Culturally Sensitive Social Robotics for Africa

A. Akinade,[1[0009–0009–3917–4204]] D. Barros,[2[0009–0009–0283–8206]]
E. Birhan,[1[0009–0008–4362–8152]] M. Danso,[1[0009–0006–4215–1431]]
Y. Haile,[1[0009–0001–0241–2426]] I. Jimoh,[1[0009–0003–1358–5271]]
C. Osano,[1[0009–0000–0969–2507]] P. Ranchod,[3[0000–0001–7157–1714]]
M. Richard,[1[0009–0004–1148–129X]] B. Rosman,[3[0000–0002–0284–4114]]
B. Shimelis Girma,[1[0000–0003–4857–2262]] T. Taye Tefferi,[1[0009–0000–9841–8496]]
and D. Vernon[1[0000–0002–9782–3788]]

[1] Carnegie Mellon University Africa, Kigali, Rwanda
[2] Technical University of Munich, Germany
[3] University of the Witwatersrand, Johannesburg, South Africa
aakinade@alumni.cmu.edu, daniel.barros@tum.de,
ebirhan@alumni.cmu.edu, mdanso@alumni.cmu.edu,
yohanneh@alumni.cmu.edu, ioj@alumni.cmu.edu,
cliffor2@alumni.cmu.edu, Pravesh.Ranchod@wits.ac.za,
muhirwar@alumni.cmu.edu, Benjamin.Rosman1@wits.ac.za,
bgirmash@alumni.cmu.edu, ttefferi@alumni.cmu.edu,
david@vernon.eu

Abstract. Respectful human-robot interaction is informed by local culture. This chapter describes an on-going project to equip Pepper humanoid social robots with the ability to interact verbally and non-verbally with people in Rwanda and South Africa in a culturally sensitive manner. We describe an exercise to acquire Rwandan cultural knowledge by means of an online survey, targeting spatial, verbal, and non-verbal communication. Each type of knowledge in the survey is structured using a cultural knowledge ontology. We describe the development of a complete system architecture which uses this cultural knowledge to interact with people, illustrating the approach with one specific example in which a Pepper robot gives visitors a short tour of a robotics laboratory. The system is implemented in the Robot Operating System (ROS) by developing a package of ROS nodes that provide functionality for overt attention (based on detection & localization of sounds, people, and faces), speech understanding and production, animate behavior, deictic, iconic, and symbolic gesture, and navigation. The robot mission is coordinated by a ROS behavior controller node that interprets the use case scenario specified by a behavior tree.

Keywords: Social robotics, Human-robot interaction, Cultural sensitivity, Pepper humanoid robot, ROS.

1 Introduction

A growing body of research underscores the importance of incorporating cultural factors in human-robot interaction (HRI) to enhance user acceptance, trust, and engagement [1, 2, 3, 4]. In Africa, a continent comprising 54 countries with rich cultural diversity, HRI must also be informed by local cultures [5, 6]. The continent's varied cultural landscape encompasses different perceptions of time and space, diverse social norms, and a multitude of languages and non-verbal communication styles. These factors significantly influence how technology, such as social robots, are perceived, trusted, and adopted across different African communities.
Culturally Sensitive Social Robotics for Africa (CSSR4Africa)[1] is a three-year research project, currently at the end of year two, with the goal of equipping social robots with the ability to interact sensitively and politely with people in Africa; see Fig. 1. The objectives of the project are (i) to identify the verbal and non-verbal social and cultural norms of human interaction that are prevalent in Africa, (ii) to encapsulate them in the behavior of social robots so that they can engage with African people in a manner that is consistent with their expectations of polite social interaction, and (iii) to demonstrate these culturally-sensitive social robot behaviors in two use cases: giving a tour of a university laboratory and giving directions to visitors at the reception of a university.

While the case for culturally competent robots is widely accepted [1, 2], and while there are studies on cultural differences in embracing robots in the West and East, e.g., [7, 8, 9], similar studies of the cultural factors that impact on acceptance in Africa have not been reported [10].[2] The overall goal of the research described here is to redress this situation.

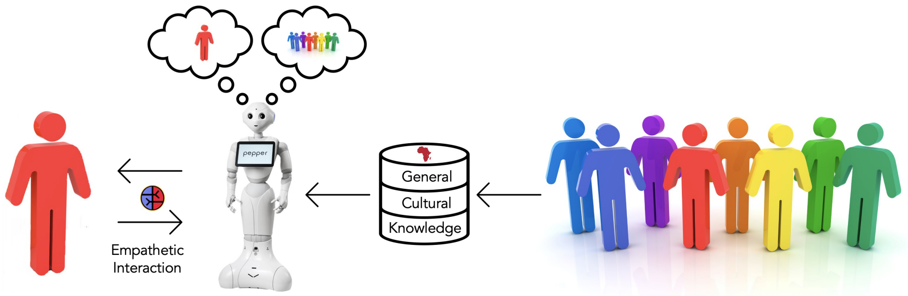

Fig. 1. Based on surveys to acquire cultural knowledge about acceptable modes of communication, the CSSR4Africa project is working to equip Pepper humanoid robots with the ability to interact sensitively and politely with people in Africa using spatial, non-verbal, and verbal modes of communication (graphic inspired by [1]).

---

1 http://www.cssr4africa.org/

2 The survey by Lim et al. mentions Egypt, Tunisia, Libya, and Sudan but only to compare perceptions with the Gulf region when interacting with an Arabic robot [11].

Given constraints in time, effort, and funding, the scope of the project is restricted to the cultures and social practices that are prevalent in Rwanda and South Africa, the countries where the principal investigators currently reside.

2 Cultural Competence vs. Cultural Sensitivity

A culturally competent robot has at least five elements [9], as follows.

1. Cultural knowledge representation.
2. Culturally sensitive planning and action execution.
3. Culturally aware multimodal human-robot interaction.
4. Culture-aware human emotion recognition.
5. Culture identity assessment, habits, and preferences.

This research project focusses on the first three of these, i.e., the generation of *culturally sensitive* robot behavior, i.e., polite, respectful behavior, adhering to the social norms of the culture in which the robot is operating. We do not address the challenge of learning these behaviors from observation, i.e., learning cultural knowledge through interaction. Instead, the approach in the CSSR4Africa project is to catalogue the behaviors based on ethnographic research, create a culture knowledge base, and use that cultural knowledge during interactions. In graphic terms, we restrict our research to the scenario on the right-hand side of the Pepper robot in Fig. 1.

3 African Cultural Knowledge

The first task to be completed before a social robot can exhibit cultural sensitivity is to compile a compendium of cultural knowledge regarding actions and behaviors that are either culturally sensitive or insensitive. We accomplished this by means of an online survey using a questionnaire created with the aid of a cultural knowledge ontology.

