The Onboard Artificial Intelligence Research (OnAIR) Platform

Evana Gizzi*[1], James Marshall[1], Alan Gibson[1], Connor Firth[2], Timothy Chase Jr[1]

[1]*NASA Goddard Space Flight Center, Greenbelt, USA*
[2]*Aurora Engineering, Potomac, USA*

In this paper, we present the NASA On-Board Artificial Intelligence Research (OnAIR) Platform, a dual-use tool for rapid prototyping autonomous capabilities in space, serving as both a cognitive architecture and a software framework. OnAIR has been used for autonomous reasoning in applications that use raw data files, simulators, embodied agents, and recently in an onboard experimental flight payload, acquiring Technology Readiness Level (TRL) 7. We briefly review the OnAIR architecture and recent applications of OnAIR for diverse reasoning in autonomy projects at NASA, and close with a discussion on intended use for the public, and future work.

1 Introduction

To keep pace with the increasing development of autonomy in space, we must streamline aerospace development approaches with industry standards for Artificial Intelligence (AI) development. Rapid prototyping has become commonplace for algorithm development with the unprecedented and consistent growth of the field of AI [1]. However, this agile style of prototyping is challenging to employ in the aerospace domain due to its multidisciplinary complexity (leading to variation in software experience and literacy), the inherent inaccessibility of space and space data, and the conservatism of spaceflight development approaches [2–4]. These factors create a barrier to entry for developing and infusing autonomy into space systems.

To support both agile algorithm development and integration of AI capabilities into space systems, we present the dual-use NASA On-Board Artificial Intelligence Research (OnAIR) platform. OnAIR is both a 1) cognitive architecture (Figure **??** and 2) rapid prototyping framework (Figure 2) that was developed to enable domain-agnostic, full-stack development of autonomous systems. In this paper, we describe the OnAIR tool, its architecture, and its historical and intended use, concluding with a discussion on future work and future use by the research community (open-sourced at `https://github.com/nasa/OnAIR`).

2 Background

OnAIR was first developed under an Internal Research And Development (IRAD) grant at NASA Goddard Space Flight Center (GSFC) which focused on generating methods for onboard fault diagnosis [5]. Researchers continued to organically reuse OnAIR as a framework for various diverse AI research applications. Project-specific needs led to unexpected and significant development time devoted to architectural/dataflow tasks. Consequently, OnAIR has also found suitable use internally as a generalized cognitive architecture to support reuse and agile development. To the best of our knowledge, OnAIR is the first cognitive architecture and rapid prototyping autonomy pipeline for aerospace applications.

OnAIR was designed for flexible use while providing the structure needed for modeling human cognition, which includes information discrimination and abstraction to emulate the function of the human brain. OnAIR is not intended to replicate the publish-subscribe architectures that are common in AI and aerospace applications, such as the Robot Operating System (ROS) [6, 7] and the Core Flight System (cFS) [8, 9]. OnAIR is intended to interface with publish-subscribe systems to streamline AI algorithm development, and active development to interface with those systems is ongoing.

2.1 OnAIR as Cognitive Architecture

Within the field of artificial intelligence (AI), cognitive architectures are used as models for human reasoning, consistent with a neurological and cognitive function [10]. While *traditional cognitive architectures* focus on underlying “rule-based” cognitive function (ACT-R or SOAR) [11, 12], more recently preferred *connectionist cognitive architectures* draw structural inspiration from neuroscience, considering both functional cognition and its underlying neurobiological basis [13]. OnAIR takes a hybrid approach, accounting for both rule-based and emergent cognitive outputs.

