A Feasibility Study of Culture-Aware Cloud Services for Conversational Robots

Carmine T. Recchiuto and Antonio Sgorbissa

***Abstract*—Cultural competence - i.e., the capability to adapt verbal and non-verbal interaction to the user's cultural background - may be a key element for social robots to increase the user experience. However, designing and implementing culturally competent social robots is a complex task, given that advanced conversational skills are required. In this context, Cloud services may be useful for helping robots in generating appropriate interaction patterns in a culture-aware manner. In this letter, we present the design and the implementation of the CARESSES Cloud, a set of robotic services aimed at endowing robots with cultural competence in verbal interaction. A preliminary evaluation of the Cloud services as a general dialoguing system for culture-aware social robots has been performed, analyzing the feasibility of the architecture in terms of communication and data processing delays.**

***Index Terms*—Social human-robot interaction, service robots, human-centered robotics.**

I. Introduction

BENEFITS provided by Big Data access and analysis, Cloud computing, collective robot learning, and human-assisted computation have recently boosted the popularity of Cloud Robotics [1]. Initiatives of companies as Google, Amazon and Willow Garage, and more than a dozen active research projects around the world, (such as the RoboEarth project [2] and DAvinCi, a Cloud Computing framework for service robots [3]), are emblematic in this sense. The field of Social Robotics has recently exploited the features offered by Cloud infrastructures. A Cloud Robotics solution aimed at improving social assistive robotics for healthy aging has been implemented in [4], [5]: the system was able to provide assistive user location-based services, by using Ambient Assisted Living (AAL) technologies and a sensorized environment. In a similar way, an Internet-of-Robotic-Things system architecture has been implemented for controlling a companion robot in [6], with the final aim of alleviating behavioral disturbances of people with dementia with a personalized and context-aware approach. Finally, in [7], context-aware dialoguing Cloud services were developed in order to embed a social robot with communication capabilities.

On these bases, this article presents the architecture and a feasibility study of a novel Cloud platform offering *culture-aware* dialoguing services for social robots. This Cloud infrastructure has been developed as a result of the CARESSES project[1] [8], a joint EU/Japan research project aimed at implementing culturally competent robots, i.e., robots able to apply an understanding of the culture, customs, and etiquette of the person they are assisting, while autonomously reconfiguring their way of acting and speaking.

In the robotic domain, research on cultural factors has mainly focussed on non-verbal aspects such as facial expressions [9], [10], greeting gestures [11], and interpersonal distance [12], which have been subjects of specific cross-cultural studies involving robots and virtual agents. The relevance of culture when designing dialogue patterns of a virtual agent has been pointed out in [13], [14], while culture-dependent speech patterns of a virtual agent speaking a meaningless gibberish language have been analyzed in [15]. As a general outcome, the results of these works suggest that cultural aspects may affect the robot's (or virtual agent's) likeability, acceptance, and persuasiveness, in the sense that people tend to prefer an artificial agent that conforms to the social norms of their own culture, both in the verbal and non-verbal behaviour. However, although underlying the importance of blending cultural factors into the Social Robotics domain, all the aforementioned approaches integrated a very small set of features that distinguish cultures from each other, and little work has been reported on how to build robots that can be easily adapted to the cultural identity of the user.

The CARESSES project has represented the first attempt to implement a robot that adapts its behavior according to the cultural identity of the person with whom it interacts, in terms of actions, actions' parameters (e.g., social distance, speech volume) [16], and dialogue patterns. In the context of the project, the evaluation of whether and how a robot using the CARESSES dialoguing framework is perceived as culturally competent has been carried out in a six-month experimental campaign, involving older persons belonging to different cultural groups, their informal caregivers and the humanoid robot Pepper [17], [18]. The evaluation results will be the subject of future publications.

This work, after briefly summarizing the underlying software architecture of the conversational framework and its functionalities [19], [20], focusses on a specific aspect: the analysis of the performance of the CARESSES system refactored as Cloud services, thus analyzing its feasibility, in terms of communication and data processing delays, as a general dialoguing framework for culture-aware social robots.

Manuscript received February 23, 2020; accepted July 22, 2020. Date of publication August 11, 2020; date of current version August 20, 2020. This letter was recommended for publication by Associate Editor H. S. Ahn and Editor D. Lee upon evaluation of the Reviewers' comments. This work was supported in part by the European Commission Horizon 2020 Research and in part by Innovation Programme under Grant 737858. *(Corresponding author: Carmine Tommaso Recchiuto.)*

The authors are with DIBRIS, University of Genova, 16145 Genova, Italy (e-mail: carmine.recchiuto@dibris.unige.it; antonio.sgorbissa@unige.it).

Digital Object Identifier 10.1109/LRA.2020.3015461

[1] http://caressesrobot.org

The article is structured as follows. Chapter II describes the general architecture of the culturally competent dialogue system, whereas Chapter III describes the proposed Cloud services, with a brief description of their motivation and some examples. Chapter IV evaluates the performance in terms of latency in communication and processing of the Cloud infrastructure. Finally, Chapter V discusses the preliminary outcomes of the proposed approach and presents conclusions.

II. Conversational Framework Architecture

In the context of the CARESSES project, a closed-loop approach between Roboticists and Transcultural Nursing researchers [21] has led to the definition of a knowledge-driven, conversational framework for culturally competent robots, which relies on two core elements:

I) A "three-layered" Ontology for storing all concepts of relevance, cultural information and statistics, person-specific information and preferences;

II) An algorithm for building culture-aware dialogue patterns, relying on I.

