Towards a Cubesat Autonomicity Capability Model

A Roadmap for Autonomicity in Cubesats

Clement Gama, Roy Sterritt, George Wilkie, Glenn Hawe
School of Computing
Ulster University
Jordanstown Campus, Newtownabbey, County Antrim BT37 0QB, United Kingdom
Gama-C1@ulster.ac.uk, R.Sterritt@ulster.ac.uk, Fg.Wilkie@ulster.ac.uk, Gi.Hawe@ulster.ac.uk

***Abstract* — The paper presents an autonomic / self-managing model, which can be used in cubesat development to make cubesats self-configuring, self-healing, self-optimizing and self-protecting. Autonomous and self-managing systems have emerged in multiple domains, e.g., autonomous ground and aerial vehicles. So far, there is no standard model as to how to design and implement self-managing, autonomous systems. In this paper, we are going to look at how autonomic computing can be applied in unmanned systems and how it can be adapted and applied to the cubesat space industry. Spacecraft always operate in remote environments whereby human intervention is infeasible and therefore making them autonomic is not a niche feature, but a required paradigm change for future satellites. An autonomic capability level model for cubesats is proposed in this paper, which can assist cubesat developers gradually increase the use of autonomic features in satellite and cubesat systems.**

Keywords - autonomic computing; autonomicity; apoptosis; cubesat; capability model.

I. Introduction

Autonomic Computing (AC) has been adopted in various technical platforms such that it is no longer about the vision that IBM had in 2001 when they first proposed AC for servers [1][2]. Multiple industries, for example the automotive industry, the Ministry of Defence, the freight industry, and space exploration - to mention a few - are all researching and developing self-managing systems specifically to address complex issues within their domains [2]. In some industries, autonomic systems are referred to as Unmanned Systems (UMSs); and examples include Unmanned Underwater Vehicles (UUVs), Unmanned Aerial Vehicles (UAVs), and Unmanned Ground Vehicles (UGVs) [3].

The purpose of this paper is to investigate the applicability of autonomic computing in cubesats, and introduce a roadmap for future autonomic cubesat development (a Cubesat Autonomic Capability Model (CACM)). Autonomicity in cubesats is a new field currently being researched by universities and other stakeholders around the world. The CACM derives inspiration from the IBM 2001 Autonomic Maturity Model, Autonomy Levels Framework, the Automotive Driving Automation Levels model, and the Capability Maturity Model Integration (CMMI).

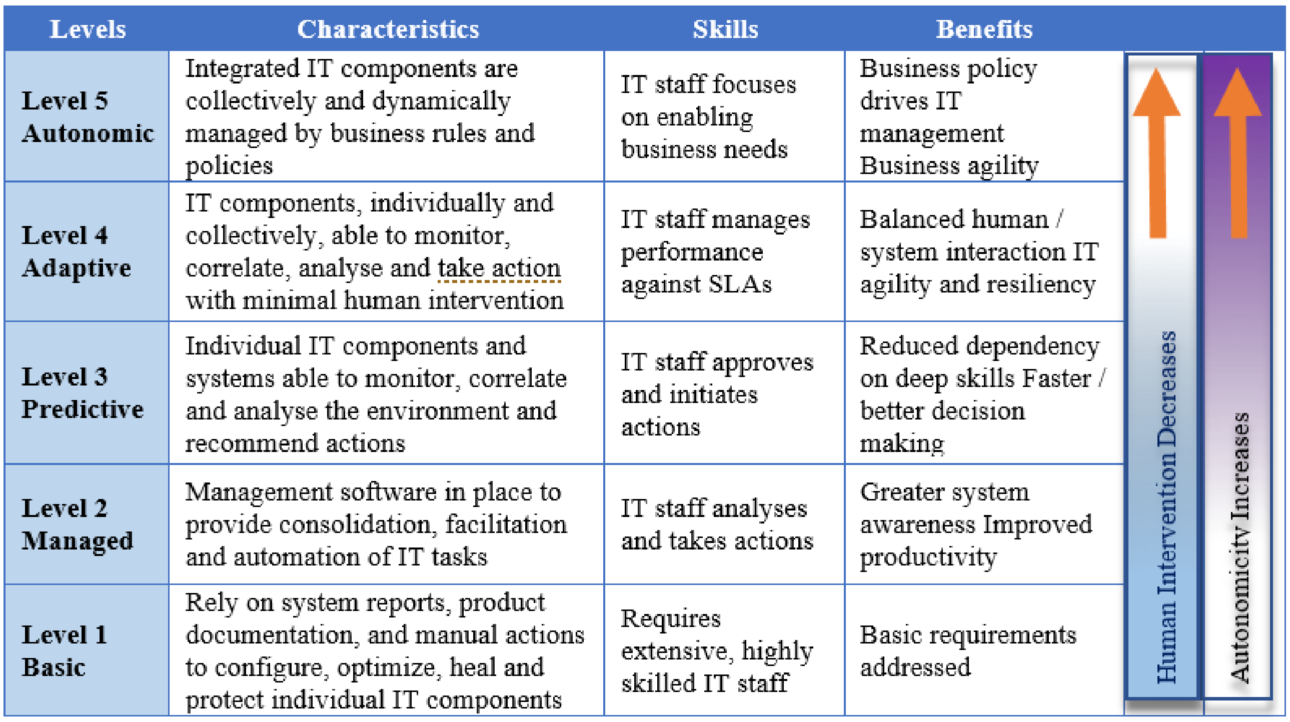
Cubesats are a type of microsatellites / nanosatellites that came out of a collaborative endeavour between California Polytechnic State University and Stanford University in 1999 [4]. The original vision for developing cubesats and standardizing them was to develop the necessary skills for creating satellites intended for Low Earth Orbit (LEO) and also limit the size and number of science instruments that could go on-board spacecraft. The cubesat form factor specification was standardized to 10cm x 10cm x10cm (1U) with a mass of about 1.33kg [5]. Other form factors include 2U (10cm x 10cm x 20cm), 3U (10cm x 10cm x 30cm), 6U, 12U, etc. The low cost and faster development of cubesats has been the result of accelerated technological advances in spacecraft miniaturization in recent years [6].

