Culturally aware Planning and Execution of Robot Actions

Ali Abdul Khaliq*, Uwe Köckemann*, Federico Pecora*, Alessandro Saffiotti*,
Barbara Bruno†, Carmine Tommaso Recchiuto†, Antonio Sgorbissa†,
Ha-Duong Bui‡ and Nak Young Chong‡

Abstract— **The way in which humans behave, speak and interact is deeply influenced by their culture. For example, greeting is done differently in France, in Sweden or in Japan; and the average interpersonal distance changes from one cultural group to the other. In order to successfully coexist with humans, robots should also adapt their behavior to the culture, customs and manners of the persons they interact with. In this paper, we deal with an important ingredient of cultural adaptation: how to generate robot plans that respect given cultural preferences, and how to execute them in a way that is sensitive to those preferences. We present initial results in this direction in the context of the CARESSES project, a joint EU-Japan effort to build culturally competent assistive robots.**

I. Introduction

Designers of personal companion robots are often faced with questions such as: "How should the robot greet a person?", "What distance should the robot keep from a person?", "What non-verbal signs should the robot use, and with what intensity?", "Should the robot avoid or encourage physical contact?", "Is there any area of the house that it should consider off-limits?". Intuitively, the correct answer to all those questions depends on the user's values, beliefs, customs and lifestyle. Of course, one may reasonably expect that robot companions will learn such user preferences over time, via interaction with the user and/or explicit configuration mechanisms. In practice, however, many assumptions must be made at design time in order to provide the robot with an initial set of working skills. This calls into question the process by which these initial assumptions are made. This paper explores the possibility of using known cultural models as the basis for this process. In particular, we propose a method to make robot companions *culturally competent*.

One of the key technical challenges in building culturally competent robots is the need to easily and automatically reconfigure the behavior of the robot accordingly to different cultural profiles. In this paper we take a plan-based approach to robot control, where robot behaviors, the way actions are sequenced, and their parameters, are encoded in an abstract domain. The result is a *culture-aware planner*, in which the domain implicitly encodes various *sub-domains* for a variety of cultural contexts.

* A.A. Khaliq, U. Köckemann, F. Pecora and A. Saffiotti are with AASS, Örebro University, Fakultetsgatan 1, S-70182 Örebro, Sweden.

† B. Bruno, C.T. Recchiuto and A. Sgorbissa are with DIBRIS, University of Genova, Via Opera Pia 13, 16145 Genova, Italy.

‡ H-D. Bui and N.Y. Chong are with JAIST, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan. Corresponding author's email: ali.abdul.khaliq@oru.se

Our work is framed in the CARESSES project [1], a joint EU-Japan effort to design culturally competent elder care robots. These robots will be able to adapt how they behave and speak to the culture, customs and manners of the person they assist. A key tenet of CARESSES is the use of a cultural knowledge base to maintain all the culture-dependent information, and to use cultural knowledge to alter how a robot behaves in the presence of users who are known to be in a particular cultural group. The knowledge base also specifies how further knowledge of the user's preferences should be elicited through verbal and non-verbal interaction. The source of knowledge used by the robot to regulate its behaviors thus shifts over time, from the "default" culture-based model to a personalized user model reflecting individual preferences, thus avoiding cultural stereotypes.

The next section provides an overview of related work, while Section III describes the CARESSES project and its general architecture for cultural awareness. Sections IV and V detail our solutions to culturally aware planning and execution, respectively. Section VI illustrates these solution on three case studies, and Section VII draws some conclusions and discusses future work.

II. Related Work

Cultural competence is the capacity to take into consideration people's cultural beliefs, behaviours and needs while interacting with them [2]. The importance of cultural competence is well known in the fields of nursing [3] and of business [4], and it has been recognized in the literature on human factors [5] and human-computer interaction [6], [7]. Cultural competence, however, has been almost totally neglected by researchers and developers in the area of social robotics. While it is increasingly possible to build robots that reliably accomplish basic services, they only address the problem of "what to do" in order to provide a service. We argue that if service robots are to be accepted in the real world by real people, they must take into account the cultural identity of the persons whom they interact with in deciding "how" to provide their services.

Several studies support the hypothesis that people from different cultures not only have different preferences concerning how the robot should be and behave [8], [9], but they also tend to prefer robots better complying with the social norms of their own culture [10], both in the *verbal* and *non-verbal* behaviour.

