# Achieving Trustworthy Autonomous Systems through Autonomic and Apoptotic Computing

## Keynote at ICAS 2020

Roy Sterritt
School of Computing, Faculty of Computing, Engineering and the Built Environment.
Ulster University
Newtownabbey, Northern Ireland
email:r.sterritt@ulster.ac.uk | ORCID 0000-0002-4035-9363

***Abstract*— This paper considers the issue of trust in Autonomous Systems. This is a challenge as these systems are already deployed across many industrial sectors in specialised and controlled conditions with little focus on trustworthiness. When unexpected or uncontrolled situations are introduced into the environment, with a probable high level of interaction with people, the resulting potential for unexpected and/or undesirable results is significant. This paper reflects upon the Autonomic Computing (self-managing) paradigm, and the Apoptotic Computing (pre-programmed death as a safety mechanism) paradigm by presenting some of our research utilizing both, as a potential contribution to achieve Assured and Trustworthy Autonomous Systems.**

***Keywords—Autonomic Computing; Apoptotic Computing; Autonomous Systems; Trustworthy Autonomy; Assured Autonomy; Autonomy.***

## I. INTRODUCTION

Autonomous systems are already developed and deployed across industrial sectors in specialised and controlled conditions with little focus on trustworthiness [1]. When autonomous systems are used in an uncontrolled environment, where there is a high level of interaction with people and a much larger number of variables, the resulting potential for unexpected and/or undesirable results is non-negligible [1]. These unanticipated events could have a very significant negative impact on the acceptability and thus compromise widespread deployment of autonomous systems [1]. For society to use and benefit from autonomous systems, people need to trust them. This means that the autonomous systems need to function as expected for their purpose, and they need to be designed and tested to ensure that they work consistently and safely and that they are appropriately developed within a legal, ethical and social context. Trust will only be enabled through technical advances conducted in specific societal circumstances [1]. To ensure that autonomous systems can be trusted, and ultimately adopted by society, fully integrated advances in both technical, and social sciences and humanities research are needed [1]. For example, research in logic, autonomy and intelligence and engineering (robotics and vehicles) is needed, but these technical developments must be carried out in the context of fundamental social sciences and humanities research across psychology, sociology, economics, ethics, philosophy, law, political science, international studies, innovation management and science and technology studies. The engagement of multiple disciplines in this endeavour, alongside regulators and the public, is key to ensuring that autonomous systems are developed to be used in real-world situations [1].

The hypothesis presented in this paper is that Trustworthy Autonomous Systems (TAS) can be (partially) achieved through Autonomic Computing extended with Apoptotic Computing.

The rest of the paper is structured as follows. In Section 2, the Autonomic Computing and Autonomic Communications paradigms are recapped, then, in Section 3, the Apoptotic Computing paradigm is summarised before presenting some of our research utilising both in Section 4. Section 5 then concludes the paper with some observations.

## II. AUTONOMIC COMPUTING AND COMMUNICATIONS

In 2001, IBM researchers predicted that by the end of the decade the IT industry would need up to 200 million workers, equivalent to the entire US labor force, to manage a billion people and millions of businesses using a trillion devices connected via the Internet. Only if computer-based systems became more autonomic—that is, to a large extent self-managing—could we deal with this growing complexity, and they accordingly issued a formal challenge to researchers [2]. Over the two decades since Autonomic Computing has become a paradigm allowing the advanced automation of system management. In effect, it is a specialisation of autonomous systems – the autonomy of the management of the system itself.

The vision of autonomic computing represents a surprising combination of revolution and retrenchment. By focusing on total costs of ownership for enterprise systems, Kephart and Chess [3] highlighted the central impact that IT systems can have on the core economics of modern businesses. Indeed, the deployment, maintenance, and evolution of enterprise systems often require enormous efforts by extremely valuable staff, whose successes add little visible business value but are nevertheless vital and whose failures can be catastrophic for the whole enterprise. Autonomic computing, in its broadest sense, seeks to reduce the need for such heroic efforts and their consequential risks.