Traditionally, spacecraft consist the main payload, which conducts space experiments or tasks, and vehicle support systems, like communications, propulsion, attitude determination and control, electric power system, and data storage [7]. Cubesats, however, can only implement some of the features of monolithic satellites due to their physical size limit, electrical power availability and processing power [7]. Nowadays, these microsatellites are utilised to accomplish LEO missions that were previously performed using monolithic satellites and this has resulted in huge mission cost savings [8].

This paper, in Section 2, delves into the review of published material on autonomic computing, autonomy in cars, Autonomy Levels For Unmanned Systems (ALFUS), CMMI and autonomy in cubesats. In Sub-Section “D. Cubesats”, we present a literature review on cubesats as a whole, starting from their inceptions to current development and proposed future use by agencies like NASA. Two hypotheses are presented in Section 3, with specific steps to be followed in investing both hypotheses. Risks in conducting this research are addressed in Sub-Sections 2 and 3. Section 4, outlines the proposed CACM, which will help guide developers who want to design self-managing cubesats. The model is summarised in Table I. Section 5 defines the application of the CACM when developing a kill switch function on cubesats. The kill switch is a de-orbiter feature required to ensure that dead cubesats do not remain in active orbit after their missions are completed or terminated [5].

II. Literature Review

Autonomic computing as defined in the IBM 2001 maturity model, as shown in Figure 1, [1] comprises five maturity levels (Level 1 Basic, Level 2 Managed, Level 3 Predictive, Level 4 Adaptive and Level 5 full Autonomic) [9]. Level 1: At the Basic level, there is heavy reliance on system reports, product documentation, and user intervention to



Levels	Characteristics	Skills	Benefits
Level 5 Autonomic	Integrated IT components are collectively and dynamically managed by business rules and policies	IT staff focuses on enabling business needs	Business policy drives IT management Business agility
Level 4 Adaptive	IT components, individually and collectively, able to monitor, correlate, analyse and take action with minimal human intervention	IT staff manages performance against SLAs	Balanced human / system interaction IT agility and resiliency
Level 3 Predictive	Individual IT components and systems able to monitor, correlate and analyse the environment and recommend actions	IT staff approves and initiates actions	Reduced dependency on deep skills Faster / better decision making
Level 2 Managed	Management software in place to provide consolidation, facilitation and automation of IT tasks	IT staff analyses and takes actions	Greater system awareness Improved productivity
Level 1 Basic	Rely on system reports, product documentation, and manual actions to configure, optimize, heal and protect individual IT components	Requires extensive, highly skilled IT staff	Basic requirements addressed

Figure 1. Derived from the IBM Autonomic Computing Adoption Levels [1].

configure, optimize, recover and protect individual IT components.

Level 2: At the Managed level, management software is used to consolidate, facilitate and automate IT tasks [9]. Level 3: The Predictive level makes limited predictions through monitoring, correlation of individual system [9] components, environmental analysis and recommends user actions. Level 4: Adaptive level, in this level individual and collections of IT components are monitored, correlated and analysed. Corrective action and reconfiguration has minimal human intervention [9]. Level 5: Autonomic level, system components are all integrated and are system managed using business rules and policies stored in a knowledge-base [9].

The lowest level of this capability model is actually not autonomic at all, but instead IT personnel run the show of configuring the AC systems, monitoring them, performing error recovery (healing), fine tuning (optimization) and protecting the systems by anticipating imminent error conditions and taking corrective action before the systems fail [10]. The fifth level, however, in contrast to the basic level, has minimal human intervention, but instead the system is more self-governing and managing. The human actor still intervenes in setting up the policies, which the autonomic system agents use to formulate tasks and goals [10].

A. *Autonomy Levels Framework*

The IBM 2001 autonomic computing maturity model is well suited for environments whereby the systems have ample computing power, enough electric power, have dedicated IT staff. In mobile UMSs, e.g., UAVs, UUVs, and spacecraft, resources are very much limited and therefore a more customised version of autonomic computing is necessary [3].

As a response to the need for a customised autonomic computing model, a voluntary Ad-Hoc Working Group sponsored by the National Institute of Standards and Technology (NIST), comprising government organisations and contractors was formed. The sole purpose of this working group was to develop the ALFUS framework, which was aimed at addressing aerial, underwater and over ground vehicles' autonomic computing issues [11]. The ALFUS framework defines unmanned autonomy using these three categories as shown in Figure 2: Mission Complexity, Environmental Complexity and Human Independence [12].

1) *Mission Complexity (MC)*

Mission complexity depends on the type of unmanned vehicle used in a mission. In ground based vehicles, it could depend on transit systems for both people and goods, which involves moving from one location to another. Mission complexity increases when route distance, optimisation, traffic congestion, and route specificity are taken into consideration [12].

Missions for space exploration and planetary science studies can be complicated by the number of instruments fitted in a vehicle, and also the science tasks and communication latencies. A space mission consists of at least two aspects: the science mission (tasks), and the spacecraft management [12].

2) *Environment Complexity (EC)*

Spacecraft environmental complexities come from space junk, solar flares, and other high energy particles that could damage the spacecraft or some of its instruments. It has to be able to autonomously avoid obstacles, i.e., space debris, other

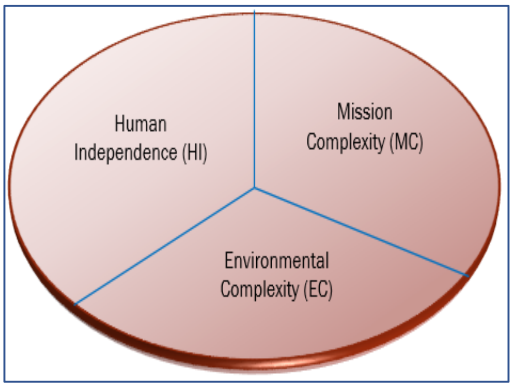


Figure 2. The Three Aspects of the ALFUS framework as defined by the working group [13].

satellites and other objects (meteors, asteroids and comets) [12][13].

