UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

HUMAN-ROBOT INTERACTION (HRI) WITH ROBOT PEPPER

Authors:
Hernan Indíbil de la Cruz Calvo.
Alejandro Zornoza Martínez.

April, 2018

Índice de contenido

1. INTRODUCTION

Actually with the evolution of technology and advances in recognition and speech algorithms, the capacibility of the robots increment so fast, that it is the reason why a lot of companies are working on robot development, robots will provide humans a lot of tools that they could make their life more easier.

With the advance accomplished in the HRI(human robot interaction), we can provide to the robots habilities that help them to interact with humans and exchange information between them.

The HRI(human robot interaction) is a multidisciplinary field with a lot of contributions from artificial intelligences, mecanics advances, robotics design, natural language understanding and social science.

These set of science are working all time from improve the capabilities of the robots, making them strongers with better motors, making them more intelligent, making them more social and finally improving their natural language understanding and better in people recognition.

We have learnt during practices how the robot acts depending diferent conditions and enviroments and now we are able to try to work with one of the most advanced robot in HRI that it is called Pepper.

Pepper is a human-shaped robot, that it was design for being a genuine day-to-day companion, which it has the hability to perceive emotions and answer depending the type of them.

The body of Pepper it is preparied for communicate with the people with the most natural and intuitive body language, giving it a friendly actitude and voice.

With the memory capability the user can personalise the robot downloading the software applications and adapts himself to user tastes and habits. We have to remember that his habilities are provided by researchers groups from around the world and the joing of all of them permits it design.

In this project we are going to try to development our personal behaviours for Pepper and learning who a robot with these caracteristics acts, especially in face recognition, speak dialogs, hear words and move around following special orders defined by us.

2. OBJECTIVES

3. PRELIMINARIES

3.1. CHOREOGRAPHE

We have to provide to the reader the enough knowledge about the development environment called choregraphe. We use it in this project to perform the human-robot interaction with Robot Pepper. One of the benefits of using this enviroment it is that the user do not need a lot of knowledge about programming in code. The reason is simple, it provides a toolbox diagram enviroment where the user can move and connect boxes that are defined to simulate the robot behavior.

We will provide to the reader a tiny guide of using this enviroment and how we use to design the robot behaviors.

First of all, we have to show the first interface page of choregraphe, but we have to remember first that the choregraphe installation package include other importants tools that we have to tell to the reader.

The installation package includes:

- Choregraphe 2.5.5 development tool: That it is the enviroment for design. We will explain more about it in other sections after that.
- NAO Documentation 2.5.5 : All relevant information about robots, enviroment, how to install, and uses are included there. It is importante documentation that the user can use to perform his hability while he is working with the robots, it includes documentation about Pepper, Nao and Romeo.
- Memory Backup 2.5.5: It provides the posibility of safe the robot state when we are working on it, it can save previusly installations and we can recover them if the user think it it necesary. This tool have two principal uses, the first one, save data installed on the robot and the second one restore data from the user computer to robot memory.
- Monitor 2.5.5: It provides the tools for monitoring and take information about memory and cameras data, we did not use it for this project.

Well, to start with we are going to explain all the information about the Choregraphe 2.5.5 development tool, and we will teach all the steps that we follow for make the project tasks.

First of all, we open the Choregraphe 2.5.5 development tool and we can see this interface:

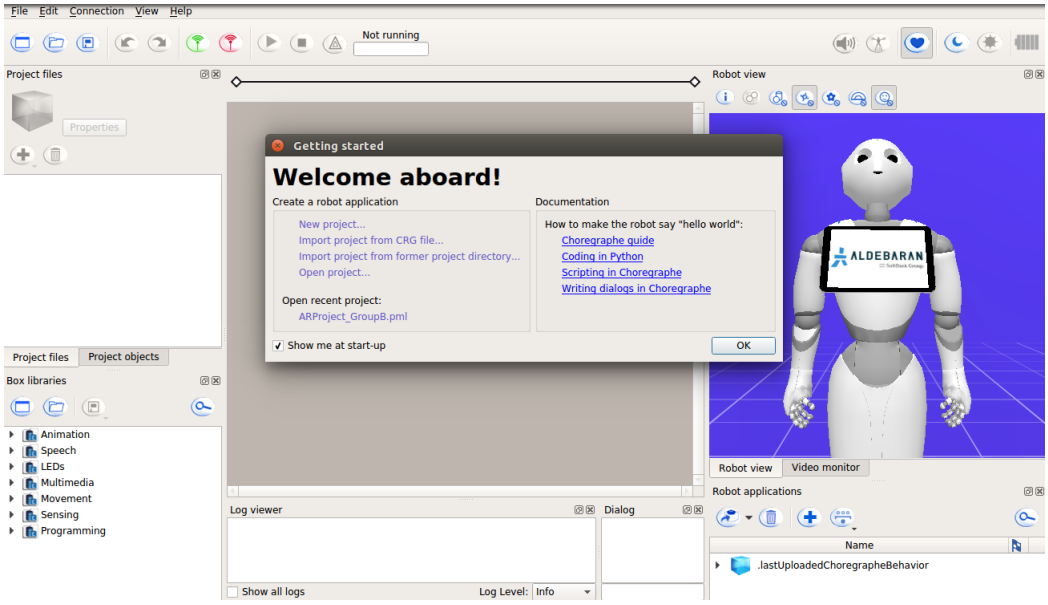

Figure 1: Image of the Choregraphe main menu interface.