The cultural knowledge ontology has three parts, corresponding to spatial interaction, verbal interaction, and non-verbal interaction [10]. Each part comprises specific elements of cultural norms and behaviors. For example, spatial interaction includes distance and positioning, while non-verbal interaction includes eye contact, facial expressions, and hand movements. Verbal interaction covers aspects like word choice, voice tone, and conversation turn-taking. The ontology is restricted to the actions that the Pepper robot can perform. It omits forms of non-verbal communication that are important in human-robot interaction, e.g., facial expressions, such as eyebrow and mouth gestures. We adopt a simple approach to defining the cultural knowledge ontology, one that represents the ontology as a tree of concepts, as shown in Fig. 2. This approach provides us with a straightforward way to specify the parameter values for each element in the ontology: we represent the cultural knowledge with a simple list

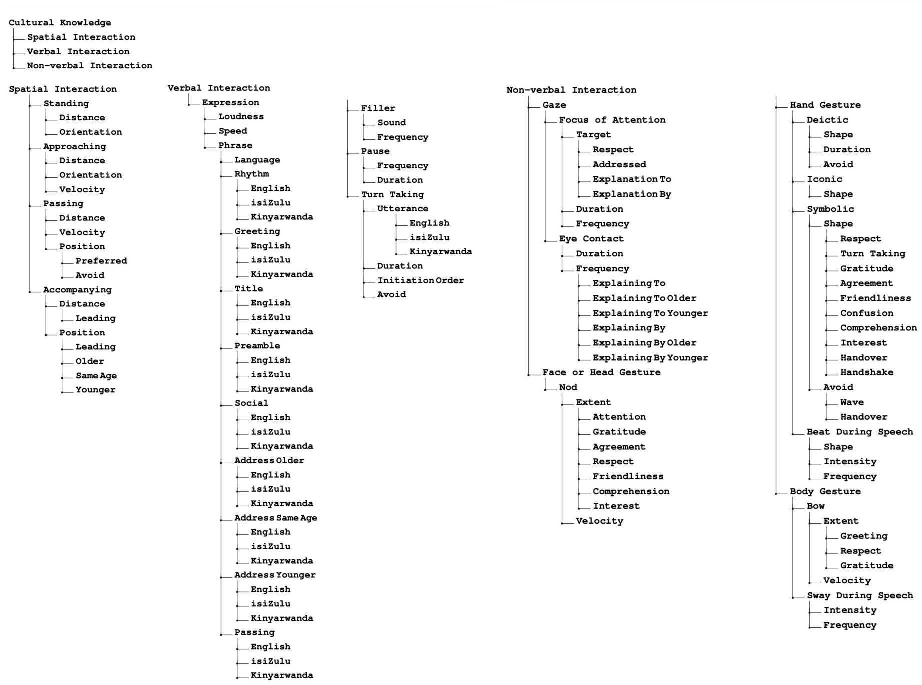

Fig. 2. Cultural knowledge ontology for spatial, verbal, and non-verbal interaction.

of key-value pairs, where a key is constructed from the name of a leaf nodes in the ontology tree and the name of its parent. The values, derived from the knowledge elicited in the survey, can be either quantitative numeric values or qualitative symbolic values, which can be interpreted to produce culturally sensitive behavior.

The survey questionnaire, in English and Kinyarwanda,[3] comprises three parts: (1) personal information for demographic balance, (2) validation of previously gathered cultural knowledge from a preliminary survey of 23 people from eight African countries [12], and (3) identification of culturally sensitive and disrespectful behaviors, allowing us to confirm and refine our understanding of Rwandan cultural norms within the broader context of African cultural knowledge.

Before launching the survey of the general public in Rwanda, it was piloted at Carnegie Mellon University Africa (CMU-Africa) over a two-month period.

We collected 108 responses in English and 35 in Kinyarwanda. The pilot also helped improve the questionnaire, particularly in clarifying cultural terminology and adjusting question sequencing.

After conducting the survey, we cleaned and organized the collected responses to identify consensus answers to each question, and then mapped each consensus answer to the cultural knowledge ontology keys. Five of the fifty-seven questions in the cultural

[3] The questionnaire is available online in Kinyarwanda (https://bit.ly/3TJX4K2) and English (https://bit.ly/4aqCVhs).

Table 1. Consensus answers to a sample of the questions in the cultural knowledge survey.

Question	Consensus Cultural Knowledge
2-1	To show respect, one should lower gaze when greeting someone older
2-6	One should use an open palm of the hand to point to people and objects
2-7	One should not point an upward facing palm of the hand at someone
2-9	To show respect, one should bow slightly when greeting someone older
2-12	One should not use the left hand to hand something to someone
2-14	To show respect, one should shake hands with the right hand and use the left arm to support the right forearm when doing so
2-16	To show respect, one should bow slightly and lower gaze when greeting someone older
2-24	One should not talk loudly to an older person
2-26	One should not walk between two or more people who are conversing
3-2	One should say ‘Muraho’ or ‘Hello’ when acknowledging someone while passing them
3-3	One should pass behind a group of two or more people

knowledge survey yielded no consensus — three in Part 2 and two in Part 3 — leaving fifty-two consensus answers to populate the culture knowledge base. This necessitated that the original ontology be extended to ensure each knowledge item, i.e., each consensus answer, has at least one corresponding entry in the ontology tree (some consensus answers gave rise to more than one entry). If several survey answers mapped to a single leaf node in the original ontology tree, the ontology was extended by appending suffixes to the leaf labels, e.g., `symbolicShapeRespect` instead of `symbolicShape`. This results in an ontology tree with at least one leaf node for each consensus answer. The ontology tree also has several leaf nodes for which there is no associated knowledge, either because no consensus answer emerged, or because no survey question addressed that element of the ontology. The extended ontology is shown in Fig 2. A sample of consensus answers to the survey questions is shown in Table 1.

4 Cultural Knowledge Base

Table 2 lists the knowledge associated with a sample of the keys derived from the ontology tree in Fig. 2. The culture knowledge base in the high-level system architecture shown in Fig. 5 replaces this knowledge with numeric and symbolic values (see Table 3) that are used by the Robot Mission Interpreter subsystem (see Section 7.6), first querying the knowledge base, and then passing the retrieved values as arguments in the ROS service requests it issues to the first task to be completed before a social robot can exhibit cultural sensitivity is to nodes in the system architecture to execute the robot mission in a culturally sensitive manner.

To prompt the user, reply to the user, or just ensure flow in the dialogue, the robot may also need to speak phrases not captured in the cultural knowledge survey. These are included in the culture knowledge base as a set of utility verbal interaction phrases, rather than embed them in the Robot Mission Interpreter software or create another knowledge base.

Table 2. Key-knowledge pairs for specifying culturally sensitive actions using the extended ontology, i.e., the ontology depicted in Fig. 2.

