Behavior Trees for Culturally Sensitive Social Robots: African Culture Case Study

Ibrahim Olaide Jimoh, Heran Semere Equbay, Clifford Onyonka, Tsegazeab Tefferi, David Vernon
Carnegie Mellon University Africa
Kigali, Rwanda
ioj, hequbay, cliffor2, ttefferi, dvernon@andrew.cmu.edu

***Abstract*—In this paper, we explore the use of behavior trees to control the behavior of culturally sensitivity social robots. Behavior trees offer a structured and modular approach to managing complex robot behaviors, enabling dynamic adaptation to different cultural contexts. We present a framework for incorporating cultural sensitivity into behavior trees, detailing the process of identifying cultural norms, modeling behaviors, and implementing dynamic adaptation mechanisms. Our approach is demonstrated through a case study in a multicultural university located in an African country setting, where the robot's interactions are tailored to respect and respond appropriately to the cultural norm of the Rwandan culture.**

***Index Terms*—Behavior tree, social robotics, cultural sensitivity**

I. Introduction

As social robots become more prevalent in various societal roles, from personal assistants to healthcare, their ability to interact respectfully and efficiently with individuals from diverse cultural backgrounds has become crucial [1]. Cultural sensitivity in social robots interaction is essential for fostering trust, ensuring respectful interactions, and enhancing user experiences. However, developing robots that can dynamically adapt to different cultural norms presents significant challenges [2].

Behavior trees, a decision-making framework which has recently been used extensively in robotics and artificial intelligence, offer a promising solution to this challenge. By organizing robot behaviors in a hierarchical, modular, and flexible structure, behavior trees facilitate the creation of complex, adaptable behavior models [3]. This paper investigates the application of behavior trees to the cultural sensitivity of social robots.

II. Cultural Sensitivity

Cultural sensitivity is critical in social robotics for fostering positive user experiences and acceptance. When robots demonstrate an understanding and respect for cultural norms and values, they are more likely to be accepted and trusted by users [4]. This acceptance is essential for the effective deployment and integration of social robots into everyday life. Furthermore, cultural sensitivity helps to build trust and comfort between humans and robots, which is particularly important in settings like healthcare, where patients may need to rely on robots for assistance and companionship [5]. From an ethical standpoint, ensuring that robots are culturally sensitive is part of broader efforts to respect human diversity and promote inclusivity [6]. By being culturally aware, robots can contribute positively to social dynamics and uphold ethical standards in their interactions.

Current approaches to social robotics involve pre-programmed behaviors that do not dynamically adapt to the cultural context of interaction, or often relying on broad generalizations that can lead to stereotyping and oversimplification. Other approaches are limited by the availability and comprehensiveness of cultural databases [7]. Developing more advanced frameworks for cultural sensitivity is crucial, and behavior trees offer a promising approach due to their modularity and flexibility. By developing a comprehensive and up-to-date cultural knowledge database and integrating these cultural norms into behavior trees and enabling dynamic adaptation, robots can achieve a higher level of cultural competence [7] [8].

III. Framework

Our proposed framework leverages behavior trees to integrate cultural norms into robot behaviors, allowing robots to dynamically adapt their actions based on cultural contexts. This framework consists of several key components: cultural norm identification, behavior tree modeling, dynamic adaptation mechanisms, and Robot Operating System (ROS) integration.

A. Cultural Norm Identification

Cultural knowledge base is currently under development, and is the subject of a separate article [submitted to the IROS 2024 Robotics in Africa Forum].

B. Behavior Tree Modelling

The behavior tree consists of nodes arranged in hierarchical tree structure with key components including root nodes, control flow nodes (sequence, selector, and decorator), conditional nodes, and action nodes [9]. Root node is the starting point of the behavior tree. The execution begins here and traverses through the tree. The control flow nodes determine the flow of execution. Selectors (fallbacks) which execute

The work described in this article was carried out in the Culturally Sensitive Social Robotics for Africa (CSSR4Africa) project as part of the Afretec Network. Afretec is managed by Carnegie Mellon University Africa and receives financial support from the Mastercard Foundation.

child nodes from left to right, returning success when one child succeeds. It is used for ”try until success” behaviors. Sequences execute child nodes from left to right, returning success only if all children succeed. It is used for ”do all tasks in order” behaviors. Decorators modify the behavior of their child nodes (e.g., repeat, invert, limit execution).

