Efficient Space Exploration through Laziness

Emil Vassev and Mike Hinchey

Lero—the Irish Software Engineering Research Centre,
University of Limerick, Limerick, Ireland
{Emil.Vassev,Mike.Hinchey}@lero.ie

Abstract. Autonomous behavior and onboard decision making is the backbone of robotic space exploration. The enormous distance and communication latency make such missions hardly controllable from Earth and external decision making may overlap and often contradict with the onboard decision making. We propose a behavior model based on some sort of "laziness" that helps spacecraft evaluate external instructions and eventually postpone their execution, or even discard some, when those are considered inappropriate by the internal spacecraft decision making.

Keywords: decision making, space exploration, autonomous spacecraft.

1 Introduction

Robotic space exploration helps NASA perform unmanned missions to reach deep space where no human can go. The enormous distance and communication latency make such missions hardly controllable from Earth. Hence, autonomous behavior and onboard decision making is the "backbone" of robotic space exploration. Contemporary robotic spacecraft have onboard intelligence based on structured knowledge and reasoning capabilities. This artificial intelligence helps spacecraft make decisions driven by factors like mission goals, safety, performance, efficiency, resource consumption, etc. Similar to the human mind, such intelligence cannot exist isolated on its own and must cope with external control provided by other sources of intelligence – human pilots, mission ground stations, or other sources of artificial intelligence like spacecraft and unmanned space stations. Therefore, external decision-making processes may overlap and often contradict with the internal spacecraft decision making. "Blind" execution of all the external control instructions might be harmful and will often lead to insufficient performance, simply because the external decision making is not that well informed about all the spacecraft issues and parameters. We propose a behavior model for spacecraft based on some sort of "laziness" that helps spacecraft evaluate external instructions and eventually postpone their execution, or even discard some, when those are considered inappropriate by the internal spacecraft decision making. Discarded or postponed external instruction may require feedback sent to the initial instruction source. Moreover, high-priority external instructions should not be evaluated, but executed immediately to assure high-authority control over the functionality of smart spacecraft. A postponed external instruction might be delayed by spacecraft just to reach a state where its execution might lead to the best possible performance.

2 Model for Efficient Space Exploration through Laziness

The basic idea behind this behavior model is to allow smart spacecraft act as "lazy workers" having their own interest and goals and often reluctant to perform external instructions if the latter do not conform to their goals. Instead of immediately performing any newly received external instruction, smart spacecraft will evaluate that instruction and decide whether to perform, postpone, or discard the same. Presuming that 1) the spacecraft goals are set by their mission objectives or driven by their safety policies; and 2) the external instructions might be obsolete due to communication delay (coming from Earth and thus travelling a long distance) and eventually incorrect, because the sender does not have complete and recent information about all the spacecraft parameters; the overall result of such behavior could be a significant performance gain. The behavior model for such *Efficient Space Exploration through Laziness* (for short *efficiency with laziness* (EL)) can be presented as a function accepting two parameters – *system knowledge* and a set of *external instructions*, and determining the spacecraft behavior concerning the incoming set of external instructions.

2.1 Formal Model for Efficiency with Laziness

Formally, the behavior model for EL can be presented as following:

$$\boldsymbol{R_{EL}: K \times L_I \rightarrow B_{EL}} \tag{1}$$

Here, $\boldsymbol{R_{EL}}$ is the EL function computing the *possible behavior* for each external set of instructions $\boldsymbol{L_I}$ considering the system knowledge $\boldsymbol{K}$. The knowledge $\boldsymbol{K}$ of smart spacecraft capable of reasoning and decision making can be formally presented as a tuple of three main knowledge components (knowledge models):

$$\boldsymbol{K = \langle K_I , K_C , K_E \rangle} \tag{2}$$

where $\boldsymbol{K_I}$ is *internal knowledge*, $\boldsymbol{K_C}$ is *control knowledge*, and $\boldsymbol{K_E}$ is *external knowledge*. The internal knowledge $\boldsymbol{K_I}$ carries information about the internal structure and capabilities of the system and it can be presented as a tuple of three components:

