Software Architecture to Generate Assistive Behaviors for Social Robots

Jason R. Wilson
jrw@fandm.edu
Franklin & Marshall College
Lancaster, Pennsylvania , USA

Yuqi Yang
yyang5@fandm.edu
Franklin & Marshall College
Lancaster, Pennsylvania , USA

ABSTRACT

To facilitate the design of socially assistive robots (SARs), we present an architecture to generate assistive behavior for social robots given a high-level description of the intent of the assistance. Our approach features an ontology of assistive intents, a hierarchical task network planner, and robot middleware. We demonstrate the behaviors on two robot platforms and compare the behaviors. While many of the behaviors are similar, challenges remain in generating behaviors that will be presented consistently across multiple platforms.

CCS CONCEPTS

• **Human-centered computing** → *Interactive systems and tools*; • **Computing methodologies** → Intelligent agents; *Robotic planning*.

KEYWORDS

socially assistive robot, behavior generation, HTN planning, ontology, software architecture

ACM Reference Format:
Jason R. Wilson and Yuqi Yang. 2024. Software Architecture to Generate Assistive Behaviors for Social Robots. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610978.3640715

1 INTRODUCTION

Socially Assistive Robots (SARs) are commonly employed in a variety of tasks. Since the assistance provided by these robots is social, the behavior of the robots is expected to be multimodal and consist of combinations of speech, gestures, emotional expressions, and more. To more easily equip a SAR with the necessary behaviors, there needs to be mechanisms for defining, selecting, and executing assistive behaviors. There is a need for an expandable set of reusable behaviors that define how they might be employed [10] and leverage experimental results and theoretical foundations in the appropriateness and effectiveness of assistive behaviors [e.g., 10, 11, 21, 23]. While researchers and designers often focus on behaviors for a single robot, a general and reusable approach needs to adopt concepts from robot architectures that are explicitly designed to be compatible with multiple robot platforms [12, 15].

Our goal is to develop a framework affords paths for easy behavior authoring for various tasks and may be deployed to different robots. Such a framework defines essential components of assistive behavior for an assortment of tasks and social settings. We introduce here an open-source layered architecture for mulitmodal, cross-platform behavior generation for SARs. This architecture includes an ontology defining SAR behavior concepts, a planning model for selecting and generating behaviors, and middleware for executing the behaviors on the Misty and NAO robots. We demonstrate the capabilities of this architecture by examining the variability of the behaviors that can be generated and comparing the expression of behaviors across robot platforms.

2 BACKGROUND

Previously, researchers have proposed *strategies*, *communication acts*, and *abstract action categories* to describe the types of behavior a social robot needs to be able to exhibit. A strategy describes a high-level behavior to accomplish goals such as initiate contact, greeting, acquaint with user, loosen up, and charge battery [17]. Many of these describe purely social interactions, whereas other researchers focus entirely on task-based communications, such as the communicative acts to propose a task or plan and announce task status [2, 12]. Strictly task-based communication may be well-suited for collaborative tasks, but social assistance requires a combination of social and task-based communications. Most similar to our approach is the abstract action categories proposed in [1]. Similar to our ontology, they relate their categories of instructions, promises, feedback, disclosures, and inquiries to illocutionary speech acts [16].

To adapt to how much help a user needs, SAR behavior can apply the concepts of graded cueing [1, 6] or levels of assistance [22]. Both approaches are adapted from occupational therapy and are designed to support the autonomy of the user by gradually increasing how much assistance the robot provides.

Behavior trees are commonly used to define, control, and execute robot behaviors that are modular and reactive [3]. A behavior tree can define a decomposition of behaviors into a sequence of steps and the conditions for selecting a particular behavior. Behavior trees are used broadly across robotics and have recently been used to generate social [17] and explanatory [7] behaviors. Since behavior trees are designed to be reactive, they have limited ability to plan ahead. The sequences we generate are more deliberative than reactive and can require careful planning to consider the appropriate combination of behaviors. Concurrently, our planning-based approach maintains the modularity available in behavior trees.

