2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

IVOrpheus

A proposal for interaction by voice commands in three-dimensional environments of information visualization

Lennon Furtado, Brunelli Miranda, Nelson Neto, Bianch Meiguins
Institute of Exact and Natural Sciences
Federal University of Pará, UFPA
Belém, Brazil
{lennonsfurtado, brunelli.miranda, dnelsonneto, bianchism}@gmail.com

***Abstract*— IVOrpheus is an information visualization tool for three-dimensional data that allows user's interaction by voice commands, mouse and keyboard input. The visualization technique used was the scatterplot 3D, which was implemented using Jmathplot API, and the speech recognition in Brazilian Portuguese was performed by Coruja software. IVOrpheus was developed in Java following the architectural pattern MVC, design patterns and open technologies. In voice interaction, some usability guidelines have been set in interface building process, making it more intuitive and contributing to lower the user cognitive effort. In addition, initial usability tests with users were performed to evaluate the application interface with and without interaction by voice. The tasks with and without voice interaction have shown similar results of time and completeness. The speech recognizer achieved a word error rate of approximately 17%.**

Keywords—IVOrpheus; Voice Recognition; Brazilian Portuguese; Information Visualization; 3D Scatterplot.

I. Introduction

According [1], Information Visualization (InfoVis) is the use of an interactive computing environment that allows visual representation of abstract data to amplify user's cognition on a set of data and their relationships. The large amount of electronic data that is stored daily by various computer systems has shown the InfoVis as an area that can help this troubled analysis.

The large amount and the dimensionality of data sets have presented challenges such as the need for more visual space in devices, data complexity, new InfoVis techniques, advanced forms of interaction, among others. The 3D environments are presented as an alternative to 2D environments by presenting the depth dimension. Thus, making available a large space for visual representation of data.

However, there is one question constantly linked to 3D environments: why does the user have difficulty when interacting in 3D environments? A first hypothesis to understand this difficulty presented by the user is the use of traditional ways of interaction - such as a mouse and keyboard - to interact with 3D environment. Another one is the quality of the interaction interface that does not guide the user properly in carrying out their tasks [2].

Thus, this work aims to present aspects of design, development and evaluation of an interface for voice commands to interact with InfoVis 3D applications. As usage scenario, we used the 3D scatterplot InfoVis technique. Besides, the Coruja API was used to perform the speech recognition task, which uses the speech recognition engine Julius to support Brazilian Portuguese.

Basic guidelines were adopted for the application design, as the interface automatic adaptability to a database and the dynamic generation of the grammar used in the speech recognition system. Finally, brief usability tests will be presented for a first evaluation of the interface and interaction proposal.

II. Application

IVOrpheus is an InfoVis tool that uses the 3D scatterplot technique, making use of three spatial dimensions (x, y, and z axes) and three visual dimensions (color, shape, and size) to represent the data. It is possible to interact by mouse and keyboard or voice commands in Brazilian Portuguese.

This section presents the used tools, aspects of interface, features, MVC (Model-View-Controller) applied in architecture design, screen flow demonstrating the application and grammars management.

A. Used tools

1) Coruja

The proposal presented in this paper uses the functionality of interaction via automatic speech recognition (ASR) in Brazilian Portuguese. For this, the application makes use of the free ASR software Coruja [3], which provides acoustic and language models, as well as a programming interface (API). Coruja was built to facilitate the task of controlling the Julius engine.

Aiming for flexibility in the platform, the latest version of Coruja supports the Java Speech API specification (JSAPI), working in both controlled grammar (command-and-control) and text-free (dictation) applications.

2) JMathPlot

JMathplot is a multi-platform API developed in Java, and it was used in IVOrpheus for the development of InfoVis module. The JMathPlot API is not limited to only 3D scatterplot; other InfoVis tecnhiques are available, such as, histogram, parallel coordinates, line chart, among others.

B. Conceptual Aspects

IVOrpheus follows the basics guidelines of a good InfoVis tool, defined by [4] and commonly referred to as the InfoVis mantra, which are: overview of data - user should have a

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.131

general idea of the data for analysis; semantic zoom - focus on a subset of the data; filters - reduce the analysis data set; and details on demand - present data that are not visually represented (hidden data).

