Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?

Roy Sterritt and Mike Hinchey

University of Ulster
School of Computing and Mathematics,
Jordanstown Campus, BT37 0QB
Northern Ireland
r.sterritt@ulster.ac.uk

NASA Goddard Space Flight Center
Software Engineering Laboratory
Greenbelt, MD 20771
USA
michael.g.hinchey@nasa.gov

Abstract. Autonomic Computing (AC), a self-managing systems initiative based on the biological metaphor of the autonomic nervous system, is increasingly gaining momentum as the way forward in designing reliable systems. Agent technologies have been identified as a key enabler for engineering autonomicity in systems, both in terms of retrofitting autonomicity into legacy systems and designing new systems. The AC initiative provides an opportunity to consider other biological systems and principles in seeking new design strategies. This paper reports on one such investigation; utilizing the *apoptosis* metaphor of biological systems to provide a dynamic health indicator signal between autonomic agents.

1. Introduction

One of the great things about being involved in the early days of development of a new paradigm is having the opportunity to look again at how things are done, and contemplate approaches not normally considered before the paradigm beds down into its evolutionary path.

Autonomic Computing is based on the biological metaphor of the Autonomic Nervous System (ANS) [1], taking the ANS as inspiration to achieve self-managing systems without 'conscious effort' from the user. IBM's initial set of self-properties (self-CHOP, configuration, healing, optimisation and protection) have been expanded to include many self-* properties leading to the adoption of the term *selfware*.

Biological systems inspire systems design in many other ways – reflex reaction and health signs [2, 3], nature-inspired systems (NIS) [4] – hive and swarm behaviour, fire flies, etc., for example.

At this stage in the emerging field of Autonomic Computing we are seeking inspiration for new approaches from (obviously, pre-existing) biological mechanisms. An obscure mechanism which is discussed in this paper is *Apoptosis* – the approach for cell self-destruction, which at first sight may seem a metaphor too far.

2. Biological Apoptosis

The biological analogy of autonomic systems has been well discussed in the literature. While reading this the reader is not consciously concerned with his[1] breathing rate or how fast his heart is beating. Achieving the development of a computer system that can self-manage without the conscious effort of the user is the vision and ultimate goal [5]. Another typical biological example is that the touching of a sharp knife results in a reflex reaction to reconfigure the area in danger to a state that is out of danger (self-protection, self-configuration, and, if damage is caused, self-healing) [6].

If one cuts oneself and starts bleeding, good training results in washing the finger, applying a bandage and carrying on with one's tasks without any further conscious thought. 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 cancer is the result of cells not dying fast enough, rather than multiplying out of control, as previously thought).

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* [8], meaning 'drop out', used by the Greeks to refer to the 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' [9], where cells are prevented from putting an end to themselves due to constant receipt of biochemical 'stay alive' signals (Figure 1).

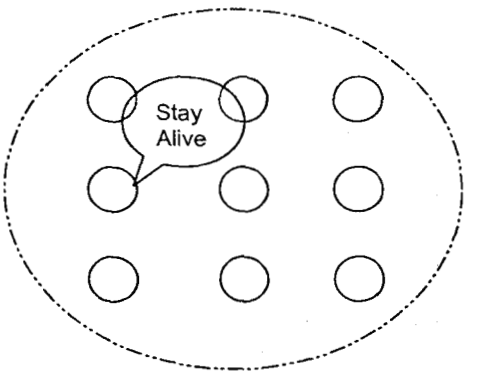

Fig. 1. Turning off the self-destruct sequence - cell receives 'stay alive' signal.

[1] Throughout this paper, for "his", read "his/her".

Further investigations into the apoptosis process [10] have discovered more details about the self-destruct programme. 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 [7]. 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 [7].

3. Autonomic Computing and Agents

Autonomic Computing is dependent on many disciplines for its success; not least of these is research in agent technologies. At this stage, there are no assumptions that agents have to be used in an autonomic architecture, but as in complex systems there are arguments for designing the system with agents [11], as well as providing inbuilt redundancy and greater robustness [12], through to retrofitting legacy systems with autonomic capabilities that may benefit from an agent approach [13].

