# Cognitive Architectural Control for Free-Flying Robots on the Lunar Orbital Platform-Gateway

**Conference Paper** · November 2020

**5 authors**, including:

Sayanti Roy
Colorado School of Mines
**21** PUBLICATIONS **26** CITATIONS
SEE PROFILE

Alexandra Bejarano
Colorado School of Mines
**10** PUBLICATIONS **11** CITATIONS
SEE PROFILE


Tom Williams
Colorado School of Mines
**128** PUBLICATIONS **1,076** CITATIONS
SEE PROFILE

# Cognitive Architectural Control for Free-Flying Robots on the Lunar Orbital Platform-Gateway

Jon Emmanuel Serrano
*MIRRORLab*
*Colorado School of Mines*
Golden, CO USA
serrano@mines.edu

Shania Jo Runningrabbit
*MIRRORLab*
*Colorado School of Mines*
Golden, CO USA
srunningrabbit@mines.edu

Sayanti Roy
*MIRRORLab*
*Colorado School of Mines*
Golden, CO USA
sayantiroy@mines.edu

Alexandra Bejarano
*MIRRORLab*
*Colorado School of Mines*
Golden, CO USA
abejarano@mines.edu

Tom Williams
*MIRRORLab*
*Colorado School of Mines*
Golden, CO USA
twilliams@mines.edu

***Abstract*—In this paper we describe a proposed integration between the DIARC robot cognitive architecture and the NASA Astrobee robot to enable goal-directed cognition and natural language control over this platform. After describing the capabilities of both architectures, we describe how the architectures can be integrated and provide examples of capabilities that would be enabled through this integration.**

***Index Terms*—Cognitive Architectures, Space Robotics, Astrobee**

## I. INTRODUCTION

The Lunar Orbital Platform-Gateway will serve as a staging point for crewed and uncrewed missions to the Moon, Mars, and beyond [4]. While the Gateway will sustain human crews for small periods of time, it will be primarily staffed by autonomous caretaker robots like the free-flying Astrobee platform [11]: the Gateway's sole residents during quiescent (uncrewed) periods [3]. This creates a unique human-technical system comprised of two categories of human teammates: ground control workers permanently stationed on earth and astronauts that may transition over time between work on Earth, the Gateway, the Moon, and Mars; and three types of machine teammates: robot workers stationed on the Gateway; robot workers stationed on the Moon and Mars; and the Gateway itself.

This distributed multi-robot system comprised of both embodied actors (e.g., robots) and minimally embodied actors (e.g., the Gateway) must be capable of (1) autonomously allocating tasks to and coordinating computation between multiple robot bodies, and (2) accepting directives and communicating feedback to human teammates, both through standard control interfaces and through natural language. Interaction of the distributed multi-robot system with human teammates can occur in a variety of ways: each individual robot and the Gateway may have individual minds and one-to-one communication with human teammates, robots and the Gateway may all share the same mind (i.e. a hive mind) and line of communication, or only a single actor has one-to-one communication with human teammates as it serves to collect feedback from all other robots and distribute commands from human teammates to them (e.g., a ground control worker on Earth communicating with a local social interface that then transmits commands to non-social robot workers stationed on the Moon.)

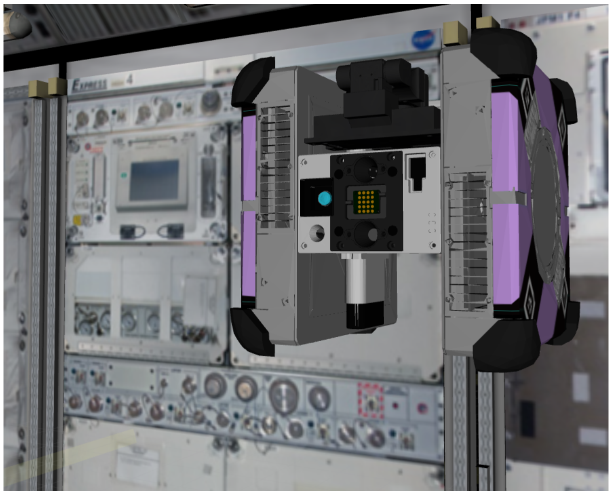

Fig. 1. Astrobee robot in the simulator.