In the context of verbal preferences, the effects of language and cultural context on the credibility of robot speech has been studied in [11]; work presented in [12] study the effect of politeness on the acceptance of robots in different cultures; the authors of [13] studied the effect of the cultural background on decision making when recommendations are provided by a robot collaborator.

The study of non-verbal preferences include the work presented in [14], that explored the proxemics in HRI and studied robot behavior when approaching humans in different cultures. In a similar vein, the interpersonal distance when a robot interacts which users of different cultural background has been studied in [15]. The work presented in [16] shows that users with different cultures interact more favorably with agents that have a similar cultural profile. Some other non-verbal studies compare the preference of different cultures for the appearance, design factor, capability, likeability, safety of the robot [17], [18], [19].

Despite these findings, very little work has been reported on how to make robots adapt easily to a given cultural identity. Notable exceptions include the works by Trovato et al. [20] and by Lugrin et al. [21], who propose frameworks for the selection of culturally appropriate greeting gestures and words; and the work by Bruno et al. [22], who uses fuzzy-logic to compute motion parameters depending on the cultural profile of the person. While these are worthy attempts, they are restricted to special cases and do no address the general problem of how to endow a robot with cultural competence.

III. The CARESSES Approach

In cultural adaptation, one can use a bottom-up or a top-down approach. In a bottom-up approach, such as [20] above, one focuses on adaptation at a personal level and then identifies cultural groups as clusters of people with similar cultural profiles. In a top-down approach, one encodes cultural information at national-level and then sees how this information influences preferences in the robot behaviours. Examples of the latter approach are [21] and [22] above, that use Hofstede's dimensions for the cultural categorization of countries [23]. Bottom-up and top-down approaches have complementary merits and drawbacks: the former makes cultural adaptation a demanding process which requires either a long time, or a large corpus of data to begin with; the latter makes adaptation faster, but it has the risk of incurring in prejudices and stereotypes.

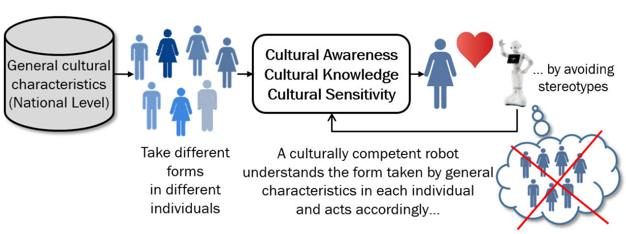

Fig. 1. The CARESSES concept of a culturally competent robot [1].

The CARESSES approach to design *culturally competent robots* combines the top-down and bottom-up approaches. When a robot interacts with a person for the first time, it uses a top-down approach to bootstrap its behaviour using a cultural profile based on their cultural group; over the course of time, the robot uses a bottom-up approach to refine this cultural profile based on the individual preferences expressed by that person. Figure 1 illustrates this concept. A culturally competent robot: (i) knows general cultural characteristics that are shared by a group of people; (ii) is aware that general characteristics take different forms in different individuals, thus avoiding stereotypes; and (iii) is sensitive to cultural differences when perceiving, reasoning, and acting. More specifically, Bruno et al. [24] have identified the following key capabilities for a robot to exhibit a culturally competent behaviour:

- *cultural knowledge representation*: the capability of storing cultural knowledge and reasoning upon it, properly managing general and user-specific information.
- *culturally-sensitive planning and execution*: the capability to produce plans and adapt such plans depending on the cultural identity of the user.
- *culture-aware multi-modal human-robot interaction*: the capability of adapting the interaction (in terms of gestures, tone of voice, etc.) to the user's cultural identity.
- *culture-aware human emotion and action recognition*: the capability to interpret sensor data acquired by the robot during interaction in light of cultural knowledge.
- *assessment of cultural identity, habits and preferences*: the capability to adapt general cultural knowledge and acquire new knowledge to better fit the individual profile of the user.

