EXPERIENCES IN USING THE PEPPER ROBOTIC PLATFORM FOR MUSEUM ASSISTANCE APPLICATIONS

Dario Allegra, Francesco Alessandro, Corrado Santoro, Filippo Stanco

Department of Mathematics and Computer Science, University of Catania
Viale Andrea Doria, 6 - 95125 Catania, Italy
{allegra, santoro, fstanco}@dmi.unict.it, francescosancono91@yahoo.it

ABSTRACT

This paper presents the software architecture of a robotic museum guide application called CUMA. It is intended to run upon the Pepper robotic platform and has the objective of guiding visitors of a museum accompanying them in the tour, explaining museum works, and interacting with them in order to gather feedback. CUMA has been partially implemented and preliminarily tested. The results reported in the paper, highlight that even if Pepper, from the structural point of view, seems particularly suited for this kind of application, the provided software platform presents some important limitations thus requiring the integration of external tools and algorithms.

***Index Terms*—** Museum Guide, Robotics, Cultural Heritage, Pepper

1. INTRODUCTION

Humanoid robots are gaining an increasing interest thanks to the recent availability, in the market, of many ready-to-use solutions. Been a matter of research institutions for a long time, many humanoid platforms are now available, equipped with their SDK that lets developers to easily implement a robotic application; these SDKs already include the typical low-level algorithms, such as motion, navigation, vision sensing, speech recognition, etc., offering high-level constructs to allow designer to rapidly prototype the robot behaviour. Humanoids are appearing in many contexts and, among them *assistive applications* represent key scenarios.

In this paper we focus on the robotic platforms *Pepper* [1] (Figure 1), a wheeled humaniod by Softbank Robotics. Among humanoid platforms, Pepper is a hybrid solution, since it has a human shape in the upper body, but omidirectional wheels in the lower part. Also the human-robot interface of Pepper is hybrid; it is able to interact using the classical human-like modes (e.g., speech recognition, face identification, arm motion), but it also equipped with a tablet PC that allows human interaction by means of a touch screen. Thanks its *multi-modal* user interface, Pepper is particularly suitable for applications that could gain value from combining text, pictures, speech and gesture. These features led us to choose Pepper to serve as robotic *museum guide*. Pepper comes with a software package including the IDE Choregraphe, to manage with Pepper behaviours through a graphical language, and a suite of libraries for Java, C/C++, Javascript and Python. By exploiting the software tools provided, we designed an assistance application to let a Pepper behave as a *touristic cultural guide*. The application, called CUMA (*Cultural hUManoid Assistant*) allows Pepper to accomplish some guidance tasks, such as to follow a visitor in its path in a museum, to give proper description about the various museum items, to provide useful information about the other touristic resources of the town, to answer questions, to gather feedback, etc. This work is part of Archeomatica (www.archeomatica.unict.it), a project about the preservation and exploitation of Cultural Heritage through Computer Science. CUMA is described in this paper together with a set of experimental results gathered during some tests. The application is mainly a proof of concepts to understand *(i)* the advantages to use a modern humanoid robot as museum guide as well as *(ii)* the strengths and limitations of the Pepper robotic platform in this context.

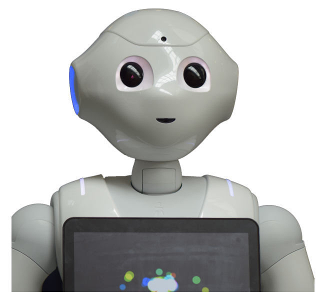

Fig. 1. The robot Pepper.

The paper is structured as follows. Section 2 provides an overview of similar papers dealing with the use of robots in cultural heritage context. Section 3 describes the software architecture of the museum guide application. Section 4 presents a qualitative evaluation of the solution developed. Section 5 provides our conclusions.