The nucleus of the conversational framework is a Description Logics Ontology, a formal representation of objects, concepts, and other entities, assumed to exist in some area of interest, and the relations that hold among them [22]. Knowledge about concepts and their mutual relations are stored in the terminological box (TBox) of the Ontology, while knowledge that is specific to instances belonging to the domain is stored in the assertional box (ABox). From an implementation perspective, the OWL-2 language [23] has been adopted. In this formalism, the TBox is composed of classes and properties, which include Data Properties, relating instances of a class to literal data (e.g., strings, numbers), and Object Properties, relating instances of a class to other instances; the ABox stores instances of classes and instances. Given these definitions, the three layers in which the Ontology is ideally divided (I) are:

- A layer that stores the terminology (TBox) required to represent all the information that may play a role in a culture-aware conversation, but without being specific for a given culture: beliefs, values, habits, preferences, objects, norms, among the others.
- A layer that stores the assertions (ABox-I), required to represent culture-specific information.
- A layer that stores the assertions (ABox-II) required to represent the unique cultural identity, preferences, social and physical environment of the assisted person.

The architecture is represented in Fig. 1. The TBox (a) encodes concepts at a generic, culture-agnostic level (e.g., the class *Beverage*). The ABox-I (b) is composed of instances of classes, encoding culturally appropriate sentences (Data Property hasSentence) and the probability that the user would have a positive attitude toward that concept, given that he belongs to that cultural group (Data Property hasLikeliness). Finally, the ABox-II (c) comprises user-specific instances: instances of this layer may encode the actual user's attitude about that concept to be updated during the verbal interaction by collecting his/her feedback (i.e., the system may discover that the attitude towards

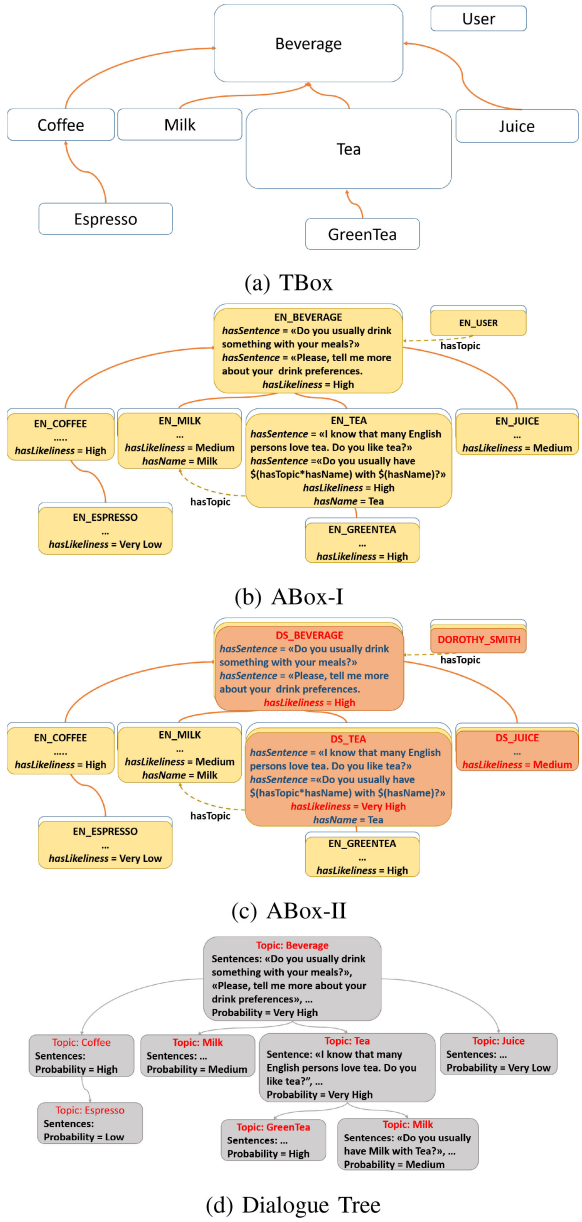

Fig. 1. The three layers of the Ontology: TBox, Abox-I (for the English culture) and Abox-II (for the user Dorothy Smith), and the Dialogue Tree generated from the Ontology structure.

tea of Mrs. Dorothy Smith is more positive than the average English person), or specific knowledge about the user (e.g., name, town of residence) explicitly added during setup.

The dialogue tree is built starting from the Ontology structure: each instance of the ABox-I layer is seen as a *conversation topic*, i.e., a node of the tree. The relation between topics is borrowed from the structure of the Ontology: specifically, the Object Property hasTopic, and the hierarchical relationships among instances, are analyzed to define the branches of the dialogue tree. In the example of Fig. 1, the instance of Tea for the English culture is connected, in the dialogue tree, to its *children* nodes GreenTea (which is a subclass of Tea in the Ontology) and Milk (since EN_MILK is a filler of EN_TEA for the Object Property hasTopic).

Based on the dialogue tree, the policies for knowledge-driven conversation can be briefly summarized as follows [24]. Each time a user sentence is acquired (Algorithm 1):

1) A keyword-based Language Processing algorithm is applied to check if the sentence may trigger one of the topics in the tree.
2) If no topics are triggered, the conversation follows one of the branches of the tree, depending on the probabilities of each node (Remark I below).
3) Whatever node has been chosen, the system:
   i) proposes some of the corresponding sentences (Remark II below));
   ii) acquires the user's feedback that can be used to update the Ontology with user-specific instances and/or determine the next node to move to.