1) *Interface*

The first guideline for construction of IVOrpheus environment is that both interaction by keyboard and mouse and interaction by voice commands will share the same interface. We considered four main points in the development [5]:

- Meaningful communication: user should easily identify the commands available for interaction and their meaning, and get help about on the available commands on the screen.
- Minimal user action: the commands should be simple on each screen. And the input data should be on the screen.
- Consistency and standardization in interaction and screens: forms of interaction and standardization of screens are maintained. For example, the same commands back or cancel on every screen, same commands in different contexts for similar operations
- Speaker-independent speech recognition.

The IVOrpheus interface is divided into three main areas, as shown in Fig. 1: the Options bar (1), the Preview area (2) and the Menu bar (3). In all areas, each option presented on the screen can be performed by voice command, or mouse click.

The following voice commands are available and visually displayed on the Option bar: “load base”, “back”, “legend” and “help”. The Menu bar is enabled only after a database is loaded into the tool, and has the following initial commands: “configure”, “filter” and “interact”. After one of these commands is used, the other sub-menus are enabled. All this commands are in Brazilian Portuguese.

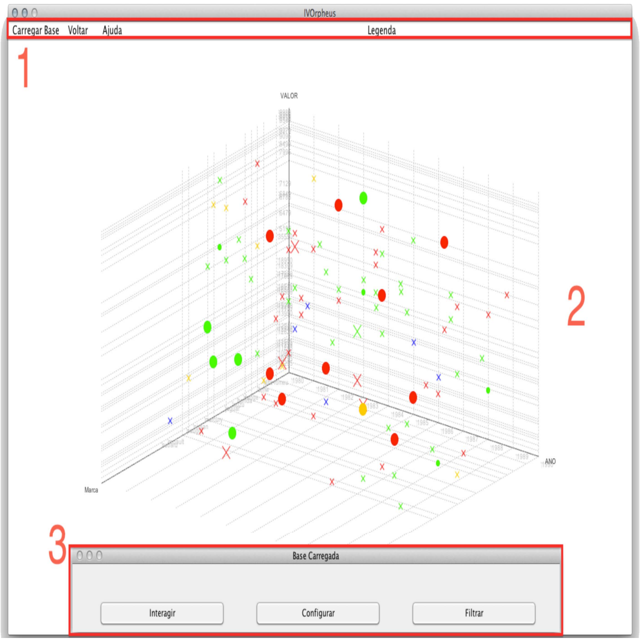


Fig. 1. IVOrpheus Interface.

2) *Architecture Pattern*

In IVOrpheus it was used the MVC (Model View Controller) architectural pattern, where the application was divided in three main components, each one with its own functionality [6]. IVOrpheus class diagram is shown in Fig. 2.

The View is responsible for showing a visual response to user actions, the Controller deals with the communication between the model and the view packages, and Model is the responsible for processing data and data logic access.

In the View package, the Interface class determines how the user interface will be created and what appears in it, having two main functions: to determine which windows should be displayed for each user input event; and displaying the data in a visualization technique in the Preview area, implemented by JMathPlot library, and represented by the diagram subsystem.

In the Controller package, the Recognizer class loads Coruja system and all grammar files. It is also responsible for managing the grammar. The Translator "listens" the user’s voice entries, and with the use of grammars, defines which entries (rules) will be passed in the form of commands to the RecognizerListener. The RecognizerListener class makes the association between a received voice command from the Translator and an abstract button, which is passed to the BtnListener to call the relevant function to the button.

The Model package, manages the database and the attributes in it. It is composed by Attribute, Directory, Grammars classes and Coruja system. The Directory class returns the names of the existing databases in the application root directory. The Attribute class reads the bases and handles the database selected in the tool by assigning a corresponding data type for each attribute, according to the existing values by attribute, thus making possible the manipulation of these values by Grammars methods. The Grammars classes represent all grammars present in IVOrpheus application and its dynamic writing system.