Key	Cultural Knowledge
`passingPositionPreferred`	Pass behind a group of two or more people
`passingPositionAvoid`	Walk between two or more people who are conversing
`expressionLoudness`	Do not talk loudly to an older person
`phrasePassingEnglish`	Say ‘Hello’ to acknowledge someone when passing them
`phrasePassingKinyarwanda`	Say ‘Muraho’ to acknowledge someone when passing them
`focusOfAttentionTargetRespect`	Lower your gaze and bow slightly
`deicticShape`	Use an open palm to point at people and objects
`deicticAvoid`	Do not point an upward-facing palm at someone
`symbolicShapeHandShake`	Handshake with the right arm supported by the left hand & bow
`symbolicAvoidHandover`	Do not use the left hand without using the right hand
`bowExtentGreeting`	Bow slightly when greeting someone older.
`bowExtentRespect`	Bow slightly

Table 3. Key-value pairs for specifying culturally sensitive actions using the extended ontology, i.e., the ontology depicted in Fig. 2.

Key	Values	Units
`passingPositionPreferred`	`BEHIND`	
`passingPositionAvoid`	`BETWEEN`	
`expressionLoudness`	`40`	% of max.
`phrasePassingEnglish`	`Hello`	
`phrasePassingKinyarwanda`	`Muraho`	
`focusOfAttentionTargetRespect`	`TORSO`	
`deicticShape`	`PALM_UPWARDS`	
`deicticAvoid`	`PALM_FORWARDS`	
`symbolicShapeHandShake`	`POLITE_HANDSHAKE`	
`symbolicAvoidHandover`	`LEFT_HAND`	
`bowExtentGreeting`	`10`	degrees
`bowExtentRespect`	`10`	degrees

5 Environment Knowledge Base

The high-level system architecture shown in Fig. 5 also has an environment knowledge base with the information required to execute the robot mission — including references to cultural knowledge — structured using a simple ontology shown in Fig. 3. Internal nodes in the ontology tree form the key in the environment knowledge base, e.g., `robotLocation`. Leaf nodes represent the data entities and their types. This allows multiple elements in a value for each key, e.g., `robotLocation 3 15.2 9.0 45.0`. The identification number value element associated with each key is the means by which the different elements of an environment location are related: robot location, location description, gesture target, pre-gesture message, post-gesture message, and the culture knowledge base keys for the cultural knowledge that is to be used at this location. The tour specification identifies the number and sequence of locations to be visited in the tour.

Table 4 lists the key-value pairs, i.e., each key and the associated multiple numeric or alphanumeric elements of the value that encapsulate the environment knowledge. These numeric or alphanumeric values can then be used directly in the Robot Mission Interpreter subsystem, and passed as arguments in the service requests it issues to the nodes in the system architecture.

```
Environment Knowledge
├── Robot Location
│   ├── Identification Number (integer)
│   └── Pose
│       ├── x (float)
│       ├── y (float)
│       └── theta (float)
├── Robot Location Description
│   ├── Identification Number (integer)
│   └── Description (string)
├── Gesture Target
│   ├── Identification Number (integer)
│   └── Position
│       ├── x (float)
│       ├── y (float)
│       └── z (float)
├── Pre-gesture Message
│   ├── Identification Number (integer)
│   └── Description(string)
│       ├── English (string)
│       ├── Isizulu (string)
│       └── Kinyarwanda (string)
├── Post-gesture Message
│   ├── Identification Number (integer)
│   └── Description(string)
│       ├── English (string)
│       ├── Isizulu (string)
│       └── Kinyarwanda (string)
├── Cultural Knowledge (string)
│   ├── Identification Number (integer)
│   ├── Cultural Knowledge Key (string)
│   ├── Cultural Knowledge Key (string)
│   ├── ...
│   └── Cultural Knowledge Key (string)
└── Tour Specification
    ├── Number of Locations (integer)
    └── Location Sequence
        ├── Identification Number (integer)
        ├── Identification Number (integer)
        ├── ...
        └── Identification Number (integer)
```

Fig. 3. Environment knowledge ontology.

Table 4. Key-value pairs for specifying environment knowledge actions using the ontology depicted in Figure 3. As noted in the text, the identification number element of the value associated with each key is the means by which the robot location, the location description, the gesture target, the pre-gesture message, the post-gesture message, and the cultural knowledge keys (from the cultural knowledge ontology) are related. The tour specification identifies the number of locations and the sequence of locations to be visited in the tour.

Key	Values	Units
robotLocationPose	<IDNumber> <x> <y> <theta>	Metres, degrees
robotLocationDescription	<IDNumber> <text>	String
gestureTarget	<IDNumber> <x> <y> <z>	Metres
preGestureMessageEnglish	<IDNumber> <text>	String
preGestureMessageIsizulu	<IDNumber> <text>	String
preGestureMessageKinyarwanda	<IDNumber> <text>	String
postGestureMessageEnglish	<IDNumber> <text>	String
postGestureMessageIsizulu	<IDNumber> <text>	String
postGestureMessageKinyarwanda	<IDNumber> <text>	String
culturalKnowledge	<IDNumber> <Key1> ... <Keyn>	String
tourSpecification	<n> <ID1>, <ID2>, ..., <IDn>	String

6 Use Case Scenario Specification

The complete system is being validated in two use cases scenarios, a lab tour guide and receptionist. Detailed walk-throughs of both scenarios, including a decomposition into a time sequence of elementary robot actions, can be found in [13]. Algorithm 1 is a brief outline of the lab tour use case in the form of a pseudo-code algorithm. This algorithm captures only the functional aspects of the tour use case, specifying what the robot is to do, not how it is to do it or where it is to do it. These non-functional aspects are handled by the Robot Mission Interpreter subsystem (see Section 7.6) by querying the culture and environment knowledge bases when interpreting a behavior tree representation of Algorithm 1 stored in the Robot Mission Specification file (see the high-level architecture in Fig. 5). For example, the pseudo-code statements “Look at the visitor and adjust pose”, “Make a symbolic gesture for welcome”, and “Make a deictic gesture toward the object” in Algorithm 1 involves querying the culture knowledge base for the relevant cultural knowledge, e.g., duration of eye contact (`eyeContactDuration` key-value pair), symbolic gesture (`symbolicShapeRespect`, `focusofAttentionTargetRespect`, and `bowExtentGreeting` key-value pairs), and deictic gesture (`deicticShape` key-value pair).

The location and descriptions of each stop on the tour, and the order in which they are to be visited, are determined by querying the Environment Knowledge Base. The map of the environment is provided separately as an input to the Gesture, Speech, and Navigation subsystem described in Section 7.5. Note that failure handling is omitted from the pseudo-code encapsulation in Algorithm 1.

Fig. 4 shows four examples of the Pepper robot using cultural knowledge when giving a tour of the Robotics Lab at Carnegie Mellon University Africa.