It shall be mentioned that the user is always allowed to give specific commands to the system (e.g., *play music*) rather than engaging in a knowledge-driven conversation: this is implemented as a parallel mechanism as it will be described in Section III.

Two additional aspects need to be remarked:

Remark 1: The probability that a node/topic is selected is given by the value of the Data Property hasLikeliness of the user-specific corresponding instance (ABox-II), or, if this does not exist, of the culture-specific corresponding instance (ABox-I). In other words, if the attitude of the user toward a concept is known, its value is taken into account (instead of the more generic culture-specific attitude) when choosing the next conversation topic. In the Fig. 1 example, since we know that Dorothy Smith does not like drinking juice, a very low probability is assigned to the topic Juice, even if it may be popular in England. The presence of the two bottom layers plays a key role in the context of *personalization* [25], and avoids a stereotyped representation of cultures.

Remark 2: A topic may have several sentences, and a sentence may be assigned to a topic in two ways: it may be encoded *as is*, as a Data Property of the instance (e.g., *I know that many English persons love tea. Do you like tea?*, Fig. 1(b)–(d)), or automatically built by relying on the Ontology structure. As an example, the Data Property hasName can be used as a variable for automatically building sentences with a common pattern. In Fig. 1(b)–(d), the sentence *Do you usually have $(hasTopic* hasName) with $(hasName)?*, encoded as a property of the instance EN_TEA, is assigned to the topic Milk, which is a child of Tea in the Dialog Tree, Fig. 1 (d). In the final sentence, the two variables, indicated with the symbol $, are replaced with the corresponding values stored in the property hasName of EN_TEA (i.e., 'Tea') and the property hasName of its filler EN_MILK for the property hasTopic (i.e., 'Milk'). This mechanism allows for building several sentences with the same pattern, by simply using the Data Properties hasSentence and hasName to produce class restrictions which are inherited by all subclasses and related instances.

Finally, it is worth mentioning that a Bayesian Network is associated with the dialogue tree for speeding up culture-aware personalization by propagating the acquired information to interconnected concepts [19].

III. Culture-Aware Cloud Services

This work analyzes the feasibility of re-engineering the conversational framework described in Section II as a portfolio of Cloud services. This choice is motivated by several reasons: among the others, the possibility of processing cultural information encoded in the Ontology using remote computing services (including massively-parallel computation), the usage of collective learning strategies for a run-time expansion of the knowledge base, and the integration of the framework with a wide range of robotic devices, such as humanoid robots, table-top robots, smartphone applications, and voice assistants. The culture-aware services have indeed been integrated with three robotic platforms and devices: the Pepper humanoid robot, realized by Softbank Robotics[2] (used during the tests of the CARESSES project), Pillo, a robotic pill dispenser with social abilities developed by Pillo Health,[3] and a custom Android smartphone app (Figure 2).

Very important, as already mentioned in Section II, please note that the culture-aware Cloud services may easily work in conjunction with existing Language Processing systems (onboard or anyway integrated with the robot), with the final aim of complementing them. In other words, the developed Cloud services are not meant at substituting the conversational capabilities that the robot is already equipped with, but rather provide a backup plan whenever onboard language processing techniques are not able to interpret what the user is saying. Usually, in commercial products, the backup plan foresees a connection to Wikipedia or other websites: the proposed Cloud services provide a culturally competent alternative, allowing any robot or system to talk with the user about thousand of different topics. In practice, the user's demand is handled locally if the request can be achieved by the robot, otherwise the user's sentence is sent to the Cloud, by implementing a simple protocol for communicating.

Algorithm 1, described in Section II, produces a mixed-initiative verbal interaction [24]: both the user and the robot can take the initiative in leading the conversation. This is aimed at providing dialogues that overcome the typical *command-only barrier* [26], which is the main limitation of home assistants and most robotic systems, typically expecting a command from the user (*Robot, tell me the weather report!*) which is then reactively executed. On the opposite, the proposed Cloud services are able to produce rich goal- and knowledge-driven dialogue patterns, making the robot able to execute different kinds of speech acts [26]. Sentences are sent from the Cloud together to an associated Tag (also encoded in the Ontology) that specifies the typology of speech act. Table I describes the possible speech acts currently encoded in the dialogue tree by composing sentences in the Ontology, together with their associated Tags.

As the reader may observe, Directive speech acts play a key role, since they explicitly request a user's reply, and therefore deserve special attention. On the Cloud side, Directive speech acts, performed by the robot to explore the user's preferences,

[2]https://www.softbankrobotics.com/us/pepper
[3]https://pillohealth.com/

TABLE I
CARESSES CLOUD SERVICES: SPEECH ACTS AND TAGS

Speech act		Example	Tag
Directive	Question	Do you like baseball?	DQ
	Goal	Do you want me to set a reminder for the baseball match?	DG
	Open	Please tell me what is your favourite baseball team.	DO
Assertive		I know that baseball is very popular in Japan.	A
Commissive		I will remember that you like baseball.	C
Expressive	Positive	I am happy to hear that you enjoy watching baseball matches	EP
	Negative	Baseball is not everyone's cup of tea.	EN

constitute the basis to update the Ontology (and thus the dialogue tree) depending on the information acquired. On the robot's side, Directive speech acts require to implement mechanisms for retrieving the user's feedback, e.g., Speech-To-Text (possibly relying on additional, dedicated Cloud Services) or touch-based GUIs.