At the Getting started window we can see diferent options:

- Create a robot application:
  - New project.
  - Import project from CRG file.
  - Import project from former project directory.
  - Open project.
- Documentation:
  - Choregraphe guide.
  - Coding in Python.
  - Scripting in Choregraphe.
  - Writing dialogs in Choregraphe.

In addition the interface contains a top bar, one left bar and one right bar.

The top bar contains the options for:

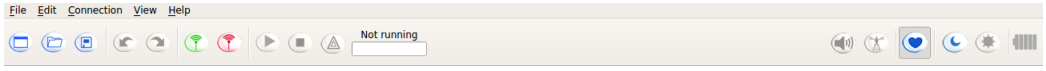

Figure 2: Image of the top bar.

- The top part contains:
  - File options.
  - Edit options.
  - Connection options.

- View options.
- Help options.

The bottom part of this bar provides:

- The first rectangle contains: Create project option, open recent project option, save project.

Figure 3: Image of the top bar first rectangle.

- The second rectangle provides the option of backward or forward in the steps that we do.

Figure 4: Image of the top bar second rectangle.

- The third rectangle provides the posibility of connect and disconnect to a robot or a simulate robot.

Figure 5: Image of the top bar third rectangle.

- The fourth rectangle provides options for upload to the robot and play, stop, debugs and errors output, the state running, not running, volume options and animation mode.

Figure 6: Image of the top bar fourth rectangle.

- The last rectangles provides options for turn autonomous life on/off, rest the robot, wake up the robot and battery information about the robot.

Figure 7: Image of the top bar fiveth rectangle.

Also, we are going to continue giving information about other parts of the main menu interface, at the left side we can see this menu blocks.

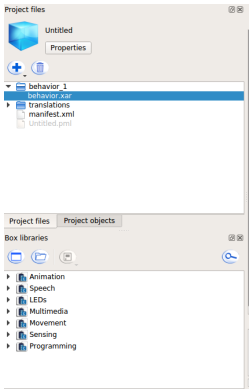

Figure 8: Image of the left bar.

In this part the reader can see two diferences zones, the first one, at the top it provides information about project files, in the plus mark, you can create new behaviors, dialogs, planar moves, directories, or import files from other selected directories.

The last one provides the user box libraries which will help the user to program the robot behaviors. It have 7 categories called:

- Animation.
- Speech.
- LEDs.
- Multimedia.
- Movement.
- Sensing.
- Programming.

We will explain better in future sections giving relevant information about the utility of these categories and the different types of boxes that they provide.

On the other side, the right bar, it provides a video image of the robot where the user can see what is doing the robot doing, objects that are in front of him, and the currently distance of them. It is a place where you can test the robot behavior and the user can validate and test the designed behaviors.

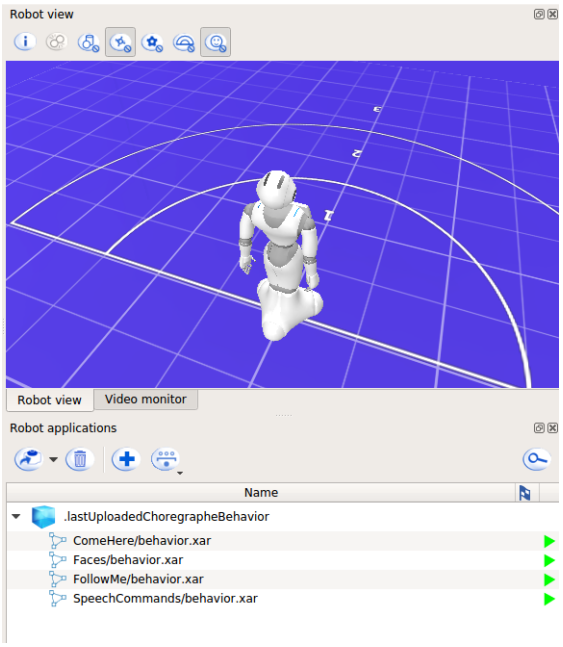

Figure 9: Image of the right bar with robot simulation.

At the bottom, the user can see packages that are installed into the robot and options for install new ones or delete one of them.

Finally,in the middle, we have the space for the block designing.