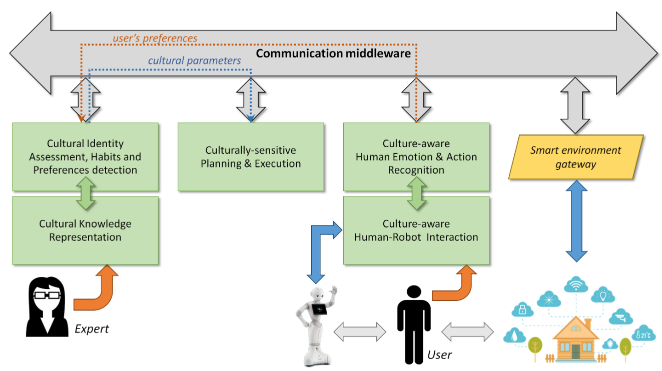

Fig. 2. A coarse view of the CARESSES functional architecture.

The above capabilities are integrated in CARESSES through the functional architecture sketched in Figure 2. The *Cultural Knowledge Base* (CKB) encodes quantitative and qualitative knowledge about different cultural groups, as well as the specific cultural identity of the subjects with whom the robot must interact. The *Culturally-sensitive Planning & Execution Module* is responsible for generating and executing robot's actions. The *Culture-aware Human-Robot Interaction Module* is responsible for implementing the motion and interaction behaviors of the robot, and for interfacing with a smart environment, if present. All the components are integrated in universAAL [25], a software platform for open distributed systems of systems. The interface to the smart environment is provided by a bridge between universAAL and ECHONET [26], the Japanese standard for home automation. In this paper we focus on *culturally-sensitive planning and execution*.

IV. Cultural Sensitivity in Action Planning

We have integrated an online Constraint-based Planner [27] into the CARESSES architecture and connected it to the *Cultural Knowledge Base (CKB)*. The core of the CKB is an ontology that encodes all elements that may play a key role in socially assistive robotic scenarios: goals, actions, cultural norms, the environment, topics of conversation. The CKB also includes information about how the cultural context influences the operators available to the planner and the parameters of actions that are executed by the robot. This strategy decouples all cultural information from the planner and the robot's executive, thus avoiding redundancy across different modules in the CARESSES architecture. When launched, the planner requests operators and actions from the CKB. During execution it listens for new goals, updates on the execution status of actions, and messages about the state of the environment and people in the environment.

The planner is modular and considers a variety of constraint types. It solves problems by searching a combined flaw-resolution search space for each type. A set of constraints of a given type may lead to a *flaw* that can be resolved in different ways by adding *resolvers* (i.e., sets of constraints). Consider a reusable resource and several statements that use it. If the temporal intervals of these statements can overlap and together their usage exceeds the resource capacity we have a resource flaw. This flaw must be addressed by adding temporal constraints as a resolver to make the overlap of statements impossible [28]. In a similar way, an open goal is seen as a constraint that constitutes a flaw that can be resolved by adding a set of operators (a plan) that achieves that goal. This view on solving problems is convenient because it allows to easily integrate a wide variety of constraint types with minimal overhead (see [27] for details on a broader selection of constraints).

We refer a collection constraints of different types as a *Constraint Database (CDB)*. In this paper we use several constraint types. *Domain constraints* specify a variable's type and each type's domains. *Statements* assign values to state-variables over temporal intervals. They allow to model events or states over periods of time. Thus, a single CDB may contain information about the past, present, and future values of a state variable. This allows, for instance, to easily delay action dispatching in case a previous action is not finished yet. These timing issues are expressed by *temporal constraints* that specify how the intervals of statements relate to each other. We express temporal constraints via quantified Allen's Interval Algebra [29], [30]. *Goals* are statements that need to be achieved by *operators*.

Operators consists of a name, a set of preconditions and effects (both statements), as well as a set of constraints (of any type). Usually, an operator contains at least a set of temporal constraints to relate itself to its preconditions and effects. Often, we add further constraints to limit the scope of situations in which the operator can be applied or to model more complicated effects. Consider the following (abridged) example of an operator used in this paper.

```
(:operator (approach-user ?R)
  (:signature t_robot)
  (:preconditions (?P approached-user false) )
  (:effects (?E approached-user true))
  (?A {?ID (type ApproachUser) (robot ?R) (
      apar ?Distance)})
  (:constraints
    (:temporal
      (before ?P ?THIS [1 inf])
      (before ?THIS ?E [1 inf])
      (equals ?THIS ?A)
```