The most widely recognized elements of autonomic systems are the so-called self-* properties: For systems to be self-managing they should be self-configuring, self-healing, self-optimizing, and self-protecting and exhibit self-awareness, self-situation (environment and context awareness), self-monitoring, and self-adjustment [4]-[6]. Despite their seeming simplicity, these goals mask a complex interaction between the

behaviors of systems and their goals, users, and relationships with the external environment. We can only optimize a system against some external criteria, so self-optimization implies that these criteria are made available in some way to the management system. Moreover, composition and analysis of systems probably imply that the criteria be explicit, symbolic, and machine-readable rather than embedded implicitly into algorithms [7].

In thinking of systems rather than simply of machines, we must also consider communications a component of the problem space [4], the most notable omission from Kephart's and Chess's vision. Mikhail Smirnov [8] propounded the notion of Autonomic Communications, not only based on IBM Autonomic Computing, but David Clark and colleagues' call for a knowledge plane for the Internet [9], and which became an active research topic in itself [10], especially in Europe, where it has received considerable support from the EU's Framework programs. Considering communications as well as computing naturally leads to an exploration of the interplay of these different aspects [7].

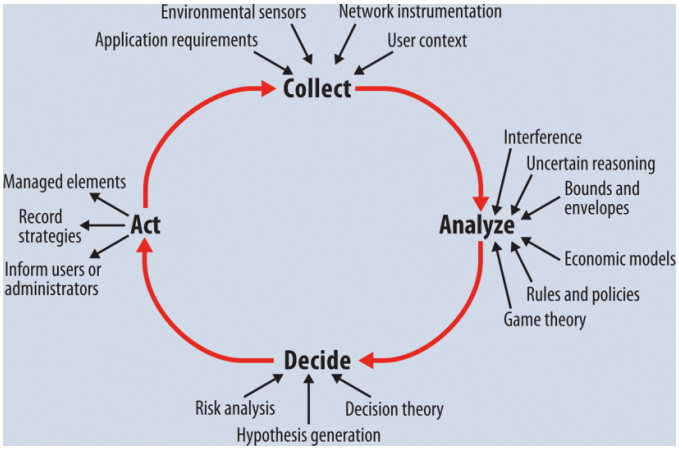


Figure 1. Collect-Analyze-Decide-Act control loop [7][10]

As Figure 1 shows [10][7], providing self-monitoring and self-control suggests the application of control theory—expressing a control action derived from a system's observed behavior against a model of intended or expected behavior. Researchers have successfully applied such techniques to, for example, power management, to achieve clear closed form representations. However, it is less clear whether the techniques can be applied more broadly in areas where the control domain changes dynamically and provide an assured or trusted autonomy.

## III. Apoptotic Computing

Apoptotic Computing and Apoptotic Communications are 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.

The Apoptotic Computing project, first started back in 2002 [11]-[15], involves working towards the long-term goal of developing Programmed Death by Default for Computer-Based Systems to provide for this foreseen future. 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) [16][17].

With biological systems, 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 [18], 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 [19]. The process has also been nicknamed 'death by default' [20], where cells are prevented from putting an end to themselves due to constant receipt of biochemical 'stay alive' signals. The key aspect of apoptosis is that the cell's self-destruction takes place in a programmed and controlled way; 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 [21].

Further investigations into the apoptosis process [18] 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 [19]. 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 [19].

Apoptotic Computing takes its inspiration from the biological apoptosis, and can be implemented as part of the self-management of Autonomic Computing. The following sections will discuss some of the research conducted into these.

## IV. Autonomic and Apoptotic Computing Case Studies

As has been stated, the hypothesis presented in this paper, is that TAS can be (partially) achieved through Autonomic Computing extended with Apoptotic Computing.

We consider to truly achieve Autonomy, design and development of Autonomous Systems benefits from separation of concerns, namely splitting the advanced automation of the task/mission/user oriented goal of the system from the advanced automation of the management and running of the actual system. The former represents self-governance/autonomy of the system (and what users focus in on) and the later represents self-management/autonomicity. A simple example of such is self-driving (autonomous) cars. The user perception is cars that drive themselves; which represents the task/mission/goal, the split in roles is that the autonomic system takes care of is the actual management of the system,

are the sensors, actuators, algorithms, and processors working correctly? Requiring re-configuring to improve performance or reflex reactions of self-protection and self-healing if a tire blows out. Division of labor into Autonomous and Autonomic layers in the design and development effort should enable a more trustworthy system. The autonomicity can be added to provide assurance at the system, application and/or component level.