In ground based vehicles, complexity comes from the transit network - closed roads, closed lanes, and closed tracks. In some urban areas bus lanes exist to be exclusively used by buses at certain times of the day, and some bus lanes are exclusive to buses all the time, so the vehicle must navigate this complex network at all times. Other constraints come from pedestrians who may or may not adhere to traffic rules and other manned vehicles can pose a threat to UMSs [12].

3) *Human Independence (HI)*

The measurement of human independence in a UMS ranges from partial human control (Hybrid), e.g., in cars, the use of the auto-cruise, which needs a human to activate it – to fully automated sky-trains and automated trams. The human aspect determines the vehicle's level of autonomy [12].

B. *Autonomy in the Automotive Industry*

The automotive industry has joined the autonomic computing development race in an attempt to make self-driving cars. Autonomy in cars has long been a science fiction phenomenon… as Gao et al [14] puts it *"the Firebird IV concept car, which, as the company explained, "anticipates the day when the family will drive to the super-highway, turn over the car's controls to an automatic, programmed guidance system and travel in comfort and absolute safety at more than twice the speed possible on today's expressways"*. This was in 1964 during the New York World's Fair exhibition by General Motors [14].

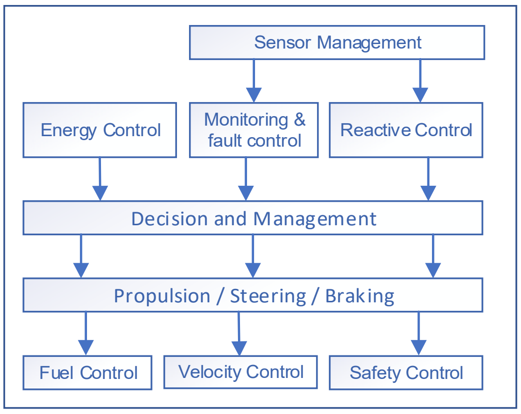


Figure 3. An adapted modular architecture of an autonomous car [16].

The idea of cruising safely on motorways was a far-fetched dream in the 1960s, but that vision is coming closer to a reality, however, there are still problems to overcome. According to the Spectrum IEEE publication [15] an autonomous car failed every 3 hours during test experiments in California in 2016. The Department of Motor Vehicles released a report in January 2017 detailing over 2,500 self-driving car failures in 2016 alone [15].

Autonomous cars control the steering wheel, acceleration, brakes, gears and the clutch using sensory information from multiple sensors. Fig. 3 below shows a diagram of an autonomic car and its various components. Autonomic cars mimic a human driver in that they understand the current situation on the roads from the live streaming of sensory values [16]. An autonomous car modular structure would take the high-level design shown in the figure below, whereby the car controls acceleration through fuel control and braking, steering and implements safety [16].

C. *Capability Maturity Model Integration (CMMI)*

CMMI is a collection of industry best practices designed to help organisations continuously improve their business processes [17]. There are 3 CMMI models comprising CMMI for Development (CMMI-DEV), CMMI for Services (CMMI-SVC) [18] and CMMI for Acquisition (CMMI-ACQ) [19]. CMMI has a framework structure required to produce models within CMMI, appraisal tools and training material [17]. The CMMI framework comprises goals and practices necessary to create CMMI constellation models. The models contain 16 process areas, which are essential to business process improvement [18]. Examples of process areas in CMMI-DEV include: Configuration Management (CM), Integrated Project Management (IPM), Organizational Training (OT), Product Integration (PI), Project Monitoring and Control (PMC), Project Planning (PP), etc. [17]. Each process area has generic goals and practices, and specific goals and practices as shown in the structure below.

CMMI [17] is an integrated approach across an enterprise, which focuses on building tools to support process improvement used to develop software systems. Process improvement helps to reduce the complexity, redundancy and costs associated with the use of separate and multiple capability maturity models (CMMs) [20].

CMMI has two streams of business process improvement (representations), namely: maturity levels – which corresponds to a staged representation, and capability levels, which correspond to a continuous representation. In a staged representation, processes are grouped and improved upon to achieve a specific maturity level. For example, in order to reach Maturity Level 1, an organisation would have to select and improve on these processes: CM, IPM, OT and PP. In continuous representation, a process to be improved on and the desired capability level are selected.

D. *Cubesats*

The original goal for developing cubesats and standardizing them was to develop the necessary skills for creating satellites intended for LEO missions and also limit the size and number of science instruments that could go on-board these spacecraft [21].

Cubesats (see an example of a 1U in Figure 4) [22] are considered low cost spacecraft – by satellite cost standards – because they tend to be designed and constructed from Commercial Off-The-Shelf (COTS) components [4].

Nowadays, such microsatellites are utilised in LEO missions previously performed using monolithic satellites and this has resulted in huge mission cost savings [8]. The International Space Station (ISS) has – among other functions – been used to deploy cubesats into orbit. The ISS uses a standardised deployer called the Poly Picosatellite Orbital Deployer (P-POD) as shown in Figure 5 [23]. Cubesats have gained international interest and adoptions, even NASA has been seriously considering reducing exploration mission costs by automating most of the tasks involved in spacecraft monitoring and control.

Traditionally, monolith spacecraft send their data (instrument data & navigation and health status data) back to earth stations for analysis. The data analysts and engineers send back commands to the craft for the next tasks to be performed. This exercise has a high cost because the number of missions has increased over the years and therefore mission personnel has had to increase [24].

Another issue with traditional satellites is communications lags between earth stations and the satellites in space. This increases the risk of mission failures because it takes more than a few minutes for the ground operators to receive the data and process it before they know what is happening out there, and by that time the spacecraft could be damaged, or the mission could have been jeopardized [24].

Cubesats were meant for the academic environment, but NASA – since the cubesat launch initiative and the Educational Launch of Nanosatellites (ELaNa) [25] - is investing in the advancement and development of the platform with the view to send cubesats into deep space in the near future [26].

NASA Goddard is actively working on propulsion systems, power sources and avionics to use on cubesats.