3 ARCHITECTURE

The architecture supports assistance generation based on processing a simple goal, characterized as a task of communicative intent. Provided with the assistive intent and the robot's understanding of the world, the components in the architecture (see Fig. 1a) handle the planning and execution of the robot's behaviors. In designing this architecture, our goals were the following:

(1) **High-level language:** provide a more abstract way of describing a robot's behavior that is independent of the task or robot.
(2) **Extendable:** allow for assistance for any task to be defined.
(3) **Reusable:** define behaviors that may be used in most tasks.
(4) **Variability:** allow for a large number of ways in which assistive behavior may be expressed.
(5) **Interoperable:** integrate with multiple robot platforms.

While our architecture is only designed for generating behaviors, it is an incremental aspect in a larger robot system that incorporates components that recognize environmental cues such as the user's task or their emotions, gaze, rapport, etc. Once there is a description of the type of assistance the robot needs to provide, then our architecture will find an appropriate behavior to express the intended assistance. Similarly, other robot systems separate the robot's actions and behaviors from the perceptive components and use components to decide on the goal or strategy of the robot's behavior [12, 15, 17]. Our architecture is designed to integrate with these decision components.

The architecture consists of an ontology describing the essential assistive intents, an HTN model and planner for planning the robot's behavior, and middleware to execute the behavior on the robot. The planner generates a sequence of actions for the robot to perform, which are mediated by the middleware for execution on multiple robot platforms: currently demonstrated on the Misty and NAO robots. The subsequent sections describe each of these.

3.1 Ontology

The ontology (see Fig. 1b) is derived from relevant literature on normative attributes of interactive agents: conducting speech acts of various intentions; taking and releasing turns in a conversation; mitigating disagreements with politeness; and utilizing nonverbal cues to develop rapport. Therefore, to construct the behavioral system for an effective social robot, the ontology describes the types of assistive intents the robot may express, an input state with necessary properties concerning the situation, and how these external elements may affect the robot's decision to exercise appropriate behavior strategies. The ontology also defines the set of actions as building blocks of robot modalities, which are to be combined to constitute distinct behavioral expressions.

Assistive Intent: While the ontology is intended to specify assistance for a particular domain, the major structural components of the ontology are expected to be task-independent and reusable across most domains. The ontology defines eight major intents, drawing from Speech Act Theory [16], to cover different categories of pragmatic functions in social assistance. Each intent is actualized through the robot's behavior in taking turns to speak during the interaction. While turn-taking in interpersonal conversations happens implicitly, there are non-verbal cues that indicate the pre-beginning and pre-ending of an utterance [4, 9]. The ontology utilizes a three-step abstraction to model turn-taking behavior to regulate conversation dynamics [14]. A take-turn and release-turn sequence signal the rotation of the conversational floor, which are widely applicable across contexts. Moreover, in addition to turn-taking behaviors, socially oriented assistance and lower-level assistance are potentially applicable in most domains.

State Descriptor: The state contains key information that describes the situational variables concerning both the progress of the task and the state of the user: next move, rapport, affect, and other social characteristics. Through the definition of such properties, the ontology establishes a requisite for the perceptive and cognitive system to recognize important attributes, which jointly enable context-dependent adaptations.

To support the autonomy of the user while enabling them to complete the task to the fullest extent [6, 22], the robot should offer help when requested, but also when an implicit need is recognized. However, being proactive could introduce 'face-threatening acts' when invoking instruction, advice, and disagreements. Accordingly, there is a tradeoff between directness and politeness [5]. The robot should balance between minimizing discomfort and optimizing its assistance of communication according to the needs of the situation.

This is supported by the levels of help, described in details [20], range from 1 to 4, where 1 indicates only the lowest level of help to maintain the user's progress and flow, and 4 is the highest level, often requiring explicit detail and elaborated instructions on what the user needs to do. For example, when the task is under smooth progression, the robot should exert minimal influence, and may simply say "Yes, that's it" to confirm their ongoing actions. However, when the user is making frequent mistakes, the robot should start with the lowest level of instructive intent, and escalate as needed.