Leaf nodes are the actionable and evaluative endpoints of the tree and can be action nodes which perform specific tasks (e.g., move to a location, pick up an object) or condition nodes that check certain conditions or states (e.g., battery level, obstacle detection).

The behavior tree was designed with Groot2, a compatible IDE with BehaviorTree.CPP library for creating and editing behavior trees and converting the tree into XML compatible with the BehaviorTree.CPP library [9].

C. Dynamic Adaptation Mechanisms

Sensors, cameras, and microphones help us to gather real-time data about the environment and user interactions. Dynamic selectors and sequences allow choosing actions based on the current cultural context. For instance, selecting an appropriate and respectful verbal greeting or gesture during interactions.

D. ROS Integration

ROS is a flexible framework for writing robot software which provides a collection of tools, libraries, and conventions designed to simplify the task of creating complex and robust robot behavior [10]. ROS provides a robust communication infrastructure, enabling smooth data exchange between the behavior tree nodes and robot subsystems. We integrate each behavior tree action node to a ROS action (either subscribing to a topic and/or calling a service) [11]. Each action node has a callback function implemented in C++ that calls on a ROS action that implements the appropriate task associated with that action node. The BehaviorTree.CPP library is utilized because it contains the necessary framework for executing behavior tree nodes [9] [12] [13] [14].

IV. Case Study

In this case study, we explore a multicultural university environment where the behavior tree is designed to accommodate the cultural norms of Rwandan culture, reflecting the local context of the university. We examine two scenarios: a lab tour guide and a receptionist robot agent. For both scenarios, we utilize the Pepper humanoid robot as the robotic agent.

The lab tour guide robot uses its attention system to gauge visitor interest and tailor its interactions accordingly. It navigates through areas of interest, provides educational content, and adjusts its behavior based on visitor engagement. In the receptionist scenario, the receptionist robot manages visitor interactions with a focus on directing visitors to the room or office they want to visit.

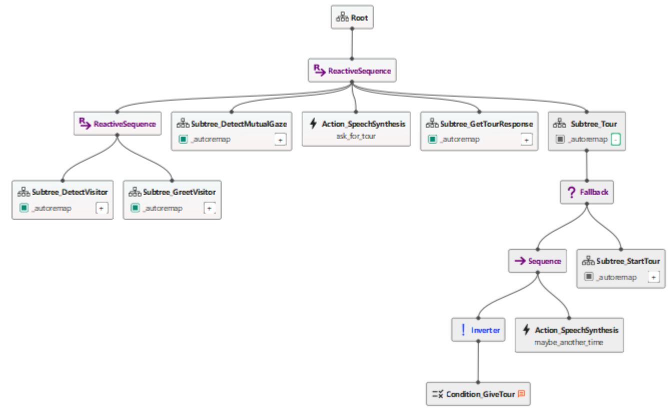

Fig. 1. Groot2 visualization of designed behavior tree for the lab tour guide scenario.

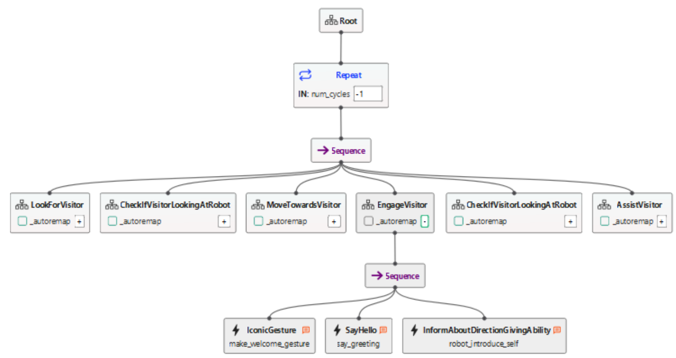

Fig. 2. Groot2 visualization of designed behavior tree for the receptionist scenario.

V. Preliminary Result

No tests have been conducted as yet; however, the planned tests in simulated environments are expected to demonstrate the adaptability and effectiveness of the robots in managing interactive tasks. The logical design of the system, integrating attention systems, speech recognition and synthesis, and gesture systems, suggests that these robots will perform effectively. The integration of a cultural knowledge base is anticipated to be pivotal in refining the robots' responses and interactions, ensuring that they are not only technically proficient but also culturally attuned. These tests, once conducted, will provide critical insights into the practical implications of our theoretical design and help identify areas for further refinement.