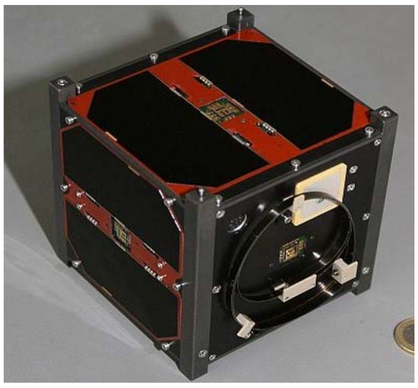


Figure 4. An example of a 1U cubesat [22].

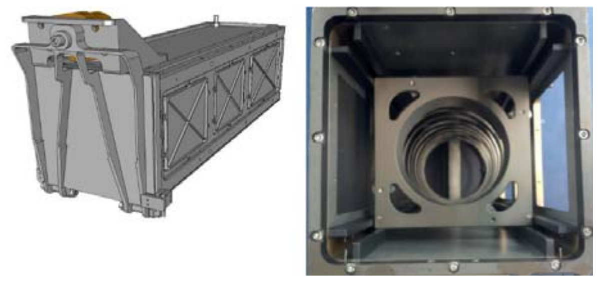


Figure 5. A 3U Poly Picosatellite Orbital Deployer (P-POD) [23].

Currently, cubesats, due to their size have the following limitations [21]:

- No long-range communications for sending data from deep space - however NASA has developed a miniature radio-communication system capable of talking directly to Earth from Mars and beyond.
- Current propulsion systems cannot fit on these small form factor satellites.
- Some scientific instruments cannot fit in the small form-factor frames. Innovators have to design instruments fit for the cubesat platform.

NASA's future strategy is to incorporate cubesats into their long-term plans for deep space exploration once the above limitations have been resolved [27].

The European Space Agency (ESA), since 2013, has had vested interest in cubesats too, mainly to test miniaturised technologies for small payload-driven missions [28]. ESA uses cubesats because of their low cost and high modularity, and therefore allows the demonstration of miniaturised technologies using System-On-Chip (SOC) integration and to demonstrate constellation configurations [28].

Autonomic computing in cubesats as of the writing of this paper is scarcely available – at best. Future endeavours will see cubesats take centre-stage in space missions with monolithic satellites [29] and therefore need to be autonomic. Cubesat design and development is still based on commands being sent from earth stations [30]. An example of a ground station controlled satellite was the SwissCube, which was designed for altitudes of 400km and 1000km and only had ground station access time of between 5 minutes and 10 minutes [31].

This shows another problem in LEO satellites, "access time" is very limited, hence the need for cubesats to be autonomic so they can perform multiple tasks without waiting for instructions from ground stations. There are several projects whereby autonomy is built-in from the beginning like the University Würzburg's Experimental satellite UWE-3, which was designed to address the following challenges [32]:

- Real-time failure detection, identification and recovery on-board, as miniaturization increases susceptibility to noise effects.
- Network control of the multi-satellite system, requiring integration of attitude and orbit control with the communication in order to be tolerant to interruptions of the link.

Current development of autonomic cubesats is still in early stages, e.g., the UWE-4 satellite incorporated orbit determination and control, and it had attitude determination and control [33]. ESA developed a similar autonomic model with levels ranging from E1 to E4 with level E1 being the lowest and level E4 being the highest whereby mission goals are run by the on-board computer.

The level of desired autonomy is not always the highest, it depends on the mission and its goals, and sometimes the best level of AC is adjustable and mixed autonomy [32].

III. Research Hypotheses

Given the young and evolving nature of the cubesat industry and hence the lack of autonomic control in cubesats, the plan in this project is to investigate autonomic computing in cubesats using the following hypotheses:

A. *Hypothesis 1:*

An autonomic capability model can be used as a tool to educate and motivate cubesat developers on the relevance and areas of application of autonomicity in space missions. By following the reasoning and use of the IBM 2001 maturity model and the automotive industry autonomic model, and CMMI, it is possible to develop a CACM that can form the basis for specifying autonomic features of relevance to future cubesat missions.

1) *Research Steps*

In studying and exploring this hypothesis, the following steps are being followed and used in the development of the CACM:

a. An evaluation of the layered models applied to autonomicity in the car industry and also the IBM 2001 maturity model has been conducted. Parallels have been drawn from the automotive industry autonomic levels and the IBM AC levels and capabilities, and currently investigating how they can be adapted and customised to suit the size and power constrained cubesat platform. This step is being performed through literature reviews of the IBM AC model and the automotive autonomic model, identifying specific capabilities in each level and key features of the progression from one level to the next.

b. Perform an in-depth literature review of the cubesat platform. This will be an intensive study of the cubesat environment, i.e., the components, form factors, current features, feature limitations, current and future cubesat mission types, and projected roadmap for development. Possible alternatives to cubesat components and computing platform will be explored because current "Commercial Of The Shelf" (COTS) components, are too expensive for most schools and universities, and therefore cheaper alternatives are a must to further drive the future of cubesat development.

c. Formulate the capability model (drawing inspiration from CMMI, the IBM Autonomic Maturity Model, the ALFUS and the Automotive Autonomy Model) to be applied to cubesats and to be incorporated in future cubesat developments. This step will involve distilling the best features of the models used as an inspiration for drawing this capability model. Some features and capabilities present in the above-mentioned models will not be applicable to cubesats due to the physical size limitations, component limitation and computing power

d. Develop an expert system to educate cubesat developers on how to use and apply the CACM to their cubesat designs and development. This system will guide developers on how to best develop and deploy the model. The system will be run remotely, and will create space mission profiles based on the interaction with developers as they respond to questions asked by the system to try and determine what level of autonomic computing is required for specific missions.

e. Formulate a set of evaluation questions to test the hypothesis, and to help improve the CACM. The questionnaire will be used to conduct surveys of cubesat developers and professional institutions in the cubesat industry. Such institutions will include private space technology companies, universities and space agencies. The surveys will collect information on how current satellite and cubesat developers think autonomic cubesats could benefit future space missions.

f. Survey a suitable number of cubesat institutions / companies to complete the evaluation. These will be planned in conjunction with the Space Mission challenges for Information Technology (SMC-IT) conference and possibly use the NASA network of contacts to distribute a questionnaire and get enough responses to test the hypothesis. A conference paper will be presented in the SMC-IT conference and delegates will be requested to fill-out the survey questionnaires. The data collected from the questionnaire forms will be used to change and tweak the model to best address any concerns obtained through the questionnaire feedback.

