AUTONOMIC & APOPTOTIC COMPUTING PROTOTYPE; PROVIDING PRE-PROGRAMMED DEATH OF CUBESATS FOR AVOIDING SPACE JUNK

Robbie Palmer and Roy Sterritt

Ulster University, Jordanstown Campus, Co. Antrim, Northern Ireland
Email: r.sterritt@ulster.ac.uk

Abstract

Apoptotic Computing is inspired by the apoptosis mechanism in biological systems. This mechanism provides security for the overall system by having a pre-programmed death and indeed a death by default at, for instance, the cellular level. It has been argued that this approach should be included in our modern ubiquitous/pervasive computer-based systems.

This paper presents a first step prototype ultilizing apoptotic computing for CubeSats, pre-programming death with the intent to ensure that CubeSats do not add to the proliferation of Space Junk. In addition, to support that apoptotic mechanism, autonomicity (self-management) is utilised and the paper considers more general swarm task assignment and control protocols.

Keywords: Emerging Technologies, Biological Inspired, Autonomic Computing, Apoptotic Computing, Self-management, Autonomy, Swarms, Constellations, Space Junk

1. Introduction

The Apoptotic Computing project, first started back in 2002 [2][3][4][5][1] involves working towards the long-term goal of developing Programmed Death by Default for Computer-Based Systems to provide an ultimate safety mechanism. It is essentially biologically-inspired by the Apoptosis mechanisms in multicellular organisms. It may be considered as a sub-area of Bio-Inspired Computing, Natural Computing or Autonomic Systems (providing the self-destruct property amongst other self* properties).

Our approach to implementing Apoptotic Computing has been through Autonomic Computing – the vision to create self-managing systems, inspired by the biological Autonomic Nervous System (ANS). Autonomic Computing we consider a sub-field of Autonomous Systems, and from a Software Engineers perspective a sub-division of labour – autonomicity (or autonomics) to provide self-management (inspired by the non-conscious efforts of the ANS) to enable better autonomy (biological inspiration; higher level thinking and conscious decision making by the brain freed from sub-management concerns).

This paper presents a prototype for CubeSats, to apply the apoptotic mechanism with the longer term vision to prevent the potential mass deployment of CubeSats adding to an already extensive man-made debris field in orbit around Earth.

2. Related Works & Inspiration

2.1 Biological Apoptosis

If one cuts oneself and starts bleeding, one treats it and carrying on with one's tasks without any further conscious thought (although pain receptors will induce self-protection and self-configuration to use the other hand!). Yet, often, the cut will have caused skin cells to be displaced down into muscle tissue [7]. If they survive and divide, they have the potential to grow into a tumour. The body's solution to dealing with this situation is cell self-destruction (with mounting evidence that some forms of cancer are the result of cells not dying fast enough, rather than multiplying out of control, as previously thought and considered by the general public).

It is believed that a cell knows when to commit suicide because cells are programmed to do so – self-destruct (sD) is an intrinsic property. This sD is delayed due to the continuous receipt of biochemical retrieves. This process is referred to as apoptosis[16], pronounced either as APE-oh-TOE-sls or uh-POP-tuh-sis and means for 'to fall off' or 'drop out', used by the Greeks to refer to the Fall/Autumn dropping of leaves from trees; i.e., loss of cells that ought to die in the midst of the living structure. The process has also been nicknamed 'death by default'[7], where cells are prevented from putting an end to themselves due to constant receipt of biochemical 'stay alive' signals (*Figure **1***). The key aspect of apoptosis is that the cell's self-destruction takes place in a programmed and controlled way (*Figure **2***); the suicidal cell starts to shrink, decomposes internal structures and degrades all internal proteins. Thereafter, the cell breaks into small membrane-wrapped fragments (drop-off) that will be engulfed by phagocytic cells for recycling. Necrosis, is the un-programmed death of a cell, involving inflammation and toxic substances leaking to the environment [8].

Further investigations into the apoptosis process [16] have discovered more details about the self-destruct program. Whenever a cell divides, it simultaneously receives orders to kill itself. Without a reprieve signal, the cell does indeed self-destruct. It is believed that the reason for this is self-protection, as the most dangerous time for the body is when a cell divides, since if just one of the billions of cells locks into division the result is a tumour, while simultaneously a cell must divide to build and maintain a body.

The suicide and reprieve controls have been compared to the dual-key on a nuclear missile [6]. The key (chemical signal) turns on cell growth but at the same time switches on a sequence that leads to self-destruction. The second key overrides the self-destruct [6].

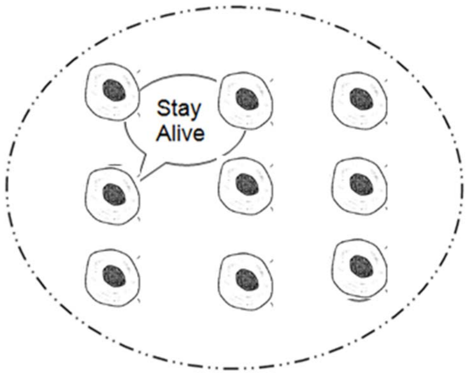


Figure 1 Turning off the self-destruct sequence - cell receives 'stay alive' signal [2].