Robot Action: The building blocks of each behavior are the basic actions of the robot. To allow the robot to express a range of verbal and nonverbal behaviors, the ontology includes actions for the robot to speak, make simple arm gestures, and express emotions.

3.2 Planning Model

The ontology describes how the robot can behave. To get the robot to perform these behaviors, we realize the ontology as a hierarchical task network (HTN) model. In this section, we describe how the ontology maps to an HTN model and provide an overview of the HTN planner that will be released as open source.

HTN Model: An HTN model is used for planning, and it describes a set of tasks to be performed, where each task can be decomposed into a sequence of more tasks. The primitive tasks are ones that cannot be decomposed any further. For each complex task, the HTN model may describe multiple methods to do the task. Methods and primitive tasks may specify preconditions that need to hold in order for the method or primitive task to be included in the plan. Primitive tasks may also define anticipated effects.

In the HTN model, we represent Assistive Intents as high-level tasks. Each task may define different methods for communicating that intent, and methods may define preconditions using any of

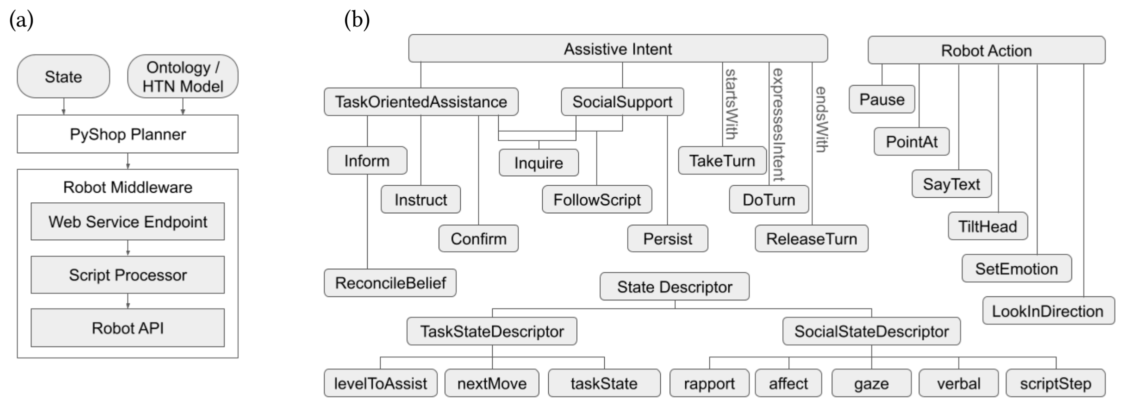

Figure 1: (a) The architecture uses a planner to find a sequence of actions for the robot to perform to express the given intent. (b) The ontology to describe assistive behavior of social robots. All edges represent an "isa" relationship unless labeled otherwise.

the State Descriptors from the ontology. The most common precondition is the levelToAssist descriptor, which allows each intent to be expressed differently based on how much assistance the robot should provide. Preconditions are represented in predicate calculus, and a condition must be found in the State given to the planner for the preconditions to be met.

Since each Assistive Intent follows a typical turn-taking pattern, each method for an intent has a sequence of three subtasks corresponding with the Take Turn, Do Turn, and Release Turn behaviors. The Do Turn subtask is the one primarily responsible for communicating the intended assistance. Since the exact nature of the assistance often needs to be specific to the domain, the methods for Do Turn are expected to need to be defined for each domain. The Take Turn and Release Turn tasks have multiple methods that change the robot's gaze direction or emotional expression. Since these behaviors tend to be purely social, these behaviors are expected to be reusable for any domain in which a SAR is helping.