2) *Risk Assessment*

Perhaps the main risk associated with testing the first hypothesis is in not being able to survey the intended number of cubesat institutions. This could potentially render testing the hypothesis by real-world cubesat developers unachievable. There is another risk that the intended conference paper might not be accepted for presentation at the SMC-IT conference, and that would result in not getting the survey performed as mentioned in the research steps.

3) *Risk Mitigation*

A workshop will have to be planned and run at one of the SMC-IT conferences if the paper is not accepted. The workshop will be used to show developers how they could benefit from applying autonomic computing in cubesat design and development for individual cubesats and in constellations.

The workshop can also show how space missions can be better managed and made cheaper if cubesats were autonomic. By using a pre-recorded video to cover the use of the CACM, in conjunction with a knowledge-based system, it is hoped that workshop attendees can quickly be brought up to speed with our work and will then be in a position to complete our survey. Participants who need additional time will be able to take the CACM the video and the survey questionnaire away with them, and will be followed-up within two weeks to ensure they complete the survey.

B. *Hypothesis 2:*

An autonomic and apoptotic solution can address the needs of cubesats in complying with the requirements associated with space junk and will act as a suitable demonstrator area to illustrate the architecture of the CACM. Using the tenets of the CACM, cubesats can be designed to comply with the international requirement to clean-up space junk and debris by de-orbiting cubesats at the end of their mission or by executing the kill-switch if a cubesat develops an irrecoverable error condition before the end of its mission. This study will seek to demonstrate an apoptotic architecture through its implementation in a working solution.

1) Research Steps

The research steps to be followed when studying and exploring this hypothesis are as follows:

a. Investigate the space junk / debris problem in light of increased deployed spacecraft, especially cubesats.

b. Investigate available space clean-up solutions both current and those being proposed. Various sizes of space junk may possibly require different methods of clean-up, in this step, we will investigate proposed solutions for cubesat clean-ups.

c. Create a de-orbiter architecture employing apoptosis. The architecture will demonstrate various levels of capability according to the CACM autonomic levels, and therefore will incorporate Self-CHOP (Self-Configuration, Self-Healing, Self-Optimisation and Self-Protection), Monitoring, Analysis, Planning and Execution (MAPE) and communications. The demonstrator will have a number of versions each illustrating specific capabilities and level of sophistication at each of the CACM levels.

d. Perform the de-orbiter evaluation, during and after development and use that outcome to reform, improve and refine the model in step C of the first hypothesis. This is a feedback mechanism to change the model and make it more relevant to actual software development of autonomic cubesats. This provides an internal assessment and fine-tuning of the CACM model.

e. Build a de-orbiter simulator application to demonstrate the kill-switch capabilities of cubesats without propulsion (ground de-orbit simulator).

f. Get feedback from domain experts about the demonstrator / simulator through conference paper presentations and get questions or paper reviewers' feedback.

2) Risk Assessment

The demonstrator application will only test the hypothesis, but getting feedback from domain experts would add weight to the CACM and the apoptotic solution. The risk is in not being able to get the domain expert opinion.

3) Risk Mitigation

A workshop to demonstrate the application of the CACM in an apoptotic solution would have to be held at a conference. The workshop would help to test both the first and the second hypotheses using questionnaires to gather expert opinion on both the CACM and the apoptotic solution.

IV. Roadmap for Autonomicity in Cubesats

The roadmap for future cubesat development, as proposed in the CACM, is divided into 2 categories, namely: - the "Inspiration from Existing Models" and the "CACM Functional Areas". These two define both the hierarchy of the model and what modules or functional areas the CACM levels apply to within the cubesat platform.

A. *Inspiration from Existing Models*

Cubesats are designed from the ground up for specific missions. The mission type and goals determine what size the cubesat has to be, what capabilities and what scientific instruments to fit in the system. All these aspects of the cubesat can be autonomic individually and collectively, and can collaborate in a constellation.

Table I below summarises completed work on the draft CACM. The model consists of 5 autonomic levels derived from the IBM 2001 autonomic maturity model and the automotive automation models [34][35][36][37]. The proposed model shows how each level implements MAPE [1] of commands, and the tenets of Self-CHOP [1]. It also shows how cubesat functions and capabilities in the various incremental autonomic levels get more sophisticated as the level increases towards AC level 5.

Based on the literature review [5][13][38][39] conducted in this study, autonomicity is not common in the cubesat platform, especially as it relates to space debris clean-up. It was therefore deemed necessary to develop a roadmap model to help educate cubesat designers of the advantages of autonomic computing. The model will be a systematic implementation guide to autonomicity in the cubesat platform.

The CACM is inspired by CMMI [17][18][19], the Automotive Automation Model (AAM) [36][37], the IBM 2001 Autonomic Computing Model [1] and the ALFUS Model [11][12]. It derives from the structures and formulation of these models to create a capability model specifically designed to be implemented on cubesats by engineers, cubesat designers and mission architects. The model's purpose is to outline, draw up a roadmap for future cubesat autonomic designs and provides means to measure autonomic capabilities of cubesats against a standardized set of tiered self-management autonomic capabilities.

Another model that inspired CACM is the car industry AAM as described by SAE International [36]. It serves as a common taxonomy and descriptions for driving automation and attempts to simplify coordination and communication in the autonomous car industry [36]. This model comprises 6 levels ranging from full human driving with no automation to fully automated driving with no human intervention [36][37].

Since a cubesat is designed from the ground up for a specific scientific mission, whether it be monitoring

TABLE I. SUMMARY OF THE CACM.