VI. Discussion

While actual tests are yet to be conducted, it is anticipated based on the system's design that behavior trees will be highly effective in managing complex, culturally nuanced interactions in robotic systems. The theoretical framework suggests a strong potential for success; however, anticipated challenges include optimizing module integration and ensuring seamless real-time response to human inputs. These challenges highlight

```
▼<root BTCPP_format="4" main_tree_to_execute="TourGuide">
  ▼<BehaviorTree ID="Subtree_DetectMutualGaze">
    ▼<RetryUntilSuccessful num_attempts="3">
      ▼<ReactiveSequence>
          <Action_LookAtVisitor/>
          <Action_DetectMutualGaze/>
        </ReactiveSequence>
      </RetryUntilSuccessful>
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_DetectVisitor">
    ...
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_EndTour">
    ...
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_GetTourResponse">
    ...
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_GetTourResponseFromSpeech">
    ...
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_GetTourResponseFromTablet">
    ...
    </BehaviorTree>
  ▼<BehaviorTree ID="Subtree_GreetVisitor">
    ▼<Sequence>
        <Action_IconicGesture name="welcome_gesture"/>
        <Action_SpeechSynthesis name="welcome_greeting"/>
        <Action_SpeechSynthesis name="introduction"/>
      </Sequence>
    </BehaviorTree>
  ▶<BehaviorTree ID="Subtree_StartTour">
    ...
    </BehaviorTree>
  ▼<BehaviorTree ID="Subtree_Tour">
    ▼<Fallback>
      ▼<Sequence>
        ▼<Inverter>
            <Condition_GiveTour _description="Node to be implemented.
          </Inverter>
          <Action_SpeechSynthesis name="maybe_another_time"/>
        </Sequence>
        <SubTree ID="Subtree_StartTour" _autoremap="true"/>
      </Fallback>
    </BehaviorTree>
  ▼<BehaviorTree ID="TourGuide">
    ▼<ReactiveSequence>
      ▼<ReactiveSequence>
          <SubTree ID="Subtree_DetectVisitor" _autoremap="true"/>
          <SubTree ID="Subtree_GreetVisitor" _autoremap="true"/>
        </ReactiveSequence>
        <SubTree ID="Subtree_DetectMutualGaze" _autoremap="true"/>
        <Action_SpeechSynthesis name="ask_for_tour"/>
        <SubTree ID="Subtree_GetTourResponse" _autoremap="true"/>
        <SubTree ID="Subtree_Tour" _autoremap="true"/>
      </ReactiveSequence>
    </BehaviorTree>
```

Fig. 3. XML representation of behavior tree for the lab tour guide scenario.

the need for thorough testing and refinement to achieve the desired operational fluency in dynamic environments.

VII. Future Work

Our initial implementation includes the development of general-purpose tour guide and receptionist robot agents. While the development of a cultural knowledge base is currently under progress, the full integration of this cultural knowledge base within the behavior tree framework is still in progress. Future developments will concentrate on embedding these cultural norms more deeply into the decision-making processes of these behavior trees. This enhancement aims to ensure that the robots can dynamically adjust their behaviors based on the cultural context of Rwanda as case study and

```
▼<root BTCPP_format="4" main_tree_to_execute="ReceptionScenario">
  ▼<BehaviorTree ID="AssistVisitor">
    ▼<Fallback>
      ▼<Sequence>
        ▼<Fallback>
          ▶<Sequence>
            ...
            </Sequence>
          ▶<Sequence>
            ...
            </Sequence>
          </Fallback>
        ▶<Sequence>
          ...
          </Sequence>
        </Sequence>
        <NoVisitorResponse name="acknowledge_no_response_from_visitor"
      </Fallback>
    </BehaviorTree>
  ▶<BehaviorTree ID="CheckIfVisitorLookingAtRobot">
    ...
    </BehaviorTree>
  ▼<BehaviorTree ID="EngageVisitor">
    ▼<Sequence>
        <IconicGesture name="make_welcome_gesture" _description="Make
        <SayHello name="say_greeting" _description="Say "Hello, I'm Pe
        <InformAboutDirectionGivingAbility name="robot_introduce_self"
      </Sequence>
    </BehaviorTree>
  ▶<BehaviorTree ID="LookForVisitor">
    ...
    </BehaviorTree>
  ▼<BehaviorTree ID="MoveTowardsVisitor">
    ▼<Inverter>
      ▼<KeepRunningUntilFailure>
        ▼<Inverter>
          ▼<Fallback>
              <PersonDistanceEstimation name="check_if_robot_is_close_
            ▼<ForceFailure>
              ▼<Sequence>
                  <BodyRotation name="rotate_body_to_face_visitor" _des
                  <Locomotion name="move_towards_visitor" _description=
                </Sequence>
              </ForceFailure>
            </Fallback>
          </Inverter>
        </KeepRunningUntilFailure>
      </Inverter>
    </BehaviorTree>
  ▶<BehaviorTree ID="ReceptionScenario">
    ...
    </BehaviorTree>
```