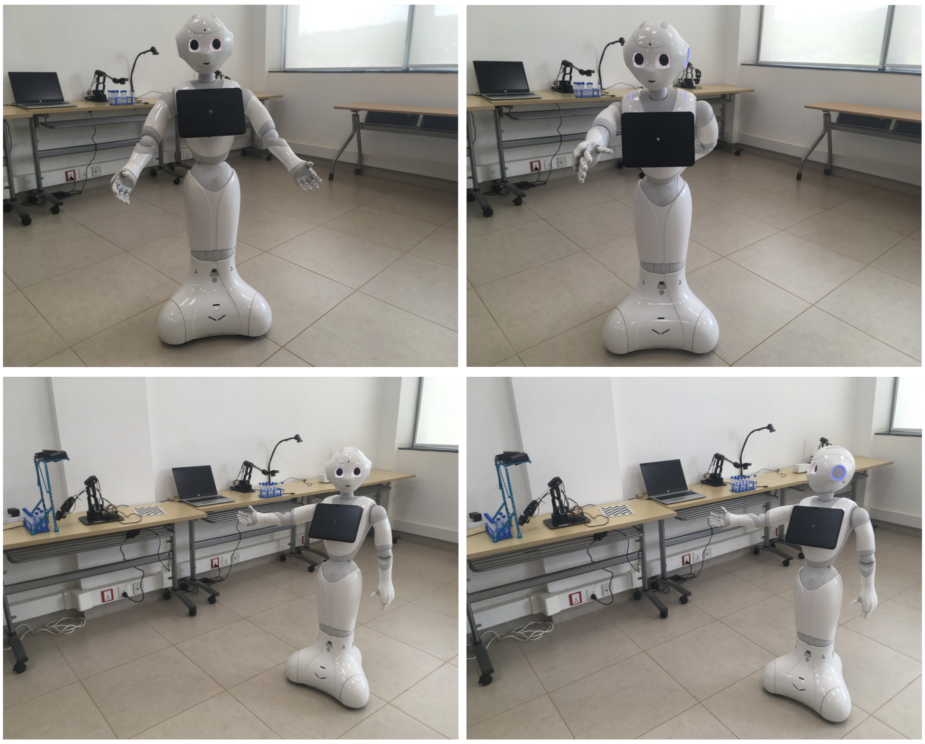

Fig. 4. The Pepper robot making four gestures. Top left: a symbolic welcome gesture. Top right: a greeting gesture with right hand extension and a slight bow. Bottom left: a deictic gesture, while maintaining mutual gaze with the user; note that the palms of the hand are open and directed upwards. Bottom right: a deictic gesture, while invoking joint attention with the user.

Algorithm 1 Robot behavior for the lab tour use case

```
Ensure: The CSSR4Africa ROS system has been launched.
Ensure: The Pepper robot stands at the greeting location in the robotics laboratory.
  repeat
    repeat
      Make natural movements
      Attention ← social mode, detecting faces
      if a face is detected then
        Look at the visitor and adjust pose
      end if
    until face detected
    Make a symbolic gesture for welcome
    Say "Hello! Welcome to the Robotics Lab!"
    repeat
      Attention ← seek mode, detecting mutual gaze
    until Mutual gaze established
    repeat
      Say "Would you like a tour of the lab?"
      Listen to the visitor and wait for a response
      if the response is not "Yes" or "No" then
        Say "I only understand yes or no"
      end if
    until the visitor says "Yes" or "No" or the robot has prompted three times
    if the visitor says "No" or does not respond then
      Say "Maybe another time"
    else
    repeat                    ▷ The robot starts the tour and visits each exhibit in turn
      Query environment knowledge base for location and description
      Query culture knowledge base for associated knowledge
      Say "Please follow me"
      Navigate to location
      repeat
        Attention ← social mode, detecting faces
        if a face is detected then
          Look at the visitor and adjust pose
        end if
      until face detected
      Say "This is <lab equipment to be described>"
      Make a deictic gesture toward the object
      Look at the object
      Look at the visitor
      Say "<description of lab equipment>"
    until All locations have been visited        ▷ The last location is where the robot
  end if                                         ▷ ends the tour and thanks the visitor
until The application is terminated
```

7 System Architecture

The design of the system architecture draws on experience in reviewing [14], designing [15,16], and implementing robot (cognitive) architectures, including the iCub [17] and DREAM architectures [18]. The system architecture is specified at two levels of abstraction: a high-level architecture, identifying major subsystems and data sources, and an implementation-level architecture, comprising a suite of interconnected ROS nodes.

The high-level architecture is shown in Fig. 5. It comprises six subsystems and three external data sources. Two of the six subsystems are concerned with sensing (Sensing & Analysis and Detection of Interaction Events), three with actuation Attention, Animate Behavior, and Gesture, Speech, and Navigation), and one with control (Robot Mission Interpreter). The three external data sources are the Culture Knowledge Base, the Environment Knowledge Base, and the Robot Mission Specification.

The implementation-level system architecture, realizing the six subsystems in the high-level architecture, are implemented in ROS Noetic on Ubuntu 20.04, each subsystem comprising one or more ROS nodes, as shown in Table 5. We describe the ROS node implementation of each subsystem in the sections that follow.

Several reasons underpinned the decision to use ROS [19] rather than other robot middleware systems, such as YARP [20]. Even though previous projects which implemented robot software with YARP were very successful [21,22,18], ROS has become the *de facto* standard realization of component-based software engineering (CBSE) [23,24], the paradigm most favored in the robotics community, and has been adopted by the two partner universities in the CSSR4Africa project for both research and teaching.

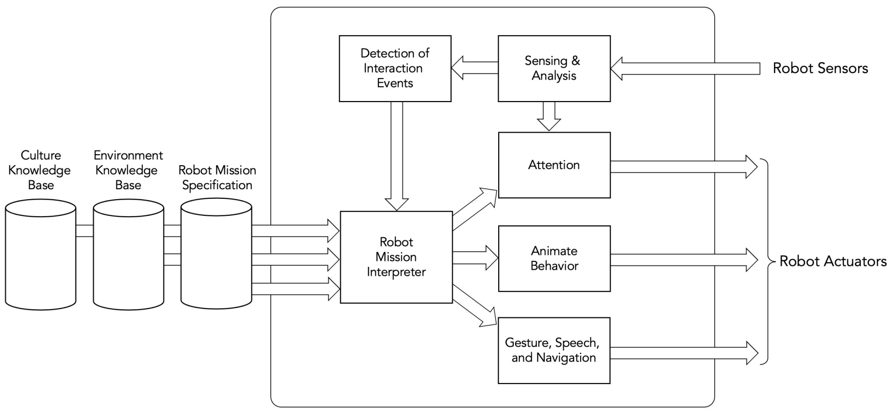

Fig. 5. High-level system architecture comprising six subsystems and three external data sources; see text for details.