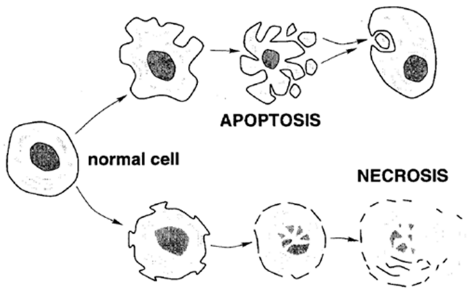


Figure 2 Programmed death by default (apoptosis) and necrosis due to injury [8]

2.2. Computer-Based System's Apoptosis

The case has been made for introducing apoptotic measures into Agent-Based Systems, Autonomic (Self-managing and adaptive) Systems and Swarm Based Space Exploration Systems [1]-[5] and is recapped in this section.

2.2.1 Agent-Based Apoptotic Systems

Agent destruction has been proposed for mobile agents, in order to facilitate security measures [10]. Greenberg et al. highlighted the scenario simply by recalling the situation where the server omega.univ.edu was decommissioned, its work moving to other machines. When a few years later a new computer was assigned the old name, to the surprise of everyone, email arrived, much of it 3 years old[10]. The mail had survived 'pending' on Internet relays waiting for omega.univ.edu to come back up.

Greenberg encourages consideration of the same situation for mobile agents; these would not be rogue mobile agents – they would be carrying proper authenticated credentials. This work would be done totally out-of-context due to neither abnormal procedure nor system failure. In this circumstance, the mobile agent could cause substantial damage, e.g., deliver an archaic upgrade to part of the network operating system, resulting in bringing down the entire network.

Misuse involving mobile agents comes in the form of: misuse of hosts by agents, misuse of agents by hosts, and misuse of agents by other agents.

From an agent perspective, the first is through accidental or unintentional situations caused by that agent (race conditions and unexpected emergent behavior), the latter two through deliberate or accidental situations caused by external bodies acting upon the agent. The range of these situations and attacks have been categorized as: damage, denial-of-service, breach-of-privacy, harassment, social engineering, event-triggered attacks, and compound attacks.

In the situation where portions of an agent's binary image (e.g., monetary certificates, keys, information, etc.) are vulnerable to being copied when visiting a host, this can be prevented by encryption. Yet there has to be decryption in order to execute, which provides a window of vulnerability [10]. This situation has similar overtones to our previous discussion on biological apoptosis, where the body is at its most vulnerable during cell division [2]. As such an agent should have an inherent pre-programmed self-destruct mechanism inbuilt for safety of the information or code it carries, either that can be self-activated if detects interference or if it has not received its stay-alive signal as it is no longer in the correct context or arrived with an non-authorized host.

2.2.2 Autonomic Apoptotic Systems

Autonomic Computing and Communications are inspired by the biological nervous system and properties of homeostasis and responsiveness The general properties of an autonomic, or self-managing, system can be summarized by four objectives—self-configuration, self-healing, self-optimization and self-protection—and four attributes— self-awareness, self-situation, self-monitoring and self-adjusting [9][14][15].

Essentially, the objectives represent broad system requirements, while the attributes identify basic implementation mechanisms.

The autonomic paradigm assigns an "autonomic manager" to a component utilizing a sensors and effectors and a closed control feedback loop to provide the self-management. Autonomic Mangers communicate and cooperate to provide system wide self-management.

AMs may communicate and cooperate through a combination of various means; self-managing event messages, heart-beats, pulse signals, RPCs, and mobile agents. The apoptosis (stay alive / self-destruct mechanism) may be utilized in this scenario as self-protection, to withdraw authorization to continue operation, for example, if the policies become out-of-date when they arrive at the autonomic manager their "stay alive" reprieve has not been received thus preventing the system changes from being enacted.

2.2.3 Swarm-Based Space Exploration Systems

Space Exploration Missions, through necessity, have been incorporating more and more autonomy and adaptability. Autonomy may be considered as self-governance of one's own tasks/goals. NASA is investigating the use of swarm technologies for the development of sustainable exploration missions that will be autonomous and exhibit autonomic properties. The idea is that biologically-inspired swarms of smaller spacecraft offer greater redundancy (and, consequently, greater protection of assets), reduced costs and risks, and the ability to explore regions of space where a single large spacecraft would be impractical.

ANTS (Autonomous Nano-Technology Swarm) is a NASA concept mission, a collaboration between NASA Goddard Space Flight Center and NASA Langley Research Center, which aims at the development of revolutionary mission architectures and the exploitation of artificial intelligence techniques and the paradigm of biological inspiration in future space exploration [13] [12]. The mission concept includes the use of swarm technologies for both spacecraft and surface-based rovers, and consists of several submissions such as SARA (The Saturn Autonomous Ring Array), PAM (Prospecting Asteroid Mission) and LARA (ANTS Application Lunar Base Activities).

In terms of ANTS missions' Autonomy, for instance, results in a worker having responsibility for its goals. To achieve these goals many self-* properties such as self-configuration will be necessary, as well as utilization of heart-beats, pulse signals and reflex reactions within AMs. NASA missions, such as ANTS, have Mission control and operations in a trusted private environment. This eliminates many of the wide range of agent and autonomic security issues briefly highlighted earlier, just leaving the particular concern is the agent operating in the correct context and exhibiting emergent behaviour within acceptable parameters, whereupon apoptosis can make a contribution.