In this paper, we argue that these capabilities are best enabled through *Robot Cognitive Architectures* such as the Distributed Integrated Affect Reflection Cognition (DIARC) architecture [10], which not only provides capabilities for goal-directed cognition and natural language interaction, but does so in a way that is naturally suited to enable interaction between multiple humans and multi-robot distributed systems. To explain how cognitive architectures could be used for control of and action with robots like the Astrobee, in this paper we will describe (1) the unique capabilities of those robots and of the DIARC architecture; (2) how the Astrobee robots can be integrated into the DIARC architecture; and (3) an example dyadic interaction that can be enabled through this integration.

## II. ASTROBEE ARCHITECTURE

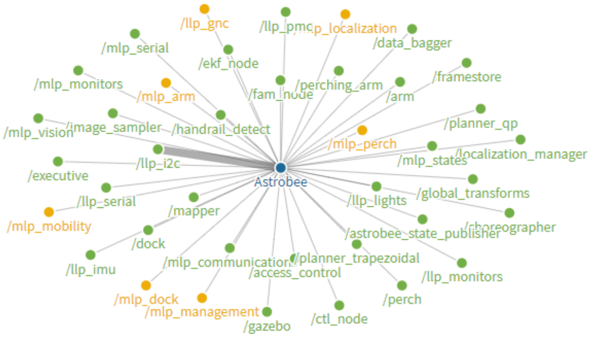

Fig. 2. Network graph of all the available Astrobee ROS nodes.

Astrobees are free-flying robots designed to work closely with astronauts and helping them with routine tasks aboard the International Space Station (ISS). The Astrobee Robot Software uses the open-source Robot Operating System (ROS) as message-passing middleware for performing vision-based localization, autonomous navigation, docking and perching, managing various sensors and actuators, and supporting user interaction via screen-based displays, light signaling, and sound. ROS enables Astrobee robots to be operated autonomously or via teleoperation, monitors timeouts and action execution progress, identifies system faults, and controls nodelet lifecycle [5]. While ROS affords Astrobee robots a wide range of capabilities, however, they currently lack natural language understanding and generation capabilities. In our research, we are working to provide such capabilities through the DIARC architecture, in order to allow astronauts to task Astrobees to execute complex tasks such as Spot Checks without requiring the use of a graphical interface.

## III. DIARC ARCHITECTURE

The Distributed, Integrated, Affect, Reflection, Cognition (DIARC) architecture is a component-based robot architecture that focuses on enabling robust goal-driven cognition and open-world spoken language interaction [10]. DIARC has previously been used for a variety of projects of relevance to or in collaboration with NASA, e.g., work on shared mental model development from [6]. Central to DIARC are a key set of language-, memory-, and action-oriented components.

Language-oriented components allow for robots to process and understand referring expressions such as definite, anaphoric and deictic expressions [13], as well as understanding and generating a variety of speech acts (including clarification requests), through DIARC's Dialogue Manager. This includes speech acts phrased as indirect speech acts [14]. For language-oriented components to allow the processing of any kind of referring expression (i.e. reference resolution), they must leverage memory-oriented components.

Memory-oriented components allow for robots to access and extract information from knowledge bases distributed across the robot architecture. This can be accomplished, for example, using either a centralized Prolog knowledge base such as DIARC's Belief Manager, or through POWER Consultants, which serve as interfaces to information stored across multiple heterogeneous knowledge bases [17], [18]. These components are in turn leveraged by action-oriented components which manage high-level goals and actions.

At the core of DIARC is its Goal Manager, which allows for goal-directed cognition, i.e., creation, satisfaction, and monitoring of logically specified one-time and maintenance goals. DIARC also provides components for physical manipulation and navigation in order to provide primitive actions for satisfying theses sorts of goals [15], but these action-oriented components are de-emphasized relative to cognition and dialogue oriented capabilities.

DIARC is implemented in the Agent Development Environment (ADE). ADE is a software Middleware that supports the development and implementation of agent architectures [8] using a distributed multi-agent system computing infrastructure (cp. [2], [12]). This framework provides dynamic, reliable, fault-recovering, remotely accessible, distributed computing and autonomic computing by treating architectural components as autonomous agents [1], [7], [9].

## IV. INTEGRATED APPROACH

By integrating the Astrobee system and DIARC robot architecture (through specific integration of the Astrobee's ROS and ADE MAS middlewares), we have produced a new robot that can both leverage the unique physical capabilities of the Astrobee Robot and the state-of-the-art linguistic capabilities of the DIARC architecture, as well as new synergistic capabilities made possible only through this integration (e.g., natural language tasking to perform unique tasks aboard the ISS).

The integration between the Astrobee software and DIARC are comprised of three types of components.