---

[*] Corresponding author. E-Mail: Evana.Gizzi@nasa.gov

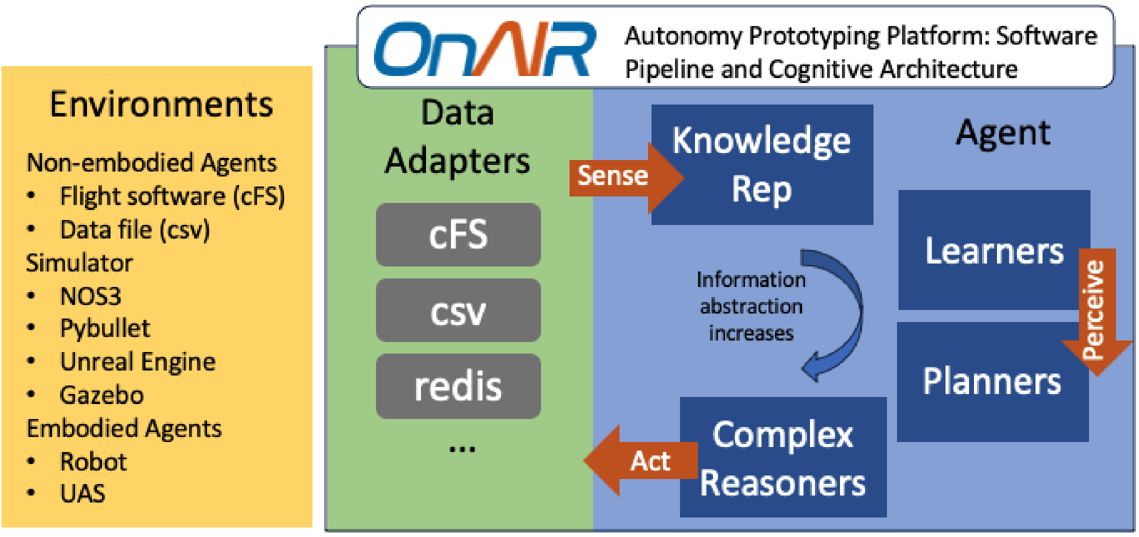

Figure 1: OnAIR is a versatile autonomy framework enabling the infusion of artificial intelligence algorithms into aerospace flight systems, facilitating rapid prototyping with diverse data environments and serving as a cognitive architecture for rational agents.

The main abstraction levels of OnAIR are housed within four main plug-in interfaces: Knowledge Representations, Learners, Planners, and Complex Reasoners. Table 1 describes each interface as a brain analog, drawing from major researchers in the fields of psychology, neuroscience, philosophy, and computer science, which collectively comprise the conglomerate of AI.

2.2 OnAIR as a Prototyping Pipeline

OnAIR provides a plug-and-play architecture that allows user code (written and received as one of the four plugin types listed above) to receive low-level data (from an external source), or high-level data (from previous plugins within the pipeline); see Figure 2. OnAIR's lightweight implementation and the self-contained nature of its data make it suitable for rapid deployment and co-development across a team of systems. It has seen significant use in NASA internal demonstrations, running onboard mobile platforms like robots and drones. The modular nature of architecture is well-suited to rapid prototyping tasks, especially to the integration of separately-developed tools. Data adapters are provided to ingest data from static sources like comma-separated values (.csv files) or dynamic data sources like Redis servers or NASA's cFS.

3 How to Use OnAIR

In this section, we provide a high-level overview of OnAIR's use from a developer perspective. Figure 2 depicts a high-level architecture diagram of the OnAIR architecture. OnAIR is built around extensibility: algorithms are implemented as user plugins in Python that are inserted into a data pipeline. The configuration of OnAIR is dictated by a configuration file that specifies the data adapter, metadata location, and plugins to load. Data is ingested by Data Adapter objects, with three types provided by default: csv, Redis, and cFS.

Each data source requires a metadata file to describe characteristics of the data such as human-readable labels, message source (in the case of live data), conversion factors, and feasibility tests. In the case of live data adapters, the metadata file specifies the Redis channel or cFS message ID. The adapter collects messages in a double frame buffer with a thread and provides that buffer to the main OnAIR execution loop upon request (at which point the write and read buffers are flipped so that plugins are not reading from the same buffer that new data is being written to).