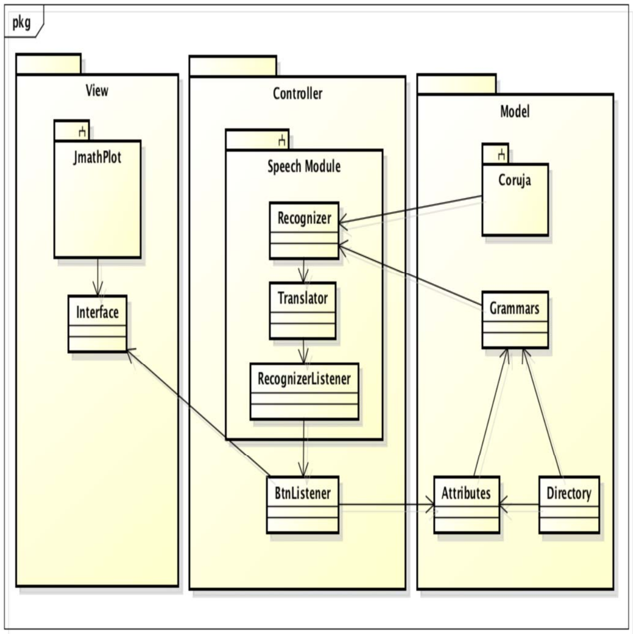


Fig. 2. IVOrpheus class diagram.

3) *Screen Flow*

This section intends to introduce the IVOrpheus application and displays via a screen flow, its features and its grammars.

In the initial screen, the user will encounter the Load Base button. When selected, i.e by mouse click or voice command. A new screen shows the available data set, where the user can select them by name or number of the data set.

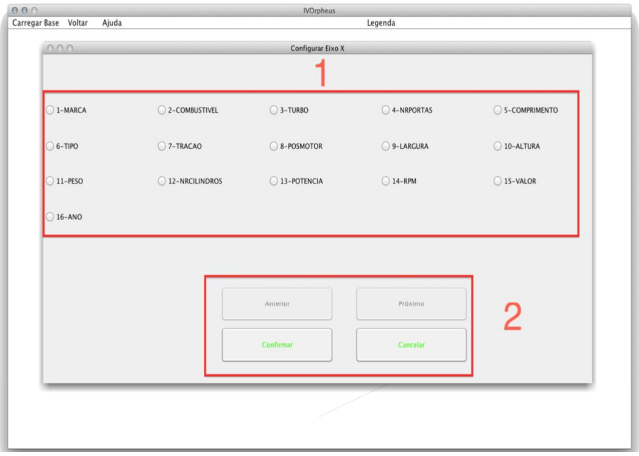
When the user loads a dataset, the grammar IFC (Interact/Filter/Configure.) is initialized allowing the user to configure the spatial dimensions x, y, z or color, shape, and size with the relevant attributes of the base. In the moment where the user set the axes (x, y and z), the dynamic grammar Atributes takes place, loading all attributes of the dataset selected. Fig. 3 shows the configuration of the X-axis, whith the following areas highlighted: 1 - the location of the selected database attributes. 2 - confirm, cancel, Next and Previous buttons. If the database has many attributes, they will be divided into tabs, which can be browsed by the Next and Previous buttons (commands).

After axes configuration, the user can apply filters in dataset, in order to extract the desired information. The IVOrpheus tool has two types of filters: the categorical filter that uses the CategoricalFilter grammar, and the continuous filter that uses the ContinuousFilter grammar. The Categorical Filter Screen is similar to the configure axis screen presented in Fig. 3, but instead of the users selecting the attributes, they select the unique values of the attribute, which depends on the axis to be filtered. For instance, if the X-axis is setted to “Fuel”, the user can select the gasoline and diesel values to be filtered.

Fig. 4 shows the Continous Filter Screen, where the highlighted areas show: 1 - the axis limits, minimum and maximum, and the buttons begin and end, that change the start and the end of the axis; 2 - A user’s guide with the commands allowed in this screen. As commands: “clear”, “delete”, “the numbers from 0 to 9” and “dot”; and 3 – the Confirm and Cancel buttons.