The name of the operator is *approach-user* and it uses a single variable $?R$ (indicated by the question mark) that is of type t_robot. Its precondition states that the robot has not approached the person during the time interval $?P$. Its effects model that the robot has approached the person during interval $?E$, as well as an action $?A$ that will be sent to the robot to achieve this effect. Note that we consider only one person in this example (i.e., the person is not an argument of the operator) and that the actions we execute are statements and not operators directly. Thus we can model several actions as part of the same operator. The first and second parameter of the action are its own ID and the name of the robot that will be used to execute it. The action has the robot approach the user until a distance $?Distance$ is reached. The variable $?THIS$ indicates the time interval associated to the operator itself. The set of temporal constraints state that the effect interval $?P$ ends before the interval $?THIS$ starts, and the interval $?THIS$ ends before the interval $?E$ starts. The distance between the respective end and start times of intervals is flexible, and it may range between 1 and ∞. Finally, we consider that the operator is being executed as long as the corresponding $ApproachUser$ action is not finished, expressed through the *equals* constraint.

The action parameters, like the approaching distance in the previous example, are decided and provided by the CKB depending on the current cultural context. Temporal constraints model the fact that the robot should connect the timing of preconditions and effects. The operator itself (which is enacted during temporal interval $?THIS$) finishes before the effect. The temporal constraints are kept as flexible as possible to keep the temporal constraint network robust towards delays during execution. The planner works online and dispatches actions for execution when the earliest start time of their intervals is reached. It also periodically tests for new flaws, for instance, to compensate for failures or delays in execution, or to satisfy new goals.

The planner itself does not employ any *ad-hoc* mechanisms for cultural sensitivity. Instead, it dynamically assembles and uses culture-specific planning domains based on information provided by the CKB. This information includes which operators are allowed in a cultural context, and the parameters of actions that are sent to the executive. This is in line with the design goal of CARESSES to keep all cultural knowledge in the CKB.

We emphasize that the cultural context may affect not only the choice of the execution parameters, but also of the planning operators. For example, in many Asiatic cultures it is considered rude to hand-over an object using only one hand, so the corresponding hand-over planning operator uses two hands. This knowledge must be encoded as an operator rather than as an execution parameter, since the planner needs to know that both hands are used in this cultural context: for instance, it should not generate a plan where a hand-over occurs while the robot carries another object.

Our planner can be easily expanded by incorporating Interaction Constraints for human-aware planning [31] that depend dynamically on the cultural context. Interaction constraints allow to hand-craft flaws and resolvers using any type of constraint. This can be used to avoid or forbid complex situations expressed via any combination of constraint types available to the planner [31] or to dynamically infer context and create new goals in specific situations [32]. We can impose, for instance, that the robot does not vacuum in a room in which a person is currently reading. If we observe a person executing a task for which the robot has the capability to assist, we can dynamically add new goals to realize this assistance. In CARESSES, these rules may depend on the cultural context and any personal preferences that were gathered to expand on or overwrite that knowledge. Conveniently, this can work exactly in the same way as for operators and actions above: the relevant set is selected and parametrized by the CKB which dynamically creates problems for the planner to solve. This expansion is part of our current work, but it has not been included in the experiments reported below.

V. CULTURAL SENSITIVITY IN ACTION EXECUTION

The above planner dispatches the actions that are required to be executed on the robot and receives status updates from the robot. This communication is done through an "executor" module, which acts as a front-end between the planner and the robot's sensing and actuation functionalities. The executor consists of three main processes: (1) receiving action execution requests; (2) executing actions; (3) sending feedback. At start, the executor waits for incoming messages. When it receives a message from the the planner that contains an action with its parameters, it decides which low level control commands should be sent to the robot to execute the action request. The executor sends two feedback messages to the planner about the state of action execution: one when the action has started, and one when the action has finished execution. Moreover, whenever the executor acquires information about user's preferences or habits, it sends a message to the CKB for updating the knowledge base.

We use the universAAL middleware (uAAL, for short) for communication. This middleware uses two busses (Context Bus and Service Bus) to transmit and receive specific types of messages. Figure 3 shows a simplified view of the software architecture used in our implementation, including the uAAL infrastructure, the planner, the executor, and the robot. The planner uAAL component and the executor uAAL component use the uAAL bus to publish and subscribe different types of messages. The planner and the executor use sockets to interact with the uAAL components. The planner end of the interface has two modules: a dispatching agent, and a situation awareness agent. The former looks at action statements in the CDB and decides when it is time to dispatch an action to the robot for execution using a so-called context publisher. This is a software module that receives information about the current state of the robot, users, and environment using a context subscriber. The robot end of the interface consists of robot control module that receives action requests from the planner using the context subscriber and realizes them by sending the appropriate command to the robot.