For the sake of clarity, a possible interaction with a robot connected to the proposed Cloud is described in the following, and depicted in the Sequence Diagram of Fig. 2. Suppose that a Japanese man, Yamada Tarō, is interacting for the first time, in English, with a robot connected to the Cloud. At first, the client program running on the robot is supposed to send all information about the user: background culture (Japanese), language (English), and personal data, such as the first name and the family name. The user's information is essential for the system to start with a culturally competent interaction, that will later take benefit of the additional information acquired in run-time. Currently, the Cloud supports three cultures: English, Indian, Japanese,[4] and three languages (English, Japanese, Italian). Please notice that cultural identities and languages are not necessarily coupled in the system, as the current example shows (in the Ontology, a culture-specific instance may have a number of Data Properties hasSentence written in different languages).

The Cloud will notify the client that the user information has been received and is being processed (*Wait* in Fig. 2). A new message (*Ready*) is then sent when the dialogue tree has been rebuilt, taking into account the user information. During the interaction, sentences pronounced by the users should be handled locally by a parallel mechanism, if they are directly aimed at starting an activiy (e.g., *Play some music*, *Show me the weather report*), or sent to the Cloud in case the robot does not know how to reply. For example, suppose that the user says *Now it is time to pray:* in this case, the client running on the robot is likely not to recognize it as a command aimed at executing an activity, and relies on the culture-aware Cloud services, forwarding the user's intention to start the conversation (*chitchat:started*), and the user's sentence (*userSays:[Now it is time to pray.]*).

On the Cloud side, the principles summarized in Algorithm 1 are implemented: the user's sentence is analyzed, and an appropriate conversation topic is chosen. The Cloud takes the

[4]This is strictly related to the cultural groups considered within the CARESSES project.

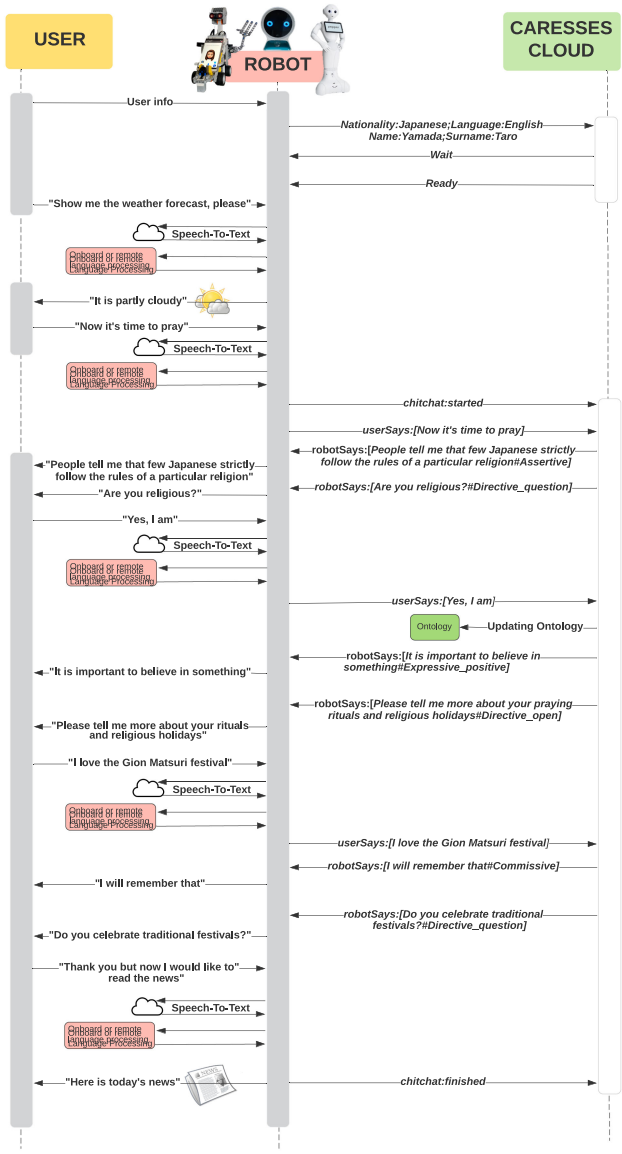

Fig. 2. Sequence Diagram describing a possible interaction with the CARESSES Cloud. The figure shows that the proposed culturally competent Cloud services are complementary to existing frameworks for Speech-To-Text and Language Processing services. The framework has been integrated with three devices: Pepper, Pillo and an Android application.

initiative of the conversation: it may reply with a sentence that shows the robot's cultural awareness (*RobotSays:[People tell me that few Japanese strictly follow the rules of a particular religion.#A*), it could ask a question related to the user's attitude with respect to the current conversation topic (*RobotSays:[Are you religious?#DQ*), or let the user freely talk about it (*RobotSays:[Please, tell me more about your praying rituals and religious holidays.#DO*). As said before, sentences are associated with Tags that specify the typology of the sentence, and if the user's feedback is required (*DQ*, *DO*, and *DG* sentences). After the user's reply, the Ontology may be updated depending on the feedback received, and the Cloud will probably send an *Expressive* (*It is important to believe in something.#EP*) or

Commissive sentence (*I will remember that.#C*), depending on what the user said.