### A. *System level Trust and Assurance Cases*

Motivated by an incident at a Smart (elderly care) Home where a resident with dementia left the building undetected, unaccompanied and not dressed for the external elements. Thankfully, the older resident was found quickly, but a google search on this incident found cases where similar events occurred nationally, where the care home (or fold) has Smart technology yet dementia residents leave undetected and unfortunately were not found before hypothermia set in resulting in death. In our case, the issue was a faulty fire alarm, where for safety the fire doors cannot be locked from the inside but are alarmed for when opened. The faulty alarm had not been detected. This obviously raises trustworthy issues for this type of autonomous system. We researched how autonomic computing helps provide assurance in this scenario.

In this research, an approach to ensuring fault tolerance in intelligent environments for the elderly through the provision of mobile sensor substitution (via a robot) in the event of the detection of anomalous static (smart home) sensor behaviour was investigated. One stream focused on the monitoring of an external door in an intelligent care home environment. A mobile robot equipped with an array of ultrasonic sensors is dispatched to monitor the door state and report a change in state to a central server. For each door state, there are consistent changes in the sensor readings identified in the course of the experiments carried out within this work. The use of ultrasonic sensors provides a viable substitution option that can assist a central system in deciding whether a care assistant or maintenance engineer is required to resolve the anomalous static sensor behaviour.

A robot to investigate static sensors and then act as "watchbot" filling in for the defected door sensor with its sonar sensors until a technician can arrive (potentially days later) and replace the faulty sensor may seem like a "sledge-hammer to crack a nut" solution but there was a wider context to this autonomic solution that the robot would also be proactively testing the sensors around the smart-home as well as determining conflict in sensor readings such as has the elderly person fallen at the front door or is it a parcel/dog lying on the sensor mat constantly alarming to the system? Figure 2 depicts a high-level overview of the autonomic solution providing trustworthiness and assurance to the autonomous system (smart home). Note NASM and EHSM in the figure stand for Normal Activity State Machine and Error Handling State Machine which were FSMs designed with novel built-in adaptability. More details can be found in [23]- [25].

Another critical autonomous system we researched from an autonomic perspective, adding assurance and trustworthiness at the system level, was a biometric enabled prison/correctional institute system [26]. This system was already extensively robust (in a FTC–fault tolerant computing way) with a watchdog/sentential polling components in the system. Yet, we investigated better (autonomic) ways of designing the system to provide more proactive than reactionary fault tolerance [27] to then attempt to move towards next-gen prison systems [28][29] beyond high granularity of prisoner tracking (essentially knowing which area they have biometrically entered/exited) to a much finer grained self-management of the system, ensuring a trustworthy system for inmates, staff and visitors [30].

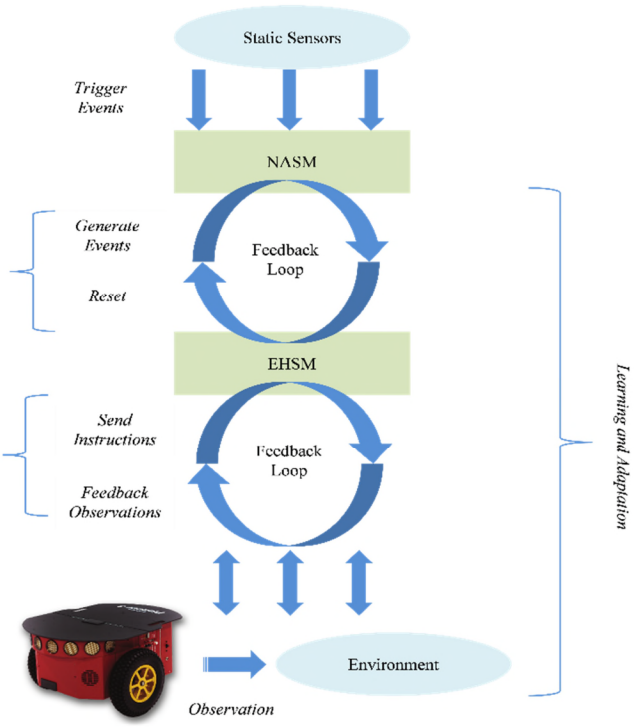


Figure 2. Autonomic Robot ensuring TAS in a Elderly care Smart Home [23].