Each new packet of data (a line from a .csv, new message from Redis, etc) is ingested as a new data frame in OnAIR. This data frame is then passed to each plugin sequentially, allowing each plugin to modify, append, or create data to add to the frame before passing it to the next plugin. From low-level data, Knowledge Representation plugins synthesize high-level knowledge and Learners update their internal knowledge

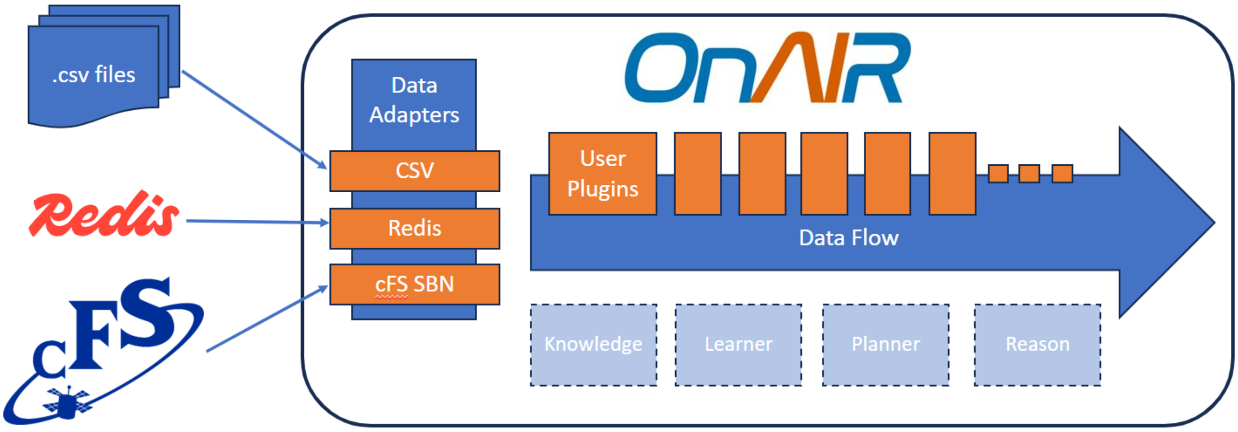

Figure 2: The OnAIR architecture and dataflow. Data is ingested from sources such as csv files or live sources such as cFS by Data Adapters. Frames of data are then propagated through a pipeline of user-created plugins. The plugins are instantiations of four types: Knowledge, Learner, Planner, or Complex Reasoner constructs.

bases. High-level data, output from both Knowledge Representation and Learner plugins, is used by Planners and Complex Reasoners to make decisions based on that knowledge.

Interfacing with cFS is complicated by the fact that cFS is not designed to allow external processes access to its internal message bus (unlike Redis). Additional open source tools are required to gain access: the Software Bus Network (SBN) (`https://github.com/nasa/sbn`) is configured to publish cFS messages to a local UDP socket. The OnAIR data adapter then uses the Software Bus Network Client (`https://github.com/nasa/sbn-client`) to connect to SBN.

4 Results

To date, OnAIR has been used to support reasoning in many diverse autonomy applications at NASA GSFC, and in concert with our collaborators. As mentioned, OnAIR was first used as an architecture for resilience research. During this time, OnAIR leveraged csv data corresponding to sounding rocket telemetry, data from Kerbal Space Program for education and research use, Dellingr [14] satellite telemetry, and basic emulated toy data. OnAIR was eventually used as a cognitive architecture to support NASA GSFC's Distributed Systems Mission (DSM) internal research and development project, led by Engineering and Technology Director Chief Technologist Michael Johnson. During this time, OnAIR was used for all reasoning, which included resilience, opportunistic science discovery, basic sensing, planning, and scheduling, among others. OnAIR was used with collected real-world science data [15–17], with **simulated data** (small satel-