Table 5. ROS nodes comprising each subsystem in the high-level system architecture; see Fig. 6 for the implementation-level system architecture specified as a network on interconnected ROS nodes.

Subsystem	ROS Nodes
Sensing & Analysis	`robotLocalization, personDetection, faceDetection, soundDetection`
Detection of Interaction Events	`speechEvent, tabletEvent`
Attention	`overtAttention`
Animate Behavior	`animateBehavior`
Gesture, Speech, and Navigation	`gestureExecution, robotNavigation, textToSpeech`
Robot Mission Interpreter	`behaviorController`

7.1 The Sensing & Analysis Subsystem

This subsystem comprises four ROS nodes: `robotLocalization`, `personDetection`, `faceDetection`, and `soundDetection`.

The `robotLocalization` node estimates the pose of the Pepper robot in a Cartesian world frame of reference. The module achieves this through a combination of relative and absolute position estimation techniques, including odometry, IMU data, and triangulation using visually detected ArUco marker visual landmarks.

The `personDetection` node detects and localizes people in the camera's field of view, while the `faceDetection` node detects and localizes human faces and determines whether mutual gaze is established between the Pepper robot and the human user through head pose estimation.

The `soundDetection` node detects and localizes conspicuous sounds within a robot's hearing range. It provides two outputs: the direction of arrival of the sound, and a filtered audio signal, thereby enabling the robot to focus its attention on sound sources. It interfaces with the `speechEvent` node in the Detection of Interaction Events subsystem, facilitating automatic speech recognition in both Kinyarwanda and English, and with the `overtAttention` node in the Attention subsystem.

7.2 The Detection of Interaction Events Subsystem

This subsystem comprises two ROS nodes: `speechEvent` and `tabletEvent`.

The `speechEvent` node deploys a speech-to-text model using deep neural networks that enables speech utterances in Kinyarwanda and English languages captured by Pepper's microphones to be transcribed into written text.

The `tabletEvent` node provides an alternative way of interacting with the user, allowing the user to respond to spoken prompts by tapping a graphic user interface instead of using automatic speech recognition provided by the `speechEvent` node.

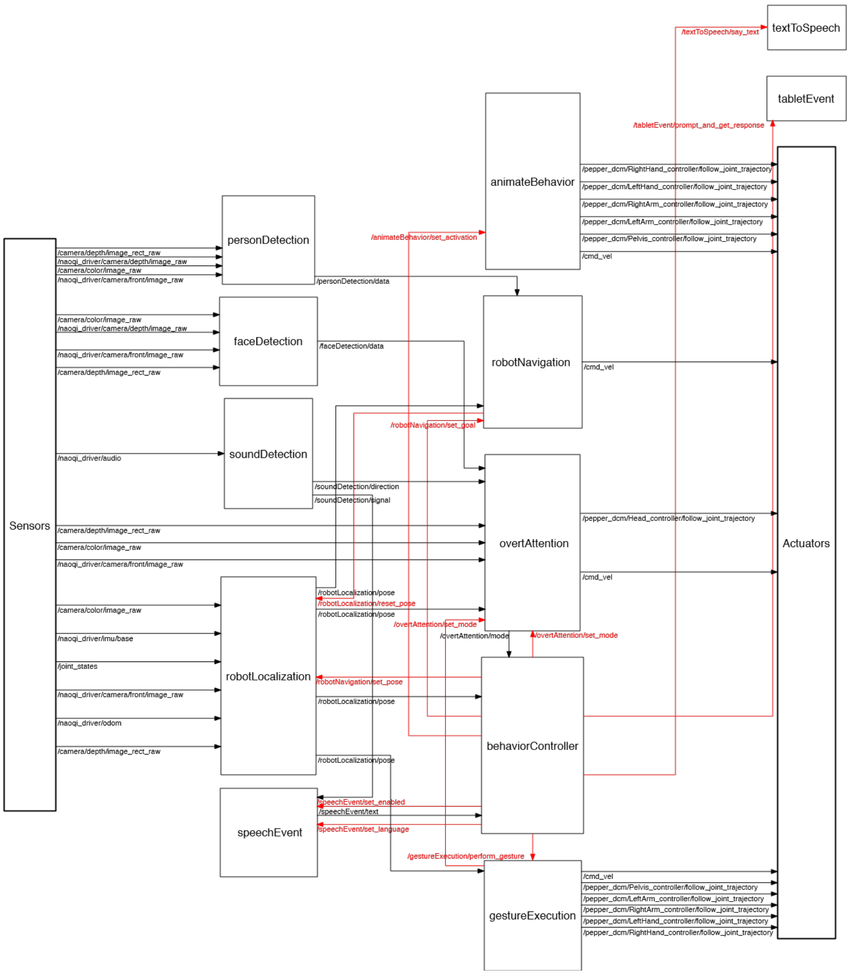

Fig. 6. Detailed system architecture comprising 12 ROS nodes. Black connectors depict published ROS topics; red connectors depict ROS service requests.

7.3 The Attention Subsystem

This subsystem comprises just one ROS node, `overtAttention`, which enables the robot to dynamically direct its attention towards salient features in its environment, primarily during social interactions, and to scan its surroundings when not actively engaged. Based on input from the `faceDetection` and `soundDetection` nodes, as well as an internally-generated saliency map, it directs the robot's gaze by controlling the head azimuth and elevation angles, and the robot pose to recentre the head after changing the head gaze. It operates in different modes, paying attention to features of general interest, to human faces in general, or to human faces exhibiting mutual gaze, i.e., the situation where the robot and the human are looking at each other. It also allows the gaze to be directed to a specific location in the robot's environment.

7.4 The Animate Behavior Subsystem

This subsystem also comprises just one ROS node, `animateBenavior`, which enhances the robot's lifelike appearance by generating subtle body movements to flex the robot's hands and slightly rotate its base around the vertical axis. Head movements are handled by the Attention subsystem.

7.5 The Gesture, Speech, and Navigation Subsystem

This subsystem comprises three ROS nodes: `gestureExecution`, `robotNavigation`, `textToSpeech`.

The `gestureExecution` node enables the Pepper robot to perform a range of hand and body gestures: deictic, symbolic, and iconic hand gestures, and bowing and nodding body movements. Hand gestures are executed so that they exhibit biological motion, since this has been shown to enhance people's engagement when interacting with the robot [25]. The node interfaces with the robot localization system, so that deictic gestures accurately indicate the location of points of interest in the robot's environment. Additionally, it interfaces with the Attention subsystem to direct the head to the same location to which the robot is making a deictic gesture.