We have extensively discussed in the literature the research with NASA into how Autonomic Computing can provide assurance for Swarm-Based Space Exploration Systems (most notably ANTS – Autonomous Nano Technology Swarms – concept mission), for instance [31]. The following section though will highlight how this also provided assurance at the application and component (or even Nano) level.

### B. *Application and Component level Trust and Assurance Cases*

Autonomic Computing can provide assurance at all levels of an autonomous system through its feedback control self-management. Apoptotic Computing, with its pre-programmed death tends to provide assurance at the component level and possibly the application level (rarely would one want a system level self-destruct (apoptotic) mechanism).

We have researched introducing apoptotic measures into Agent-Based Systems, Autonomic (Self-managing and adaptive) Systems and Swarm Based Space Exploration Systems as highlighted earlier [11]-[17]. At an application level, we have applied this to Robotics (Apoptotic Robotics) [32]. In the wider view of this stream of research, Autonomic

Robotics, we have carried out several case studies investigating self-* healing strategies and a confirmatory case study;

- Robot Wheel Alignment Fault [33][34]
- Robot Sonar Sensor Faults [35]
- Robot Battery Degradation Fault [36]
- Stereo Vision Camera Fault – Confirmatory Case Study [37]

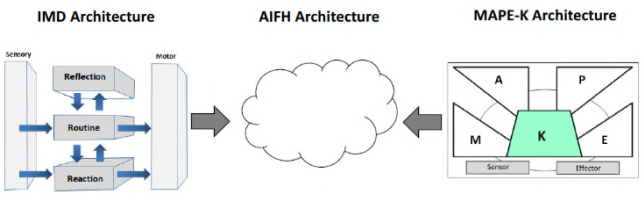


Figure 3. Towards an Autonomic Robotics Architecture.

The lessons learnt from these case studies enabled an Autonomic Robotics Architecture to be derived from IMD (Robotics) and MAPE (Autonomic Computing) architectures (Figure 3 and Figure 4), which is also referred to as AIFH: Autonomic Intelligent Fault Handling architecture. More detail can be found in [37][38].

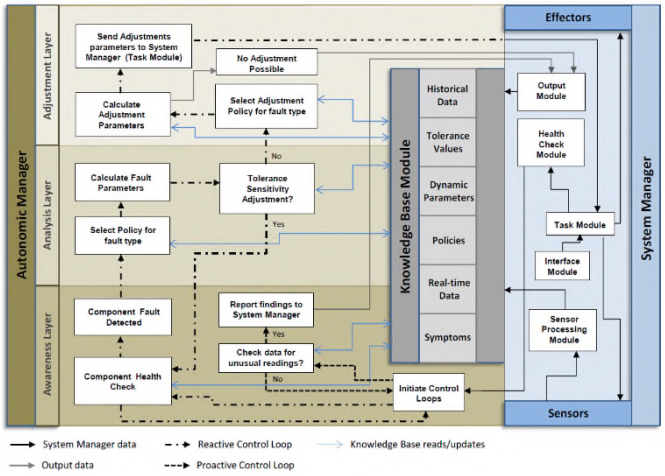


Figure 4. Autonomic Architecture for Fault Handling in Mobile Robots

More recently, we have been investigating it with application to CubeSats/NanoSats/PicoSats.

Figure 5. Autonomic and Apoptotic CubeSat research.

In the first instance, the research was to build in the apoptotic pre-programmed death (component level) to the CubeSat in an attempt to prevent adding to the proliferation of Space Debris/Space Junk (Figure 5) [39].