CUBESAT AUTONOMIC CAPABILITY MODEL (CACM)	
Autonomic Capability Level	Autonomic Cubesat Level Description
AC1 **Incorporate Specific Capabilities**	Mission is fixed Limited on-board capability to transmit data and health signals Constellation: Information only. No propulsion
AC2 **Standardize Capabilities**	Mission is pre-scheduled, mission operations on-board. Transmits data to ground station on a schedule. Constellation: Information only. No propulsion
AC3 **Human & Machine Shared Capabilities**	Mission is pre-scheduled, mission operations on-board. Transmits data to ground station on a schedule. Some internal systems are autonomous Kill switch autonomously executed and by ground station. Mission goals can be adapted mid-mission
AC4 **Machine Delegated Capabilities**	Execution of goal-oriented mission operations on-board. Autonomic internal systems operations. Send health status to ground station and constellation. Mission goals can be adapted mid-mission Allows ground station to veto kill-switch execution. Propulsion Autonomic Avionics and collision avoidance – human instructions are optional
AC5 **Full Autonomic Capabilities**	Goal-oriented mission operations on-board. Can self-re-initialize OS and internal systems – no human intervention Sends health status to ground stations. Only receives new mission from ground station. Kill switch notification with error details

missions, exploration missions, defence / offence missions, and environmental study missions, cubesat functional specifications are therefore always mission specific. The CACM is product (cubesat) specific, and therefore, it would not follow the CMMI process areas, which are organisation specific [17]. However, it is envisaged that cubesat development can benefit from adopting this model because the CACM will form a basis for a systematic process of designing and developing cubesats incorporating autonomic behaviour from the ground up depending on mission specifics. Also, the CACM will be useful in setting the validation criteria for cubesat autonomic capabilities.

The drive to reduce space missions' costs is a strong catalyst in the development of safer, cheaper, smart, self-deorbiting cubesats and successful missions by implementing autonomic behaviour in cubesats. This will result in cubesats that will reduce space debris by deorbiting themselves and burn-up upon entry into the earth's atmosphere or any other planet's atmosphere. Part of the CACM capability definitions will be safe cubesats in that they can self-protect against potential dangers, e.g., change course if a cubesat is on a collision course with another object or if solar storms are predicted to be heading towards the satellite, the cubesat can self-shutdown. Self-protection helps to extend cubesat life-span, thereby allowing more successful missions on first take, resulting in low-cost missions.

Currently, there are no standard practices followed by all cubesat developers in reference to automation, autonomy and autonomicity. Every manufacturer seems to follow their own standards based on mission requirements and in CMMI terms, this would be similar to project management that is not standardized throughout an organisation.

B. *CACM Functional Areas*

Every cubesat contains the following components and functional areas, which form the core elements of a cubesat. Every cubesat implementation of the core components will vary depending on mission objectives. Their level of sophistication will be determined by the science mission objectives and the level of autonomicity the designers will want to implement. The main tasks and specific goals of each functional area are listed under each component. These tasks are high level, more details will be included as the model is further developed.

1) *Mission Control (MC)*
   - Hardcode mission objectives.
   - Mission tasks run on fixed schedules.
   - Software controlled mission objectives.
   - Change mission objectives mid-flight.
   - End / Terminate mission.
2) *Communication and Data Transmission (C&DT)*
   - Send science and internal systems data to ground stations.
   - Receive constellation communications and data.
   - Receive ground stations commands – including new mission plans and commands.
3) *Health Monitoring (HM)*
   - Monitor internal sensors.
   - Monitor internal modules (collection of sensors).
   - Send health status to ground stations and constellation.
4) *Ground Station (GS)*
   - Receive and analyse satellite data.
   - Plan new missions.
   - Change current missions.
   - Upload new missions.
5) *Management*
   - Monitor all systems on the spacecraft.
   - Manage and coordinate all sub-systems.
   - Collaborate with all spacecraft modules.
6) *Launch and Deployment (L&D)*
   - Power-up cubesat 2 hours after deployment.
   - De-tumble and stabilise cubesat after power up.
   - Deploy antennas and solar panels.
   - Collaborate with internal control systems.
7) *Electric Power Supply (EPS)*
   - Monitor available battery power.
   - Monitor power drainage rate.
   - Collaborate with Planning about schedules.
   - Regulate available power for components.
8) *Attitude Determination and Control System (ADCS)*
   - Use various methods to determine attitude.
   - Collaborate attitude determination in constellation.
   - Use available mechanism to alter attitude to meet mission goals.
9) *Orbit Determination and Control (ODC)*
   - Use various instruments and algorithms to determine orbit length, and inclination angle.
   - Collaborate orbit determination with constellation.
   - Use available propulsion mechanism to change orbit.
10) *Position Control (PC)*
    - Monitor spacecraft position relative to mission specification.
    - Use propulsion to manoeuvre the cubesat to various pre-planned and ad-hoc positions.
    - Move cubesat to specific positions in constellation.
11) *Scientific Instrumentation*
    - Monitor all scientific experiment instruments.
    - Validate instrument data.
    - Collaborate with Data Transmission.
12) *Kill Switch*
    - Enable provisional kill switch.
    - Override ground station kill switch commands.
    - Collaborate with Self-CHOP, De-Orbit Control and Management.
13) *De-Orbit Control*
    - Monitor Kill Switch status.
    - Collaborate with Orbit Determination and Control, Attitude Determination and Control, to quickly degrade the craft's orbit.
    - Collaborate with Constellation Management.
14) *Constellation*
    - Monitor individual members' health status.
    - Collaborate and coordinate flying formation through individual members' Attitude Determination and Control, Orbit Determination and Control, Position Control, and Data Transmission.

V. Kill Switch – Exemplar Application

The kill switch functional area is designed to address the space debris problem of defunct satellites remaining in active orbit for many years after their missions have ended. We propose that all cubesats should implement a kill switch feature / function, which will shut down all onboard equipment, and deorbit the cubesat into the graveyard orbit, or cause it to deorbit towards the earth and burn up in the earth's atmosphere.