2. RELATED WORK

The effectiveness of communicating historical information plays a key role on the ability to engage museum visitors [2]. In this respect, robots provided of social inference can give a notable contribution [3]. However, the use of robot in museum environment has a long story. In 1999, Burgard et al. [4, 5] presented RHINO a tour-guided robot deployed for six days at Deutsches Museum Bonn. The architecture consists of different modules to perform localization in dense crowds, collision avoidance and the merging of first order logic and numerical robot control. The robot is also equipped with a mixed-media interface that integrates text, graphics, pre-recorded speech and sound. The museum robot MINERVA [6, 7], was presented in 2000 by Thrun et al. Differently from RHINO, it allows to learn the map from sensor data and presents an improved interaction system with the users. Moreover, MINERVA is equipped with a camera for localization purpose and it is able to compose simple tour on-the-fly by employing data recorded in the past. Nourbakhsh et al. proposed the robot SAGE at the Carnegie Museum of Natural History [8]. Its goal was to provide educational content to museum visitors to improve their museum experience. SAGE is able to build no internal representation of the environment to distinguish between permanent obstacle and temporary ones. In 2003, Nourbakhsh et al. [9] published the results of a five-year experiment about three tour-guided robot. They mainly focus on the educational contribution of the installed robots. At the end, although their study demonstrated real educational efficacy, the robots have been shut down because of their high cost. During the last decade, Computer Vision algorithms gave the opportunity to the robots to understand human behavior. In [10], it is proposed a robot Robovie-R3, able to proactively initiate interaction with target people and offer them an explanation about any particular painting. This system used multiple video camera for detecting and tracking people's visual focus of attention. Recently, the importance of robotic technologies for museum is remarked in [11]. Additionally, in [11], it is proposed a telepresence robot to explore inaccessible areas of the heritage.

3. THE CUMA PROJECT

3.1. Overview

The overall objective of the CUMA project is to develop a humanoid assistant in order to accomplish the following tasks:

- to welcome a museum visitor and gather some data to recognize her/him during the guided tour;
- To organize the plan and the schedule of the tour and to accompany visitors along the itinerary;
- To provide information about the history, the culture and the context of the various museum works, also showing additional pictures or video contents;
- To answer to questions concerning museum items;
- To gather a feedback from visitors about the quality of the service offered by itself and the museum.

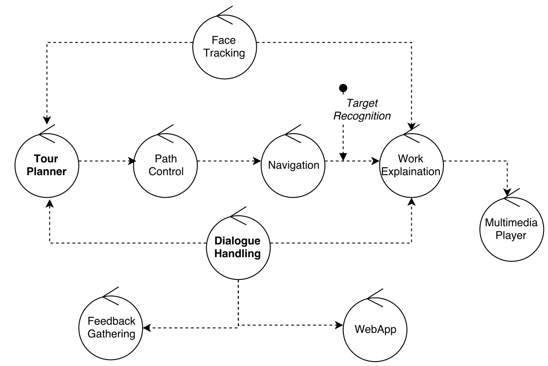

Fig. 2. Software Architecture of the CUMA Application

3.2. Software Architecture

To support the tasks described above, we designed an application made of a set *behaviours* whose architecture is depicted in Figure 2. In the figure, each circle corresponds to a specific behaviour while arrows represent a dependence, meaning that a behaviour controls another one, or interact by simply exchanging data. Each behaviour contains the code to activate and interpret the proper sensors, to let the robot performs the required actions and to interact with other behaviours.

Tour Planner and **Dialogue Handling** can be considered as the *main behaviours*, since they are responsible to the working scheme of the overall assistance application. They, in turn, interacts with the other behaviours. **Tour Planner** is a deliberative behaviour which makes Pepper able to determine the specific tour to follow. It performs a planning process by selecting a set of museum items to be visited and composing the tour accordingly; in addition, some pre-configured tours can be loaded (e.g., by an operator) to let the robot makes a selection among one of them. The selected tour is thus represented by an ordered list of the museum items to be visited. According to the characteristics of the group of visitors, selection can be made either by an automated process performed by CUMA on the basis of a pre-established *decision tree*, or by a human museum operator. The automatic strategy should be preferred since it can exploit the various sensing abilities of Pepper; indeed,its library functions in SDK can use data from cameras to perform tasks like counting number of people in the group, guessing the age and the gender of people sight, or guessing their mood; these information that can be used to drive the decision process. These specific sensing activities are in charge of the **Face Tracking** behaviour that, as shown in Figure 2, can drive the Tour Control; such sensing data are used not only in the first tour planning phase, but also