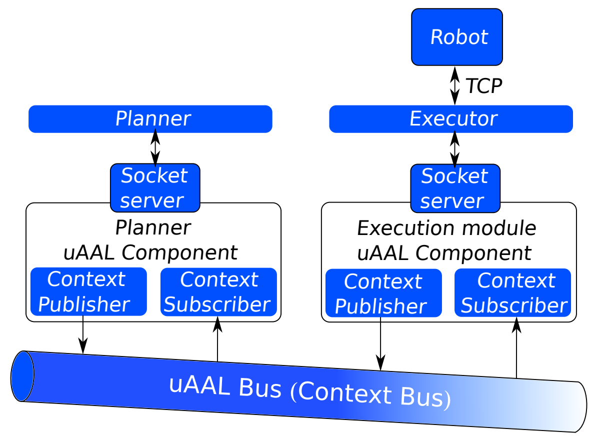

Fig. 3. Illustration of the software architecture, including the uAAL infrastructure, the planner, the executor, a robot.

Just like in the case of the planner, cultural information is not directly encoded in the executor. Instead, the culturally sensitive parameters to be used for actions are sent from the CKB to the planner, and the latter sends them to the executor along with the planned actions. This design choice increases modularity, since all cultural knowledge is created, maintained and updated in one single place.

VI. DEMONSTRATION CASE STUDIES

In this section we look at three cases of culturally aware planning and execution. These cases will be run for three different users, John, Kabir, and Kenji, who belong to three different cultural groups, respectively, English, Indian and Japanese. The preferences of each group with respect to the operators and actions that we consider in this paper are based on the guidelines complied by experts in trans-cultural nursing within the CARESSES project [33] starting from existing guidelines for culturally competent nursing care. Each case was executed on a Pepper robot [34]. Figure 6 shows examples of execution for the three individuals.[1].

Figure 4 shows the overall flow of the examples demonstrated in this section. It includes three main operators: (1)

[1] A video showing a real execution of these case studies is available at http://caressesrobot.org/en/2018/03/20/caresses-culturally-aware-planner-in-action/

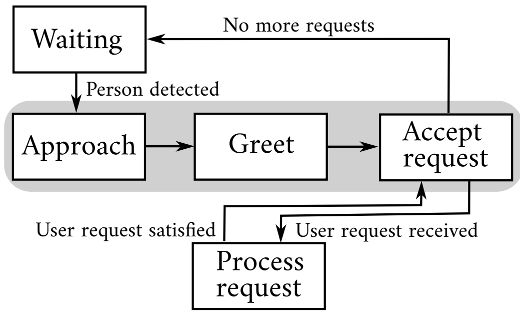

Fig. 4. Overall flow of the demonstrated case studies. Inside the gray box are the three actions generated by the planner.

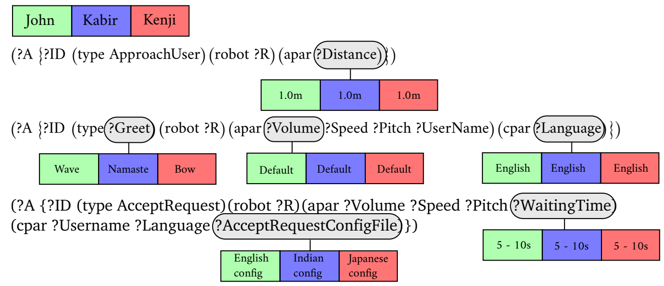

Fig. 5. Actions generated by the planner. Gray: names of the culturally specific parameters. Green: values of the culturally specific parameters for John. Blue: values of the culturally specific parameters for Kabir. Red: values of the culturally specific parameters for Kenji.

the *Approach* operator contains an action to approach the user; (2) the *Greet* operator is used to greet the user; and (3) the *Accept request* operator accepts the user requests and processes them. Initially, the robot is in a waiting state. Whenever the robot detects a user, it approaches them, greets them, and starts receiving the user requests. This may involve a dialogue to acquire details about what the user wants. The robot goes back to the Accept request state after processing the user request. If no more requests are made, the robot goes to the waiting state.

