Artificial Thought is Required for Sustained Autonomy
Lunar Surface Innovation Consortium (LSIC) Spring Meeting 2023

Fernando Figueroa
NASA Stennis Space Center, Autonomous Systems Laboratory (fernando.figueroa@nasa.gov)

Introduction: Current literature primarily addresses what is referred to as "Systems Thinking." Dr. Marie Morganelli from Southern New Hampshire University states that *"Systems thinking is a holistic way to investigate factors and interactions that could contribute to a possible outcome [1]. A mindset more than a prescribed practice, systems thinking provides an understanding of how individuals can work together in different types of teams and through that understanding, create the best possible processes to accomplish just about any-thing."* So, a "Thinking System" is a system that is capable of "systems thinking," as it should be able to " … create [utilize] the best possible processes, and possess the [intelligence] to accomplish just about anything." To achieve this capability, hu-man-like thinking is required. A truly autonomous system must be one that is capable of human-like thinking. Sustained autonomy requires "Thinking Autonomy" (TA) that is enabled by a "Thinking System."

What needs to be done: The autonomy R&D community must address this functionality and its implementation, including software capabilities needed. The NASA Platform for Autonomous Systems (NPAS) foundationally encompasses a software architecture and core infrastructure to enable TA. NPAS is a long-term project that has been funded by NASA's Advanced Exploration Systems (AES) and currently receives continuing support from the Mars Campaign Office [2] (https://techport.nasa.gov/view/94884).

Thinking systems will enable a fundamental change on how Artificial Intelligence (AI) and autonomy are implemented. Thinking Systems will change from a scripted or "brute force" approach that results in one-time implementations that are minimally intelligent or autonomous, to a "thinking" approach where implementations evolve continuously and enable powerful intelligence and autonomy on systems of high complexity as well as on systems-of-systems.

A Paradigm Solution: To achieve "Artificial Thought" (AT) a sensible paradigm is to try to achieve artificial human-like thinking. This requires an architecture encompassing a framework and elements that enable the artificial "thought" process. A foundational architecture to achieve thinking behavior should include the following systems or elements: Understanding, Intellect, Reason, Analysis, Decision, Communications, Presentation, Sensing, and Will (Figure 1). The architecture formally compartmentalizes knowledge and information within the corresponding elements, also called systems of the thought process. In this manner, a clear understanding is provided about the content and function of each element.

Thinking involves contemplation (assessment of the state, and running of what-if scenarios), learning, being receptive to instructions and orders, and being able to execute all those functions. Thinking involves making rational decisions without human input, and acting upon those decisions; as well as being able to communicate in the HUMAN LANGUAGE with people and other autonomous systems; and among its own internal systems. The systems of a "Thought" System compartmentalize data, knowledge, and information so that a formal use of the contents may be implemented with a proper ontology.

An Approach to a Technology Solution: How does one implement the above paradigm? Moreover, how may the implementation be application-agnostic to a high degree? Typically thinking behavior is not built into an application (say a space habitat) in a way that the application is able to think on-board; rather, thinking is done off-line by teams of experts. As described in [3] … "*This method [Brute Force Autonomy] could never truly be comprehensive, since there will inevitably be cases that are not apparent, imagined and/or just missed, thereby limiting the fundamental degree of autonomy. Therefore, brute-force autonomy implementations. are application instance specific (hard-coded), and hence are by their own nature exorbitantly expensive to develop, deploy, maintain, and evolve for anything beyond the specific application for which the code was designed. For these reasons, BFA has limited capability and reusability – critical requirements for developing affordable autonomous systems - and therefore not a viable solution for the sustained evolution of autonomy. "Thinking Autonomy," in contrast, implies that analysis and reasoning are evoked by the system in real time in order to derive information, knowledge, and optimal solutions."* Table 1 describes the contrast between Brute Force and Concept/Model Based autonomy capability (Thinking Autonomy).