The `robotNavigation` node enables the Pepper robot to traverse its environment autonomously. It employs different path planning algorithms, including Breadth-First Search (BFS), Dijkstra, and A*. Navigation is executed by identifying waypoints along the planned path and controlling the robot to locomote from waypoint to waypoint. A key feature of this node is the capacity to incorporate culturally sensitive proxemics, derived from the Rwandan culture knowledge base. This functionality ensures that the robot maintains appropriate social distances when navigating around humans, enhancing its adaptability and acceptance in human-centered environments. It interfaces with the `robotLocalization` node to ensure the navigation path is registered with the robot's environment, continuously updating its pose with real-time data.

The `textToSpeech` node provides the functionality for converting English and Kinyarwanda text to speech.

7.6 The Robot Mission Interpreter Subsystem

The Robot Mission Interpreter subsystem comprises a single ROS node: the `behaviorController`. This is a central component in the CSSR4Africa system architecture. It interprets and executes the robot mission, specified using a behavior tree [26,27,28] in the Robot Mission Specification file, and it queries the Environment Knowledge Base and Culture Knowledge Base to determine the tour locations, descriptions, and associated cultural knowledge values, as described in Sections 5 & 6.

It interfaces directly with eight ROS nodes: `animateBehavior`, `gestureExecution`, `overtAttention`, `robotLocalization`, `robotNavigation`, `speechEvent`, `tabletEvent`, and `textToSpeech`, subscribing to their ROS topics for sensor data and issuing ROS service requests to control the robot's action, passing the values retrieved from the environment and culture knowledge bases as arguments.

7.7 Programming by Demonstration

A stand-alone ROS application has also been developed to provide the robot with the ability to learn gestures through manual teleoperation or human demonstration, employing RGB-D camera technology to map human skeletal movements onto the robot's joint system, i.e., programming by demonstration [29,30], also known as learning from demonstration [31].

In addition there are a suite of unit tests to verify that sensor data is successfully acquired on each sensor topic, and to verify the accurate and reliable functioning of the robot actuators: head, arms, hands, legs, and wheels.

8 Use Case Scenario Implementation

Broadly speaking, there are two complementary approaches to specifying a robot mission: state machines and behavior trees. Gzhouli et al. [28] make a compelling case for the use of behavior trees and, consequently, this is the methodology we have adopted for specifying the robot mission, i.e., the lab tour use case scenario. A behavior tree consists of nodes arranged in hierarchical tree structure with key components including a root node, control flow nodes (sequence, selector, and decorator), conditional nodes, and action nodes [28]. The root node is the starting point of the behavior tree. The execution begins here and traverses depth-first through the tree. The control flow nodes determine the flow of execution. Sequences execute child nodes from left to right, returning success only if all children succeed. They are used for "do all tasks in order" behaviors. Selectors (also known as fallbacks) execute child nodes from left to right, returning success when one child succeeds. They are used for "try all tasks in order until any succeeds" behaviors. Decorators modify the behavior of their child nodes (e.g., repeat, invert, and 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 or make a deictic gesture to an object) or condition nodes that check certain conditions or states (e.g., battery level or obstacle detection). Recall from Section 6 that Algorithm 1 captures only the functional aspects of the tour use case. This is also true of the behavior tree which implements Algorithm 1. The non-functional aspects are handled by the `behaviorController` ROS node that comprises the Robot Mission Interpreter subsystem (see Section 7.6) by querying the Environment Knowledge Base and Culture Knowledge Base when interpreting the behavior tree.

The location and descriptions of each tour location, including the cultural knowledge to be used at that location, and the order in which the locations are to be visited, are determined by querying the Environment Knowledge Base. The map of the environment is provided separately as an input to the Gesture, Speech, and Navigation subsystem described in Section 7.5. Note that failure handling is omitted from the pseudo-code encapsulation in Algorithm 1.

The XML specification of the behavior tree is stored in the third external file system in the high-level system architecture shown in Fig. 5, i.e., the Robot Mission Specification.

The behavior tree for the lab tour use case was designed with Groot2, an IDE for the `Behavior-Tree.CPP` library [26,27] allowing interactive creation and editing of behavior trees. It produces an XML specification of the behavior tree that is read and executed by the `Behavior-Tree.CPP` library, linked from the `behaviorController` ROS node. Each behavior tree action node interfaces with one of the ten ROS nodes mentioned in the previous section, either subscribing to ROS topics or calling ROS services. The `Behavior-Tree.CPP` library was chosen because, among those surveyed by Ghzouli et al. [28], it is the only currently-maintained open-source library that supports ROS and is implemented in C++, the preferred programming language at the time the choice was made.

To facilitate an efficient mission design process and enhance readability and clarity, the overall mission is structured as eight distinct subtrees, each representing a logical segment of the overall task defined in the use case scenario. These are the root TourGuide subtree, and its constituent DetectVisitor, EngageVisitor, QueryVisitorResponse, VisitExhibit, EndTour, NavigateToLocation, and GoHome subtrees. The subtree leaf nodes, which include both action and condition nodes, are where the custom functionality is implemented. Combined with control flow nodes, these building blocks enable the definition of the desired behaviors. A total of 21 action and condition nodes were defined for the lab tour scenario, with many of these nodes reused multiple times throughout the behavior tree. The complete specification of the behavior tree is documented in [32].

9 Discussion

As noted above, we have adopted a simple approach to specifying the cultural knowledge ontology. There are superior approaches, e.g., the OWL-2 language used by Bruno et al. [33]. Similarly, there are also more powerful ways to construct a culture knowledge base than the ontology-based key-value approach we have adopted, e.g., using first-order logic to represent the culture facts and the rules that define how this knowledge can be used, e.g., as the CRAM cognitive architecture does [34]. However, we defer exploring these more sophisticated approaches until the first proof-of-principle CSSR4Africa system has been fully tested.

When planning the work described in this paper, we assumed that a ROS package existed which would provide all the software required to access the Pepper sensors and control the Pepper actuators. This assumption proved to be invalid. Most developers use Choreograph, the visual programming interface provided with the robot. This necessitated the development of a complete suite of ROS modules to provide the requisite functionality. These are now encapsulated in the `pepper_interface_tests` ROS package and used in the `cssr_africa` ROS package, both of which are freely available on the CSSR4Africa software respository on GitHub [35]. We also assumed that the quality of the data produced by the sensors on the Pepper robot would be sufficient for our needs. This also turned out not to be valid, requiring the installation of a RealSense camera speaking, there are two complementary approaches to specifying a robot [36] on the head of the robot and a Lidar laser rangefinder near the base of the robot to provide reliable sensor data.