The ANTS architecture is itself inspired by biological low level social insect colonies with their success in the division of labour. Within their specialties, individual specialists generally outperform generalists, and with sufficiently efficient social interaction and coordination, the group of specialists generally outperforms the group of generalists. Thus systems designed as ANTS are built from potentially very large numbers of highly autonomous, yet socially interactive, elements. The architecture is self-similar in that elements and sub-elements of the system may also be recursively structured as ANTS [11], and as such the self-management architecture with at least an AM per ANT craft can abstractly fit with that portrayed for the Autonomic Systems paradigm.

The revolutionary ANTS paradigm makes the achievement of such goals possible through the use of many small, autonomous, reconfigurable, redundant element craft acting as independent or collective agents[11].

Let us consider the role of the self-destruct property, inspired by apoptosis, in the ANTS mission: suppose one of the worker agents was indicating incorrect operation, or when co-existing with other workers was the cause of undesirable emergent behaviour, and was failing to self-heal correctly. That emergent behaviour (depending on what it was) may put the scientific mission in danger. Ultimately the stay-alive signal from the ruler agent would be withdrawn [2].

Also, if a worker, or its instrument, were damaged, either by collision with another worker, or (more likely) with an asteroid, or during a solar storm, a ruler could withdraw the stay-alive signal and request a replacement worker. Another worker could self-configure to take on the role of the lost worker; i.e., the ANTS adapt to ensure an optimal and balanced coverage of tasks to meet the scientific goals.

If a ruler or messenger were similarly damaged, its stay-alive signal would also be withdrawn, and a worker would be promoted to play its role.

3. CubeSat Apoptosis & Autonomic Prototype / Proof of Concept

We had discussions with a potential client in the space industry who recognizing the risk and financial impact of failure with large expensive satellites, wished to evaluate the potential of using many cheap satellites (such as

CubeSats) for their use case. The Autonomic (self-managing) paradigm we have been advancing and advocating obviously fits this vision to create collaborating "SmartSats". Though the first focus for the prototype we developed was to prove the Apoptotic feature so that the cheaper (and potentially more numerous) sats would not add to the already existing extensive amount of artificially created space debris.

It is also envisioned that traditional challenges can be overcome by use of many small, cheap and redundant satellites if these can collaborate (constellations or even swarms), manage themselves autonomously and have the intelligence to dispose of themselves if they are jeopardizing the mission.

The potential client described a scenario in which they would benefit from a collaboration of many small, intelligent satellites; for instance if a satellite's ability to take a reading at a particular set of coordinates is obstructed, then rather than waiting until it reaches that point again, it should be able to contact other satellites, one of which would change its trajectory to pass this point and collect a reading.

3.1. Proof of Concept Hardware

To meet the requirements of the potential client each satellite must be:

Small; Cheap; Able to communicate; Able to adjust orbit; Intelligent; Autonomous; Autonomic; Able to carry out apoptosis; Developed with redundancy in mind; Solar powered - to be able to operate in space; Energy efficient – to utilize the limited energy available.

Therefore the requirements for the prototype/proof of concept (PoC) equipment were;

Require a low powered, basic computer; No operating system; Need low level control e.g. control with C++; Arduino microcontrollers; Small; Cheap; Designed for prototyping; Large, helpful online community; Many open source libraries available.

Arduinos are carrying out basic control of CubeSats in space currently and as such this seemed the obvious choice. A satellite shell/frame can be 3D printed and we obtained the modelling files required to do so.

We worked with 3x Arduinos for controlling the equipment programmatically and acquired the necessary equipment for the solar power system.

The micro-propulsion systems are impractical to use for a proof of concept, as such we built a rover onto which the equipment can be attached to demonstrate the concept of control of movement.

The Nano-communication systems are impractical to use for proof of concept. We have implemented an infrared communication system instead. We gathered a number of sensors which could be used in a proof of concept demonstration.

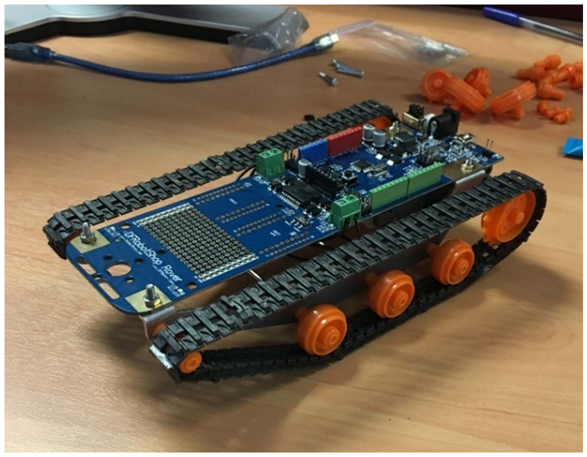


Figure 3 Proof of Concept Hardware, Arduinos

3.2 Proof of Concept Software (developed)

3.2.1 Intelligence – Choosing an agent

To implement the desired system from a software perspective, we needed a communication protocol and a way to determine which recipient of the signal should be allocated the mission.