A Technology Solution: A technology solution to implement Thinking Autonomy where thinking is done on-board by the application, needs to encapsulate Data Information and Knowledge (DIaK) defined by the systems in the architecture of a "Thought" System (Figure 1). In order to fulfill the Thought System architecture, software platforms must be developed that embody the following:

1. Tools to create a comprehensive description of the application system that will possess "Thinking" behavior. This description should be a Digital Twin (DT) of the application, encompassing structure and behavior.
2. Tools to create the mission of the application encompassing tasks, timelines, and their execution.
3. A framework and tools to incorporate the Will, Reason, Intellect, Understanding, and Decision systems of the "Thought" System.
4. A framework and tools to incorporate thought patterns and strategies that focus on assessing complete condition awareness and solutions leading to achieving a mission.

NASA Stennis Space Center (SSC) Autonomous Systems Laboratory (ASL) has developed an autonomy software platform, NPAS [3], that is consistent with the Thought System architecture and concept of operations. NPAS continues to evolve, but has already been used for multiple prototype implementations on ground and space systems [4]. Currently NPAS is planned for integration as a payload rider that will have the opportunity to conduct some autonomous operations of a satellite that will be launched in October 2023 [5]. Figure 2 shows the software architecture of NPAS. All four items above are included in the architecture. The DT created is denominated "NPAS system application knowledge domain model (blue box), and the mission is denominated "Autonomous mission plans, schedules, and execution (blue box)." "Thought" is enabled by communicating DIaK among the systems of the "Thought" architecture; whereby communication modalities and framework (taxonomy and language) permits a fluid conversation with purpose, across the "Thought" systems. "Thought" is pervasive and is ignited by needs of the two primary goals: (1) determination of the condition of the system, and (2) achievement of a mission.

Ultimately … *"thinking — pure thinking —is considered a system of concepts. Logic is in turn the science of pure thinking. Thus, in the design of the autonomous system, we can originate concepts from the Reason, the Intellect, and even the Presentation [6]."* In fact, the way to enable "Thought" by NPAS is by establishing concepts and using them in models and strategies that are part of the architecture systems.

A Use-Case of Thinking Autonomy implemented using NPAS: An example use-case will provide an understanding of how "Thinking" is enabled by NPAS. The use case is about trying to determine if a sensor may be faulty. The topology/structure of the system is shown in Figure 3, which is a graphical representation of the application system's digital twin (in NPAS the graphical topology is captured as graphical code). In this case the sensor is any sensor that meets the following constraint:

- The sensor is installed in a "Flow Section" of a commodity distribution system (CDS) as defined by the topology. The "concept" of a Flow Section is defined as the collection of connected objects (elements) that are bounded by commodity distribution control (CDC) objects (e.g. Flow Section 2 is bounded by 4 valves).

In carrying out "Thought," behavior models engage concepts to arrive to knowledge or information or data of value to the mission. In this case, the explicit model says that when a commodity fills a Flow Section, parameters such as temperature are expected to be the same throughout the Flow Section. So, one way to check if a temperature sensor is nominal or may be suspected to be off-nominal is to apply the following strategy: Identify Flow Sections across the system, check if more than one temperature sensor are installed in the Flow Section, and compare their measurements; if the measurements are reasonably close, then both sensors are likely to be nominal; if the measurements are sufficiently different, then both are suspect to be faulty.

The strategy incorporated is a generic model that itself incorporates another model that uses the concept of "Redundant" data sources (more than one temperature sensor).

What is important about the implementation described is that a broad range of application systems (fluid, electrical) can carry out the same activity (try to determine if a sensor may be faulty) without hard-coding the concepts and behavior models for a one-time, one-system, one process case. That is, once NPAS is used to create the DT, the concepts and the strategy can be applied, regardless of the specific topology (the concepts are identified in any topology). Thinking is being done on-board, by the system. Thinking is enabled by a system of concepts and logic.

Another concept that is used to detect "leak" is the concept of "isolated subsystem" (ISS). An ISS is defined as a set of connected Flow Sections whereby the boundary CDC objects of the collection are in a Closed State. Again, NPAS includes this concept to implement the following leak-detection strategy: For any ISS, if the ISS includes a pressure sensor which measurement is increasing or decreasing, then the ISS is suspect of leak, consequently, the leak may be happening at any member object of the ISS.