Fig. 4. XML representation of behavior tree for the receptionist scenario.

user interactions, allowing for more nuanced and appropriate responses.

VIII. Conclusion

This project illustrates the potential of behavior trees in developing culturally sensitive robot agents. The ongoing development and future testing will further contribute to the practical and theoretical advancements in human-robot interaction.

References

[1] B. Louie, E. A. Bj örling, A. C. Kuo, and P. Alves-Oliveira, “Designing for culturally responsive social robots: An application of a participatory framework,” Frontiers in Robotics and AI, vol. 9, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2022.983408

[2] P. Zantou and D. Vernon, “Culturally Sensitive Human-Robot Interaction: A Case Study with the Pepper Humanoid Robot,” in 2023 IEEE AFRICON. Nairobi, Kenya: IEEE, Sep. 2023, pp. 1–6.

[3] M. Colledanchise and P. Ögren, Behavior Trees in Robotics and AI: An Introduction, Jul. 2018, arXiv:1709.00084 [cs]. [Online]. Available: http://arxiv.org/abs/1709.00084.
[4] P. D. Chatzoglou, V. Lazaraki, S. D. Apostolidis, and A. C. Gasteratos, “Factors Affecting Acceptance of Social Robots Among Prospective Users,” International Journal of Social Robotics, Jul. 2023.
[5] Vagnetti, R., Di Nuovo, A., Mazza, M. et al. Social Robots: A Promising Tool to Support People with Autism. A Systematic Review of Recent Research and Critical Analysis from the Clinical Perspective. Rev J Autism Dev Disord (2024). https://doi.org/10.1007/s40489-024-00434-5.
[6] L. Battistuzzi, A. Sgorbissa, C. Papadopoulos, I. Papadopoulos and C. Koulouglioti, “Embedding Ethics in the Design of Culturally Competent Socially Assistive Robots,” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1996-2001, doi: 10.1109/IROS.2018.8594361.
[7] Bruno, B., Recchiuto, C.T., Papadopoulos, I. et al. Knowledge Representation for Culturally Competent Personal Robots: Requirements, Design Principles, Implementation, and Assessment. Int J of Soc Robotics 11, 515–538 (2019). https://doi.org/10.1007/s12369-019-00519-w.
[8] Lim, V., Rooksby, M. and Cross, E.S. Social Robots on a Global Stage: Establishing a Role for Culture During Human–Robot Interaction. Int J of Soc Robotics 13, 1307–1333 (2021). https://doi.org/10.1007/s12369-020-00710-4.
[9] R. Ghzouli, T. Berger, E. B. Johnsen, S. Dragule, and A. Wasowski. 2020. Behavior trees in action: a study of robotics applications. In Proceedings of the 13th ACM SIGPLAN International Conference on Software Language Engineering (SLE 2020). Association for Computing Machinery, New York, NY, USA, 196–209. https://doi.org/10.1145/3426425.3426942.
[10] J. M. O’Kane, “A gentle introduction to ROS” 2016. [Online]. Available: https://api.semanticscholar.org/ CorpusID:13746214.
[11] Cañas, J. M., Perdices, E., García-Pérez, L., and Fernédez-Conde, J. (2020). A ROS-based open tool for intelligent robotics education. Applied Sciences, 10(21), 7419.
[12] D. Faconti and M. Colledanchise, “BehaviorTree.CPP library documentation,” https://www.behaviortree.dev, 2018.
[13] D. Faconti and M. Colledanchise, “BehaviorTree.CPP library tutorials,” https://www.behaviortree.dev/tutorial 01 first tree/, 2018.
[14] D. Faconti. 2019. “BehaviorTree.CPP library”. https://github.com/BehaviorTree/BehaviorTree.CPP