The highest level of autonomicity for single cubesat missions is level 3. This level, corresponds to the IBM autonomic level 5 [1], whereby the cubesat is "self-sufficient". Levels 4 and 5, only apply in constellations configurations. In all levels of autonomicity in the CACM, ground station has full access to the cubesat, but as the levels increase there is less need for human / ground station intervention. The sophistication of one CACM level, is built on the previous level's capabilities, features and functions. Each level adds more features and functions to the previous level's defining capabilities, and functions that are performed by the cubesat, thereby, reducing human intervention in the cubesat mission management process.

In level 1, the cubesat's monitoring subsystem polls sensors for heartbeats at fixed intervals and if no heartbeats are detected, it alerts the ground station. If ground station does not respond after 1 full orbital time, the cubesat restarts the faulty sensors. The restarts of the sensors are performed up to a maximum number of times, if the errors are not cleared. After the restart threshold is reached, then the cubesat executes the kill switch. All data validation processes are performed on the ground station systems. At this level, the only autonomic feature of the cubesat is this apoptotic function, which means if the cubesat loses contact with the ground station, it will, by default, deorbit.

Level 2 adds the ability for sensors to send their heartbeats and data at parameterised intervals to the monitoring subsystem of the cubesat. A high-level data validation, is performed by the monitoring subsystem. If, and when sensors become faulty, with errors that cannot be cleared, the cubesat can turns off those faulty sensors and continues the mission until all sensors are defunct. The cubesat will reboot itself to try and clear faults if all of its sensors have become faulty. If errors are not cleared after a maximum number of reboots, the kill switch will be executed.

Since level 3 is the highest CACM level for single cubesat missions, the cubesat implements all possible autonomic features within the limits of its processing power, and electrical power resources. As in previous CACM level, the features in this level are a cumulative addition to the above-mentioned levels 1 and 2. At this level, a cubesat can reload mission tasks from storage and recalibrate sensors. If the reload attempt threshold is reached and errors persist, the kill switch is executed. If the cubesat detects an imminent collision, it navigates around the obstacle without waiting for ground station instructions.

CACM level 4, introduces constellation configurations for cubesats i.e., a level 4 cubesat, is self-aware and constellation aware. It can announce itself with its functions and capabilities, it can join and leave the constellation, as and when required. It communicates with the constellation via the constellation coordinator / manager. A constellation member, as an individual device, implements the CACM level 3 of autonomicity. The science data and health status information, in a level 4 cubesat, is sent to the ground station via the constellation manager. If the cubesat's sensors all fail, the cubesat removes itself from the constellation – it notifies the constellation coordinator - and follow the CACM level 3 recovery process. If, and when the cubesat's sensors are working again, it re-joins the constellation.

At level 5, in addition to level 4 capabilities and features, cubesats are grouped according to features and capabilities in order to form redundancy groups. Cubesats at this level, can request to delegate some or all of their tasks to other members via the constellation manager. The constellation manager delegates other members' tasks to individual members of the same redundancy group. Cubesats at this level will take over other members' tasks in the same redundancy group at the request of the constellation manager. Members must be environment aware, i.e., be able to move in formation with others, and keep appropriate distances among themselves and avoid collisions.

The constellation manager can remove members with faulty statuses or corrupt data from the constellation, in order to protect the constellation's integrity. Constellation management delegation is performed by vote, i.e., if the primary manager and the initial backup manager go offline or fail, other members take a vote to elect the next constellation manager.

Similarly, CACM level 5 cubesats implement and operate at CACM level 3, at the individual level. The kill switch is executed if errors persist even after going through the recovery process as defined in CACM level 3.

VI. Conclusion and Future Work

This paper has presented the background literature review on autonomic computing as used in normal computing platforms, in the auto-industry, in unmanned systems, in monolithic satellites and cubesats. A review of the autonomic models used in the automotive industry and other unmanned systems has shown that autonomy is still in its infant stages, and therefore, more work still needs to be done.

An autonomic capability model geared towards advancing cubesats and their functionality has been proposed and is under development. A brief summary of the model has been presented in Table I, and it is a very high level view of the model. The model development has drawn inspiration from other models, e.g., the IBM Autonomic Maturity Model, CMMI, ALFUS and others. These models will continue to be used to fine-tune the CACM and through domain experts feedback collected through surveys in conferences.

The current CACM is in early stages of development, it is still a high-level work-in-progress document. Further development will be carried out as per the listed steps in both hypotheses. When the model has the details in each task and autonomic level, an exemplar application will be developed to illustrate practical applicability of the model. Challenges encountered during the application development phase will be used to modify and fine-tune the model. This process will be on-going until the end of the study.

The model will require cubesat developers to create components that are manageable through application software in order to implement full autonomic features. This will require an increase in the number of device sensors to enable the cubesats to monitor more of their internal systems and their surrounding environment.

References

[1] IBM, "Autonomic Computing White Paper: An Architectural Blueprint for Autonomic Computing," *IBM White Paper*, no. June, p. 34, 2005.

[2] T. O. Eze, R. J. Anthony, A. Soper, C. Walshaw, A. Computing, and U. Kingdom, "A Technique for Measuring the Level of Autonomicity of Self-Managing Systems," no. c, pp. 8–13, 2012.

[3] H.-M. Huang, E. Messina, and J. Albus, "Autonomy Levels For Unmanned Systems (ALFUS) Volume II : Framework Models," *Framework*, vol. II, no. December, 2007.

[4] A. Toorian, E. Blundell, J. Puig-Suari, and R. Twiggs, "CubeSats as Responsive Satellites," *Space 2005*, no. September, pp. 1–14, 2005.

[5] P. Marks, "CubeSat craze could create space debris catastrophe," *New Scientist*, Sep-2014.

[6] S. Sadeghi and M. R. Emami, "Multi-spacecraft Studies of the Auroral Acceleration Region: From Cluster to

Nanosatellites," *Advances in Space Research*, vol. 59, no. 5, pp. 1173–1188, 2016.

[7] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Space Research*, vol. 47, no. 4, pp. 663–684, 2011.

[8] E. P. Caillibot, C. C. Grant, and D. D. Kekez, "Formation Flying Demonstration Missions Enabled by CanX Nanosatellite Technology," *Small Satellite Conference*, 2005.