References:

[1] Marie Morganelli (2020) *https://www.snhu.edu/about-us/newsroom/business/what-is-systems-thinking*. [2] *https://techport.nasa.gov/view/94884*. (2023). [3] Fernando Figueroa, Lauren Underwood, Mark Walker, and Jonathan Morris (2019) *AIAA Scitech 2019 Forum, 7-11 January 2019, San Diego, CA, USA*. [4] Fernando Figueroa, Lauren Underwood, Jonathan Morris, Mark Walker, and Rane Brown. (2022) *"Risk-Reduction Autonomy Implementation to Enable NASA Artemis Missions," IEEE Aerospace Conference 2022, March 5-12, 2022, Big Sky, Montana, USA*. [5] https://www.businesswire.com/news/home/20230308005327/en/Sidus-Space-Awarded-New-Contract-for-Next-Phase-of-NASA-ASTRA-Project. [6] *The Autonomous System: A Foundational Synthesis of the Sciences of the Mind*, by Szabolcs Michael de Gyurky, Mark A. Tarbell, John Wiley & Sons, Oct 8, 2013

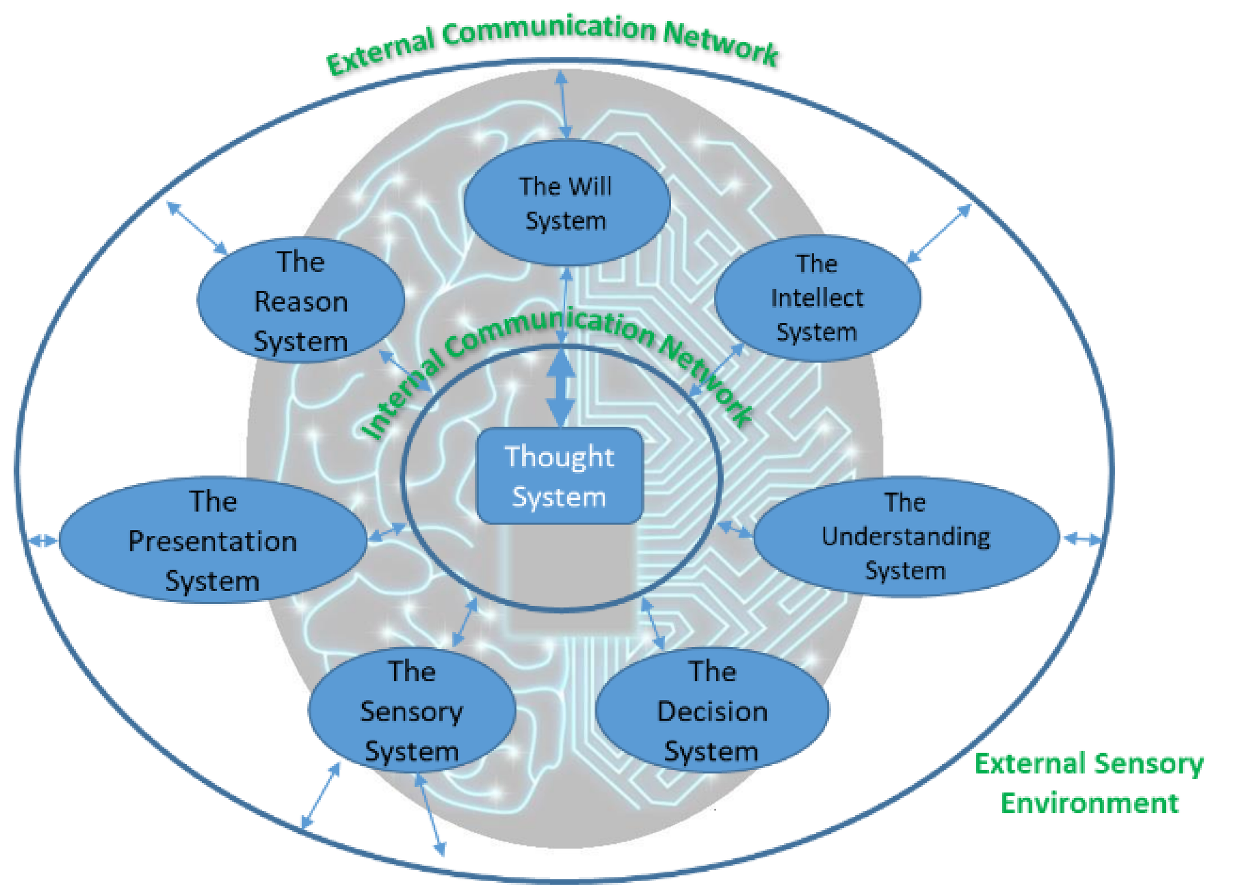


Figure 1 Architecture and Systems of a Thought System (The Autonomous System: A Foundational Synthesis of the Sciences of the Mind, by Szabolcs Michael de Gyurky, Mark A. Tarbell, John Wiley & Sons, Oct 8, 2013).