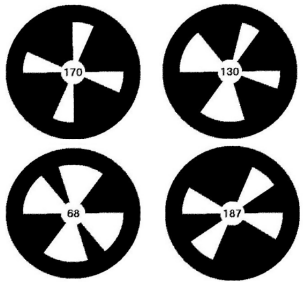

Fig. 3. Examples of landmarks used in our work.

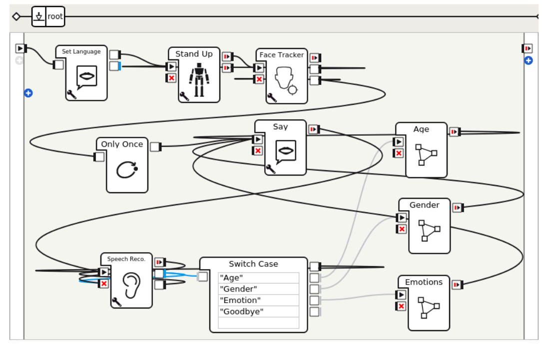

Fig. 4. A Screenshot of the Choreographe Tool

during the tour itself, in order to catch emotions like fatigue, boredom or happiness and re-plan the tour accordingly.

Once the tour has been planned, it is passed to the **Path Control** behaviour that has the task of determining the optimal path, within the museum, that lets accompanied tourist visit the works selected. Such a path is provided to the **Navigation** behaviour that is in charge of properly driving the Pepper motion system. Navigation is performed using both a map of the environment that has been previously created and some specific *landmarks* (see Figure 3) that can help Pepper to determine its position in the museum. Landmarks are also used to identify museum items: while their position in the museum is known (by the Tour Planner and the Path Control), a landmark permits to detect the precise location of the robot w.r.t. the museum item and lets Pepper to properly approach it.

Once the target landmark is detected and the work is approached, the execution of the **Work Explanation** behaviour is triggered (see Figure 2); Pepper starts reciting the description of the work (by means of text-to-speech) and showing multimedia contents—if available—on the tablet PC, activity that is performed by the **Multimedia Player** behaviour. Once the description of the work is completed, the Path Control and Navigation behaviours intervene driving Pepper towards the next work until the overall visit is over.

During the execution of the various tasks related to the visit, another behaviour, the **Dialogue Handling**, runs in parallel. This behaviour has the objective of managing a speech-based dialogue with visitors: people may ask questions to Pepper using natural language or provide a feedback on the tour or on the services offered by the museum. The Dialogue Handling may also trigger an additional **WebApp** that, by exploiting the tablet PC and by means of a classical web-based navigation, has the task of showing touristic resources of the town along with speech-image-video description of the places. Feedbacks are instead managed by the **Feedback Gathering** behaviour that, by means of an interview based on natural language interaction and using a set of pre-defined questions, can acquire the visitor evaluations, storing them into a proper database for further analysis.

3.3. Implementation Remarks

The CUMA application, designed as detailed in the previous subsection, has been partially implemented in our labs in order to test and verify its effectiveness. As it has been introduced in Section 1, the Pepper platform has several options to program the behaviours of the robot; the most common way is by means of Choreographe, a graphical tool exploiting a form of data-flow diagrams based on functional blocks that can be connected through "data wires" (see Figure 4). Indeed, while using Choreographe allows a rapid prototyping of a Pepper application, in the long term, the resulting diagram becomes quite hard to read and maintain; for this reason, we decided to use the API of the SDK and develop the application using the Java programming language, for the behavioural part, and HTML/JavaScript for the apps running on the Tablet PC.