The planner generates three actions using the above mentioned operators. These actions are sent to the executor for execution on the robot. The action are depicted in Figure 5 together with their parameters, and should be executed according to the preferences of each user. In this example, these are the default preferences of each cultural group, as the robot has never interacted with the users before. The *AcceptRequest* has a "config file" as one of its parameters: this makes the robot suggest possible requests according to a given probability distribution. These are only suggestions and do not limit the set of requests that the user can make. Note that some parameters do not vary across the three cultural groups, reflecting the fact these aspects of behavior are not known to vary across cultures [33]. This is also an indication that the robot should acquire knowledge about user preferences regarding these parameters. Preferences are encoded in the choice of operators and parameters provided by the CKB, and thus do not increase the complexity of planning. We now show see how these preferences influence the behaviour of the robot in accordance to the relevant cultural group.

A. John (English group)

The robot is provided with knowledge of John's cultural background — highlighed in green in Figure 5. First the robot executes the *ApproachUser* action where the cultural parameter, *Distance*, is the minimum proximity distance between the robot and John. After the completion of this action, the robot executes the *Greet* action. Since the cultural background of John in known, this action is executed with Wave. The robot waves at John and greets him while keeping the cultural parameter *Volume* at default level. The *Language* parameter is set to English.[2] After the completion of the *Greet* action, the robot executes the *AcceptRequest* action, suggesting to the user possible requests in relation to the content of the English config file: for example, it will probably ask John if he wants to set a reminder for feeding the cat. In relation to the user's inputs, the robot will execute his requests or it will engage John in a conversation using probabilities on conversation topics coming from the CKB for asking questions that are specific to John's English background. For example, if John says something about breakfast, the robot may ask the question "Do you usually have typical English breakfast?". The outcomes of this conversation may in turn be used to update the cultural profile of John in the CKB. The parameter *WaitingTime* regulates how long the robot should wait to get an answer from John, and ends the action if no reply is received: this parameter might, once again, be culturally dependent.

B. Kabir (India group)

The robot is provided with knowledge related to Kabir's default cultural background, namely, the Indian group — highlighed in blue in Figure 5. The robot executes the *ApproachUser* action with parameter *Distance*. The *Greet* action is executed by performing the Nameste gesture. The parameter *Volume* is set to Default level and the *Language* is set to English. The *AcceptRequest* is executed and, among the options, the robot will probably suggest that Kabir sets a reminder for visiting the temple. As before, conversation topics and probabilities stored in the CKB allows the robot to engage in a culturally-specific conversation. For example, Kabir may say "I am waiting for the Diwali festival". The robot then continues talking about the Diwali festival. As done for John, the robot waits before ending the action using the *WaitingTime* parameter.

C. Kenji (Japanese group)

Similar to the above two cultural groups, the robot knows that Kenji is Japanese, and hence uses that set of culture-specific knowledge — highlighted in red in Figure 5. The

[2]For the sake of understandability, the Language parameter was set to English for all cultural groups.

(a) (b) (c)

Fig. 6. Snapshots of the demonstrations. (a) person acting as John; (b) person acting as Kabir; (c) person action as Kenji.

robot executes the action *ApproachUser* with cultural parameter *Distance*, followed by the *Greet* action involving the Bow gesture. The parameters *Volume* and *Language* are set to Default and English respectively. The action *AcceptRequest* is executed by the robot with the Japanese config file, having suggestions such as watching the sumo wrestling match on TV. Given the probabilities of conversation topics for the Japanese culture, the robot will likely asks questions such as "Do you usually have Japanese food for breakfast?" if Kenji shows interest in chitchatting about breakfast. Finally, the parameter *WatitingTime* is set to 5–10 seconds.

VII. Discussion and Conclusions

To the best of our knowledge, CARESSES is the first attempt to systematically address the automatic adaptation of robot systems to different cultures and preferences. We believe that cultural competence is a necessary, although so far understudied ingredient for any social, personal or assistive robot.

In this paper, we have shown how we can achieve cultural sensitivity in planning and execution by obtaining knowledge about preferences from a cultural knowledge base. The main advantage in this approach is that knowledge about culture does not need to be directly encoded into the planner or the execution layer. This approach decouples the cultural information from the planner and the execution layer, which avoids redundancy across different modules in the CARESSES architecture. The planner receives cultural dependent goals and the CKB suggest different parameters depending on the cultural profile of the person. Beside increasing modularity, this strategy allows the possibility to use different, off-the-shelf planners with minimal modifications.