- **DIARC Components:** ADE components that only exist within the DIARC architecture, and are only aware of components implemented in the ADE middleware.
- **Astrobee Components:** Astrobee components made up of ROS nodes that sends and gets data from other nodes using the publish/subscribe model, either publishing topics or subscribing to them.
- **Dual-Citizen Components:** Components that exist within both architectures, and can communicate both with ADE components and Astrobee nodes. In our current implementation a single Dual-Citizen Component is used, but the architecture is sufficiently flexible to allow for an arbitrary number of Dual-Citizen Components to effect different points of interface between the architectures.

As shown in Figure 3 the Astrobee operates using a large number of ROS nodes. To effect our integration, a DIARC utility is used to automatically generate wrappers to connect between DIARC and ROS. For example, a wrapper component

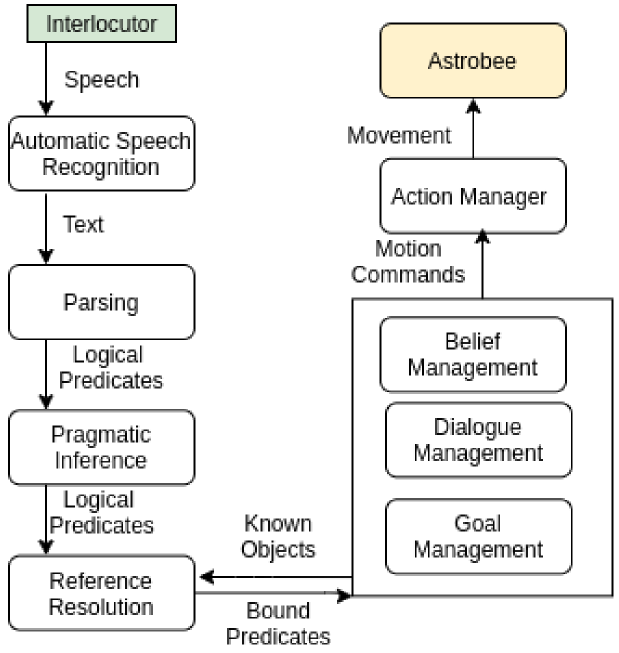

Fig. 3. Diagram of our proposed integrated Architecture with relevant components and their information flow.

is generated for the Astrobee's MLP Mobility node so that movement goals can be sent from DIARC through publishing to the topics used by that node. This also provides DIARC with access to pose representations that can be used to determine where the robot is on the station. In addition, the Astrobee has a number of unique sensors and effectors used for functionality beyond point-to-point motion, which are exposed through its ROS nodes and can be wrapped in this way. These include the Astrobee's cameras, laser pointer, touch screen, arm, speaker/microphone, and flashlights.

The Dual-Citizen component used in our integration is implemented as a Java class that extends the ADE Component interface and uses service calls to communicate with other ADE nodes. The component then imports these generated wrappers described above in order to participate in publish/subscribe communication with the wrapped ROS nodes.

By integrating Astrobee and DIARC, each can leverage the other's capabilties, resulting in new synergistic capabilities and behaviors. DIARC can leverage Astrobee's Free-flying navigational capabilities and situated spatial knowledge in order to discuss, reason about, and travel through the ISS, and the Astrobee can leverage DIARC's linguistic capabilities, to enable the Astrobee robots to travel to locations that are loosely specified; an instruction such as "I need you to survey the ship for leaks," for example, is not literally a direct command, and does not clearly specify a location, yet implies an intended command to travel around the ship, and, when paired with Astrobee's knowledge of its environment, may imply locations around the ship to which the robot should travel.

## V. EXAMPLE FUNCTIONALITY

In this section we present an example of how we envision our integrated approach playing out. While the implementation has not yet been fully verified end-to-end, each constituent piece has been verified. For an example of how a similar demonstration was implemented on the Vulcan robotic wheelchair, we direct the reader to Williams et al. [16]. In this example, the Astrobee is told "Astrobee, go to Control Panel C". After recognition, DIARC's ASR component passes this utterance to its NLP component, which performs parsing and reference resolution. This utterance is parsed into the utterance form $Statement(moveto(X))$ with supplemental semantics $controlpanelc(X)$.