$$\boldsymbol{K_I = \langle C , F , R \rangle} \tag{3}$$

where $\boldsymbol{C}$ states for system components, $\boldsymbol{F}$ is system functionality, and $\boldsymbol{R}$ is system resources. There could be added more parameters to this tuple, such as *interdependencies* (if not present in the knowledge about the system components $\boldsymbol{C}$), *system architecture*, etc. Further, the control knowledge $\boldsymbol{K_C}$ gives the system knowledge about its control parameters and mission and it can be presented as a tuple of a few parameters:

$$\boldsymbol{K_C = \langle\, \Pi\,,\, M\langle G\,,\, T\rangle\,,\, S\,,\, A\,,\, H_{Si}\langle S_H,\, A_H\rangle\,,\, L\,\rangle} \tag{4}$$

where:

- $\boldsymbol{\Pi}$ states for behavior policies (safety, performance, etc.) driving the system in particular situations.
- $\boldsymbol{M}$ is the mission knowledge such as goals $\boldsymbol{G}$ (or objectives, e.g., service-level objectives), time constraints $\boldsymbol{T}$, etc.
- $\boldsymbol{S}$ is a set of all known possible states the system can take.
- $\boldsymbol{A}$ is a set of all possible actions the system can undertake (actions are functions over system's functionality involving consumption of system resources).
- $\boldsymbol{H_{Si}}$ is a history of situations the system ended up in. A situation connects past states $\boldsymbol{S_H}$ to past actions $\boldsymbol{A_H}$ performed by the system to get out from those states. Note that $\boldsymbol{H_{Si}}$ provide the necessary information for *reinforcement learning*.
- $\boldsymbol{L}$ is alphabet of an interface language used to communicate with the system.

Finally, the external knowledge $\boldsymbol{K_E}$ is to provide the spacecraft with information about the surrounding environment (environmental factors), e.g., solar system, solar storms, planetary systems, asteroids, gravity force of the near space objects, etc. Considering all the knowledge elements, we further reveal Definition (2):

$$\boldsymbol{K = \langle\, K_I\langle C, F, R\rangle\,,\, K_C\langle \Pi, M\langle G, T\rangle, S, A, H_{Si}\langle S_H, A_H\rangle, L\rangle\,,\, K_E\,\rangle} \tag{5}$$

Further, the set of external instructions in Definition (1) shall be expressed with the alphabet provided by the interface language $\boldsymbol{L}$. The determined by the function $\boldsymbol{R_{EL}}$ (see Definition (1)) EL behavior $\boldsymbol{B_{EL}}$ can be one of the following:

- *execute* – all the external instructions are immediately executed with the highest possible priority;
- *postpone* - the external instructions are scheduled for execution but after the execution of more important and locally decided instructions;
- *discard* – the external instructions are discarded.

When the spacecraft postpones or discards external instructions, it may notify the instructions' sender about this behavior with the appropriate reasons. In order to decide on the behavior $\boldsymbol{B_{EL}}$, the function $\boldsymbol{R_{EL}}$ follows the following algorithm:

1) Check whether the external instructions $\boldsymbol{L_I}$ are high priority instructions:
   - Yes – perform.
   - No - continue with 2).
2) Check whether the external instructions $\boldsymbol{L_I}$ are obsolete:
   - Yes – discard.
   - No – continue with 3).