In the second instance, with a broader perspective, this research has widened into developing a "Cubesat Autonomicity Capability Model (CACM)" as a roadmap for future autonomic cubesat development including autonomic cooperation in constellations, thus addressing trust at the system level once again, while having a "killswitch" (Apoptotic Computing) pre-

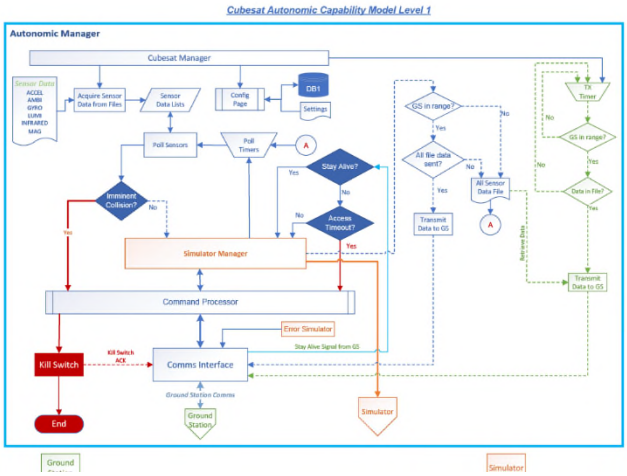


Figure 6. Part of CubeSat Autonomic Capability Model (CACM) – Level 1

programmed at the component level [40][41], for instance, Figure 6.

## V. Conclusion

The hypothesis presented in this paper, was that Trustworthy Autonomous Systems (TAS) and Assured Autonomous Systems can be (partially) achieved through Autonomic Computing extended with Apoptotic Computing.

The research carried out in the noughties on Autonomic and Apoptotic Computing with NASA GSFC, briefly recapped here, started in the first instance as expanding on the NASA Formal Approaches to Swarm Technologies (FAST) project which was funded by the NASA Office of Systems and Mission Assurance (OSMA) through its Software Assurance Research Program (SARP). The concern that was attempting to address here was the future concept missions of potentially 1000s of autonomous adaptive craft and how can you assure their operation. The apoptotic (pre-programmed nano-craft death) became the ultimate assurance with autonomic paradigm ensuring the trustworthy self-management of the mission assets. This work lead to 16 patents [43], such as [44].

This assurance and trustworthy via autonomic and apoptotic computing theme carried on throughout reflection on our other research; from elderly care smart homes to prison systems, robotics and returning to space with cubesats and the derivation of a generic architecture and a capability model. Yet the larger, more difficult task of combining these point solutions into wider autonomous systems remains. More consideration must be given to integrating solutions, and to choosing solutions from the range of possibilities— to *trustworthy and assured autonomous and autonomic systems*

*engineering*, in other words. Without the development of such an approach, we will simply rediscover the risks of feature interaction at a higher level, and in a way that is so dynamic as to be resistant to debugging and testing. We are confident, however, that the foundation exists to construct a systems theory and practice from which we can engineer trustworthy autonomous solutions for the next generation of enterprise and sensor systems.

## ACKNOWLEDGMENT

The author is supported by the Ulster University's Computer Science Research Institute and 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, formerly NASA GSFC) through NASA and assigned to the US government. Thanks to all colleagues and in particular my PhD students who have done a lot of the heavy lifting, in particular during the tens.

## REFERENCES

[1] EPSRC, "Trustworthy Autonomous Systems Nodes –Outline funding call", ver.9, Oct. 2019

[2] P. Horn, "Autonomic Computing: IBM's Perspective on the State of Information Technology," 15 Oct. 2001, IBM Research.

[3] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," Computer, Jan. 2003, pp. 41-50.