[9] IBM, "Autonomic Computing Toolkit," *User's Guide*, 2004. [Online]. Available: https://www.ibm.com/developerworks/autonomic/books/fpu0mst.htm#ToC_16. [Accessed: 15-Jan-2018].

[10] M. C. Huebscher and J. a McCann, "A survey of autonomic computing—degrees, models, and applications," *ACM Computing Surveys*, vol. 40, no. 3, pp. 1–28, 2008.

[11] H.-M. Huang, K. Pavek, J. Albus, and E. Messina, "Autonomy Levels for Unmanned Systems (ALFUS) Framework: Safety and Application Issues," *Proc. SPIE*, vol. 5804, pp. 439–448, 2005.

[12] G. T. McWilliams *et al.*, "Evaluation of autonomy in recent ground vehicles using the autonomy levels for unmanned systems (ALFUS) framework," *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems - PerMIS '07*, pp. 54–61, 2007.

[13] E. Levin, J. Pearson, and J. Carroll, "Wholesale debris removal from LEO," *Acta Astronautica*, vol. 73, pp. 100–108, 2012.

[14] P. Gao, R. Hensley, and A. Zielke, "A Road Map to the Future for the Auto Industry," *McKinsey & Company*, no. 4, pp. 42–53, 2014.

[15] M. Harris, "The 2,578 Problems With Self-Driving Cars - IEEE Spectrum," *Spectrum IEEE*. 2017.

[16] T. S. Kim, J. C. Na, and K. J. Kim, "Optimization of an Autonomous Car Controller Using a Self-Adaptive Evolutionary Strategy," *International Journal of Advanced Robotic Systems*, vol. 9, 2012.

[17] CMMI Product Team, "CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). Software Engineering Institute.," 2010.

[18] CMMI Product Team, "CMMI for Services, Version 1.3 (CMU/SEI-2010-TR-034). Software Engineering Institute," 2010.

[19] M. Phillips, "CMMI for Acquisition (CMMI-ACQ) Primer, Version 1.3," 2011.

[20] L. Greiner, "Capability Maturity Model Integration (CMMI) Definition and Solutions | CIO," *Tutorial Capability Maturity Model Integration (CMMI) Definition and Solutions*, 2007. [Online]. Available: http://www.cio.com/article/2437864/process-improvement/capability-maturity-model-integration--cmmi--definition-and-solutions.html. [Accessed: 15-Jan-2018].

[21] E. Gibney, "CubeSats Set for Deep Space - If They Can Hitch a Ride.," *Nature*, vol. 535, no. 7610, pp. 19–20, Jul. 2016.

[22] Team Compass, "File:COMPASS-1 PR-Bild.jpg - Wikimedia Commons," *Wikimedia Commons*. [Online]. Available: https://commons.wikimedia.org/wiki/File%3ACOMPASS-1_PR-Bild.jpg. [Accessed: 15-Jan-2018].

[23] A. Chin, R. Coelho, L. Brooks, R. Nugent, and J. Puig-Suari, "Standardization Promotes Flexibility: A Review of CubeSats' Success," *Aerospace Engineering*, vol. 805, pp. 756--5087, 2008.

[24] W. Truszkowski, L. Hallock, J. Karlin, J. Rash, and G. Michael, "Autonomous and Autonomic Systems," p. 289, 2010.

[25] A. Heiney, "About ELaNa," 2015. [Online]. Available: https://www.nasa.gov/content/about-elana. [Accessed: 15-Jan-2018].

[26] A. Babuscia, K.-M. Cheung, D. Divsalar, and C. Lee, "Development of cooperative communication techniques for a network of small satellites and CubeSats in deep space: The SOLARA/SARA test case," *Acta Astronautica*, vol. 115, pp. 349–355, 2015.

[27] N. Cheeks, "From the Chief," *NASA Goddard Tech Transfer News*, vol. 11, no. 2, pp. 1–24, 2013.

[28] R. Walker, "Technology CubeSats," 2016. [Online]. Available: http://www.esa.int/Our_Activities/Space_Engineering_Technology/Technology_CubeSats. [Accessed: 15-Jan-2018].

[29] E. Gottzein *et al.*, "Challenges in the control and autonomy of communications satellites," *Control Engineering Practice*, vol. 8, no. 4, pp. 409–427, 2000.

[30] M. A. Viscio *et al.*, "Interplanetary CubeSats system for space weather evaluations and technology demonstration," *Acta Astronautica*, vol. 104, no. 2, pp. 516–525, 2014.

[31] M. Noca *et al.*, "Lessons Learned from the First Swiss Pico-Satellite: SwissCube," *23rd Annual AIAA/USU Conference on Small SatellitesConference on Small Satellites*, 2009.

[32] S. Zieba, P. Polet, F. Vanderhaegen, and S. Debernard, "Principles of adjustable autonomy: A framework for resilient human-machine cooperation," *Cognition, Technology and Work*, vol. 12, no. 3, pp. 193–203, 2010.

[33] I. Kronhaus, K. Schilling, S. Jayakumar, A. Kramer, M. Pietzka, and J. Schein, "Design of the UWE-4 Picosatellite Orbit Control System using Vacuum-Arc-Thrusters," no. Eit 1, pp. 1–11, 2013.

[34] R. W. Proud and J. J. Hart, "The Function-Specific Level of Autonomy and Automation Tool," *Engineering*.

[35] C. D. Wickens, A. S. Mavor, R. Parasuraman, and J. P. Mcgee, *The Future of Air Traffic Control: Human Operators and Automation*. 1998.

[36] SAE International, "Automated Driving: Levels Of Driving Automation Are Defined In New SAE International Standard J3016," *Global Ground Vehicle Standards*, 2016.

[37] M. Blanco *et al.*, "Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts," *(Report No. DOT HS 812 182)*, no. August, p. 300, 2015.

[38] J. Su and Z. Lixin, "The European Union draft Code of Conduct for outer space activities: An appraisal," *Space Policy*, vol. 30, no. 1, pp. 34–39, 2014.

[39] F. Alby *et al.*, "The European space debris safety and mitigation standard," *Advances in Space Research*, vol. 34, no. 5, pp. 1260–1263, 2004.