PyShop Planner: The PyShop planner produces a sequence of actions for the robot to perform based on the HTN model and a description of the state. PyShop, which is a derivative of Pyhop [13], is an HTN planner written in Python, which allows for easy integration into many systems. An important enhancement is the supported format of the HTN model. Where Pyhop relies on Python functions to implement actions and methods, PyShop uses a declarative representation of the HTN model. The core algorithm remains nearly identical to Pyhop, but PyShop has a parser to read in the HTN model encoded in HDDL [8]. Using this format provides an established format as a human readable interface to the architecture.

3.3 Robot Middleware

The robot middleware provides a mechanism for communicating with a specific robot platform and executing the plan generated by the planner. Each robot platform requires its own instantiation of the middleware, but there exists a single common interface for communicating with a robot.

Common Interface: The middleware is always designed to receive a plan in the form of an *Action Script*, which is a translation of the plan from PyShop into the following JSON syntax.

```
{
  "intent": <intent>,
  "description": <any text>,
  "actionList":
     [{
        "name": <action name>,
        "args": [ list, of, argument, values ]
      },
      {
        "name": <action name>,
        "args": [ list, of, argument, values ]
      }, ...
     ]
}
```

The `actionList` is processed by the ScriptProcessor, and the `intent` and `description` fields are designed for traceability, debugging, and documentation.

Middleware Design. Each instantiation of the middleware has the same design consisting of two components. As shown in Fig. 1a, the middleware consists of a web service endpoint to receive the action script, a script processor to iterate through the Action Script and execute each action one at a time, and a connection to the robot API. Since all robot actions are implemented as non-blocking functions, parallel actions can be simulated. The Pause action then waits for an action to complete before continuing the behavior.

On the Misty, all of the middleware runs onboard as a Skill. The Skill provides an event trigger, which may be triggered via a web service. When the event is received, the Action Script is handled by a Script Processor that directly calls Misty API.

For the NAO, most of the middleware is not run on the robot. A Python web service provides an endpoint to receive the Action Script. That application also implements the Script Processor. To execute each action in the Action Script, the Script Processor uses NAOqi to manage the connection to the robot and make API calls.

4 DEMONSTRATION

To demonstrate our functionality, we look at the variability of the behaviors possible and the differences across robot platforms. We focus on how a social robot would assist a user in completing a tangram puzzle. A tangram is a set of seven pieces that can be

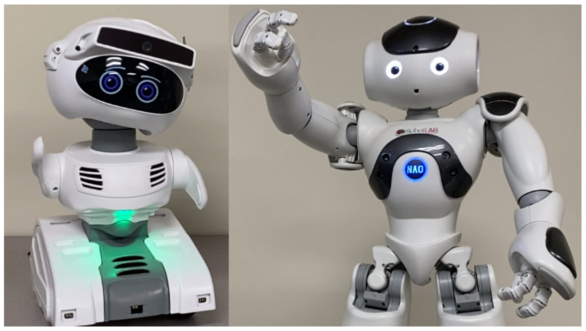
Figure 2: The Misty (left) and NAO (right) robots perform a behavior to suggest the user rotates a piece.

assembled in a variety of way to resemble shapes like a house, rabbit, or sail boat. The shape being assembled does not matter since the robot's assistance helps the user move, rotate, flip, etc. For this task, we developed an HTN model consisting of 121 methods on 32 tasks. Given combinations of assistive intent, level to assist, rapport, affect, and gaze, over 1,500 unique behaviors can be generated.

In this proof of interoperability of the behavior design, we examine six behaviors performed on two types of robots (see video[1]). The set of behaviors includes five assistive intents and three levels of assistance. While this does not cover each feature in the ontology, it is sufficient to allow us to begin to identify some of the similarities and differences in the delivery of assistive behavior across platforms. Overall, the execution across both robots appears to be fairly similar. There are two areas with more significant differences:

- The most noticeable difference is in the **emotional expression**. The eye colors do not express the same emotions as Misty's face. Even if more ideal colors were chosen, the face of the Misty has more expressive capability.
- The Misty does a **head tilt**, which the NAO is not capable of doing. In most cases, this difference appears to be minor, but the gesture for rotating the piece is an exception (see Fig. 2). The head tilt with Misty provides a more complete gesture that is coordinated with the arm movements.