In our proof of concept system, we need to determine which recipients can propel themselves towards the target location, and evaluate which is best positioned to do this

We created a metric to determine the effort required for each CubeSat to reach the objective point.

We developed C++ code to develop the system and loaded it onto the Arduinos. The system is implemented just as it would be on the real system in space, calculating its orbital velocity based on a given altitude, and using the laws of motion to determine the force, and from that the energy required to change to an appropriate orbit.

3.2.2 CubeSat Displacement Effort Logic

Each CubeSat calculates how many degrees of longitude and degrees of latitude it would have to travel to reach the objective point.

The CubeSat takes into account its direction of travel, and determines if waiting until it has passed over one of the Earth's poles would lead to an easier path.

Each CubeSat calculates the energy required to propel itself into an orbit which will pass the objective point. This value is then adjusted based on the amount of energy available in each CubeSat's propulsion system, and to further penalize those which require more energy. Each CubeSat then takes into account the total time taken to reach the objective coordinates, and so produces the final metric value. This value is comparable between CubeSats allowing for the most suitable to be chosen. The calculations assume that the satellite was moving in an orbit with a fixed longitude. They also assume that to reach the objective coordinates, the CubeSat will use a propulsion system to alter its longitude, and wait until its orbital speed brings it to the correct latitude. The orbital speed of the CubeSat is calculated based on the CubeSat's altitude and therefore the gravitational force acting upon it.

The time available to move to the objective coordinates is therefore the time taken to travel the latitude distance at the orbital speed. For simplicity, the system assumes the method of travel along the longitudinal distance is by accelerating to the mid-point, and then decelerating until the agent stops at the objective point. Therefore the total force required is four times the distance, times the mass, divided by the time available squared. The energy required is the force times the distance.

We normalized the produced value to be between zero and one, whereby those requiring less energy receive a higher score, and those which require more energy than is available in the propulsion system receive a score of zero. The results are displayed in Fig. 4 as the Proportional Displacement Effort.

The time taken is also important, not just the energy required. We developed a Proportional Time metric by taking the proportion of a full orbit required to reach the objective coordinates, shown in Fig. 4 as the Proportional Time Metric.

The Agent Candidacy Score is calculated by multiplying together the Displacement and Time metrics (Fig. 4).

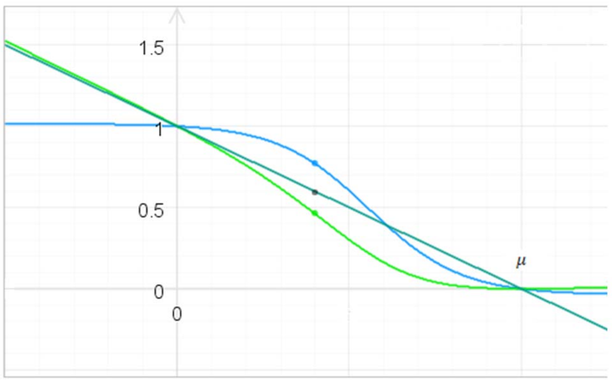


Figure 4 Agent Candidacy Score

Key:
Blue line: Proportional Displacement Effort Metric
Dark Green: Proportional Time Metric
Light Green: Agent Candidacy Score
μ = The energy available in the propulsion system

3.2.3 Communication System

Now that a metric was developed, the focus moved on to the means by which the CubeSats would communicate with each other, compare metric scores and allocate the task.

The CubeSats must take into account that; they will not always be in communication range with each other; some CubeSats may already have a task allocated to them and; some CubeSats may be malfunctioning or destroyed.

We implemented a system which establishes a temporary connection between all satellites within range. The agents then determine who is functioning correctly, and those who are not carry out apoptosis (pre-programmed death).

The agents which are in range and are functioning then calculate a score for themselves, and the agent with the best score has the mission allocated to them. The CubeSat which detects a disturbance broadcasts its coordinates. Those which receive the coordinates respond with their scores. The initiator of the communication responds to scores with acknowledgements.

The initiator compares all scores and contacts the most viable candidate to tell them to undertake the mission.