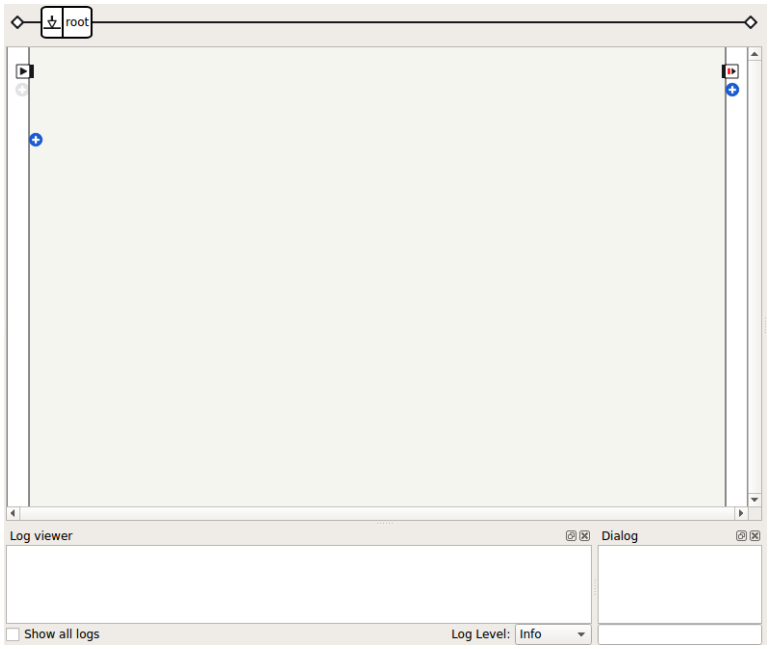

Figure 10: Image of the block diagram design section.

At the top the reader can see timelines blocks, it contains block that will show other block designing space. In the middle the user can drop blocks for making his behaviours, he have to connect the first one to the left arrow and the last one in the other side, at the right arrow.

At the bottom the reader can see the log viewer for see error details, he can use the dialog text for writting messages to the robot, it provides to the user that he do not need to talk if he want to test for example a speech recognition block.

3.2. HOW TO CONNECT TO PEPPER

In this section we are going to teache

4. SPEECH RECOGNITION

4.1. ANIMATIONS

4.2. MOVEMENT

5. FACE RECOGNITION

6. DIALOGUE

7. CONCLUSIONS

7. REFERENCES

[1] Apache Cordova, cordova.apache.org. Consultado el 10 de Abril del 2018. https://cordova.apache.org/docs/en/latest/

[2] ArcGIS for Developers, developers.arcgis.com. Consultado el 14 de Abril del 2018. https://developers.arcgis.com/

[3] Bouskela, Mauricio., et al (2016),*La ruta hacia las Smart Cities: Migrando de una gestión tradicional a la ciudad inteligente*. Banco Interamericano de Desarrollo.

[4] Dirección General de Tráfico, www.dgt.es. Consultado el 13 de Abril de 2018. http://revista.dgt.es/es/categorias/distracciones-moviles.shtml

[5] IBM Cloudant, www.ibm.com. Consultado el 14 de Abril del 2018. https://www.ibm.com/cloud/cloudant

[6] IBM Bluemix, www.ibm.com. Consultado el 14 de Abril del 2018. https://www.ibm.com/cloud-computing/bluemix/es

[7] Ionic, www.ionicframework.com. Consultado el 11 de Abril del 2018. https://ionicframework.com/docs/

[8] Marco de Desarrollo de la Junta de Andalucia, *Conceptos sobre Escalabilidad*, Consultado el 10 de Abril de 2018. http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220

[9] Node-RED, www.nodered.org. Consultado el 16 de Abril del 2018. https://nodered.org/docs/platforms/bluemix

[10] Pandorafms, *Bases de Datos NoSQL*, pandorafms.org. Consultado el 12 de Abril de 2018. https://blog.pandorafms.org/es/bases-de-datos-nosql/

[11] Universidad de Castilla-la Mancha. www.uclm.es. Consultado el 9 de Abril de 2018. https://eventos.uclm.es/19833/detail/hack-for-albacity.html

[12] URBIOTICA (2008), www.urbiotica.com. Consultado el 18 de Abril del 2018. http://www.urbiotica.com/producto/u-spot/

[13] W3Schools (1998), www.w3schools.com. Consultado el 15 de Abril del 2018. https://www.w3schools.com/nodejs/default.asp

ANNEXES

ANNEX I ENLACES DE CONTENIDO

- **Enlace a la WebApp:**
  `https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-Web`
- **(Aplicación de mapas en ejecución):**
  `https://hackaton-skyhold-web.eu-gb.mybluemix.net/`
- **Documentación (presentación y memoria):**
  `https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-Web/tree/master/DOCS`
- **Node-RED:**
  `https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-RED`
- **(Editor de Node-RED):**
  `https://hackaton-skyhold-red.eu-gb.mybluemix.net/red/`
- **Repositorio de la Aplicación:**
  `https://github.com/Mowstyl/Hackaton-Skyhold-App`

ACLARACIÓN: *Los diferentes servicios serán públicos o se proveerán los accesos necesarios a los mismos.*