At the start of this interaction, the robot's $ShortTermMemory$ and $FocusofAttention$ are both empty, and thus the robot's $LongTermMemory$ is searched for a suitable referent to bind to the variable $X$. The property $controlpanelc(X)$ can be advertised by the Astrobee component if that component is implemented as a POWER consultant and populated with knowledge of relevant objects, including their properties and locations. If Control Panel C is represented in the Astrobee component with memory trace $astrobee_5$, this trace will be bound to $X$, producing $Statement(moveto(astrobee_5))$, which is passed to DIARC's pragmatic reasoning component. This component has a rule with implicative content: $Statement(moveto(X)) \rightarrow goal(at(self, X))$, resulting in the goal $at(astrobee_5)$ being adopted. If the Astrobee Component exposes a method with the effect $at(self, X)$ using Action Effect Annotations, then the robot will identify that this method can be used to achieve $at(astrobee_5)$. If this method has access to the metric location of $astrobee_5$, then the Astrobee Component can broadcast a MLP Mobility goal to travel to that location. ROS's Mobility node can then receive this goal and initiate a new motion planning task to go to the specified coordinates.

## VI. CONCLUSIONS AND FUTURE WORK

In this brief workshop paper we have demonstrated how the DIARC architecture and the ROS-based Astrobee robot can be integrated in order to enable goal-directed natural language control over the Astrobee robot. While we have not yet evaluated the demonstration described in an end-to-end fashion, each constituent piece of the demonstration has been shown to work either in our recent work or through our past ten years of work on DIARC. In future work, we aim to complete this demonstration and produce a configuration that will allow natural language tasking of *spot checks*, as described by Bualat et al. [3].

## REFERENCES

[1] Virgil Andronache and Matthias Scheutz. Ade—an architecture development environment for virtual and robotic agents. *International Journal on Artificial Intelligence Tools*, 15(02):251–285, 2006.

[2] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Jade–a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.

[3] Maria G Bualat, Trey Smith, Ernest E Smith, Terrence Fong, and DW Wheeler. Astrobee: A new tool for iss operations. In *2018 SpaceOps Conference*, 2018.

[4] Jason C Crusan, R Marshall Smith, Douglas A Craig, Jose M Caram, John Guidi, Michele Gates, Jonathan M Krezel, and Nicole B Herrmann. Deep space gateway concept: Extending human presence into cislunar space. In *2018 IEEE Aerospace Conference*, pages 1–10. IEEE, 2018.

[5] Lorenzo Fluckiger and Brian Coltin. Astrobee robot software: Enabling mobile autonomy on the iss. 2019.

[6] Felix Gervits, Terry W Fong, and Matthias Scheutz. Shared mental models to support distributed human-robot teaming in space. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, page 5340, 2018.

[7] James Kramer and Matthias Scheutz. Ade: A framework for robust complex robotic architectures. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4576–4581. IEEE, 2006.

[8] James Kramer and Matthias Scheutz. Development environments for autonomous mobile robots: A survey. *Autonomous Robots*, 22(2):101–132, 2007.

[9] Matthias Scheutz. Ade: Steps toward a distributed development and runtime environment for complex robotic agent architectures. *Applied Artificial Intelligence*, 20(2-4):275–304, 2006.

[10] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. In Maria Isabel Aldinhas Ferreira, Joao S.Sequeira, and Rodrigo Ventura, editors, *Cognitive Architectures*. 2019.

[11] Trey Smith, Jonathan Barlow, Maria Bualat, Terrence Fong, Christopher Provencher, Hugo Sanchez, and Ernest Smith. Astrobee: A new platform for free-flying robotics on the international space station. In *Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2016.

[12] Katia Sycara, Massimo Paolucci, Martin Van Velsen, and Joseph Giampapa. The retsina mas infrastructure. *Autonomous agents and multi-agent systems*, 7(1-2):29–48, 2003.

[13] Tom Williams, Saurav Acharya, Stephanie Schreitter, and Matthias Scheutz. Situated open world reference resolution for human-robot dialogue. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 311–318. IEEE, 2016.

[14] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going beyond literal command-based instructions: Extending robotic natural language interaction capabilities. In *AAAI*, pages 1387–1393, 2015.

[15] Tom Williams, Rehj Cantrell, Gordon Briggs, Paul Schermerhorn, and Matthias Scheutz. Grounding natural language references to unvisited and hypothetical locations. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*. Citeseer, 2013.

[16] Tom Williams, Collin Johnson, Matthias Scheutz, and Benjamin Kuipers. A tale of two architectures: A dual-citizenship integration of natural language and the cognitive map. In *Proceedings of the 16th International Conference on Autonomous Agents and Multi-Agent Systems*, 2017.

[17] Tom Williams and Matthias Scheutz. Power: A domain-independent algorithm for probabilistic, open-world entity resolution. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1230–1235. IEEE, 2015.

[18] Tom Williams and Matthias Scheutz. A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. In *AAAI*, pages 3958–3965, 2016.