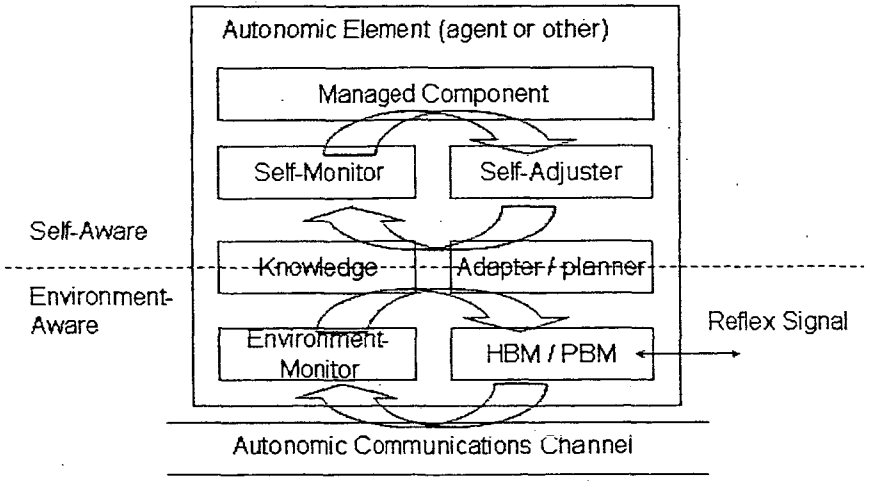

Fig. 2. Autonomic Element (agent or other) consists of a managed component and an autonomic manager. Control loops with sensors (self-monitor) and effectors (self-adjuster) together with system knowledge and planning/adapting policies allow the autonomic element to be self-aware and to self-manage. A similar scheme facilitates environment awareness (allowing self-managing if necessary, but without the immediate control to change the environment – this is effected through communication with other autonomic managers that have the relevant influence, through reflex or event messages).

Emerging research suggests that the autonomic manager may be an agent itself, for instance, an agent termed a *self-managing cell* (SMC) [14], containing functionality for measurement and event correlation and support for policy-based control.

Essentially, the aim of autonomic computing is to create robust dependable self-managing systems [15]. To facilitate this aim, fault-tolerant mechanisms such as a heart-beat monitor ('I am alive' signals) and pulse monitor (urgency/reflex signals) may be included within the autonomic element (Figure 2) [2, 16]. The notion behind the pulse monitor (PBM) is to provide an early warning of a condition so that preparations can be made to handle the processing load of diagnosis and planning a response, including diversion of load. Together with other forms of communications it creates dynamics of autonomic responses [17] – the introduction of multiple loops of control, some slow and precise, others fast and possibly imprecise, fitting with the biological metaphor of reflex and healing [2].

The major motivating factor for formal approaches to agent-based systems is to prevent race conditions and undesirable emergent behaviour. In this situation, Self-Destruction of the agent may be viewed as a last resort situation to prevent further damage; in other situations, such as security of the agent, Self-Destruction may be used as an intrinsic part of the process.

Agent destruction has been proposed for mobile agents to facilitate security measures [18]. Greenberg *et al.* highlighted the situation 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 [19]. 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 behaviour), the later two through deliberate or accidental situations caused by external bodies acting upon the agent. The range of these situations and attacks have been categorised 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 [19]. This situation has similar overtones to our previous discussion on biological apoptosis, where the body is at its most vulnerable during cell division.

4. Autonomicity in NASA Missions

New paradigms in spacecraft design are leading to radical changes in the way NASA designs spacecraft operations [20]. Increasing constraints on resources, and greater focus on the cost of operations, has led NASA to utilize adaptive operations and move towards almost total onboard autonomy in certain classes of mission operations [21, 22].

NASA missions, particularly those to deep space, where manned craft will not at present be utilized, are considering the use of almost wholly autonomous decision-making to overcome the unacceptable time lag between a craft encountering new situations and the round-trip delay (of upwards of 40 (earth) minutes) in obtaining responses and guidance from mission control.