5 DISCUSSION

We show that our architecture consisting of an ontology, HTN model, planner, and robot middleware provides a large degree of variability in the behaviors that may be expressed while also being able to execute on two robot platforms. While in many cases the behaviors across both robots appeared similar, the differences suggest areas for improvement. Changing the eye color of the NAO was intended to match the expression of the Misty in that only the eyes were changing. However, the Misty face is far more expressive, and portraying similar emotions may require using additional modalities. The NAO can show emotions through a combination of voice, body movement, body pose and gesture [19].

Arm gestures by the robots appeared to be similar, but no behaviors use any lateral arm movement. Since the behaviors were initially designed for the Misty, this type of movement was not initially considered. In some cases, rotating the Misty's body may allow it to simulate the lateral arm movement of the NAO.

[1]https://youtu.be/YDRcyy3Wsvc

Similar to [12], our approach currently features only deliberative behaviors that are planned out to accomplish an explicit intent. Our architecture is designed to integrate with systems that have a component to decide the robot's assistive intent. For a natural and effective interaction, it will be necessary to support reactive behaviors that are more closely connected to perceptions and do not require decision-making, such as gaze following and backchanneling. As we extend our architecture for these behaviors, we will draw from recent approaches that have combined deliberative and reactive behaviors [17].

The ontology is expected to continue to develop. Given that the assistive intents cover most speech acts from the literature, we do not expect to be more top-level types. However, more specific examples, similar to the Reconcile Belief intent may be useful. Where the ontology is most expected to grow is in the robot actions and state descriptors. For example, we have focused on stationary social robots, and thus the ontology does not account for moving to locations or managing proxemics.

The behaviors for the tangram task were designed to be effective in helping complete the task, support the autonomy of the user, and develop human-robot rapport [23]. For example, there is a Take Turn behavior specific to the precondition that the user and robot currently have low rapport. Based on literature on developing human-human rapport [18], this behavior shifts the robot's gaze to the user. While considerable work remains to validate the behaviors, this architecture can play a critical role in designing controlled experiments to study effects on the user. The HTN model can ensure an autonomous robot consistently shifts its gaze or never does, or an interface for a wizard-of-oz experiment may directly call our architecture with conditions to control the robot's gaze.

5.1 Future work

In addition to extending the ontology and developing validated, reusable behaviors, another important area of development will be expanding the range of social robot platforms supported. Our goal is to continue being able to define the assistive behaviors at a high level, invariant to the capabilities of each platform. As such, we intentionally do not support any conditioning of behaviors based on the type of robot. Then, the challenge going forward, as we develop support for more platforms (including integrations with ROS), is to identify commonalities across platforms while also finding creative solutions to provide the intended behavior.

6 CONCLUSION

We introduce an ontology defining SAR behavior concepts, a planning model for selecting and generating behaviors, and middleware for executing the behaviors on the Misty and NAO robots. This architecture may be used to ease experimentation on effects of social robot behaviors, can facilitate wizard-of-oz operation of SARs, and and may be integrated with large robot systems that have perceptive and decision-making capabilities.

ACKNOWLEDGMENTS

A special thanks to Nhi Phan and Robi Jenik for their work in implementing the middleware for the Misty and NAO robots, respectively.

REFERENCES

[1] Caitlyn Clabaugh, Kartik Mahajan, Shomik Jain, Roxanna Pakkar, David Becerra, Zhonghao Shi, Eric Deng, Rhianna Lee, Gisele Ragusa, and Maja Matarić. 2019. Long-term personalization of an in-home socially assistive robot for children with autism spectrum disorders. *Frontiers in Robotics and AI* 6 (2019), 110.

[2] Aurélie Clodic, Rachid Alami, Vincent Montreuil, Shuyin Li, Britta Wrede, and Agnes Swadzba. 2007. A study of interaction between dialog and decision for human-robot collaborative task achievement. In *RO-MAN 2007-The 16th IEEE international symposium on robot and human interactive communication*. IEEE, 913–918.