The dialogue may possibly continue by exploring the branches of the dialogue tree more deeply, according to Algorithm 1, always taking into account the user's cultural identity, through a sequence of educated guesses, based on topics' probabilities. (*RobotSays:[Are you a Shintoist?#DQ*), *RobotSays:[Are you a Buddhist?#DQ*). In the case of positive feedback from the user, the Cloud will probably propose a new cultural competent sentence (*RobotSays:[I know that Shinto shrines are the places of worship and the homes of kami.#A*), request more information to the user (*RobotSays:[Please, tell me more about your beliefs as a Shintoist.#DO*), or suggest to propose an activity related to the current topic of conversation (*RobotSays:[Do you want me to help you to pray now? We could listen to the prayer for heaven and earth together.#DG*). Please notice that this latter kind of sentence is sent only if the robot is able to implement that goal, by relying on information added on the Cloud in the setup phase. Finally, when the conversation on that specific topic has ended, the Cloud notifies the robot that the dialogue pattern is over (*chitchat:finished*).

Very important, and according to Algorithm 1, the user may also take back the initiative at any moment, by saying something that triggers a different conversation topic on the Cloud (*userSays:[I love the Gion Matsuri festival!]*), or by directly asking for a robot's task (*userSays:[I would like to read the news.]*). As already mentioned, this kind of request shall be handled locally, by notifying the Cloud that the conversation is over through the message *chitchat:finished*.

For a more detailed description of the knowledge-driven dialogue algorithm, please refer to [19], [20] and [24].

IV. Evaluation and Results

As mentioned in Section I, the hypotheses about the impact of culturally competent interaction on the user have been tested and evaluated in the experimental phase of the CARESSES project [18], and they are the subject of ongoing publications. In the following, the analysis of Cloud services will be focused only on technical aspects, so as to evaluate the feasibility of the framework as a Cloud infrastructure. In particular, the performance of the CARESSES Cloud has been assessed by analyzing a specific parameter referred to as Chitchat Overall Latency (COL): i.e., the round trip time, measured on the client side, which incurs between sending the user's sentence (already converted into text) and receiving the robot's reply (the lowest COL the best, as this is key to keep verbal interaction natural and engaging). The round trip time has already been adopted as an evaluation parameter of Cloud services for assistive robots in some related works, such as [4], [5]. In these scenarios, a mean round trip time of less than 135 ms was assessed and evaluated as satisfactory for AAL applications.

For this evaluation, data have been collected over ca. 150 interactions, resulting from a full-day conversation between a volunteer person and the system. The complete Ontology, used during the experimental tests of the CARESSES project, which generates around 3000 topics, has been adopted for this

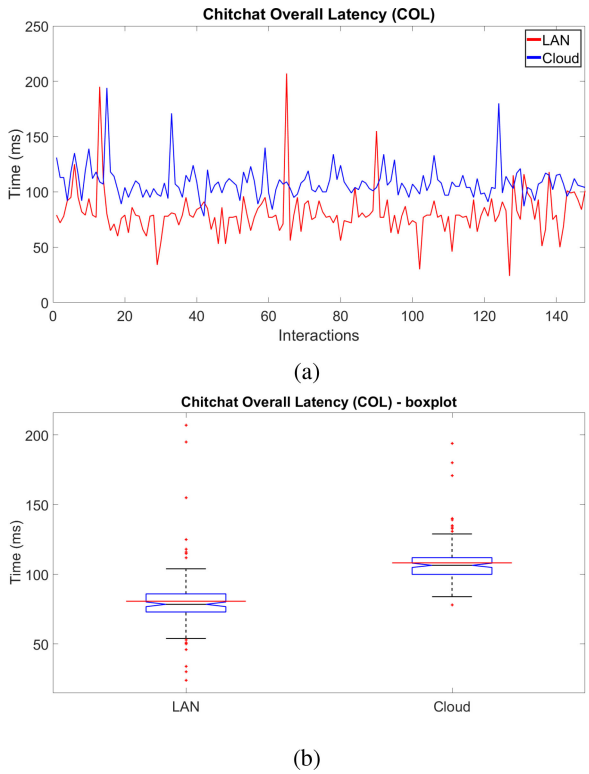

Fig. 3. Chitchat Overall Latency versus interactions (a) and boxplot (b) in the LAN and Cloud scenarios.

evaluation. Moreover, only the smartphone app was used: the COL, being the sum of the communication time and the Cloud processing time, only depends on the Cloud system, since none of these latencies is influenced by the device used. Please notice that, in the following, all results are graphically shown both in terms of COL versus interactions (Figure a) and are mediated to produce boxplots (Figure b). In the latter case, red lines represent the average values, the center of the notch is the median value, the bottom and top edges of the box indicate the 25th and 75th percentiles and the whiskers extend to the most extreme data points not considered outliers, while the outliers are plotted individually.

The COL has been evaluated and compared in two different scenarios:

- devices locally connected in the same LAN;
- devices remotely connected to the Cloud.

Tests have been executed on a i7-8550 U CPU, with 16 GB RAM, running Windows 10 while locally executing the system, and on a i7-6700 CPU, with 24 GB RAM, running Ubuntu 16.04 for the Cloud case. The results are shown in the following Table and Figures.