More and more NASA missions will, and *must*, incorporate autonomicity as well as autonomy [23, 27].

4.1 Previous Missions

Two of the first notable missions to use autonomy are DS1 (Deep Space 1) and the Mars Pathfinder [24].

The *Beacon Monitor* concept, first used in the DS1 mission work [25] automates the routine task of health monitoring and transfers the process of monitoring from ground to the spacecraft [16]. With beacon monitoring, the spacecraft sends a signal to the ground that indicates how urgent it is to track the spacecraft for telemetry.

This concept involved a paradigm shift for NASA from its traditional routine telemetry downlink and ground analysis, to onboard health determination and autonomous data summarization [25].

In terms of high-level concepts, the beacon monitor is analogous to the heartbeat monitor, but with the addition of a tone to indicate the degree of urgency involved: *nominal*, *interesting*, *important*, *urgent* and *no tone* [26].

Some long-term drawbacks of this approach have been discovered. Since one of the primary goals of beacon monitoring was to reduce the amount of data sent to the ground (achieved by eliminating the download of telemetry data), operators lost the ability to gain an intuitive feel for the performance and characteristics of the craft and its components, as well as losing the ability to run the data through simulations [20].

As such, to fully benefit from beacon monitoring, the *fast loop* of real-time health assessment must be supplemented by a *slow loop* to study the long-term behaviour of the spacecraft. This *engineering data summarization* is where the spacecraft creates a second set of abstractions regarding the sensor telemetry, which is then sent back to ground to provide the missing context for operators.

This dual approach has conceptually much in common with the reflex and healing approach [2, 16].

4.2 A Future Mission

The Autonomic Computing initiative has been identified by NASA as having potential to contribute to their goals of autonomy and cost reduction in future space exploration missions [22, 23, 27].

ANTS, Autonomous Nano-Technology Swarm, is a mission that will launch sometime between 2020 and 2030 ("any day now" in terms of NASA missions). The mission is viewed as a prototype for how many future unmanned missions will be developed and how future space exploration will exploit autonomous and autonomic behaviour.

The mission will involve the launch of 1000 pico-class spacecraft swarm from a stationary factory ship, on which the spacecraft will be assembled. The spacecraft will explore the asteroid belt from close-up, something that cannot be done with conventionally-sized spacecraft.

As much as 60% to 70% of the spacecraft will be lost on first launch as they enter the asteroid belt. The surviving craft will work as a swarm, forming smaller groupings of *worker* craft (each containing a unique instrument for data gathering), a coordinating *ruler*, that will use the data it receives from workers to determine which asteroids are of interest and to issue instructions to the workers and act as a coordinator, and *messenger* craft which will coordinate communications between the swarm and between the swarm and ground control. Communications with earth will be limited to the download of science data and status information, and requests for additional craft to be launched from earth as necessary.

A current project (FAST) is studying advanced technologies for the verification of this incredibly complex mission; the reader is directed to [22, 27] for a more detailed exposition of the ANTS mission and the FAST (Formal Approaches to Swarm Technologies) project.

5. The Role of Apoptosis

The discussions so far have established the concepts of:

- Heart-Beat Monitor (HBM) *I am alive*: a fault-tolerant mechanism which may be used to safeguard the autonomic manager to ensure that it is still functioning by periodically sending 'I am alive' signals.

- Pulse Monitor (PBM) *I am healthy*: extends the HBM to incorporate reflex/urgency/health indicators from the autonomic manager representing its view of the current self-management state. The analogy is with measuring the pulse rate instead of merely detecting its existence.

- Apoptosis *Stay alive*: a proposed additional construct used to safeguard the system and agent; a signal indicates that the agent is still operating within the correct context and behaviour, and should not self-destruct.

The title of this paper (purposely) raises the question of whether there is a role for the apoptosis metaphor within the development of autonomic agents. Additionally, in the introduction, we prompted the consideration of whether perhaps it is a metaphor too far.