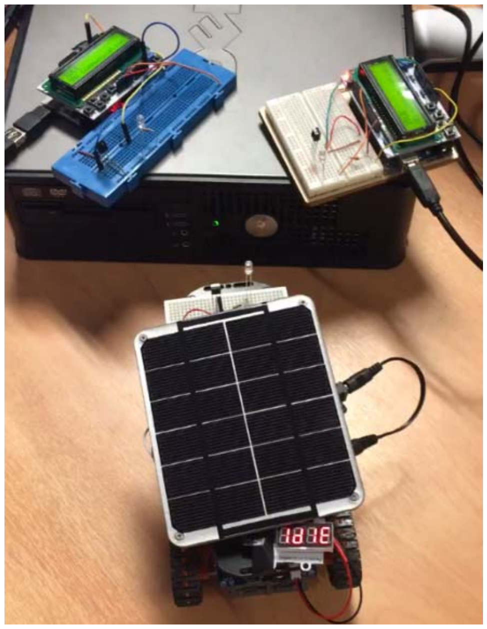
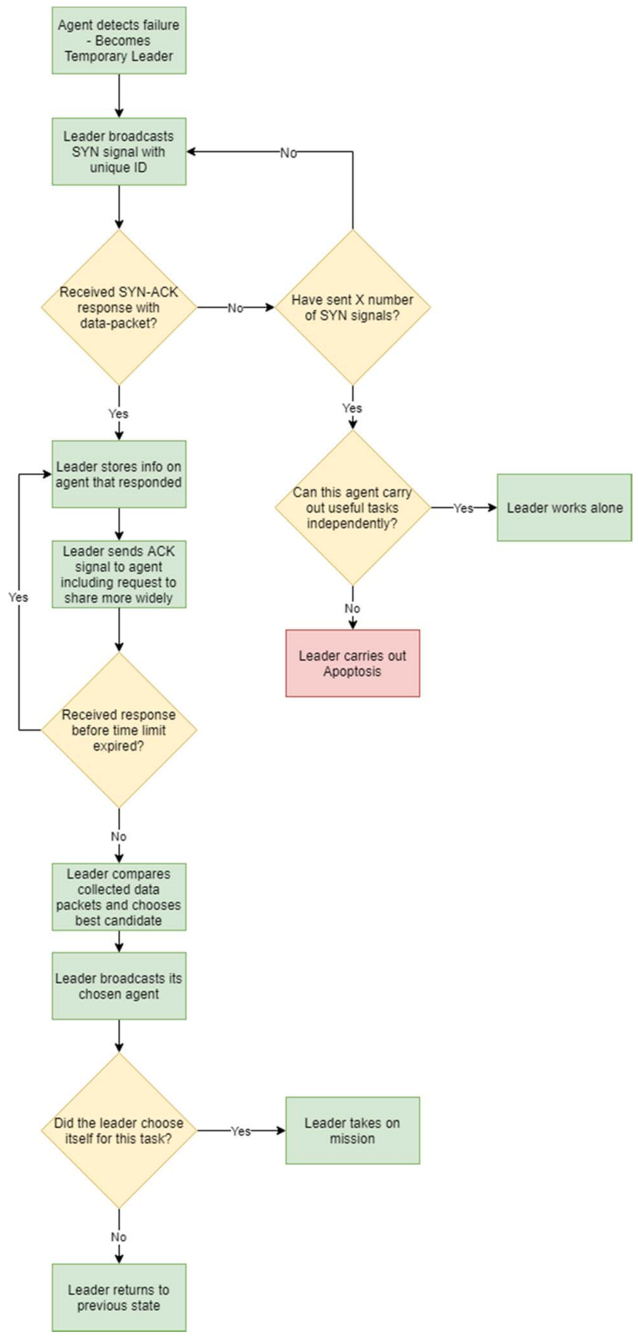


Figure 5 “idle” An idle ‘CubeSat’

If no scores are received, all other CubeSats must be out of range, or malfunctioning or the initiator's transmitter/receiver is broken. If it can be established its transmitter is broken, the initiator carries out apoptosis.

If a CubeSat which transmits a score doesn't receive an acknowledgement, its transmitter is broken, or the initiator's receiver is broken. If it can be established that the CubeSats transmitter is broken it will carry out apoptosis.

Figure 6 Temporary Leader Control Flow



Agent detects failure - Becomes Temporary Leader
Leader broadcasts SYN signal with unique ID
No
Received SYN-ACK response with data-packet?
No
Have sent X number of SYN signals?
Yes
Yes
Leader stores info on agent that responded
Can this agent carry out useful tasks independently?
Yes
Leader works alone
Leader sends ACK signal to agent including request to share more widely
Yes
No
Leader carries out Apoptosis
Received response before time limit expired?
No
Leader compares collected data packets and chooses best candidate
Leader broadcasts its chosen agent
Did the leader choose itself for this task?
Yes
Leader takes on mission
No
Leader returns to previous state

3.2.4 Agent Selection Control Flows

A sample of the control flows underlying the agent selection process are presented in Figures 6 & 9.

3.2.5 Demonstration 1

The first demonstration of the PoC software in action demonstrating the autonomic (self-managing and self-coordinating of this small constellation) we had 3x Arduino connected to an infrared transmitter, an infrared receiver and have a screen to display the state that they are currently in (in Fig. 5). There are 5 potential states. The Arduino is idle if it does not have a mission and is not undergoing communication. It is the leader if it is transmitting the objective coordinates, receiving the scores from the other Arduinos, transmitting acknowledgements, comparing the scores and transmitting the mission details to chosen candidates. It is in the candidate state between receiving the objective coordinates and receiving the ID of the chosen candidate. It is in the active state if it has been chosen for a mission and is moving towards the objective coordinates, and finally, it is dead if it has identified a non-recoverable fault and has carried out apoptosis.

In this first demonstration all 3x Arduino's are in the idle state. A disturbance was injected (press of a button) and that Arduino becomes the leader, the others receive the objective coordinates and become candidates. The Arduino on the right has been given the closet coordinates to the objective location with the leader now being the furthest away. The Arduino on the right becomes active and carries out the mission. At the next stage of the demonstration the Arduino on the right is already carrying out a mission, so the Arduino attached to the Rover is the closest non-active agent and gets assigned the next mission. Finally all but the leader is active so the leader must choose itself for the mission.