3) Check whether the external instructions $\boldsymbol{L_I}$ require the execution of actions $\boldsymbol{A'} \subset \boldsymbol{A}$ that will harm the mission goals $\boldsymbol{G}$ or contradict with the spacecraft policies $\boldsymbol{\Pi}$ (e.g., for safety). To do that, the spacecraft must compute the probability of the spacecraft occupying one of the undesired states $\boldsymbol{S'} \subset \boldsymbol{S}$ (where some of the mission goals or spacecraft policies are violated) after the execution of actions $\boldsymbol{A'}$. Here,

$$N\colon S \times A \rightarrow P(S) \tag{6}$$

is a *state-transition function* giving for each state $s \in S$ and action $a \in A$ a probability distribution. Here, $N\ (s;\ a;\ s')$ computes the probability of ending in state s', given that the start state is s and the spacecraft takes action **a**, $p\ (s'\ |\ s;\ a)$. Therefore, the spacecraft knows the probability of ending in one of the undesired states $\boldsymbol{S'}$ if actions $\boldsymbol{A'}$ are executed from the current state s. The computed probability p is a scalar value in the range [0..1]. A special EL policy may decide on the two thresholds: 1) what probability level is sufficient a set of external instructions $\boldsymbol{L_I}$ to be postponed, e.g., $p \in [0..0{,}5)$; and 2) what probability level is sufficient a set of external instructions $\boldsymbol{L_I}$ to be discarded, e.g., $p \in [0{,}5..1]$.

There could be different "levels of laziness" depending on the probability ranges, which determine those two thresholds. The theoretical foundation for the probability assessment is the so-called Markov Chains [1].

2.2 Probability Assessment

The EL model requires computation of probability values for ending in possible undesired states when particular actions are executed. In this subsection, we present a *model for assessing probability* applicable to the computation of EL probability values. In our approach, the *probability assessment* is an indicator of the number of possible execution paths spacecraft may take, meaning the amount of certainty (excess entropy) in the spacecraft behavior. To assess that behavior prior to implementation, it is important to understand the interactions among the *spacecraft components* and also the complex interactions with the surrounding environment (space). This can be achieved by modeling the behavior of the *individual reactive spacecraft components* and the behavior of the *environmental factors* (e.g., solar storm, gravity of a planet, etc.), together with the *global system behavior* as *Discrete Time Markov Chains* [1], and by assessing the level of probability through calculating the probabilities of the state transitions in the corresponding models. We assume that the component interactions and the environment-system interaction are stochastic processes where the events are not controlled by the spacecraft and thus, their probabilities are considered equal.

The theoretical foundation for our Probability Assessment Model is the property of Markov chains, which states that, *given the current state of the whole spacecraft system, its future evolution is independent of its history*, which is also the main characteristic of a reactive and autonomic spacecraft [2, 3].

An algebraic representation of a Markov chain is a matrix (called *transition matrix*) (see Table 1) where the rows and columns correspond to the states, and the entry p_{ij} in the ith row, jth column is the transition probability of being in state s_j at the stage following state s_i.

Table 1. Transition matrix ***P***

	s1	**s2**	**...**	**sj**	**...**	**sn**
s1	p_{11}	p_{12}	...	p_{1j}	...	p_{1n}
s2	p_{21}	p_{22}	...	p_{2j}	...	p_{2n}
...	...	...	...	...	...	...
si	p_{i1}	p_{i2}	...	p_{ij}	...	p_{in}
...	...	...	...	...	...	...
sn	p_{n1}	p_{n2}	...	p_{nj}	...	p_{nn}

We need to build such a transition matrix taking into account both the system components and environmental factors influencing the system behavior. The following property holds for the calculated probabilities:

$$\Sigma_j\, p_{ij} = 1 \tag{7}$$

We contend that probability should be calculated from the *steady state* of the Markov chain. A steady state (or *equilibrium state*) is one in which the probability of being in a state before and after a transition is the same as time progresses. Here, we define probability for a spacecraft system composed of k components and taking into account x environmental factors as the level of certainty quantified by the source excess entropy, as follows.

$$P = \Sigma_{i=1,k} H_i + \Sigma_{e=1,x} H_e - H \tag{8}$$

$$H_i = -\Sigma_j\, p_{ij} \log_2(p_{ij}) \tag{9}$$

$$H_e = -\Sigma_j\, p_{ej} \log_2(p_{ej}) \tag{10}$$

$$H = -\left(\Sigma_i\, v_i\, \Sigma_j\, p_{ij} \log_2(p_{ij}) + \Sigma_e\, v_e\, \Sigma_j\, p_{ej} \log_2(p_{ej})\right) \tag{11}$$