As is evident from the ROS system architecture depicted in Fig. 6, the CSSR4Africa system is complex,[4] involving twelve concurrently operating ROS nodes, and many topics and services, all communicating in real time. Dealing effectively with this complexity required the adoption of software engineering standards, and rigorous adherence to them in the design, development, implementation, testing, and documentation of each ROS node, and the adoption of a strict process of quality assurance when submitting CSSR4Africa software for integration. The practice, which was pivotal to the success of the system engineering aspect of the endeavour, is supported by several resources, including a reference manual of software engineering standards, a comprehensive installation manual providing step-by-step instructions for setting up the development environment and controlling the Pepper robot in both physical and simulated environments, the procedures used in CSSR4Africa to validate and test software prior to integration into the CSSR4Africa software repository. All of these resources are available on the CSSR4Africa website [38].

As noted in Section 1, this chapter describes an ongoing project that has just entered its third and final year. Due to the unforeseen delays caused by the need to develop from scratch the package of ROS nodes and integrate them in a complete system, we have focussed to date mainly on the functional aspects of the lab tour use case, while providing the capability for the Pepper robot to exhibit culturally sensitive behavior by querying the Culture Knowledge Base at run time. However, this capability is only used to a limited extent at present and, while eye contact duration, deictic gesture shape, symbolic gesture shape, bow extent cultural knowledge, for example, are specified in the Culture Knowledge Base, these values are currently hard-coded in the `behaviorController`, rather than being dynamically retrieved at run time. The immediate goal in the remaining year is to fully exploit this run-time capability in the `behaviorController`, as described in Sections 5 and 6, and utilize all the cultural knowledge produced by the survey described in Section 3 and currently encapsulated in the Culture Knowledge Base. We also plan on conducting a user study employing the Robotic Social Attributes Scale (RoSAS) [39] to assess the degree to which culturally sensitive behavior enhances the perceptions of users, similar to a study we conducted for a related project to determine the extent to which biological motion aids gestural communication by social robots [25].

---

[4] We use the term "complex" in the loose sense of a system with many parts and interconnections, rather than the alternative sense of a system that is self-organizing and emergent [37].

10 Sustainable Practices

In 2015, the UN general assembly outlined a global agenda to address urgent world issues by 2030 [40]. These goals, known as the Sustainable Development Goals (SDGs), comprise 17 interconnected objectives, including poverty eradication (SDG 1), zero hunger (SDG 2), good health and well-being (SDG 3), and quality education (SDG 4). Robotics has an important role to play in realizing many of the SDGs [41].

The Fourth Industrial Revolution, also referred to as Industry 4.0, is marked by innovation in AI and robotics. Through technological innovation, AI and robotics have the potential to address hunger by improving food security, sustainable agriculture, and healthcare systems. These technologies can assist doctors in making informed decisions, accurately predict and treat cancers, enhance banking systems, education, and the lives of individuals with disabilities. Unlike previous technological revolutions that primarily benefited the global north, the fourth industrial revolution, with the SDGs as its compass, has the potential to bring about disruptive and positive change in the global south, including countries in Africa, provided AI and robotics are embraced and leveraged effectively [42,43]. Innovation (rather than mere invention) requires adoption [44] which in turn depends on trust [45]. In the case of social robots, trust and acceptance are fostered if the robots exhibit cultural competence [6]. Social robots must be inclusive of the people they serve. Otherwise, the benefits that such technologies can bring will not be realized, and the many sustainable development goals to which robotics can contribute may not be achieved. In short: inclusion drives sustainable development, and cultural sensitivity is a prerequisite for inclusion [46]. This is core motivation for the research described in this chapter.

11 Conclusion

The overall goal of the CSSR4Africa project is to redress the lack of focus on African cultures when developing social robots with cultural sensitivity and cultural competence. The work done to date lays a solid foundation for the achievement of this goal. As noted above, the pilot survey yielded 108 responses in English and 35 in Kinyarwanda. While this is certainly useful, a more extensive survey of the general population is needed before we can claim that the culture knowledge base is comprehensive and representative. The survey also needs to be conducted in South Africa with isiZulu speakers. In the meantime, our objective is to promote awareness of the work through the African Engineering and Technology Network (Afretec) [47], a pan-Africa alliance of nine technology-focussed universities, by dissemination to a broader audience through the CSSR4Africa website [38], and by making the software outlined in this paper openly available on the CSSR4Africa software repository [35].

Acknowledgments. This research was carried out as part of the Afretec Network. Afretec is managed by Carnegie Mellon University Africa and receives financial support from the Mastercard Foundation.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this chapter.

References

1. Bruno, B., Chong, N.Y., Kamide, H., Kanoria, S., Lee, J., Lim, Y., Pandey, A.K., Papadopoulos, C., Papadopoulos, I., Pecora, F., Saffioti, A., Sgorbissa, A.: Paving the way for culturally competent robots: A position paper. In: 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). pp. 553–560. Lisbon, Portugal (2017)
2. Khaliq, A., Köckemann, U., Pecora, F., Saffiotti, A., Bruno, B., Recchiuto, C., Sgorbissa, A., Bui, H.D., Chong, N.: Culturally aware planning and execution of robot actions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 326–332 (2018)
3. Lugrin, B., Rehm, M.: Culture for socially interactive agents. In: The Handbook on Socially Interactive Agents: 20 Years of Research on Embodied Conversational Agents, Intelligent Virtual Agents, and Social Robotics Volume 1: Methods, Behavior, Cognition, chap. 13, pp. 463–494. Association for Computing Machinery (2021)
4. Ornelas, M., Smith, G., Mansouri, M.: Redefining culture in cultural robotics. AI & Society (2022)
5. Akinade, A., Haile, Y., Mutangana, N., Tucker, C., Vernon, D.: Culturally competent social robots target inclusion in Africa. Science Robotics 8(85) (2023)
6. Vernon, D.: An African perspective on culturally competent social robotics: Why DEI matters in HRI. IEEE Robotics and Automation Magazine 31(4), 170–200 (2024)
7. Kaplan, F.: Who is afraid of the humanoid? Investigating cultural differences in the acceptance of robots. International Journal of Humanoid Robotics 1(3), 1 – 16 (2004)
8. Bartneck, C., Nomura, T., Kanda, T., Suzuki, T., Kennsuke, K.: Cultural differences in attitudes towards robots. In: Proceedings of the AISB Symposium on Robot Companions: Hard Problems and Open Challenges in Human-Robot Interaction. pp. 1–4 (2005)
9. Bruno, B., Chong, N.Y., Kamide, H., Kanoria, S., Lee, J., Lim, Y., Pandey, A.K., Papadopoulos, C., Papadopoulos, I., Pecora, F.: The CARESSES EU-Japan project: Making assistive robots culturally competent. In: arXiv 1708.06276 (2017)
10. Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., Sabanovic, S.: Human-Robot Interaction – An Introduction. Cambridge University Press (2020)
11. Lim, V., Rooksby, M., Cross, E.S.: Social robots on a global stage: Establishing a role for culture during human–robot interaction. International Journal of Social Robotics 13, 1307–1333 (2021)
12. Zantou, P., Vernon, D.: Culturally-sensitive human-robot interaction: A case study with the Pepper humanoid robot. In: Proc. IEEE Africon. Nairobi, Kenya (2023)
13. Vernon, D.: CSSR4Africa Deliverable D2.1 User Case Scenario Definition. https://cssr4africa.github.io/deliverables/CSSR4Africa_Deliverable_D2.1.pdf (2025)
14. Vernon, D.: Cognitive architectures. In: Cangelosi, A., Asada, M. (eds.) Cognitive Robotics, pp. 191–212. MIT Press (2022)
15. Vernon, D., von Hofsten, C., Fadiga, L.: Desiderata for developmental cognitive architectures. Biologically Inspired Cognitive Architectures 18, 116–127 (2016)
16. 16. Vernon, D.: Two ways (not) to design a cognitive architecture. In: Chrisley, R., Müller, V.C., Sandamirskaya, Y., Vincze, M. (eds.) Proceedings of EUCognition 2016, Cognitive