We implemented some parts of the described architecture by concentrating, above all, on Face Tracking, Dialogue Handling, Feedback Gathering and WebApp. At the current stage, these behaviours are implemented exploiting the sensing and recognition algorithms provided by the API, in particular *face recognition*, *age/gender recognition*, *mood recognition*, *marker identification*, *SLAM* (for environment mapping and navigation) and *chat* (for natural language interaction). Figure 5 shows a picture of the WebApp which is display a choice of some of the best touristic places of the town, allowing the visitor to touch a choice or say the associated caption in order to perform navigation.

The WebApp is written in HTML and JavaScript and runs on the Tablet PC. Here, HTML is used to perform information display and touch-based navigation while, by means of JavaScript, the WebApp interacts with other behaviours: indeed there is a strict interaction between the WebApp and the Dialogue Handling since navigation can be also performed by using natural language.

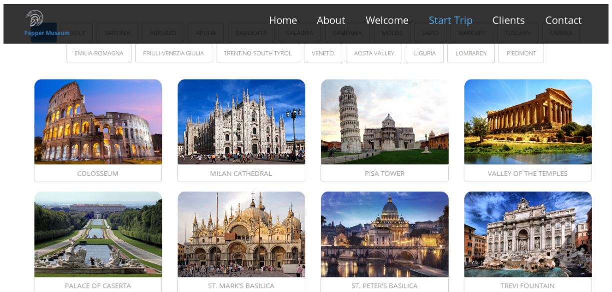

Fig. 5. A Screenshot of the WebApp

4. DISCUSSION

Behaviour programming for humanoid robots is one of the most difficult software development tasks since the robot is expected to exhibit a certain form of human-like behaviour: the robot is expected to possess the ability to well understand phrases in natural languages and reply accordingly, to avoid repetitive actions, to move in the environment without particular problems, etc. We tested our implementation with some people, observed the behaviour of both the people and the robot, and gathered a feedback with the objective of understanding to what extent the software platform of Pepper (and in particular the API and the algorithms provided) is able to meet user expectations. On this basis, we made a preliminary and qualitative evaluation leaving to future works a quantitative/numerical performance analysis.

The first set of features we evaluated is the one related to computer vision, i.e. face recognition, age/gender recognition, mood recognition. All the provided algorithms related to these features work as expected: faces are well recognized even if the learning stage requires a long time (5 seconds) to let a new face be properly learned; gender recognition performs well and age guessing is able to identify the range.

Things go worse for natural language interaction: while *text-to-speech* is quite effective (also allowing specification of accents or prosody by means of an ad-hoc syntax), the same cannot be said for *speech-to-text*; indeed, the Pepper API has not a real speech-to-text module but a *chat module* whose functions can be programmed to perform the recognition of certain specific keywords within a phrase and (in turn) directly answer with other programmed phrases, in a pure reactive way. The absence of a module able to directly generate the string from the speech impedes to implement a form of "more rational" answers, that could be generated on the basis of e.g. a NLP engine or a deep learning module; as a consequence, speech-based interaction with Pepper results in a bare chat with pre-defined replies that, in the long term, bores and annoys the speaker. Moreover, the recognition of the Italian language is not of the same quality as the case of English, thus featuring a lot of false positive or misunderstanding, above all when the diction of the speaker is not perfect or there is a fair amount of noise in the room. To overcome this drawback, we tried to use speech-to-text services available as cloud services, like Google Speech API or Bing Services: with this approach we experienced quite better results, even if the latency of the network affects sometimes the promptness of the application.

Another feature that needs to be improved is navigation. Pepper library algorithms allow the creation of a map of the environment (by exploiting Pepper's distance sensors) that can be in turn used to let the robot navigate through the mapped zone. However navigation requires a manual set-up of the initial position of Pepper and works only in a limited environment, e.g. a room: changing the room implies to load another map and (manually) re-localize the robot in the new zone. So there is no way (with the tools natively offered by the Pepper platform) to have a knowledge of a global position of the robot, a feature that instead is desirable in a museum since the environment is particularly large and complex.

As a result, in our experience, the Pepper platform "as-is" is not properly suitable for the purpose of assistance projects like CUMA, and it needs to be integrated with additional libraries and/or sensors, or be integrated with cloud services, in order to add more precision, flexibility and "intelligence". This kind of aspects is currently under our investigation and will be the aim of our future works.