Brute Force **Minimal or no analysis and decisions by the System**	**Concept/Model Based** **Most analysis and decisions by the System**
Problems solved by the expert off-line, and made available to computer code to make the most appropriate choice of solutions (lookup tables).	***Problems solved by the autonomous system that is able to apply generic models of processes taking place in the system, in real time***.
Solutions must be found for ***every possible case defined by every object, every process that involves the object, and every configuration of the system where the object belongs***; and for every process that may occur in the system (nominal and off-nominal).	The autonomous system analyses the system behavior at a high level of abstraction, by applying generic models (e.g. physics models) that employ definitions and concepts used in the models; Solutions are determined with a relatively small set of models.
Coverage is not comprehensive***; it is limited by the thoroughness of the analysis by expert off-line programmers***.	Coverage is comprehensive for each generic model used. So total comprehensive coverage is feasible, since first principles cover all behaviors.
The autonomous system does not "think" it merely makes use of what has been "thought" by humans.	***The autonomous system "thinks" to provide solutions that need not be solved off-line by humans.***

Table 1 Contrast between Brute-Force (prescriptive) Autonomy and Concept/Model Based Autonomy (Thinking Autonomy).

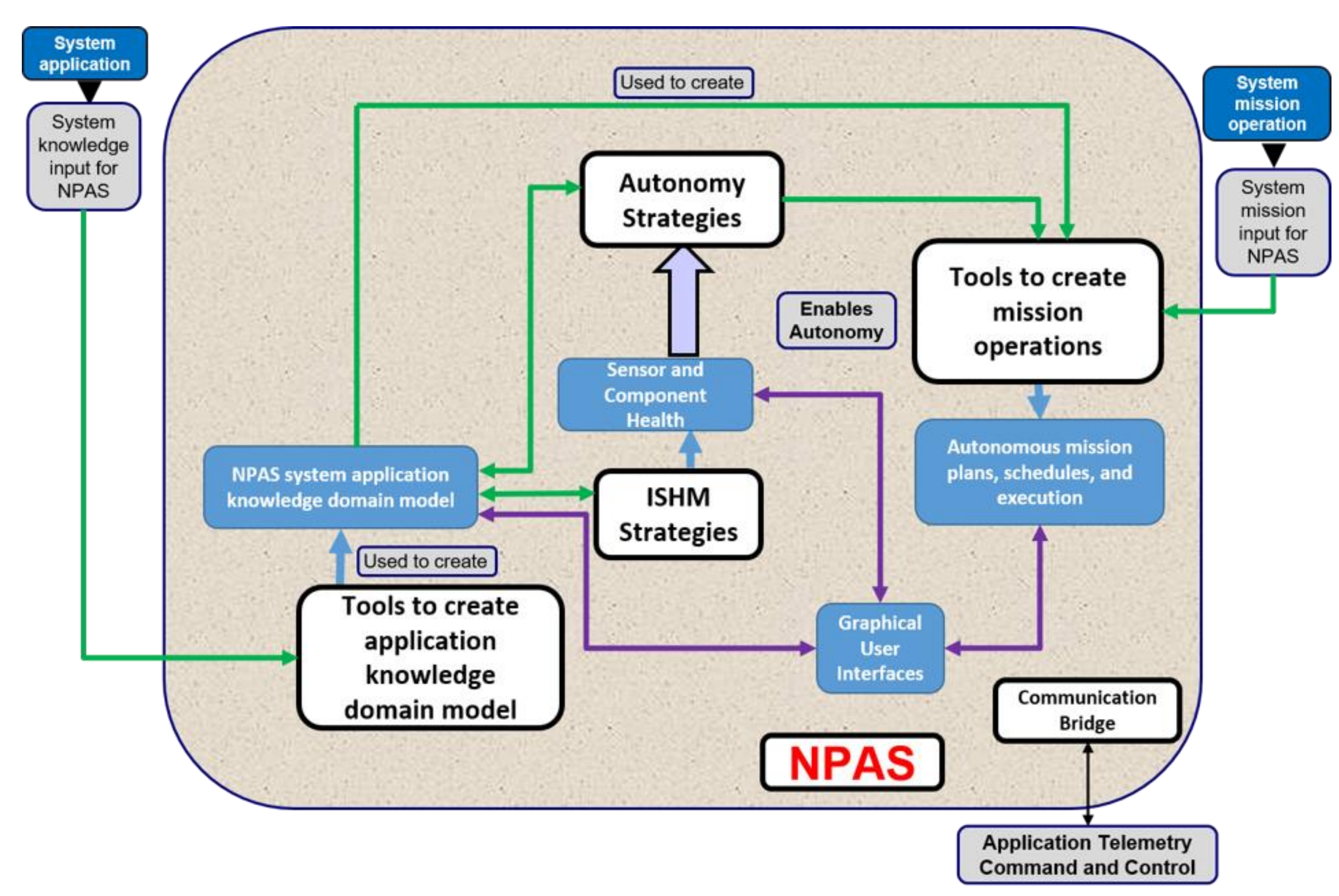
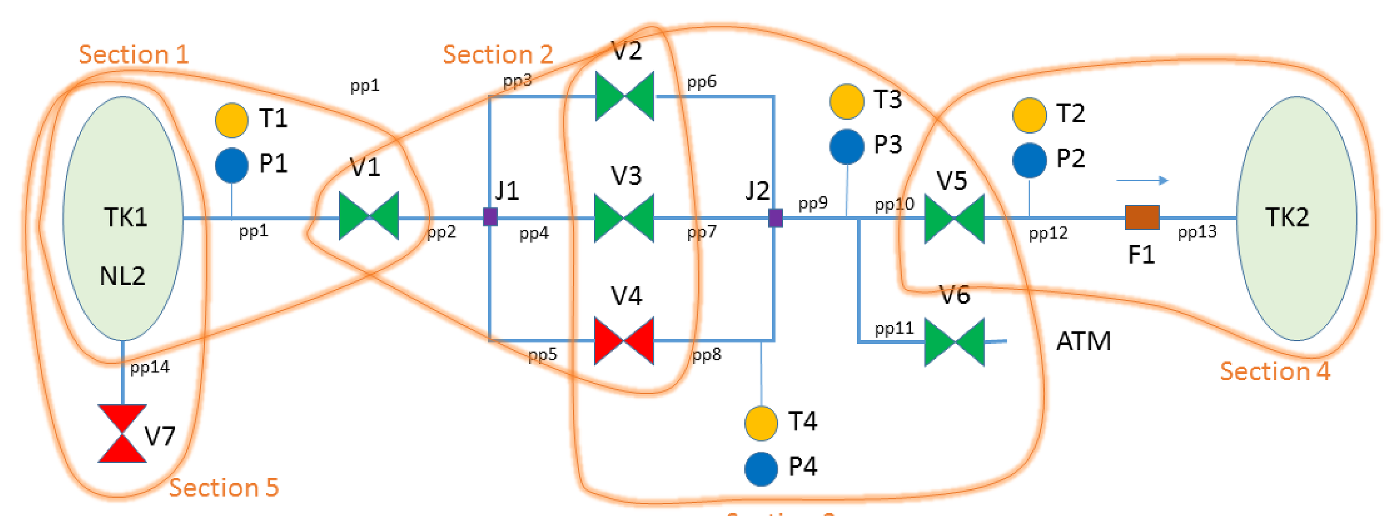


Figure 2 Software Architecture, NASA Platform for Autonomous Systems (NPAS) [3].