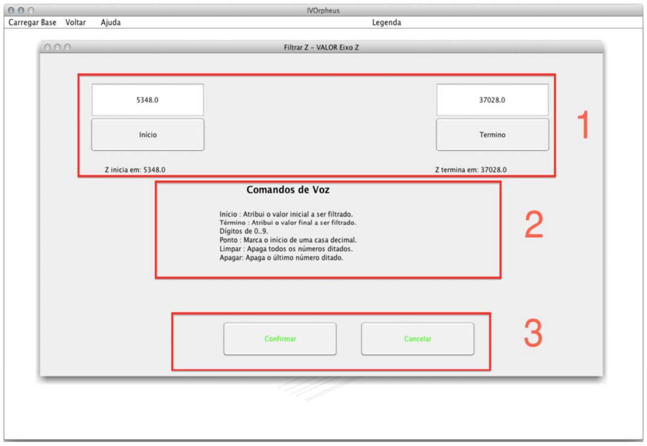
The Fig. 5 helps to understand how to configure visual dimensions such as color, shape and size. To illustrate an example, the three spatial dimensions assume respectively, X-Brand, Y-Price, Z-Year. The three visual dimensions assume the following attributes, Color - Type (Hatch, Sedan, Wagon and convertible cars), Shape - Doors number (2 or 4) and Size – Fuel (Diesel or Gasoline). This setup allows user to easily see some trends, like the hatch cars (Red/Color), with 2 doors (dot/Shape) and with gasoline engine (Small/Size), was one of the most expensive car of that decade (80`s).

With 6 dimensions applied in data representation, the use of a Legend panel is necessary. For better visualization of data by the user, the IVOrpheus software makes use of two legend panels, namely, the color panel that lies west of the screen and the panel shape / size that is on the east side of the screen. As can be seen in Fig. 5, in highlighted area 1- Color panel, which displays colors arranged in viewing and the percentages of each color; 2 – Similarly, the shape / size panel displays their respective percentages; and 3 - shows the button (command) legend that is available in all application screens.



○ 1-MARCA ○ 2-COMBUSTIVEL ○ 3-TURBO ○ 4-NRPORTAS ○ 5-COMPRIMENTO
○ 6-TIPO ○ 7-TRACAO ○ 8-POSMOTOR ○ 9-LARGURA ○ 10-ALTURA
○ 11-PESO ○ 12-NRCILINDROS ○ 13-POTENCIA ○ 14-RPM ○ 15-VALOR
○ 16-ANO

Fig. 3. Setting X axis.



5348.0

37028.0

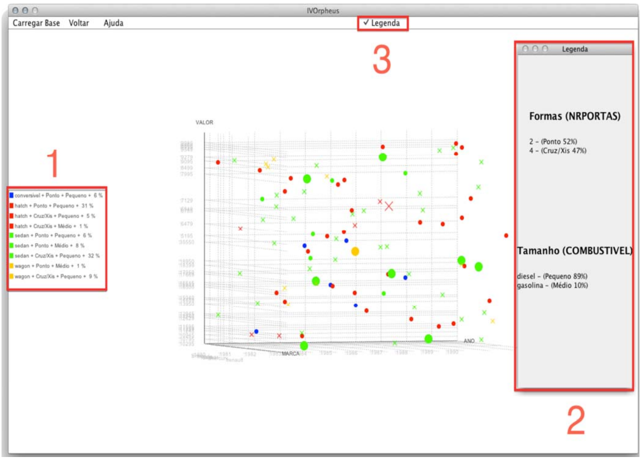
Z inicia em: 5348.0

Z termina em: 37028.0

Comandos de Voz

Início : Atribui o valor inicial a ser filtrado.
Término : Atribui o valor final a ser filtrado.
Dígitos de 0..9.
Ponto : Marca o início de uma casa decimal.
Limpar : Apaga todos os números ditados.
Apagar: Apaga o último número ditado.

Fig. 4. Continuos Filter.