Section 3 clearly highlights the general problem of agent security, whether from the agent's or host's perspective. In terms of generic contribution to autonomic agents development, with many security issues the lack of an agreed standard approach to agent-based systems prohibits further practical development for generic autonomic systems. As such, the proposal can only be 'put out there' as a concept.

Of course, within NASA missions, such as ANTS, we are not considering the generic situation. Mission control and operations is a trusted private environment. This eliminates many of the wide range of agent security issues discussed earlier, just leaving the particular concerns; is the agent operating in the correct context and showing emergent behaviour within acceptable parameters, where upon *apoptosis* can make a contribution.

For instance, in ANTS, 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.

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* (from Earth, if necessary). 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.

All of the spacecraft are powered by batteries that are recharged by the sun using solar sails [22, 27]. Although battery technology has greatly advanced, there is still a "memory loss" situation, whereby batteries that are continuously recharged eventually lose some of their power and cannot be recharged to full power. After several months of continual operation, each of the ANTS will no longer be able to recharge sufficiently, at which point their 'stay alive' signals will be withdrawn, and new craft will need to be assembled or launched from Earth.

6. Conclusions

Autonomic Computing [1] has been gaining ground as a significant new paradigm to facilitate the creation of self-managing systems to deal with the ever increasing complexity and costs inherent in today's (and tomorrow's) systems.

In terms of the Autonomic Computing initiative, agent technologies have the potential to become an intrinsic approach within the initiative [28], not only as an

enabler (e.g. ABLE agent toolkit [29]), but also in terms of creating autonomic agent environments.

Formal approaches to agent-based systems [30, 31] have a primary focus of identifying race conditions, highlighting undesirable emergent behaviour, and verifying the correctness of systems that are far too complex to ever test correctly. However, the practicality of mobile agents is predicated on the existence of realistic security techniques [19].

We have described the Heart-Beat Monitor (HBM) and Pulse Monitor (PBM) and proposed a logical addition which has an analogy from biological systems, *Apoptosis* and *Self-Destruct*, which we believe will be valuable in future autonomic systems.

Acknowledgements

The development of this paper was supported at University of Ulster by the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative. Acknowledgement is also due to the University of Ulster MSc Informatics student Gerry Clarke.

Part of this work has been supported by the NASA Office of Systems and Mission Assurance (OSMA) through its Software Assurance Research Program (SARP) project, Formal Approaches to Swarm Technologies (FAST), and by NASA Goddard Space Flight Center, Software Engineering Laboratory (Code 581).

References

1. P. Horn, "Autonomic computing: IBM perspective on the state of information technology," IBM T.J. Watson Labs, NY, 15th October 2001. Presented at AGENDA 2001, Scottsdale, AR (available at http://www.research.ibm.com/autonomic/), 2001.
2. R. Sterritt, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID," Proceedings of IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003, Banff, Alberta, Canada, 22-23 August 2003, pp 433-440.
3. T. Bapty, S. Neema, S. Nordstrom, S. Shetty, D. Vashishtha, J. Overdorf, P. Sheldon "Modeling and Generation Tools for Large-Scale, Real-Time Embedded Systems," Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS'03), Huntsville, Alabama, USA, April 7-11 2003, IEEE CS Press, pp 11-16.
4. R. J. Anthony, "Natural Inspiration for Self-Adaptive Systems," Proceedings of IEEE DEXA 2004 Workshops - 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04), Zaragoza, Spain, August 30th - September 3rd 2004, IEEE, pp 732-736.
5. J. O. Kephart and D. M. Chess. "The vision of autonomic computing". Computer, 36(1):41–52, 2003.
6. R. Sterritt, D.W. Bustard, "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, pp 694-698.