Here,

- $\boldsymbol{H}$ is an entropy that quantifies the level of uncertainty in the Markov chain corresponding to the entire spacecraft system;
- $\boldsymbol{H_i}$ is a level of uncertainty in a Markov chain corresponding to a spacecraft component;
- $\boldsymbol{H_e}$ is a level of uncertainty in a Markov chain corresponding to an environmental factor, e.g., distance to ground base, solar storm, gravity force of a planet, etc.;
- $\boldsymbol{v}$ is a steady state distribution vector for the corresponding Markov chain;
- $\boldsymbol{p_{ij}}$ values are transition probabilities in the extended state machines modeling the behavior of the $\boldsymbol{i}^{th}$ component;
- $\boldsymbol{p_{ej}}$ values are transition probabilities in the extended state machines modeling the behavior of the $\boldsymbol{e}^{th}$ environmental factor.

Note that for a transition matrix $\boldsymbol{P}$, the steady state distribution vector $\boldsymbol{v}$ satisfies the property $\boldsymbol{v*P = v}$, and the sum of its components $\boldsymbol{v_i}$ is equal to $\boldsymbol{1}$.

Interpretation. The level of uncertainty $\boldsymbol{H}$ is exponentially related to the number of *statistically typical paths* in the Markov chain. Having an entropy value of $\boldsymbol{0}$ means that there is no level of uncertainty in a Markov system for a specific unit's behavior. A higher value of probability implies less uncertainty in the model.

2.3 Modeling the Behavior Policies with Probability

To allow for EL-driven behavior, we need to develop the behavior policies $\boldsymbol{\Pi}$ (see Definition 5) taking into consideration the probability distribution for the possible state transitions (see Section 2.2). Note that in another project of ours, we have developed a model for *autonomic policies* that uses probability distributions to rate the probability of policies being executed in specific situations and under specific conditions [4]. Because, each policy is associated with possible actions, the behavior realized by the actions executed in the environment will be guided by the policies with the highest probability where the probability rates are recomputed after every action execution.

We can elaborate on this approach to adapt it to the EL model where policies should consider both the internal components and environmental factors.

2.4 Awareness for EL

In general, an EL-based system engages in interactions with the ground base on Earth and with its operational environment. When interacting with the environment, the EL spacecraft also perceive important structural and dynamic aspects of the same [3]. To become interaction-aware, such a system needs to be aware of its physical environment and whereabouts and its current internal status. This ability is defined as *awareness* and it helps intelligent computerized systems to sense, draw inferences for their own behavior and react. The notion of awareness should be generally related to perception, recognition, thinking and eventually prediction [5]. Recall that the EL mechanism requires relevant knowledge (see Definitions 1 and 2) that helps the system autonomously determine states. For example, the EL approach requires the system to be aware about the environmental factors such as gravity, solar storms, etc. This can be achieved via a mechanism called "Pyramid of Awareness" [5] where a complex chain of functions shall be implemented to control the EL awareness process via monitoring, recognition, assessment, and learning.

3 Example of Efficient Space Exploration through Laziness

In this short example, we use as a case study the prospective NASA's Autonomous Nano-Technology Swarm (ANTS) space exploration mission [6], which is a novel approach to asteroid-belt resource exploration (see Figure 1).