🟦 conversivel + Ponto + Pequeno + 6 %
🟥 hatch + Ponto + Pequeno + 31 %
🟥 hatch + Cruz/Xis + Pequeno + 5 %
🟥 hatch + Cruz/Xis + Médio + 1 %
🟩 sedan + Ponto + Pequeno + 6 %
🟩 sedan + Ponto + Médio + 8 %
🟩 sedan + Cruz/Xis + Pequeno + 32 %
🟨 wagon + Ponto + Médio + 1 %
🟨 wagon + Cruz/Xis + Pequeno + 9 %

Formas (NRPORTAS)

2 - (Ponto 52%)
4 - (Cruz/Xis 47%)

Tamanho (COMBUSTIVEL)

diesel - (Pequeno 89%)
gasolina - (Médio 10%)

Fig. 5. Color/Shape/Size and Legend Painels.

IVOrpheus as well as all information visualization tools must have the basic functionality to meet the definition established by [4]: "Overview first, zoom and filter, then detail on demand." All of these features have been contemplated by IVOrpheus application.

C. Grammars Management

All the words that will be recognized by Coruja API are organized into grammar files. The Grammar files are a set of rules used in speech recognition, where each word is a rule. These grammars were divided into two categories: dynamic and static. The dynamic grammars have the quality to load the rules to be recognized dynamically, while in the other hand, the static grammars have a fixed number of rules from the beginning to end of the application. Both the static and dynamic grammars have global and local rules, and when in global level, these rules are present in all grammars. The global rules are the following: load base, legend, back, and help. The local rules are the specific words of a grammar. Fig. 6 presents an overview of grammar organization in IVOrpheus, including the following grammars:

- Load: this grammar starts together with the application, and is responsible for containing the initial commands. Being also a dynamic grammar, the Directory component returns the names of databases stored in the application root directory, and the Recognizer component write this information on Load grammar. In addition, the Load grammar has the local rules Confirm and Cancel.
- IFC: A static grammar that has the local rules: Interact, Filter and Configure. And as implied, triggers the Interact and FC grammars.
- Interact: has Move, Rotate and Zoom as local rules.
- MR: acronym for Move and Rotate. Such grammar is used to apply translation or rotation in the information visualization technique. It is made by local rules to move and rotate left, right, up, down, front and aback.
- Zoom: grammar that has local rules of zoom in and zoom out. The minimum zoom allows an overview of the visualization, while the maximum zoom allows the user to have more details about a point or set of points.
- FC: acronym for filter and configure. Serves as an intermediary to filter or set an attribute in the view. Has the local rules: color, shape, size, x, y, and z axes.
- Attributes: dynamic grammar which acts similarly to load grammar, generating local rules by taking the attributes of the user-chosen database. Has the local rules confirm and cancel.
- ContinuosFilter : static grammar activated when the attributes to be filtered are float values or integer with more than 20 unique values. With the local rules: digits 0..9, begin, end, delete, dot / comma, confirm and cancel.
- CategoricalFilter: dynamic grammar that has in its local rules unique values of the chosen attribute to be filtered. The filter is activated when the categorical attributes are text or numerical integers with a number of unique values less than 20.

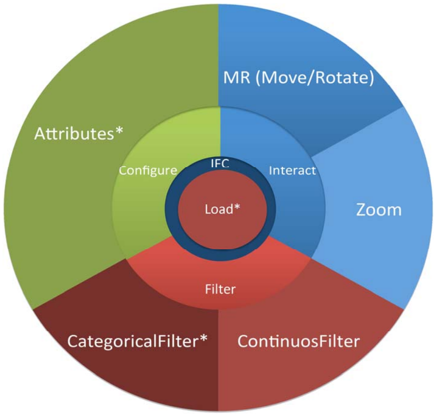


Fig. 6. Grammars Overview (*Dynamic grammars).

III. Testing with users

A. Usuability Test

In this section, we present the test with users, and the obtained results.

1) Volunteers

Eight volunteers participated in the tests, where four users perform the test using the natural user interface, while the other four used the standard interface, ie. Mouse. The volunteers were divided into two groups, the first one formed by four non-student (lay users) and the second group, formed by four students in computing. Two members of each group performed the tests using voice input while the remaining 2 used the input by mouse. The average age of lay users was 32 years and the group of students was 23 years.