[4] R. Sterritt, "Towards Autonomic Computing: Effective Event Management," Proc. 27th Ann. NASA Goddard Software Eng. Workshop (SEW 02), IEEE CS Press, 2002, pp. 40-47. doi: 10.1109/SEW.2002.1199448

[5] R. Sterritt and D. W. Bustard, "Towards an autonomic computing environment," 14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings., Prague, Czech Republic, 2003, pp. 694-698. doi: 10.1109/DEXA.2003.1232103

[6] R. Sterritt, Innovations Syst Softw Eng (2005) 1: 79. doi: 10.1007/s11334-005-0001-5

[7] S. Dobson, R. Sterritt, P. Nixon and M. Hinchey, "Fulfilling the Vision of Autonomic Computing," in Computer, vol. 43, no. 1, pp. 35-41, Jan. 2010. doi: 10.1109/MC.2010.14

[8] M. Smirnov, Autonomic Communication: Research Agenda for a New Communications Paradigm, tech. report, Fraunhofer FOKUS, 2004.

[9] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski "A Knowledge Plane for the Internet," Proc. 2003 Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (Sigcomm 03), ACM Press, 2003, pp. 3-10.

[10] S. Dobson et al., "A Survey of Autonomic Communications," ACM Trans. Autonomous and Adaptive Systems, vol. 1, no. 2, 2006, pp. 223-259.

[11] R. Sterritt and M. G. Hinchey, "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", SpringerVerlag, pp. 262-270, doi: 10.1007/978-3-540- 30960-4_18

[12] R. Sterritt and M. G. Hinchey, "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, pp. 506-511 doi: 10.1109/ECBS.2005.36

[13] R. Sterritt and M. G. Hinchey, "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, pp. 143-150

[14] R. Sterritt and M. G. Hinchey, "BiologicallyInspired 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, 8-12 May 2006, doi: 10.1007/978-3-642-04879-1_22

[15] R Sterritt and M. G. Hinchey, "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 SMCIT), 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, pp. 119-125, doi: 10.1109/EASe.2010.29

[16] R. Sterritt, "Apoptotic computing: Programmed death by default for computer-based systems," in Computer, vol. 44, no. 1, pp. 59-65, Jan. 2011. doi: 10.1109/MC.2011.5

[17] R. Sterritt and M. G. Hinchey, " Apoptotic Computing: Programmed Death by Default for Software Technologies," in "Software Technology: 10 Years of Innovation in IEEE Computer: 10 Years of Innovation, First,", Wiley, 2018, doi: 10.1002/9781119174240.ch5

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

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

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

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

[22] G. Brady, R. Sterritt and G. Wilkie, "An Investigation into the Viability of a Mobile Ultrasonic Array as a Sensor Substitute in an Autonomic Intelligent Environment," 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, 2013, pp. 577-582, doi: 10.1109/SMC.2013.104

[23] G. Brady, R. Sterritt, and G.Wilkie, "An adaptive approach to self-healing in an intelligent environment," in Proceedings for ADAPTIVE 2014, The Sixth International Conference on Adaptive and Self-Adaptive Systems and Applications, IARIA, May 2014. ISBN 978 1 61208 341 4

[24] G. Brady, R. Sterritt, G. Wilkie, (2015). Mobile Robots and Autonomic Ambient Assisted Living. Paladyn, Journal of Behavioral Robotics, 6(1), 205-217, doi: 10.1515/pjbr-2015-0013

[25] G. Brady, "Autonomic Robot for Assisted Living: Supporting Smart Environment Occupants through Sensor Substitution", PhD diss., Ulster University, 2016.

[26] P. O'Hagan, E. Hanna and R. Sterritt, "Addressing the Corrections Crisis with Software Technology," in Computer, vol. 43, no. 2, pp. 90-93, Feb. 2010. doi: 10.1109/MC.2010.29

[27] R. Sterritt, G. Garrity, E. Hanna, and P. O'Hagan, "Autonomic Agents for Survivable Security Systems", Proceedings of 1st IFIP Workshop on Trusted and Autonomic Ubiquitous and Embedded Systems (TAUES 2005) at EUC'05, in "LNCS 3823", Springer, December 6-9 2005.