Robot Architectures. vol. CEUR-WS Vol-1855, pp. 42–43. European Society for Cognitive Systems, Vienna (2017)
17. Vernon, D., Metta, G., Sandini, G.:The iCub cognitive architecture: Interactive development in a humanoid robot. In: Proceedings of IEEE International Conference on Development and Learning (ICDL). Imperial College, London (2007)
18. Vernon, D., Billing, E., Hemeren, P., Thill, S., Ziemke, T.: An architecture-oriented approach to system integration in collaborative robotics research projects – an experience report. Journal of Software Engineering for Robotics 6(1), 15–32 (2015)
19. Robot Operating System (ROS): https://www.ros.org/
20. Metta, G., Fitzpatrick, P., Natale, L.: Yarp: yet another robot platform. International Journal on Advanced Robotics Systems 3(1), 43–48 (2006)
21. Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Santos-Victor, J., Bernardino, A., Montesano, L.: The iCub Humanoid Robot: An Open-Systems Platform for Research in Cognitive Development. Neural Networks, special issue on Social Cognition: From Babies to Robots 23, 1125–1134 (2010)
22. Vernon, D., von Hofsten, C., Fadiga, L.: A Roadmap for Cognitive Development in Humanoid Robots, Cognitive Systems Monographs (COSMOS), vol. 11. Springer, Berlin (2011)
23. Brugali, D., Scandurra, P.: Component-Based Robotic Engineering (Part I). IEEE Robotics and Automation Magazine pp. 84–96 (December 2009)
24. Brugali, D., Shakhimardanov, A.: Component-Based Robotic Engineering (Part II). IEEE Robotics and Automation Magazine pp. 100–112 (March 2010)
25. Akinade, A., Barros, D., Vernon, D.: Biological motion aids gestural communication by humanoid social robots. International Journal of Humanoid Robotics 22(02) (2025)
26. Faconti, D., Colledanchise, M.: BehaviorTree.CPP library documentation. https://www.behaviortree.dev (2018)
27. Faconti, D.: Behaviortree.CPP Library. https://github.com/BehaviorTree/BehaviorTree.CPP (2019)
28. Ghzouli, R., Berger, T., Johnsen, E.B., Wasowski, A., Dragul, S.: Behavior trees and state machines in robotics applications. IEEE Transactions on Software Engineering 49(9), 4243 – 4267 (2023)
29. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Springer Handbook of Robotics, pp. 1371–1394 (2008)
30. Dillmann, R., Asfour, T., Do, M., Jäkel, R., Kasper, A., Azad, P., Ude, A., Schmidt-Rohr, S., Lösch, M.: Advances in robot programming by demonstration. Künstliche Intelligenz 24(4), 295–303 (2010)
31. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57, 469–483 (2009)
32. Vernon, D., Taye Tefferi, T.: CSSR4Africa Deliverable D6.1 User Case Implementation. https://cssr4africa.github.io/deliverables/CSSR4Africa_Deliverable_D6.1.pdf (2025)
33. Bruno, B., Recchiuto, C.T., Papadopoulos, I., Saffiotti, A., Koulouglioti, C., Menicatti, R., Mastrogiovanni, F., Zaccaria, R., Sgorbissa, A.: Knowledge representation for culturally competent personal robots: requirements, design principles, implementation, and assessment. International Journal of Social Robotics 11(3), 515–538 (2019)
34. Beetz, M., Kazhoyan, G., Vernon, D.: Robot manipulation in everyday activities with the CRAM 2.0 cognitive architecture and generalized action plans. Cognitive Systems Research 92(101375) (September 2025)
35. The CSSR4Africa Software Repository: https://github.com/cssr4africa/cssr4africa
36. Intel RealSense Depth Camera: https://www.intelrealsense.com/depth-camera-d435i

37. Poli, R.: A note on the difference between complicated and complex social systems. Cadmus 2(1) (2013)
38. Culturally Sensitive Social Robotics for Africa (CSSR4Africa): http://www.cssr4africa.org
39. Carpinella, C.M., Wyman, A.B., Perez, M.A., Stroessner, S.J.: The robotic social attributes scale (RoSAS): Development and validation. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 254–262. HRI '17, Association for Computing Machinery, New York, NY, USA (2017)
40. United Nations: Sustainable development goals. https://sdgs.un.org/goals (2015)
41. Mai, V., Vanderborght, B., Haidegger, T., Khamis, A., Bhargava, N., Boesl, D., Gabriels, K., Jacobs, A., Moon, A.J., Murphy, R., Nakauchi, Y., Prestes, E., Bhavani, R.R., Vinuesa, R., Mörch, C.M.: The role of robotics in achieving the United Nations sustainable development goals – the experts' meeting at the 2021 IEEE/RSJ IROS workshop. IEEE Robotics & Automation Magazine 29(1), 92–107 (2022)
42. Vernon, D.: Robotics and artificial intelligence in Africa. IEEE Robotics & Automation Magazine 26(4), 131–135 (2019)
43. Vernon, D.: Robotics in Africa is trending upward and has a bright future. Science Robotics 10(101) (2025)
44. Rose, J.: Software Innovation: eight work-style heuristics for creative software developers. Software Innovation, Dept. of Computer Science, Aalborg University (2010)
45. Alupo, C.D., Omeiza, D., Vernon, D.: Realizing the potential of AI in Africa. In: Ferreira, M.I.A., Tokhi, O. (eds.) Towards Trustworthy Artificial Intelligence Systems. Intelligent Systems, Control and Automation: Science and Engineering, Springer (2022)
46. Zantou, P., Vernon, D.: Inclusion drives sustainable development: The case of social robotics for Africa. https://cssr4africa.github.io/posters/2023_Zantou_Vernon_COMPASS.pdf (August 2023)
47. The African Engineering and Technology Network (Afretec): http://www.afretec.org/