2) Proceeding

Participants were introduced to IVOrpheus tool through a five minutes long training video, which presented the interface and functionalities as dataset configuration, filters and how to identify responses by 3D scatterplot. After that, the training was initiated, where it was allowed to consult the researcher for any doubt not answered in the video. During the execution of the tasks, the user time and their accuracy were recorded for later analysis. After tasks completion, the NASA-TLX questionnaire was administered to measure users' workload.

3) Tasks

A sheet with five tasks was delivered for each volunteer. The user had a total time of 30 minutes to complete the test. The database used in the test was about cars of the 80's (artificial base) that contained 789 records and 16 attributes (7 continuous and 9 discrete).

Moreover, to measure user performance quantitative, the tasks were timed. The time spent to complete the tasks by each lay/student user is showed in Fig. 7, in which the Y-axis is represented in minutes. For the tasks used in this test, the first three are low complexity tasks and the other two are medium complexity tasks, as showed below Fig. 7.

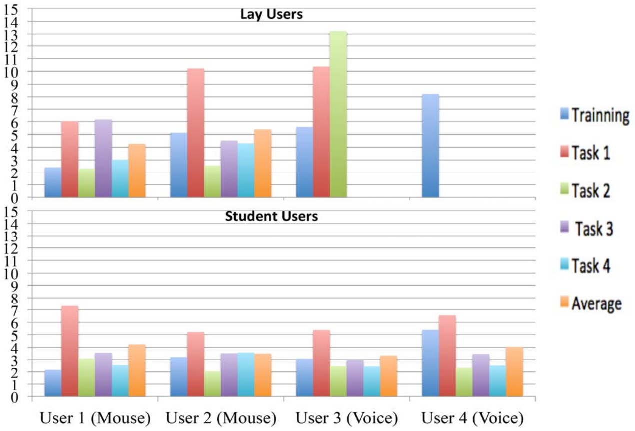


Fig. 7. Graph representing time spent in tasks and training completion by lay and student users.

a) Are there any car type convertible with rear-wheel driven diesel fuel?

b) Most cars with a width between 67 and 70 cm with 6-cylinder are moved by gasoline?

c) Most diesel and turbo powered cars have two doors?

d) What are the car brands with value greater than 30760 manufactured in the years 1983-1985?

e) What are the kind of car's traction with power of 262 RPM and 5000?

4) *Accuracy*

To measure accuracy the tests were recorded, thus allowing the researcher to observe the amount of unrecognized words by Coruja API and the amount of words spoken during the execution of the application. In Fig. 8, we depict the percentage of hits and misses of all voice users as a bar graph, with the first two bars representing the lay users and the other two representing the student users.

The amount of hits and misses was related to two factors. The first is the rate with the voice commands were inserted in the application. A voice command is recognized between 1 to 2 seconds after spoken, consecutively, if the users do not respect this condition and insert another command within this period, the system will interpret the new command, generally producing a wrong hypothesis. The second factor that influenced the outcome of the accuracy was short words with few phonemes, due to Coruja API issues.

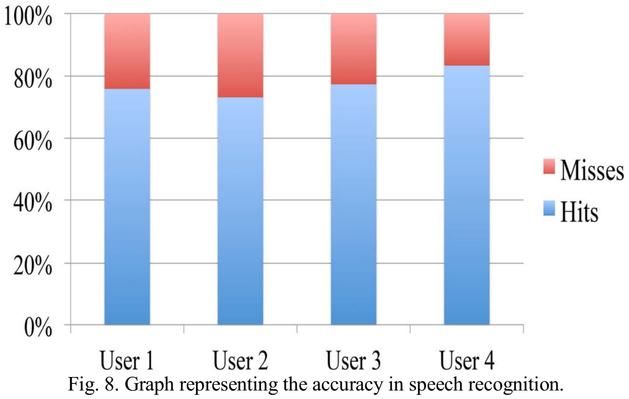


Fig. 8. Graph representing the accuracy in speech recognition.