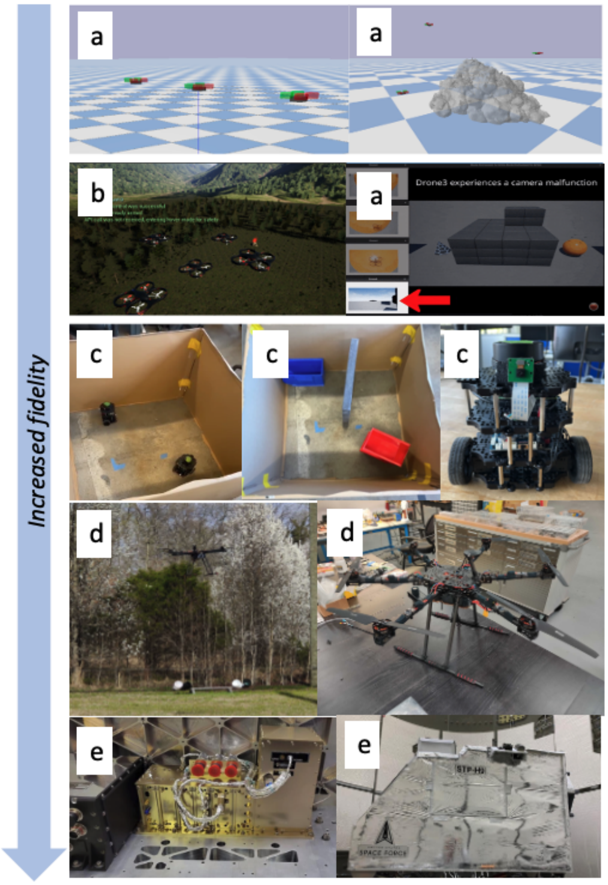

Figure 3: Applications of OnAIR for autonomy projects at NASA GSFC encompass specialized simulations developed in Pybullet (a) and Unreal Engine (b), as well as embodiments in Turtlebot (c), GSFC sUAS (d), and SCENIC (e) in-flight platforms.

	Knowledge Rep	Planners and Learners	Complex Reasoners
Description	Takes in low-level data (spacecraft telemetry) and creates needed information representations for the brain	Takes in low-level data needed for AI learning and planning algorithms and outputs labels (learners). Takes in high-level data to determine possible actions (planners)	Traverses over all synthesized high-level outputs to make a combined, logical, informed decision
Neurosymbolic Representation (input → output)	Sub-symbolic → Symbolic	Sub-symbolic (learners) or Symbolic (planners) → Symbolic	Symbolic → Symbolic
Neurobiological Premise	Neurons firings to output neural circuitry patterns	Input circuitry firings in a specific brain region for specialized cognitive function. Example: Amygdala (emotional response), Hippocampus (memory recall), ...	Outputted behavior. Example: Prefrontal Cortex (process a stimulus and remember that it is not a threat – proceed with action)
Freud Levels of the Mind	Unconscious	Pre-Conscious	Conscious
Russel & Norvig Rational Agent	Sense	Perceive	Act

Table 1: Each OnAIR plug-in interface can be mapped to philosophical, psychological and neurological brain functions.

lite flights using the NASA Operational Small Satellite Simulator [18, 19]), with publicly available robot simulators (PyBullet, Gazebo, JMAVSim, Unreal Engine, AirSim), and on **embodied agents** (Turtlebots and GSFC custom built small unmanned aerial vehicles (sUAS), ModalAI Starlings). OnAIR will be used to support reasoning in the upcoming NAMASTE (Network for Assessment of Methane Activity in Space and Terrestrial Environments, principal investigator: Dr. Mahmooda Sultana) **field campaign** to take place in Fairbanks Alaska, USA. OnAIR was recently **run onboard** to support a basic Kalman-filter experiment on the STP-H9-SCENIC payload [20] under principal investigator Dr. James Marshall to test onboard tractability, bringing OnAIR to TRL 7.

5 Discussion

In this paper, we reviewed the OnAIR tool, its architecture, and described recent applications of OnAIR for onboard reasoning for autonomy projects at NASA. OnAIR has been used in a highly interdisciplinary and application-agnostic manner. We aim for researchers to utilize OnAIR for their autonomy experiments, both within the aerospace domain and in other fields. OnAIR was intentionally open-sourced, and created with ease of use in mind for this sought widespread use. It is our goal to reduce the barrier to entry for increased partnership across academic, private, and public sectors, and we believe the creation and use of OnAIR is a promising first step. In the future, we plan on using OnAIR at NASA for future distributed systems missions research, across applications in Earth Science, Planetary Science, Astrophysics, and Heliophysics to support the most recently available decadal survey goals which seek distributed passive and active observatories [21–24].

6 Acknowledgements

We would like to acknowledge the following OnAIR contributors: Nicholas Pellegrino, Jeffrey St. Jean, Hayley Owens, Gabriel Rasskin, Charles Zhao, James Staley, and Ibrahim Haroon.