From the analysis of Fig. 3, it may be seen how the average latency (red lines in the boxplot) is similar in the LAN and the Cloud cases, being only 27.54 ms (34%) higher when using the system as a Cloud service. On the contrary, the standard deviation is definitely smaller when using the Cloud solution. This is probably due to the hardware used for executing the Cloud services, slightly more performant of the one used for the LAN version, and it runs a different Operating System (which implies

TABLE II
AVERAGE VALUES AND STANDARD DEVIATIONS OF LATENCIES WHILE VERBALLY INTERACTING WITH THE SYSTEM

Test Case	Overall Latency		CKB Processing		Communication Latency	
	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)
Local Connection	80.72	21.48	53.31	20.60	27.41	9.39
Cloud Connection	108.26	14.99	35.63	9.92	72.63	15.07

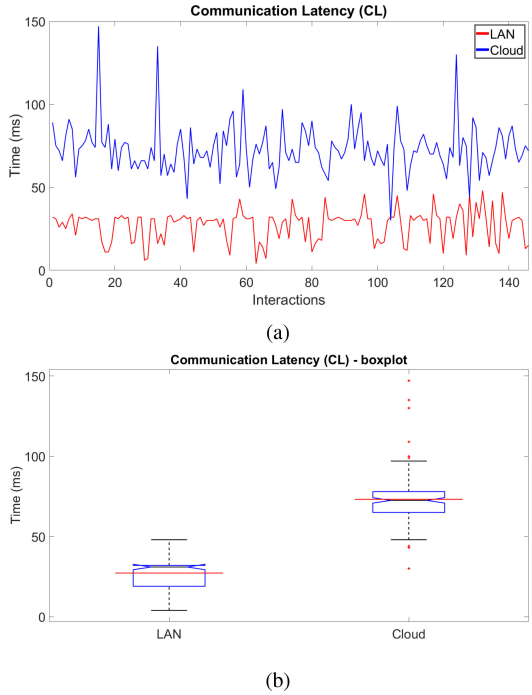

Fig. 4. Communication Latency versus interactions (a) and boxplot (b) in the LAN and Cloud scenarios.

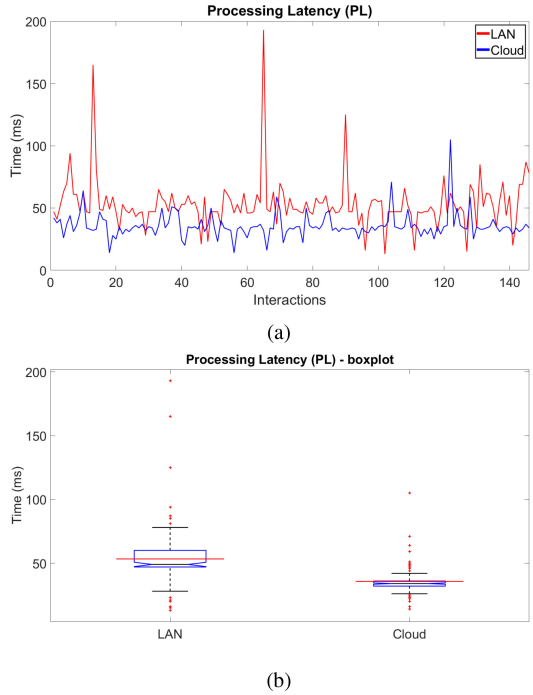

Fig. 5. Processing Latency versus interactions (a) and boxplot (b) in the LAN and Cloud scenarios.

different scheduling). However, this comes directly from the idea of having a Cloud architecture, that could be, in principle, executed in different and eventually more performant machines.

The analysis has been further refined by studying how these delays originate, i.e., by separately analyzing the delays due to communication between the two components (*Communication Latency* (CL), Fig. 4) and the delays due to the system's processing for updating the Ontology with the user's feedback and choosing the next appropriate sentence to be said (*Processing Latency* (PL), Fig. 5).

Table I summarizes all information showing at a glance COL, CL and PL in the two aforementioned scenarios. Some additional considerations can be drawn:

- Generally speaking, the PL due to the framework activity and the CL due to communication between Cloud and client have the same order of magnitude. Thus, the proposed system does not introduce a conspicuous runtime overhead during the interaction with the robot.
- Comparing the two scenarios, Fig. 4 confirms the expectations that the Cloud version leads to a higher CL, and with higher variance, with respect to the LAN version. However, Fig. 6 shows that the PL can be easily reduced by using Cloud services.
- The PL tends to be constant during the evaluated interactions (Fig. 5(a)), without any noticeable difference

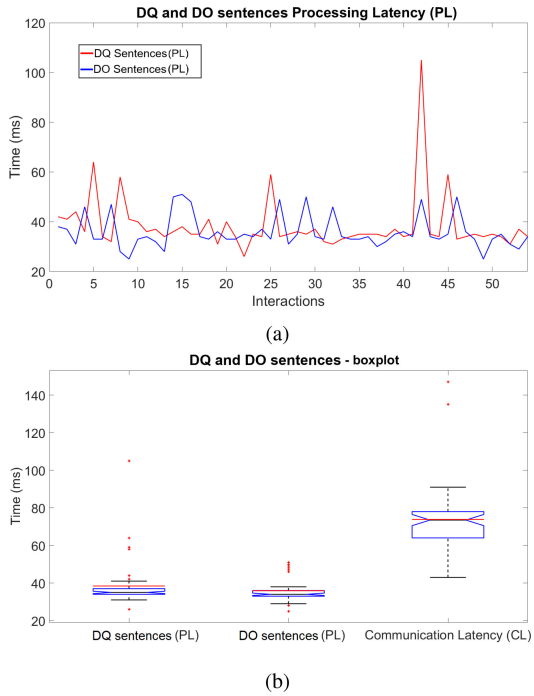

Fig. 6. Communication Latency and Processing Latency versus interactions (a) and boxplot (b), in the Cloud scenario, for two different types of speech acts (*Directive-Question* and *Directive-Open* sentences).