From a commercial perspective, cultural customization is crucial in overcoming the barriers to marketing robots across different countries. The parameters related to cultural preferences in different countries can be loaded by the manufacturer in the CKB, and can be later adapted in interaction with the user to avoid stereotyped representations. As the parameters are updated, they can also be used to change the person-specific area of the CKB (but not the culture-specific: what we learn from the interaction is not used to make general inferences on a culture, only about the individual person).

The cultural knowledge used in the cases we shown in this paper reflects those differences between cultures that are deemed sufficiently meaningful (e.g., greeting gestures), while it purposefully ignores properties that are likely to be more influenced by personal preference than culture. As part of the CARESSES project, we will model more extensive cultural knowledge and we will validate it with real elderly users in a wide experimental campaign. It is important to note that the provided cultural knowledge, although compiled by experts, is only used to give an initial set of skills to the robot. Moreover, the mechanism we have described is agnostic with respect to the knowledge contained in the CKB. Our contribution here is to show the technical solution underlying cultural awareness, and not to validate the cultural models themselves.

On the technical side, we plan to increase the sophistication of the planner by introducing culturally dependent interaction constraints [31]. These can be used to encode cultural norms to express, for instance, that in some cultures it is not acceptable for the robot to interrupt an ongoing conversation, while in others it might be tolerable for urgent matters. Another future research direction involves enhancing the monitoring of plan execution, i.e., enriching the feedback related to the status of the action execution from the Executor side as well as replanning on failures from the planner side.

Acknowledgements

This work has been supported by the European Commission Horizon2020 Research and Innovation Programme under grant agreement No. 737858, and from the Ministry of Internal Affairs and Communication of Japan. We thank all the members of the CARESSES consortium, listed in the project homepage [1].

References

[1] The CARESSES Consortium, "Caresses: Culture-aware robots and environmental sensor systems for elderly support," Project home page: http://www.caressesrobot.org, last accessed on July 28, 2018.

[2] I. Papadopoulos, *Transcultural health and social care: development of culturally competent practitioners*. Elsevier Health Sciences, 2006.

[3] M. Leininger and M. Mcfarland, *Transcultural nursing: concepts, theories, research and practice. 3rd ed.* New York, NY: McGraw-Hill, 2002.

[4] F. Trompenaars and C. Hampden-Turner, *Riding the waves of culture: Understanding diversity in global business*. Nicholas Brealey International, 2011.

[5] T. Plocher, P.-L. P. Rau, and Y.-Y. Choong, "Cross-cultural design," in *Handbook of human factors and ergonomics*, G. Salvendy, Ed. John Wiley & Sons, 2012, ch. 6.

[6] T. Clemmensen and K. Roese, "An overview of a decade of journal publications about culture and human-computer interaction (HCI)," in *Human Work Interaction Design: Usability in Social, Cultural and Organizational Contexts. Second IFIP WG 13.6 Conference*, D. Katre, R. Orngreen, P. Yammiyavar, and T. Clemmensen, Eds. Springer, 2010, pp. 98–112.

[7] R. Heimgärtner, "Reflections on a model of culturally influenced human–computer interaction to cover cultural contexts in HCI design," *International Journal of Human-Computer Interaction*, vol. 29, no. 4, pp. 205–219, 2013.

[8] V. Evers, H. Maldonado, T. Brodecki, and P. Hinds, "Relational vs. group self-construal: untangling the role of national culture in HRI," in *HRI 2008*, 2008, pp. 255–262.

[9] T. Nomura, "Cultural differences in social acceptance of robots," in *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*. IEEE, 2017, pp. 534–538.

[10] G. Trovato, J. R. Ham, K. Hashimoto, H. Ishii, and A. Takanishi, "Investigating the effect of relative cultural distance on the acceptance of robots," in *International Conference on Social Robotics*. Springer, 2015, pp. 664–673.

[11] S. Andrist, M. Ziadee, H. Boukaram, B. Mutlu, and M. Sakr, "Effects of culture on the credibility of robot speech: A comparison between English and Arabic," in *HRI 2015*, 2015, pp. 157–164.

[12] M. Salem, M. Ziadee, and M. Sakr, "Marhaba, how may i help you?: Effects of politeness and culture on robot acceptance and anthropomorphization," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, 2014, pp. 74–81.