7. J. Newell, "Dying to live: why our cells self-destruct," Focus, Dec. 1994.
8. R. Lockshin, Z. Zakeri, "Programmed cell death and apoptosis: origins of the theory," Nature Reviews Molecular Cell Biology. 2:542-550, 2001.
9. 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.
10. J. Klefstrom, E.W. Verschuren, 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.
11. N.R. Jennings, M. Wooldridge, "Agent-oriented Software Engineering," in J. Bradshaw (ed.), *Handbook of Agent Technology*, AAAI/MIT Press, 2000.
12. M.N. Huhns, V.T. Holderfield, R.L.Z. Gutierrez, "Robust software via agent-based redundancy," Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, pp 1018-1019.
13. G. Kaiser, J. Parekh, P. Gross, G. Valetto, "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems," Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop, Seattle, USA, June 2003.
14. E. Lupu, et al., EPSRC AMUSE: Autonomic Management of Ubiquitous Systems for e-Health, 2003.
15. R. Sterritt, D.W. Bustard, "Autonomic Computing: a Means of Achieving Dependability?," Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS'03), Huntsville, Alabama, USA, April 7-11 2003, pp 247-251.
16. R. Sterritt, "Towards Autonomic Computing: Effective Event Management," Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW), Maryland, USA, December 3-5 2002, IEEE Computer Society Press, pp 40-47.
17. R. Sterritt, D.F. Bantz, "PAC-MEN: Personal Autonomic Computing Monitoring Environments," Proceedings of IEEE DEXA 2004 Workshops - 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04), Zaragoza, Spain, August 30 – 3 September, 2003.
18. J.D. Hartline, "Mobile Agents: A Survey of Fault Tolerance and Security," University of Washington, 1998.
19. M.S. Greenberg, J.C. Byington, T. Holding, D.G. Harper, "Mobile Agents and Security," IEEE Comms. Mag. July 1998.
20. M.A. Swartwout, "Engineering Data Summaries for Space Missions," SSDL, 1998.
21. J. Wyatt, R. Sherwood, M. Sue, J. Szijjarto, "Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment," 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99), ESTEC, Noordwijk, The Netherlands, 1-3 June 1999.
22. W. Truszkowski, M. Hinchey, J. Rash and C. Rouff, "NASA's Swarm Missions: The Challenge of Building Autonomous Software," IEEE IT Professional mag., September/October 2004, pp 51-56.
23. W. Truszkowski, M. Hinchey, C. Rouff and J. Rash, "Autonomous and Autonomic Systems: A Paradigm for Future Space Exploration Missions," *submitted for publication.*
24. N. Muscettola, P. P. Nayak, B. Pell, and B. Williams, "Remote Agent: To Boldly Go Where No AI System Has Gone Before," Artificial Intelligence **103**(1-2):5-48, 1998.
25. J.Wyatt, H. Hotz, R. Sherwood, J. Szijjaro, M. Sue, "Beacon Monitor Operations on the Deep Space One Mission," 5th Int. Sym. AI, Robotics and Automation in Space, Tokyo, Japan, 1998.

26. R. Sherwood, J. Wyatt, H. Hotz, A. Schlutsmeyer, M. Sue, "Lessons Learned During Implementation and Early Operations of the DS1 Beacon Monitor Experiment", Third International Symposium on Reducing the Cost of Ground Systems and Spacecraft Operations, Tainan, Taiwan, 1999.
27. W. Truszkowski, J. Rash, C. Rouff and M. Hinchey, "Asteroid Exploration with Autonomic Systems," Proceedings of IEEE Workshop on the Engineering of Autonomic Systems (EASe 2004) at the 11th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2004), Brno, Czech Republic, 24-27 May 2004, pp 484-490.
28. J. McCann, M. Huebscher, "Evaluation issues in Autonomic Computing," Imperial College, 2004.
29. J.P. Bigus, et.al., "ABLE: a toolkit for building multiagent autonomic systems," IBM Systems J., **41**(3):350-371, 2002.
30. C.A. Rouff, M. G. Hinchey, W. Truszkowski, J.L. Rash and D. Spears, editors, *Agent Technology from a Formal Perspective*, NASA Monographs in Systems and Software Engineering, Springer Verlag, London, 2005.
31. W. Truszkowski, C.A. Rouff, H.L. Hallock, J. Karlin, J.L. Rash, M.G. Hinchey and R. Sterritt, *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, NASA Monographs in Systems and Software Engineering, Springer Verlag, London, 2005.