3.2.6 Demonstration 2

Following on from Demo 1, in this demonstration (Fig. 6), all the Arduinos are idle but the Rover has lost its transmitter (simulated by physically pulling it out of the board). All Arduinos will receive the signal, calculate the score and try to transmit it, but the Rover identifies its transmitter is broken and carries out apoptosis. The Arduino on the right is assigned the mission as it has the highest score and the objective coordinates are transmitted again. Since the Rover is dead and the other Arduino is on a mission, only the leader is available and so carries out this mission.

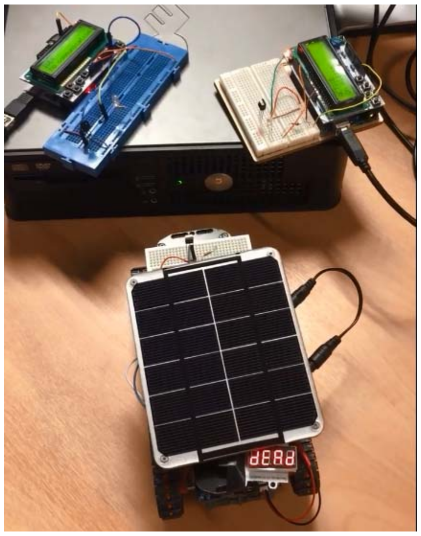


Figure 7 "**dead**"; A damaged sat that has lost its right to operate and has had its apoptotic mechanism triggered (pre-programmed self-destruct)

3.2.6 Demonstration 3

In this demonstration (in Fig. 7), all agents start in the idle start, but the agent which detects a disturbance has lost its ability to communicate. When the leader agent tries to transmit the objective coordinates, it will wait for responses from the other agents, realise it is unable to communicate and carries out apoptosis, not affecting any of the other satellites.

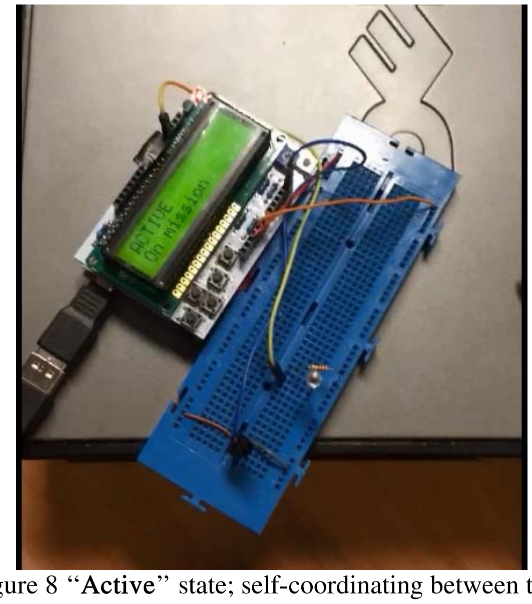


Figure 8 "**Active**" state; self-coordinating between the Arduinos.

3.2.7 Demonstration 4

This demonstration shows an implementation of the concept of soft apoptosis or autonomic quiescence. There may come times when a satellite needs to reserve power for critical functions and should base its power usage on its battery level or the intensity of light available for power generation by solar panels. It may need to temporarily go in to a non-responsive, no power state. Here the rover based agent is monitoring the light intensity and has found it is too low for its needs. It transfers into the dormant sleep state and will not process communications from other agents. Only the other available satellites will be considered for carrying out upcoming missions.
We then increase the light intensity so the Rover has access to more power from the solar panel. It therefore, awakens from its dormant state, receives the mission signals, finds it is the best candidate and is allocated the task.

3.2.8 Collision Avoidance

With the available propulsion systems, sensors and programmatic control over the satellites actions, we could take the autonomous self-management of the system further.

Space junk is an issue for all satellites. NASA recently stated that it had to move satellites to avoid collisions with CubeSats [19].

Therefore we considered the issue of collision detection and avoidance as part of the autonomic self-management system and implemented a demonstration of how with sensors and a propulsion system, a CubeSat can detect obstacles and adjust their course to avoid them.

3.3 Proof of Concept Outcomes

The code we have developed can be used in a real system, as we have implemented it on Arduinos which are currently in orbit controlling CubeSats. We have sensory monitoring, a communication protocol, general autonomic management, swarm coordination and apoptotic implementations.

Since we cannot demonstrate real propulsion systems in action, we decided to use the Arduinos to control miniature rovers instead. This would be representative of its actions to control its own orbit. However unlike the rest of the code, any code developed to control the rover would not apply to the real system; and rovers moving in two dimensional space do not provide an accurate demonstration of a system moving in three dimensional space, in orbit around the Earth.

We have successfully controlled sensors and actuators with equipment used in similar systems in space.

We have successfully integrated a solar powered approach into the system.

We have created a TRL-level 3 prototype of a fully implemented system which had attracted interest from a potential client.

We have developed libraries which could be used in a fully implemented system, applying the previous novel research (TRLs-1&2) on autonomic and autonomous computing and apoptosis.