[28] C. L. Mulholland, R. Sterritt, P. O'Hagan, and E. Hanna, (Mar 2008) "Tagging and Tracking System for Prisons and Correctional Facilities – A Design Roadmap", Proceedings of Fifth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2008), Belfast, Northern Ireland, 31st March - 4th April, IEEE CS Press, pp 143-153

[29] C. McFarland, R. Sterritt, P. O'Hagan and E. Hanna, "Interfacing with Next Generation Tagging and Tracking Systems for Prisons and Correctional Facilities," 2010 Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems, Oxford, 2010, pp. 43-50, doi: 10.1109/EASe.2010.14

[30] C. McFarland, R. Sterritt, D. W. Bustard, S. I. McClean, and P. O'Hagan, "AARCTIC: Autonomic Analytics Research for Corrections Technology, Institutional and in the Community", 11th IEEE International Conference and Workshops on the Engineering of Autonomic & Autonomous - APL, Laurel, Maryland, USA. Sep 2014.

[31] R. Sterritt, M. Hinchey, C. Rouff, J. Rash and W. Truszkowski, "Sustainable and autonomic space exploration missions," 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06), Pasadena, CA, 2006, 8 p, doi: 10.1109/SMC-IT.2006.78

[32] R. Sterritt, "Apoptotic Robotics: Programmed Death by Default," 2011 Eighth IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems, Las Vegas, NV, 2011, pp. 107-113. doi: 10.1109/EASe.2011.21

[33] M. Doran, R. Sterritt, and G. Wilkie, "Autonomic Wheel Alignment for Mobile Robots". At: 11th IEEE International Conference and Workshops on the Engineering of Autonomic & Autonomous , APL, Laurel, Maryland, USA. IEEE CS. 6 p., 2014

[34] M. Doran, R. Sterritt, and G. Wilkie, Autonomic Self-Adaptive Robot Wheel Alignment. Adaptive 2016: The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications pp. 27-33.

[35] M. Doran, R. Sterritt, and G. Wilkie, Autonomic Sonar Sensor Fault Manager for Mobile Robots. ICACCE 2017 : 19th International Conference on Autonomic Computing and Computer Engineering London, UK, Mar 2017

[36] M. Doran, R. Sterritt, and G. Wilkie, Autonomic Management for Mobile Robot Battery Degradation. ICACCE 2018 : 20th International Conference on Autonomic Computing and Computer Engineering - London, UK, May 2018

[37] M. Doran, R. Sterritt, and G. Wilkie, 'Autonomic Architecture for Fault Handling in Mobile Robots', Innovations in System and Software Engineering, a NASA Journal, Springer Publications, 26 p., ISSE-D-19-00010R1, Apr. 2020. doi:10.1007/s11334-020-00361-8

[38] M. Doran, "Autonomic Architecture for Fault Handling in Mobile Robots", PhD diss., Ulster University, 2020.

[39] R. Palmer and R. Sterritt, "Autonomic & Apoptotic Computing Prototype; Providing Pre-Programmed Death of Cubesats for Avoiding Space JUNK", 2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT), Pasadena, CA, USA, 2019, pp. 78-86, doi: 10.1109/SMC-IT.2019.00015

[40] C. Gama, R. Sterritt, G. Wilkie, and G Hawe, Towards a Cubesat Autonomic Capability Model (CACM): A Road Map. At 2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT 2017), Alcala de Henares, Sept. 2017

[41] C. Gama, R. Sterritt, G. Wilkie, and G Hawe, "Towards a Cubesat Autonomicity Capability Model A Roadmap for Autonomicity in Cubesats", The Tenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2018), Feb. 2018.

[42] R. Sterritt, G. Wilkie, C. Saunders, M. Doran, C. Gama, G. Hawe, and L. McGuigan, "Inspiration for Space 2.0 from Autonomic-ANTS (Autonomous NanoTechnology Swarms) Concept missions" 17th BIS Reinventing Space Conference, 12-14 November 2019, Belfast, Northern Ireland.

[43] NASA, R. Sterritt et al. Patents, https://pure.ulster.ac.uk/en/persons/roy-sterritt/publications/?type=%2Fdk%2Fatira%2Fpure%2Fresearchoutput%2Fresearchoutputtypes%2Fpatent%2Fpatent. [retrieved : July 2020]

[44] NASA, R. Sterritt and M. G. Hinchey, Autonomic and Apoptopic Systems in Computing, Robotics, and Security. [US Patent 8983882], 17th March 2015.