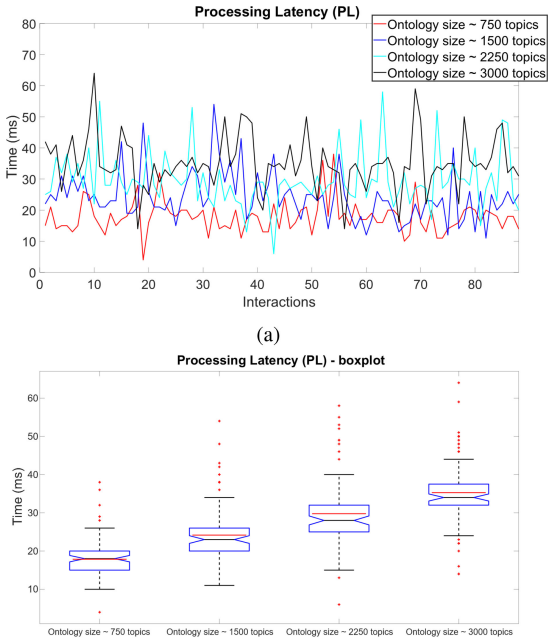

Fig. 7. Processing Latency versus interactions (a) and boxplot (b), with Ontologies of different sizes.

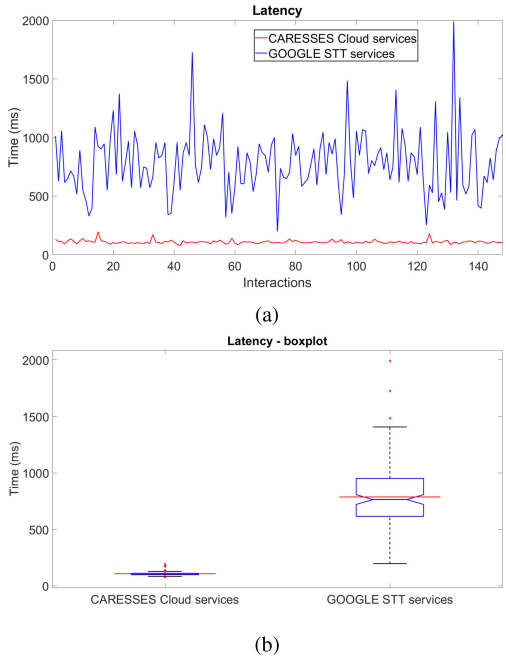

Fig. 8. Latency of the CARESSES Cloud services and of the Google Speech-To-Text (STT) services.

depending on different states in which the conversation algorithm may be from time to time. Notably, if the algorithm switches to a different conversation topic, or if the user's reply is used to update the Ontology, no higher latency is reported.

Concerning this last aspect, a more detailed analysis has been carried out. The PL has been evaluated (only in the Cloud scenario) during two different *Directive* speech acts that require a user's reply: questions directly investigating user's preferences (Directive-Question, Tag: DQ), whose reply is typically used to update the Ontology (by adding the user-specific information) and *open* questions (Directive-Open, Tag:DO), whose reply may contain a trigger for jumping onto a different conversation topic.

Fig. 6 shows that no particular differences exist in latencies between the two cases (2.59 ms on average), the CL between components being the more substantial contribution to the overall delay (72.68 ms).

As mentioned before, all these results have been collected with an Ontology comprising around 3000 topics. In order to evaluate the impact of the complexity of the Ontology on the COL, thus giving some insights about how the performance would scale, tests have also been done with reduced versions of the Ontology, generating dialogue trees of 750, 1500, and 2250 topics, analyzing the effect of scaling on the PL. The results are shown in Fig. 7. It may be observed how there is an almost perfect linear relationship between the size of the Ontology and the PL, where the smallest Ontology needs, on average, 17.84 ms to be processed, about half the PL of the complete one. In other words, adding 1000 conversation topics to the system causes an additional delay of about 8 ms, which is an order of magnitude lower than the CL.

Finally, the COL measured when using CARESSES Cloud services has been compared to the delays introduced by the widespread Google Speech-To-Text Cloud services (Fig. 8), computed during the same verbal interaction. This comparison is not aimed at directly confronting the latencies of the two systems, since they offer services that are completely different from each other: the CARESSES Cloud provides culturally competent dialogue patterns, while Google Cloud STT converts audio files to text strings. Notice that Google STT services may require more or less time to process audio depending on its length: thus, generally speaking, it is reasonable to expect that the latency introduced by Google STT, as well as its standard deviation, will be high.

The good news is that all Social Robots for verbal interaction with people are likely to already make use of some sort of STT Cloud service. This is also shown in the Sequence Diagram of Fig. 3, where STT is always required before using the complimentary, culturally competent Cloud services. Then, the latency due to STT shall be considered as a matter of fact, and we shall rather be concerned about the additional delay introduced by CARESSES Cloud services to produce a culturally competent reply to what the user said. Based on the above analysis, it can be seen how the runtime overhead introduced by the usage of the CARESSES services (108.26 ms) is negligible with respect to the one due to STT services (766.11 ms), thus not compromising the naturalness and pleasantness of the conversation.

V. Conclusion

The article presents the design and the implementation of the CARESSES Cloud, originally developed in the context of the joint EU-Japan research project CARESSES, offering culturally

competent dialogue patterns for Social Assistive Robots. The presented work does not aim to describe in detail the rationale behind the CARESSES software architecture or to validate the cultural services offered by the system and their impact on the user, which are the subject of previous and ongoing publications, but it has the purpose of describing and evaluating the potentialities of these novel Cloud services with different robotic platforms.

In particular, a technical assessment of the communication and data processing delays arising from the usage of the proposed culture-aware Cloud service has been carried out, showing how these latencies are satisfactory for verbal interaction in the Social Robotics domain, being also negligible with respect to widespread Speech-To-Text Cloud Services.