We appreciate much of this is qualitative claims concerning successful experiments. To move further up the TRL levels we will need to develop this further on a wider range of CubeSat equipment while the industry waits on affordable breakthroughs on SmallSat propulsion systems.

4. Conclusion and Future Works

We have made the case previously that all computer-based systems should be Autonomic [18] and Apoptotic [17], especially as we increasingly move into a vast pervasive and ubiquitous environment. This should cover all levels of interaction with technology from data, to services, to agents, to robotics. With recent headline incidents of credit card and personal data loses by organizations and governments to the Sci-Fi nightmare scenarios now being discussed as possible future, programmed death by default becomes a necessity.

We're rapidly approaching the time when new autonomous computer-based systems and robots should undergo tests, similar to ethical and clinical trials for new drugs, before they can be introduced, the emerging research from Apoptotic Computing and Apoptotic Communications may offer that safe-guard.

Acknowledgements

The authors were supported by the Ulster University's Computer Science Research Institute & School of Computing. Some of the research described in this paper is patented by Roy Sterritt (Ulster University) and Mike Hinchey (Lero—the Irish Software Research Centre) through NASA and assigned to the US government.

References

[1] .R Sterritt, MG Hinchey, (Mar 2010) "SPAACE IV: Self-Properties for an Autonomous & Autonomic Computing Environment – Part IV A Newish Hope", Proceedings of AA-SES-IV: 4th IEEE International Workshop on Autonomic and Autonomous Space Exploration Systems (at SMC-IT), Pasadena, CA, USA, June 2009, in "Proceedings of the Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems (EASe 2010)", IEEE CS Press, Pages 119-125

[2] R Sterritt, MG Hinchey, (Dec 2004) "Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?", Proceedings of Third NASA-Goddard/IEEE Workshop on Formal Approaches to Agent-Based Systems (FAABS III), Washington DC, April 26-27, 2004, in "LNAI 3228", Springer-Verlag, Pages 262-270, doi: 10.1007/978-3-540-30960-4_18

[3] R Sterritt, MG Hinchey, (Apr 2005) "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions", Proceedings of IEEE Workshop on the Engineering of Autonomic Systems (EASe 2005) at 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005), Greenbelt, MD, USA, , 3-8 April, 2005, Pages 506-511

[4] R Sterritt, MG Hinchey, (Jun 2005) "From Here to Autonomicity: Self-Managing Agents and the Biological Metaphors that Inspire Them", Proceedings of Integrated Design & Process Technology Symposium (IDPT 2005), Beijing, China, 13-17 June, Pages 143-150

[5] R Sterritt, MG Hinchey, (May 2006) "Biologically-Inspired Concepts for Autonomic Self-Protection in Multiagent Systems", Proceedings of 3rd International Workshop on Safety and Security in Multi-Agent Systems (SASEMAS 2006) at AAMAS 2006, Hakodate, Japan, 8 Ma

[6] J. Newell, "Dying to live: why our cells self-destruct," Focus, Dec. 1994.

[7] Y. Ishizaki, L. Cheng, A.W. Mudge, M.C. Raff, "Programmed cell death by default in embryonic cells, fibroblasts, and cancer cells," Mol. Biol. Cell, 6(11):1443-1458, 1995.

[8] M Sluyser, (ed) "Apoptosis in Normal Development and Cancer". Taylor & Francis, London, 1996

[9] R Sterritt, DW Bustard DW, "Towards an Autonomic Computing Environment", Proceedings of IEEE DEXA 2003 Workshops - 1st International Workshop on Autonomic Computing Systems, Prague, Czech Republic, September 1-5, 2003, Pages 694-698

[10] M.S. Greenberg, J.C. Byington, T. Holding, D.G. Harper, Mobile Agents and Security, IEEE Communications, July 1998.

[11] S. A., J. Mica, J. Nuth, G. Marr, M. Rilee, M. Bhat, ANTS (Autonomous Nano-Technology Swarm): An Artificial Intelligence Approach to Asteroid Belt Resource Exploration, Curtis, International Astronautical Federation, 51st Congress, October 2000.

[12] P.E. Clark, S. Curtis, M. Rilee, W. Truszkowski, J. Iyengar, H. Crawford, "ANTS: A New Concept for Very Remote Exploration with Intelligent Software Agents", Presented at 2001 Spring Meeting of the American Geophysical Union, San Francisco, 10-14 December 2001; EOS Trans. AGU, 82 (47), 2001.

[13] NASA ANTS Online, http://ants.gsfc.nasa.gov/

[14] Sterritt, R., Towards Autonomic Computing: Effective Event Management, *Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW)*, Maryland, USA, December 3-5, IEEE Computer Society, pp 40-47.

[15] Sterritt, R. and Bustard, D.W., Autonomic Computing: a Means of Achieving Dependability? *Proceedings of 10th IEEE International Conference on the Engineering of*

Computer Based Systems (ECBS '03), Huntsville, Alabama, USA, April 7-11, IEEE CS Press, pp 247-251.