5) *Nasa-TLX*

The user's workload was evaluated with the NASA Task Load-Index [7], which is used to identify the overall workload in the different tasks and the main sources of workload. The NASA TLX estimates six subscales (physical demands, mental, time, effort, frustration and performance). Since the lower value, smaller is the workload represented and vice versa, and in performance scale, in which lower the number, better is the performance.

B. *Analysis of results*

During user tests, it was possible to observe and monitor trends among different groups. Among these trends, we observed that the accuracy level directly affected in users frustration level and consequently, their performance. An example of this can be seen in table I below, where the lay user 4, that is the same user 2 in Fig. 8, had the lowest accuracy found in the tests and this generated the highest frustration index of the entire test, affecting his performance as shown in the results of NASA-TLX.

Meanwhile, the student user 4 in table II, which is the same user 4 in Fig. 8, had the better results in using voice as input with the highest accuracy of 83.4% hits. This resulted in better performance and lower effort, frustration, and TLX-Score. Furthermore, as seen in tests, the accuracy of ASR directly affects the user workload.

When comparing users who obtained the best average time, student 2 (Mouse) and student 3 (Voice) in Fig. 7, it can be noticed that even student 3 (Voice) had the lowest average time recorded in the tests, by comparing the dimensions of effort, frustration, performance and TLX-Score (table II). All these scales print a greater workload in voice user. This implies that even the voice having a better time, there was no decline in cognitive load in reference to the mouse.

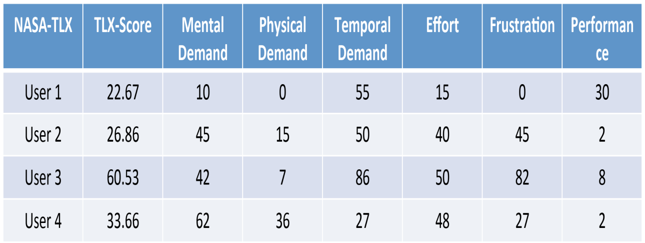
Finally, analyzing the TLX-Score average (table III), we notice that the voice had cognitive load 50% higher than the mouse. Therefore, we concluded that the voice, as a means of mimicking the mouse, implies in a greater cognitive load.

However, it is important to note that the lowest average time in performing the tasks, namely the third user (student), had an average of 3 minutes and 34 seconds using voice input, while the best average time using the mouse as input, the second student, had an average of 3 minutes and 49 seconds. So, even with the voice input causing a greater level of workload, it is shown as effective as a mouse input.

TABLE I. NASA-TLX NON-SUTDENT (LAY) USERS RESULTS.

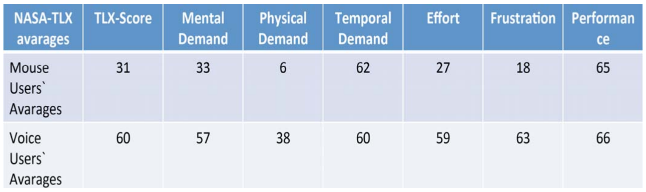
NASA-TLX	TLX-Score	Mental Demand	Physical Demand	Temporal Demand	Effort	Frustration	Performance
User 1	35.73	42	7	50	41	22	24
User 2	38.83	36	2	93	12	8	82
User 3	53.16	43	62	53	50	61	50
User 4	75.83	84	49	74	89	84	75

TABLE II. NASA-TLX STUDENT USERS RESULTS.



NASA-TLX	TLX-Score	Mental Demand	Physical Demand	Temporal Demand	Effort	Frustration	Performance
User 1	22.67	10	0	55	15	0	30
User 2	26.86	45	15	50	40	45	2
User 3	60.53	42	7	86	50	82	8
User 4	33.66	62	36	27	48	27	2

TABLE III. NASA-TLX AVARAGES.



NASA-TLX avarages	TLX-Score	Mental Demand	Physical Demand	Temporal Demand	Effort	Frustration	Performance
Mouse Users` Avarages	31	33	6	62	27	18	65
Voice Users` Avarages	60	57	38	60	59	63	66

IV. FINAL CONSIDERATIONS

In this work, an InfoVis tool for three-dimensional environment was developed, using the scatterplot visualization technique with a voice input-based interface. In the tool's development were applied guidelines from speech recognition systems, information visualization systems and HCI general area, as well as good programming practices in the Java language. Initial usability tests were performed with two different user profiles, in order to determine the tool's efficiency.