REFERENCES

[1] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Trans. Automat. Sci. Eng.*, vol. 12, no. 2, pp. 398–409, Apr. 2015.

[2] M. Waibel *et al.*, "Roboearth-a world wide web for robots," *IEEE Robot. Automat. Mag. (RAM), Special Issue Towards WWW Robots*, vol. 18, no. 2, pp. 69–82, Jul. 2011.

[3] R. Arumugam *et al.*, "Davinci: A cloud computing framework for service robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 3084–3089.

[4] M. Bonaccorsi, L. Fiorini, F. Cavallo, A. Saffiotti, and P. Dario, "A cloud robotics solution to improve social assistive robots for active and healthy aging," *Int. J. Social Robot.*, vol. 8, no. 3, pp. 393–408, 2016.

[5] L. Fiorini *et al.*, "Enabling personalised medical support for chronic disease management through a hybrid robot-cloud approach," *Auton. Robots*, vol. 41, no. 5, pp. 1263–1276, 2017.

[6] P. Simoens *et al.*, "Internet of robotic things: Context-aware and personalized interventions of assistive social robots (short paper)," in *Proc. 5th IEEE Int. Conf. Cloud Netw.*, 2016, pp. 204–207.

[7] J.-Y. Huang, W.-P. Lee, and T.-A. Lin, "Developing context-aware dialoguing services for a cloud-based robotic system," *IEEE Access*, vol. 7, pp. 44 293–44 306, 2019.

[8] B. Bruno *et al.*, "Paving the way for culturally competent robots: A position paper," in *Proc. 26th IEEE Int. Symp. Robot Human Interactive Commun.*, 2017, pp. 553–560.

[9] C. Chen *et al.*, "Equipping social robots with culturally-sensitive facial expressions of emotion using data-driven methods," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2019, pp. 1–8.

[10] T. Koda and Z. Ruttkay, "Eloquence of eyes and mouth of virtual agents: cultural study of facial expression perception," *AI Soc.*, vol. 32, no. 1, pp. 17–24, 2017.

[11] G. Trovato *et al.*, "A novel greeting selection system for a culture-adaptive humanoid robot," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 4, pp. 34–46, 2015.

[12] G. Eresha, M. Häring, B. Endrass, E. André, and M. Obaid, "Investigating the influence of culture on proxemic behaviors for humanoid robots," in *Proc. IEEE RO-MAN*, 2013, pp. 430–435.

[13] Z. Yu, X. He, A. W. Black, and A. I. Rudnicky, "User engagement study with virtual agents under different cultural contexts," in *Proc. Int. Conf. Int. Virtual Agents*, 2016, pp. 364–368.

[14] L. Yin and T. Bickmore, "Culturally-aware healthcare systems," in *Advances in Culturally-Aware Intel. Systems and in Cross-Cultural Psychological Studies*. Berlin, Germany: Springer, 2018, pp. 97–110.

[15] B. Endrass, I. Damian, P. Huber, M. Rehm, and E. André, "Generating culture-specific gestures for virtual agent dialogs," in *Proc. Int. Conf. Int. Virtual Agents*, 2010, pp. 329–335.

[16] A. A. Khaliq *et al.*, "Culturally aware planning and execution of robot actions," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2018, pp. 326–332.

[17] C. T. Recchiuto *et al.*, "Designing an experimental and a reference robot to test and evaluate the impact of cultural competence in socially assistive robotics," in *Proc. IEEE 28th Int. Conf. Robot Human Interactive Commun.*, pp. 1–8, 2019.

[18] C. Papadopoulos *et al.*, "The caresses study protocol: testing and evaluating culturally competent socially assistive robots among older adults residing in long term care homes through a controlled experimental trial," *Archives Public Health*, vol. 78, no. 1, pp. 1–10, 2020.

[19] B. Bruno *et al.*, "Knowledge representation for culturally competent personal robots: Requirements, design principles, implementation, and assessment," *Int. J. Social Robot.*, vol. 11, no. 3, pp. 515–538, 2019.

[20] B. Bruno, R. Menicatti, C. T. Recchiuto, E. Lagrue, A. K. Pandey, and A. Sgorbissa, "Culturally-competent human-robot verbal interaction," in *Proc. IEEE 15th Int. Conf. Ubiquitous Robots*, 2018, pp. 388–395.

[21] R. Menicatti *et al.*, "Collaborative development within a social robotic, multi-disciplinary effort: the caresses case study," in *Proc. IEEE Workshop Adv. Robot. Social Impacts*, 2018, pp. 117–124.

[22] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" in *Handbook on Ontologies*, Berlin, Germany: Springer, 2009, pp. 1–17.

[23] B. Motik *et al.*, "Owl 2 web ontology language: Structural specification and functional-style syntax," *W3C Recommendation*, vol. 27, no. 65, pp. 159–291, 2009.

[24] C. Recchuto *et al.*, "Cloud services for culture aware conversation: Socially assistive robots and virtual assistants," in *Proc. IEEE 17th Int. Conf. Ubiquitous Robots*, 2020, pp. 270–277.

[25] A. Sgorbissa, I. Papadopoulos, B. Bruno, C. Koulouglioti, and C. Recchiuto, "Encoding guidelines for a culturally competent robot for elderly care," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2018, pp. 1988–1995.

[26] N. Mavridis, "A review of verbal and non-verbal human–robot interactive communication," *Robot. Auton. Syst.*, vol. 63, pp. 22–35, 2015.