[16] J. Klefstrom, E.W. Verschuren, G.I. Evan, c-Myc Augments the Apoptotic Activity of Cytosolic Death Receptor Signalling Proteins by Engaging the Mitochondrial Apoptotic Pathway, *J. Biol Chem*,. **277**:43224-43232, 2002

[17] Sterritt R, "Apoptotic Computing", IEEE Computer, January 2011 . pp. 37-43

[18] Sterritt, R. and M. Hinchey, "Why computer-based systems should be autonomic," 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05), Greenbelt, MD, USA, 2005, pp. 406-412. doi: 10.1109/ECBS.2005.75

[19] Witze, A., "The quest to conquer Earth's space junk problem", Nature, 5th Sept. 2018.

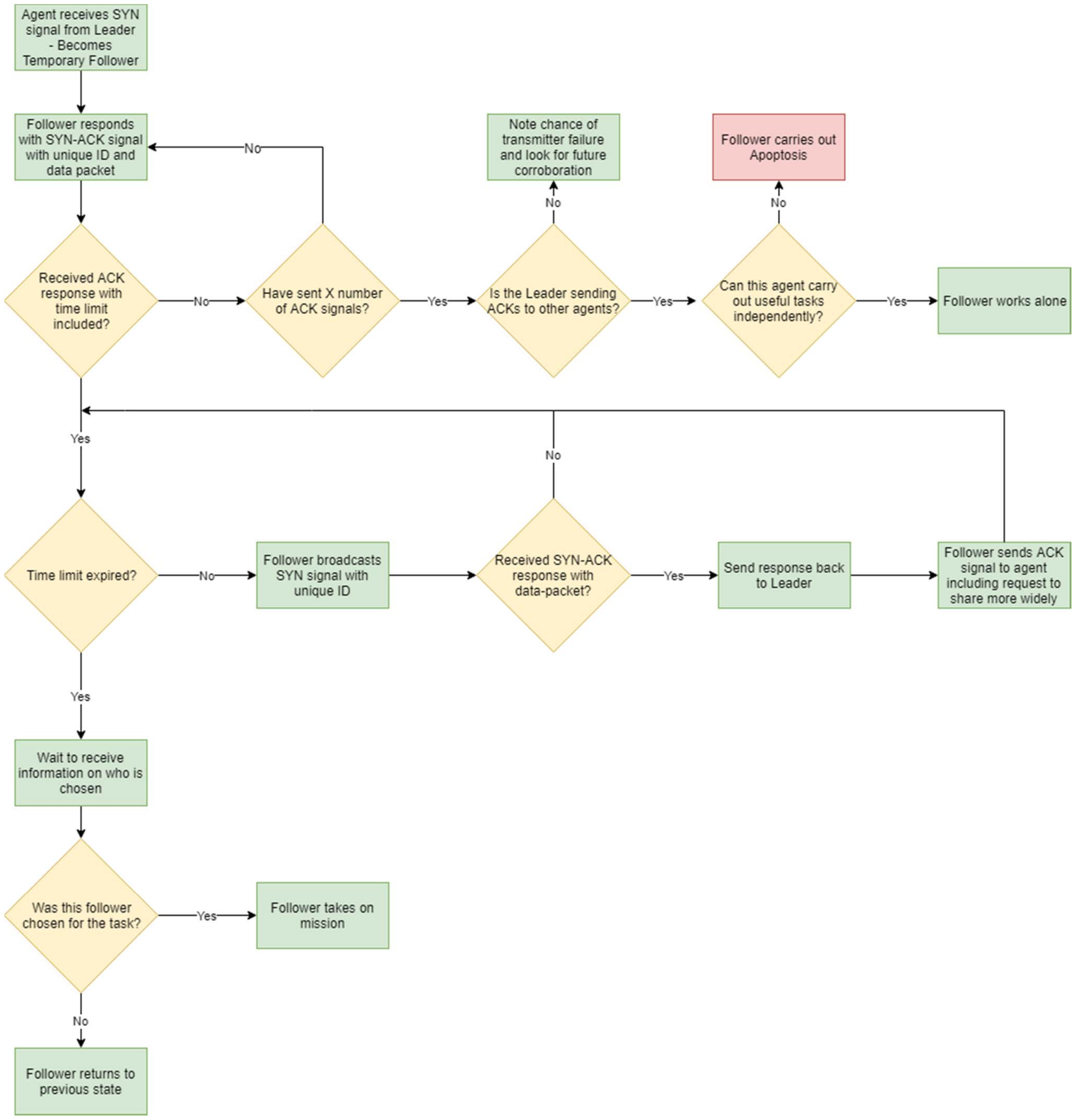

Agent receives SYN signal from Leader - Becomes Temporary Follower
Follower responds with SYN-ACK signal with unique ID and data packet
No
Note chance of transmitter failure and look for future corroboration
Follower carries out Apoptosis
No
No
Received ACK response with time limit included?
No
Have sent X number of ACK signals?
Yes
Is the Leader sending ACKs to other agents?
Yes
Can this agent carry out useful tasks independently?
Yes
Follower works alone
Yes
No
Time limit expired?
No
Follower broadcasts SYN signal with unique ID
Received SYN-ACK response with data-packet?
Yes
Send response back to Leader
Follower sends ACK signal to agent including request to share more widely
Yes
Wait to receive information on who is chosen
Was this follower chosen for the task?
Yes
Follower takes on mission
No
Follower returns to previous state

Figure 9 Temporary Follower Control Flow