References

1. Perrault, R. & Clark, J. Artificial Intelligence Index Report 2024 (2024).
2. Ribeiro, J. E. F., Silva, J. G. & Aguiar, A. Weaving Agility in Safety-Critical Software Development for Aerospace: from Concerns to Opportunities. *IEEE Access* (2024).
3. VanderLeest, S. H. & Buter, A. *Escape the waterfall: Agile for aerospace* in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference* (2009), 6–D.

4. Monteiro, J. P., Rocha, R. M., Silva, A., Afonso, R. & Ramos, N. Integration and verification approach of ISTSat-1 CubeSat. *Aerospace* **6,** 131 (2019).
5. Gizzi, E. *et al. Autonomous System-Level Fault Diagnosis in Satellites using Housekeeping Telemetry* 2022.
6. Koubâa, A. *et al. Robot Operating System (ROS).* (Springer, 2017).
7. Macenski, S., Foote, T., Gerkey, B., Lalancette, C. & Woodall, W. Robot Operating System 2: Design, architecture, and uses in the wild. *Science robotics* **7,** eabm6074 (2022).
8. McComas, D. *NASA/GSFC's Flight Software Core Flight System* in *Flight Software WorkshopFlight Workshop* (2012).
9. McComas, D., Wilmot, J. & Cudmore, A. *The core flight system (cFS) community: Providing low cost solutions for small spacecraft* in *Annual AIAA/USU Conference on Small Satellites* (2016).
10. Russell S., N. P. Artificial Intelligence: A Modern Approach, 935 (2011).
11. Ritter, F. E., Tehranchi, F. & Oury, J. D. ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science* **10,** e1488 (2019).
12. Jones, R. M., Lebiere, C. & Crossman, J. A. *Comparing modeling idioms in ACT-R and Soar* in *Proceedings of the 8th international conference on cognitive modeling* (2007), 49–54.
13. Petersen, S. E. & Sporns, O. Brain networks and cognitive architectures. *Neuron* **88,** 207–219 (2015).
14. Clagett, C. *et al.* Dellingr: NASA Goddard Space Flight Center's First 6U Spacecraft (2017).
15. Theiling, B. P. *et al. A science-focused artificial intelligence (AI) responding in real-time to new information: Capability demonstration for ocean world missions* in *2024 Astrobiology Science Conference* (2024).
16. Firth, C., Greenlee, L., Barrie, A., Theiling, B. & Clough, L. *The Module for Event Driven Operations on Spacecraft: Applying MEDOS to Science Data* in *23rd Meeting of the American Geophysical Union (AGU)* (2023).
17. Williams, C., McKinney, L., Clough, L. A., Theiling, B. & McKinney, B. *Autonomous Science: Simulated Solar System Mission to Enceladus, Icy Ocean Moon of Saturn* in *The University of Tulsa Student Research Colloquium* (2024).
18. Grubb, M., Morris, J., Zemerick, S. & Lucas, J. Nasa operational simulator for small satellites (nos3): Tools for software-based validation and verification of small satellites (2016).
19. Geletko, D. M. *et al.* NASA operational simulator for small satellites (NOS3): the STF-1 cubesat case study. *arXiv preprint arXiv:1901.07583* (2019).
20. Geist, A. *et al.* NASA SpaceCube Next-Generation Artificial-Intelligence Computing for STP-H9-SCENIC on ISS (2023).
21. Of Sciences, N. A. *et al. Thriving on our changing planet: A decadal strategy for Earth observation from space* (National Academies Press, 2019).
22. Of Sciences, N. A. *et al. Visions into Voyages for Planetary Science in the Decade 2013-2022: A Midterm Review* (National Academies Press, 2018).
23. National Academies of Sciences, E., on Engineering, M. D., on Physics, P. S. S. S. B. B., for a Decadal Survey on Astronomy, A. C. & (Astro2020), A. 2. *Pathways to Discovery in Astronomy and Astrophysics for the 2020s* (National Academies Press, 2023).
24. Council, N. R. *et al. Solar and space physics: A science for a technological society* (National Academies Press, 2013).