5. CONCLUSIONS

In this paper we proposed a new architecture for a tour-guided robot in museums. The humanoid robot Pepper [1] has been equipped with several modules, which implements a set of behaviors: Tour Planning, Face Tracking, Path Control and Work Explanation. Additionally, we provided a Dialogue Handling module that can be triggered through a speech recognition system or a novel WebApp. Because of the long time required for a proper quantitative evaluation, we preliminarily provided qualitative result of our system. Overall, we found out that Computer Vision modules worked well while the speech recognition and navigation strategy have to be improved.

In future works, we plan to integrate additional tools and libraries that can help to overcome basic Pepper platform limitation. As for Dialogue Handling, the interaction with cloud-based speed-to-text services is still in progress, and we are also investigating the integration with an AI engine, such as IBM Watson [12], in order to provide more rational answers by means of NLP and deep learning. Moreover, for behaviour programming, we will analyze tools based on the BDI (*Belief-Desire-Intention*) paradigm [13, 14] with the aim of giving Pepper a better form of rationality. Finally, inspired by [15], we are considering to add an Augmented Reality module to make Pepper able to improve user educational experience.

6. REFERENCES

[1] Softbank Robotics, "Pepper home page," *WWW, https://www.ald.softbankrobotics.com/en/robots/pepper*, 2018.

[2] C. Goulding, "The museum environment and the visitor experiences," *European Journal of Marketing*, vol. 34, pp. 261–278, 2000.

[3] F. Hegel, C. Muhl, B. Wrede, M. Hielscher-Fastabend, and G. Sagerer, "Understanding social robots," in *International Conferences on Advances in Computer-Human Interactions*, February 2009, pp. 169–174.

[4] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot," in *Artificial Intelligence/Innovative Applications of Artificial Intelligence*, 1998, pp. 11–18.

[5] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, pp. 3–55, October 1999.

[6] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: a second-generation museum tour-guide robot," in *International Conference on Robotics and Automation*, 1999, vol. 3.

[7] S. Thrun, M. Bennewitz, M. Beetz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *The International Journal of Robotics Research*, vol. 19, pp. 972–999, 2000.

[8] I. R. Nourbakhsh, A. Soto, J. Bobenage, S. Grange, R. Meyer, and R. Lutz, "An effective mobile robot educator with a full-time job," *Artificial Intelligence*, vol. 114, pp. 95–124, October 1999.

[9] I. R. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: a five year experiment," in *International Conference on Intelligent Robots and Systems*, October 2003, vol. 4, pp. 3636–3641.

[10] M. G. Rashed, R. Suzuki, A. Lam, Y. Kobayashi, and Y. Kuno, "A vision based guide robot system: Initiating proactive social human robot interaction in museum scenarios," in *International Conference on Computer and Information Engineering*, November 2015, pp. 5–8.

[11] C. Germak, M. L. Lupetti, L. Giuliano, and M. K. Ng, "Robots and cultural heritage: New museum experiences," *Journal of Science and Technology of the Arts*, vol. 7, pp. 47–57, 2015.

[12] IBM, "Ibm puts watson into softbank pepper robot," *https://www.ibm.com/blogs/nordic-msp/pepper-robot/*, 2018.

[13] L. Fichera, F. Messina, G. Pappalardo, and C. Santoro, "A python framework for programming autonomous robots using a declarative approach," *Science of Computer Programming*, vol. 139, pp. 36–55, 2017.

[14] F. Messina, G. Pappalardo, and C. Santoro, "Integrating cloud services in behaviour programming for autonomous robots," in *Lecture Notes in Computer Science*, 2013, vol. 8286, pp. 295–302.

[15] F. Stanco, D. Tanasi, G. Gallo, M. Buffa, and B. Basile, "Augmented perception of the past - the case of hellenistic syracuse," *Journal of Multimedia*, vol. 7, no. 2, pp. 211–216, 2012.