes it possible to develop a decisional voice tree structure to route the call or correctly inform the caller.
The effectiveness of the voice server will depend on the rigor with which the tree structure of all dialogues with the voice server is designed. This tree structure will be in graphical form, ie an organization chart detailing all the possible routes for all incoming calls.
Not having this tree structure generator, we will do it manually here depending on whether we have circumscribed the problem.

---

[12] J-P, ARMSPACH, P-C, FREDERIQUE O, WAERZEGGERS, *LINUX Initiation et utilisation, Paris 2e édition Dunod, p. 99.*

[13] J-M DEFRANCE, *PHP/MySQL avec Dreamweaver 8*, Paris, Eyrolles, 2008, p. 104.

International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 11 Issue 04 April 2023, Page no. – 3371-3379

Index Copernicus ICV: 57.55, Impact Factor: 7.362

DOI: 10.47191/ijmcr/v11i4.10

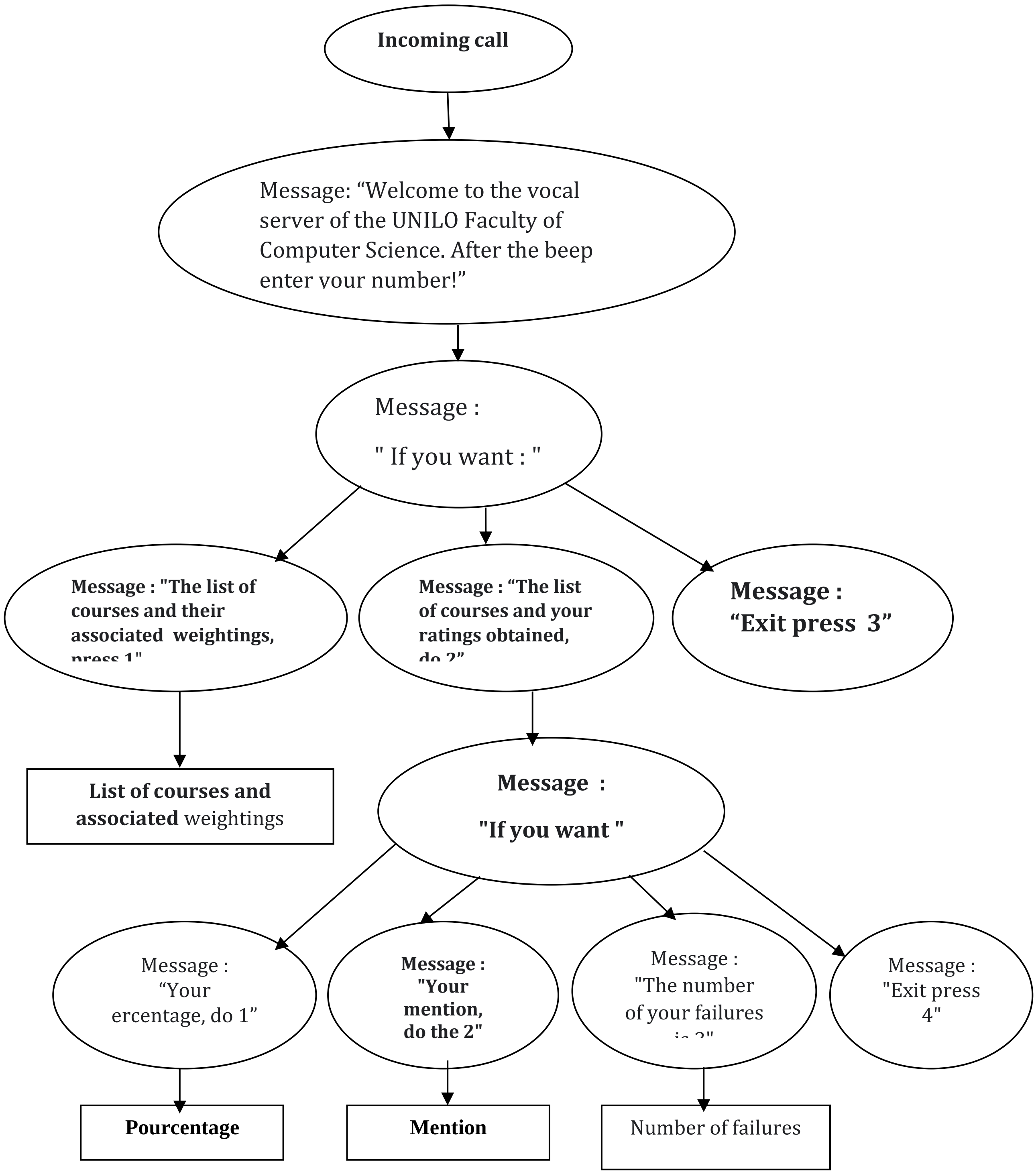


Figure 1: Application Tree

CONCLUSION

At the end of our study with the mission of:

- Study the different solutions allowing the implementation of interactive voice servers ;
- Make a choice of solution and implement the voice application for the publication of the results of deliberations.

We can come to the following conclusions :

Voice servers are appearing everywhere, conquering both the professional world and the general public, interfering in our entertainment, in training, at work, in assistance, etc.…

However, they present some significant limitations at present. Indeed it is very difficult to decide the quantity (or quality) of information that users need.

Too little information can create ambiguity, misunderstandings and the feeling of a rather artificial approach. Conversely, too much detail can lead to confusion and the user can then waste time trying to correct their mistakes. Therefore, a good voice application represents a balanced compromise that avoids too much or too little information.

As perspectives we can say that:

Many improvements should be considered, particularly in terms of automatic speech processing technologies, such as voice synthesis or recognition, used in interactive voice servers. Indeed, the latter must be the subject of an in-depth study, especially if they must be in national languages in order to allow access to information to the rural world in accordance with the general objectives of the project. Regarding the application we want to design, we will have to think about implementing the solution not at the software level (application) but at the system level, i.e. within Asterisk. In short, we can say that voice servers have a bright future in the sense that they offer solutions accessible to a greater number of users but require a greater investment to be able to derive maximum benefit from them.

REFERENCES

1. B. ORIOLA, N. VIGOUROUX, G. PERENNOU, P. TRUILLET *‘’Une interface multimodale de consultation de documents électroniques pour déficients visuels’’*, *Forum des Recherches en Informatique* – Ecole Polytechnique, 02-03 Juin 1993, pp. 90-91.
2. D. KAMBEYANDA, S. CRONK, L. SINGERA, *Potential Problems associated with use of speech recognition products*, RESNA'96 Proceedings, pp. 119-22.
3. G, PUJOLLE, Les *Réseaux*, Paris, Edition Eyrolles, 2008.
4. G, PUJOLLE, *Réseaux et Télécoms avec Exercices Corrigés*, Paris, 3e Edition Eyrolles, 2008.
5. J. BERNSTEIN, *Applications of speech recognition technology in rehabilitation. In J. E. Harkins and B. M. Virvan, editors, Speech to text: Today and tomorrow*, Washington D.C., Gallaudet University Research Institute. Monograph Series B., n°2. 1988.
6. J-M DEFRANCE, *PHP/MySQL avec Dreamweaver 8*, Paris, Eyrolles, 2008.
7. J-P, ARMSPACH, P-C, FREDERIQUE O, WAERZEGGERS, *LINUX Initiation et utilisation*, Paris, 2e édition Dunod, *2005.*
8. M. GROSSGLAUSER, et D. Tse, Mobility increases the capacity of ad-hoc wireless networks. Dans les actes de IEEE/ACM Transactions on Networking, volume 10, 2002, pages 477-486.
9. R. PAWLAK, J.-P. RETAILLE, L. SEINTURIER. *Programmation orientée aspect pour java/J2EE*, Eyrolles, Paris, juin 2004, ISBN 978-2-212-11408-9.
10. S. OVIATT, *Taming recognition errors with a multimodal interface*, Communications of the ACM, Septembre 2000, 43 (9), 45-51.
11. S. SENEFF, *A Natural Language System for Spoken Language Applications*, *Computational Linguistics*, Volume 18, N°1, 1992, pp. 61-86.
12. VILLANEAU, RIDOUX, O. ANTOINE, *“Compréhension de l'Oral Spontané : Présentation et Evaluation des Bases Formelles de LOGUS”*, *Revue d'Intelligence Artificielle (RSTI-RIA’04)*, Volume 18, N°5-6, 2004 pp. 709-742.