[13] L. Wang, P.-L. P. Rau, V. Evers, B. K. Robinson, and P. Hinds, "When in Rome: the role of culture & context in adherence to robot recommendations," in *HRI 2010*, 2010, pp. 359–366.

[14] M. P. Joosse, R. W. Poppe, M. Lohse, and V. Evers, "Cultural differences in how an engagement-seeking robot should approach a group of people," in *CABS 2014*, 2014, pp. 121–130.

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

[16] S. Mascarenhas, N. Degens, A. Paiva, R. Prada, G. J. Hofstede, A. Beulens, and R. Aylett, "Modeling culture in intelligent virtual agents," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 5, pp. 931–962, 2016.

[17] C. Bartneck, "Who like androids more: Japanese or US Americans?" in *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, 2008, pp. 553–557.

[18] K. S. Haring, D. Silvera-Tawil, Y. Matsumoto, M. Velonaki, and K. Watanabe, "Perception of an android robot in Japan and Australia: A cross-cultural comparison," in *Social Robotics*. Springer International Publishing, 2014, pp. 166–175.

[19] H. R. Lee and S. Sabanović, "Culturally variable preferences for robot design and use in South Korea, Turkey, and the United States," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, 2014, pp. 17–24.

[20] G. Trovato, M. Zecca, M. Do, Ö. Terlemez, M. Kuramochi, A. Waibel, T. Asfour, and A. Takanishi, "A novel greeting selection system for a culture-adaptive humanoid robot," *Int J Adv Robot Syst*, vol. 12, 2015.

[21] B. Lugrin, J. Frommel, and E. André, "Modeling and evaluating a bayesian network of culture-dependent behaviors," in *Culture Computing 2015*, 2015, pp. 33–40.

[22] B. Bruno, F. Mastrogiovanni, F. Pecora, A. Sgorbissa, and A. Saffiotti, "A framework for culture-aware robots based on fuzzy logic," in *IEEE Int Conf on Fuzzy Systems*, 2017, to appear.

[23] G. Hofstede, G. J. Hofstede, and M. Minkov, *Cultures and organizations: Software of the mind*. Citeseer, 1991, vol. 2.

[24] B. Bruno, N. Y. Chong, H. Kamide, S. Kanoria, J. Lee, Y. Lim, A. K. Pandey, C. Papadopoulos, I. Papadopoulos, F. Pecora, A. Saffiotti, and A. Sgorbissa, "Paving the way for culturally competent robots: A position paper," in *Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017, pp. 553–560.

[25] E. Ferro, M. Girolami, D. Salvi, C. Mayer, J. Gorman, A. Grguric, R. Ram, R. Sadat, K. M. Giannoutakis, and C. Stocklow, "The universAAL platform for AAL (ambient assisted living)," *J Intell Syst*, vol. 24, no. 3, pp. 301–319, 2015.

[26] S. Matsumoto, "Echonet: A home network standard," *IEEE Pervasive Computing*, vol. 9, no. 3, pp. 88–92, 2010.

[27] U. Köckemann, "Constraint-based methods for human-aware planning," Ph.D. dissertation, Örebro university, 2016.

[28] A. Cesta, A. Oddi, and S. Smith, "A constraint-based method for project scheduling with time windows," *Journal of Heuristics*, vol. 8, pp. 109–136, 2002.

[29] J. Allen, "Towards a general theory of action and time," *Artificial Intelligence*, vol. 23, pp. 123–154, 1984.

[30] I. Meiri, "Combining qualitative and quantitative constraints in temporal reasoning," *Artificial Intelligence*, vol. 87, pp. 260–267, 1996.

[31] U. Köckemann, F. Pecora, and L. Karlsson, "Grandpa hates robots — interaction constraints for planning in inhabited environments," in *Proc, of the 28th Conf. on Artifical Intelligence (AAAI 2014)*, 2014.

[32] ——, "Inferring context and goals for online human-aware planning," in *IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, 2015.

[33] I. Papadopoulos, C. Koulouglioti, H. Kamide, A. Sgorbissa, L. Merton, and N. Y. Chong, "Basic guidelines for a culturally competent robot," CARESSES public project deliverable D1.2, Tech. Rep., 2017.

[34] A. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: Pepper: The first machine of its kind," *IEEE Robotics Automation Magazine*, pp. 2–10, September 2018.