[3] Michele Colledanchise and Petter Ögren. 2018. *Behavior trees in robotics and AI: An introduction*. CRC Press.

[4] Starkey Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of personality and social psychology* 23, 2 (1972), 283.

[5] Daena J Goldsmith. 2007. Brown and Levinson's politeness theory. *Explaining communication: Contemporary theories and exemplars* (2007), 219–236.

[6] Jillian Greczek, Edward Kaszubski, Amin Atrash, and Maja Matarić. 2014. Graded cueing feedback in robot-mediated imitation practice for children with autism spectrum disorders. In *The 23rd IEEE international symposium on robot and human interactive communication*. IEEE, 561–566.

[7] Zhao Han, Daniel Giger, Jordan Allspaw, Michael S Lee, Henny Admoni, and Holly A Yanco. 2021. Building the foundation of robot explanation generation using behavior trees. *ACM Transactions on Human-Robot Interaction (THRI)* 10, 3 (2021), 1–31.

[8] Daniel Höller, Gregor Behnke, Pascal Bercher, Susanne Biundo, Humbert Fiorino, Damien Pellier, and Ron Alford. 2020. HDDL: An extension to PDDL for expressing hierarchical planning problems. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 9883–9891.

[9] Judith Holler and Kobin H Kendrick. 2015. Unaddressed participants' gaze in multi-person interaction: optimizing recipiency. *Frontiers in psychology* 6, 98 (2015), 1–14.

[10] Chien-Ming Huang and Bilge Mutlu. 2012. Robot behavior toolkit: generating effective social behaviors for robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. 25–32.

[11] James Kennedy, Paul Baxter, and Tony Belpaeme. 2017. The impact of robot tutor nonverbal social behavior on child learning. *Frontiers in ICT* 4 (2017), 6.

[12] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. 2017. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence* 247 (2017), 45–69.

[13] Dana Nau. 2013. Game applications of HTN planning with state variables. In *Planning in Games: Papers from the ICAPS Workshop*.

[14] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*. Elsevier, 7–55.

[15] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. 2019. An overview of the distributed integrated cognition affect and reflection diarc architecture. *Cognitive architectures* (2019), 165–193.

[16] John R Searle. 1975. A taxonomy of illocutionary acts. (1975).

[17] Sonja Stange, Teena Hassan, Florian Schröder, Jacqueline Konkol, and Stefan Kopp. 2022. Self-explaining social robots: An explainable behavior generation architecture for human-robot interaction. *Frontiers in Artificial Intelligence* 5 (2022), 87.

[18] Linda Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological inquiry* 1, 4 (1990), 285–293.

[19] Myrthe Tielman, Mark Neerincx, John-Jules Meyer, and Rosemarijn Looije. 2014. Adaptive emotional expression in robot-child interaction. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. 407–414.

[20] Jason R Wilson, Phyo Thuta Aung, and Isabelle Boucher. 2022. When to Help? A Multimodal Architecture for Recognizing When a User Needs Help from a Social Robot. In *International Conference on Social Robotics*. Springer, 253–266.

[21] Jason R Wilson, Nah Young Lee, Annie Saechao, Sharon Hershenson, Matthias Scheutz, and Linda Tickle-Degnen. 2017. Hand gestures and verbal acknowledgments improve human-robot rapport. In *Social Robotics: 9th International Conference, ICSR 2017, Tsukuba, Japan, November 22-24, 2017, Proceedings 9*. Springer, 334–344.

[22] Jason R Wilson, Linda Tickle-Degnen, and Matthias Scheutz. 2020. Challenges in designing a fully autonomous socially assistive robot for people with parkinson's disease. *ACM Transactions on Human-Robot Interaction (THRI)* 9, 3 (2020), 1–31.

[23] Tracy Yang, Allison Langer, Lauren Howard, Peter J. Marshall, and Jason R. Wilson. 2023. Towards An Ontology for Generating Behaviors for Socially Assistive Robots Helping Young Children. In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction*.