During the implementation process, the following guidelines were generated. Sort of the voice commands in global or local command tool, take into account possible speech recognition process restrictions, like foreign words and some phonemes, as well as choose natural language commands.

Based on the results of usability test, it was observed that even computing student users had a level of stress and high cognitive effort for the completion of tasks using speech. This is related to the speech recognition process, that occasionally generated incorrect hypothesis due to ambient noise or the user speak pace, leading the authors to analyze and answer a question, "Which approach in the use of voice in information visualization technique for dispersion of points in three dimensions has greater efficiency in reducing the cognitive load of the user?".

As can be seen, the approach of using voice with the aim of mimic a mouse has lower efficiency and higher cognitive effort compared to the use of a standard interaction interface (mouse). Nevertheless, even the voice input causing a greater level of workload, it is shown as effective as a mouse input. Moreover, the voice input has a chance of meeting a wider range of users, since users with motor problems, or without sensibility on the hands, can make use of IVOrpheus system to interact with a database. Thus, reaching the objective presented by Alan Kay [8], which is, the more "friendly" is man/machine interaction a wider range of people will be reached.

These observations guide the future of IVOrpheus application, to meet a proposed interaction for more efficient voice that requires less cognitive effort from the user. The next version of IVOrpheus system will address the use of voice interaction through dictation (Dialogue), using the principle set forth in the works [9], [10] and [11], leaving the interaction interface invisible to the user, where it should not stick to command interface, such as buttons and panels. Just enter a question and the system will show a visualization response. Furthermore, as future work, there will be certain improvements and additions in the current tool features and extended user tests of IVOrpheus system, with comparisons to current user tests.

REFERENCES

[1] S. Card, J. Mackinlay, and B. Shneiderman, “Readings in Information Visualization - Using Vision to Think,” Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999.

[2] J. Jankowski, M. Hachet, “A survey of interaction techniques for interactive 3D environments,” In: Eurographics 2013 state of the art reports, pp. 65–93, 2013.

[3] P. Silva, P. Batista, N. Neto, and A. Klautau, “An open-source speech recognizer for Brazilian Portuguese with a windows programming interface,” The International Conference on Computational Processing of Portuguese (PROPOR), 2010.

[4] B. Shineiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in Visual Languages, 1996. Proceedings., IEEE Symposium on , vol., no., pp. 336-343, 3-6 Sep 1996.

[5] K. Lee, R. Grice, "The Design and Development of User Interfaces for Voice Application in Mobile Devices," in International Professional Communication Conference, 2006 IEEE , vol., no., pp. 308-320, 23-25 Oct. 2006.

[6] F. Buschman, “Pattern-Oriented Software Architecture Volume 1: A System of Patterns,” Willey, pp. 125, 1996.

[7] S. Hart, L. Staveland, “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research,” In P. A. Hancock and N. Meshkati (Eds.) Human Mental Workload. Amsterdam: North Holland Press, 1988.

[8] A. Kay, “A personal computer for children of all ages,” In Proceedings of the ACM Annual Conference— Volume 1, 1972.

[9] R. Sharma, M. Yeasin, N. Krahnstoever, I. Rauschert, G. Cai, I. Brewer, A. Maceachren, K. Sengupta, “Speech-Gesture Driven Multimodal Interfaces for Crisis Management,” Proc. IEEE, Vol. 91, No. 9, pp. 1327-1354, Sep. 2003.

[10] Y. Sun, J. Leigh, A. Johnson, S. Lee, "Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations," Proceedings of the 10th international conference on Smart graphics, EUA, pp. 184-195, 2010.

[11] K. Cox, R. Grinter, S. Hibino, L. Jagadeesan, D. Mantilla, “A Multi-Modal Natural Language Interface to an Information Visualization Environment,” J. of Speech Technology 4, pp. 297–314 , 2001.