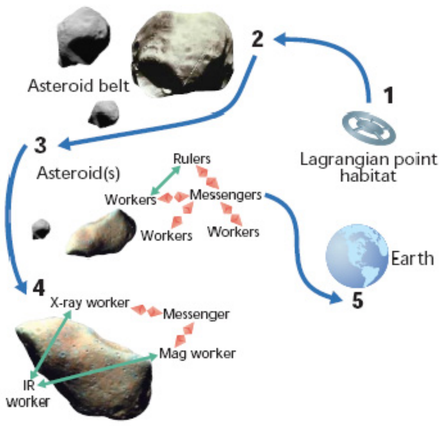

Fig. 1. ANTS mission concept [6]

An ANTS system is composed of individual spacecraft where each spacecraft is equipped with a solar sail and relies primarily on power from the sun, using only tiny thrusters to navigate independently. Moreover, each spacecraft also has onboard computation, artificial intelligence, and heuristics systems for control at both the individual and system levels. To explore a new asteroid, ANTS needs to form a team of spacecraft units called *workers* and carrying special instruments. A team is formed and coordinated by a special spacecraft unit called *ruler* [6]. Special messengers are needed to connect the team members when they cannot connect directly, due to long distances or a barrier. To form a team, a ruler broadcasts instructions to workers.

In our approach, the EL mechanism helps workers evaluate those instructions and make their decision how to proceed, so the overall ANTS goals are not harmed and the overall efficiency is the highest possible. For example, EL will help 1) only idle workers join the team; and 2) to complete the team with the sufficient number of workers. When the team is complete, the other idle workers will not join the team, because the EL mechanism will evaluate this as an act harming the overall ANTS goals. Moreover, the EL mechanism will prevent workers from joining the team if the probability of having a solar storm is high which can jeopardize the overall mission.

4 Conclusion

In this paper, we proposed a behavior model for efficient space exploration through laziness. The basic idea is to allow smart spacecraft act as "lazy workers" having their own interest and goals and often be reluctant to perform external instructions if the latter do not conform to spacecraft goals and if the environment imposes considerable hazards. The approach requires the spacecraft to compute the probability of occupying an undesired state after following external instructions. Note that as undesired state is considered a situation where either mission goals or spacecraft behavior policies are violated. Based on the computed probability, the external instructions can be immediately executed, postponed for later execution or completely discarded. The probability assessment is an indicator of the number of possible execution paths spacecraft may take, meaning the amount of certainty in the spacecraft behavior. In our approach the probability assessment is based on Markov Chains. Finally, the EL approach requires an awareness mechanism to keep the EL knowledge relevant and up-to-date by taking into consideration internal, control and environmental factors. The proposed awareness mechanism is the so-called "Pyramid of Awareness", which we are currently developing for another project of ours.

Future work is concerned with further and complete development of the EL mechanism, including knowledge representation for modeling EL behavior, probability assessment and awareness for EL.

Acknowledgment. This work was supported by Science Foundation Ireland grant 03/CE2/I303_1 to Lero—the Irish Software Engineering Research Centre at University of Limerick, Ireland and by the European Union FP7 Integrated Project Autonomic Service-Component Ensembles (ASCENS).

References

1. Ewens, W.J., Grant, G.R.: Stochastic processes (i): poison processes and Markov chains. In: Statistical Methods in Bioinformatics, 2nd edn. Springer, New York (2005)
2. Vassev, E., Sterritt, R., Rouff, C., Hinchey, M.: Swarm Technology at NASA: Building Resilient Systems. IT Professional 14(2), 36–42 (2012)
3. Vassev, E., Hinchey, M.: The Challenge of Developing Autonomic Systems. IEEE Computer 43(12), 93–96 (2010)
4. Vassev, E., Hinchey, M., Gaudin, B.: Knowledge Representation for Self-Adaptive Behavior. In: Proceedings of C* Conference on Computer Science & Software Engineering (C3S2E 2012), pp. 113–117. ACM (2012)
5. Vassev, E.: Building the pyramid of awareness. Awareness Magazine - Self-awareness in Autonomic Systems (July 2012), doi: 10.2417/3201207.004320
6. Truszkowski, W., Hinchey, M., Rash, J., Rouff, C.: NASA's swarm missions: The challenge of building autonomous software. IT Professional 6(5), 47–52 (2004)