Flow Section Concept Examples
Section 1: Part of (pp1, T1, P1) Bounding: (TK1, V1)
Section 2: Part of (pp2, J1, pp3, pp4, pp5) Bounding: (V1, V2, V3, V4)
Section 3: Part of (pp6, pp7, pp8, J2, pp9, T3, P3, T4, P4, pp10, pp11) Bounding: (V2, V3, V4, V5, V6)
Section 4: Part of (T2, P2, pp12, F1, pp13) Bounding: (V5, TK2)

Model: "Throughout each flow section there is one commodity, and some physical variables of the commodity such as temperature should be similar throughout the piping section"
Similar value set 1: T3, T4

Figure 3 Concept/model Thought implementation for reasoning about a sensor health state (nominal, off-nominal). The concept implemented is denominated "Flow Section."

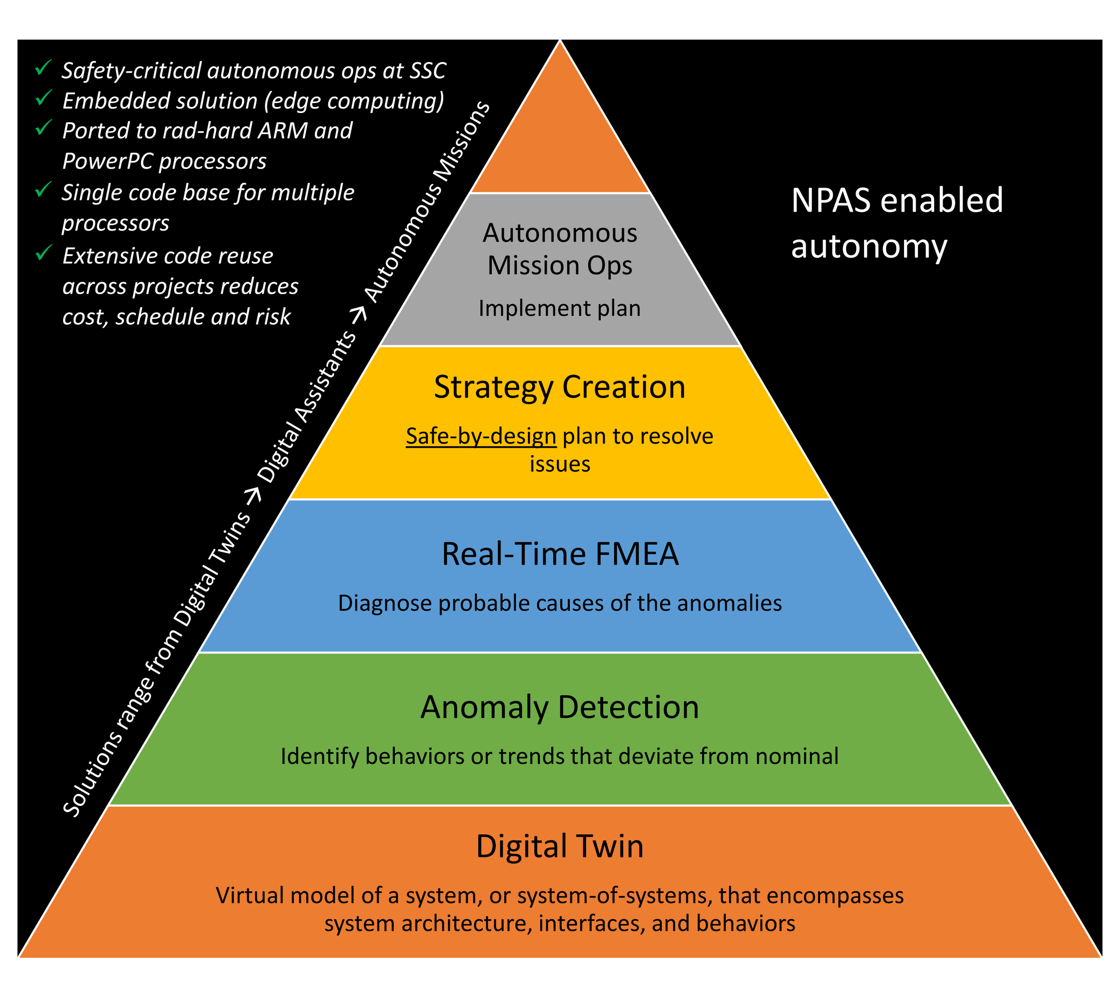


Figure 4 NPAS Solution Pyramid (Duane Armstrong, NASA Stennis Space Center).