Biologically-Inspired Concepts for Autonomic Self-Protection in Multiagent Systems

Roy Sterritt[1] and Mike. Hinchey[2]

[1] Computer Science Research Institute, University of Ulster, Jordanstown Campus, Northern Ireland, r.sterritt@ulster.ac.uk
[2] Lero–the Irish Software Engineering Research Center, University of Limerick, Limerick, Ireland, mike.hinchey@lero.ie

Abstract. Biologically-inspired autonomous and autonomic systems (AAS) are essentially concerned with creating self-directed and self-managing systems based on metaphors from nature and the human body, such as the autonomic nervous system. Agent technologies have been identified as a key enabler for engineering autonomy and autonomicity in systems, both in terms of retrofitting into legacy systems and in designing new systems. Handing over responsibility to systems themselves raises concerns for humans with regard to safety and security. This paper reports on the continued investigation into a strand of research on how to engineer self-protection mechanisms into systems to assist in encouraging confidence regarding security when utilizing autonomy and autonomicity. This includes utilizing the apoptosis and quiescence metaphors to potentially provide a self-destruct or self-sleep signal between autonomic agents when needed, and an ALice signal to facilitate self-identification and self-certification between anonymous autonomous agents and systems.

Keywords: multiagent systems, autonomic computing, biologically-inspired computing, self-managing systems.

1 Introduction

The field of Biology changed dramatically in 1953, with the determination by Francis Crick and James Dewey Watson of the double helix structure of DNA. This discovery changed Biology for ever, allowing the sequencing of the human genome, and the emergence of a "new Biology" focused on DNA, genes, proteins, data, and search. Computational Biology and Bioinformatics heavily rely on computing to facilitate research into life and development.

Simultaneously, an understanding of the biology of living organisms indicates a parallel with computing systems: molecules in living cells interact, grow, and transform according to the "program" dictated by DNA.

Moreover, paradigms of Computing are emerging based on modeling and developing computer-based systems exploiting ideas that are observed in nature. This includes building self-management and self-governance mechanisms that are inspired by the human body's autonomic nervous system into computer systems,

modeling evolutionary systems analogous to colonies of ants or other insects, and developing highly-efficient and highly-complex distributed systems from large numbers of (often quite simple) largely homogeneous components to reflect the behaviour of flocks of birds, swarms of bees, herds of animals, or schools of fish.

This new field of "Biologically-Inspired Computing", often known in other incarnations by other names, such as: Autonomic Computing, Pervasive Computing, Organic Computing, Biomimetics, and Artificial Life, amongst others, is poised at the intersection of Computer Science, Engineering, Mathematics, and the Life Sciences. Successes have been reported in the fields of drug discovery, data communications, computer animation, control and command, exploration systems for space, undersea, and harsh environments, to name but a few, and augur much promise for future progress.

2 Safety and Security in Biologically-Inspired Systems

It is often joked that researchers in the security domain view safety as being a subset of security, while researchers in the safety domain see security as being a special case of safety. In fact, there is a certain degree of truth in both views, and valid cases can be made to support either position.

It is certainly true that various techniques from reliability engineering, safety engineering, and related areas, can be adapted to address issues in security. Similarly, protocols, analysis mechanisms, and other techniques from the security domain have been demonstrated to have useful application in safety-critical systems.

The classes of system that we're concerned with in this paper have their own particular issues vis-à-vis security and safety, however. Such systems are evolving, and adapting to the circumstances in their environment. More importantly, these systems are self-directed — we cannot necessarily tell *a priori* what situations they will be expected to address, nor necessarily what actions they will take to address them.

The FAST (Formal Approaches to Swarm Technologies) project looked at deriving a formal development method for swarm-based systems, a particular class of biologically-inspired system where (usually) a large number of components (whether software or physical devices) collaborate to achieve a common goal [21, 22]. As its example "test-bed", FAST used the ANTS (Autonomous Nano-Technology Swarm) concept mission, described in more detail in Section 4.

The project found (unsurprisingly) that no single formal development notation was sufficient to address all of the issues (in the case of ANTS, these were primarily safety-related issues, although security is not entirely discounted and likely to be a more important issue in actual operation). Moreover, it found that a realistic formal approach would require the use of a notation that made some allowance for the expression of probabilities and frequencies of operations. To this end, the FAST project proposed the combination of several formal notations, one of which is a probabilistic variant of a popular process algebra. The interested reader is directed to [25] for further details.

Further investigation, however, highlighted the fact that most of these probabilities and frequencies would be little more than guesswork, with a lot of the probabilities being so tiny (unlikely) that their combination would result in so many combinations of operations for which the probably was so close to zero that they couldn't be distinguished. We believe that this is likely to be the case with other types of biologically-inspired systems also. We simply do not have the experience to realistically estimate probabilities, nor are we ever likely to, since such systems are expected to "learn" and improve their operation over time.

As a result, any approach to the development of such systems (whether formal or otherwise) will be limited. That is not to say that there are not benefits from the use of formal approaches. In fact, FAST demonstrates that properties (safety properties, security properties, and others) may be proposed and proven to hold (or otherwise), giving certain degrees of assurance as to how the system will operate under *certain* conditions. Such an approach also allows for a significant amount of "what-if" analysis, where conditions can be formulated and in many cases it can be demonstrated that the system will be able to avoid, or if necessary, recover from, these conditions.

The reality, however, is that we cannot possibly foresee *all* such conditions and eventualities, and biologically-inspired systems must, as a consequence, have a greater number of prevention mechanisms built in, in order to ensure correct, safe, and secure operation.

3 Biologically-Inspired Computing Concepts

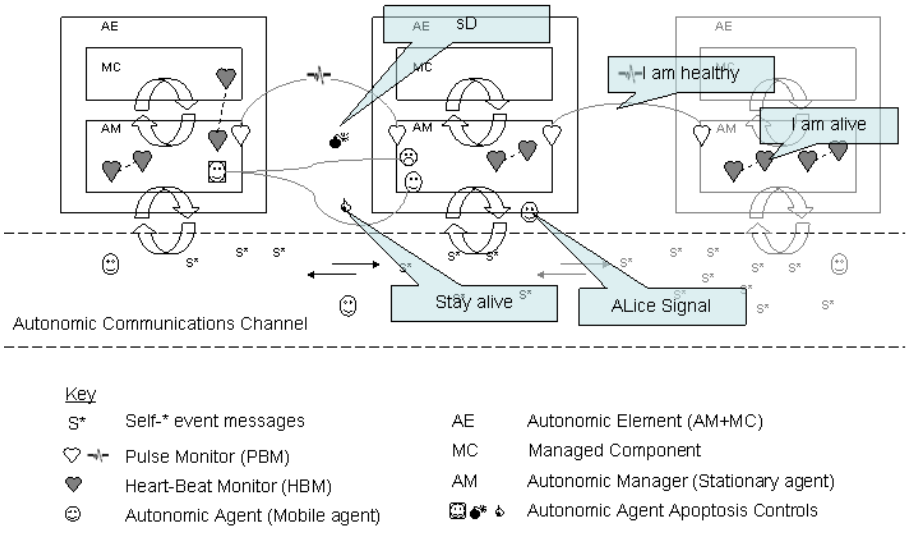

Fig. 1. Autonomic and Autonomous Computing Environment

3.1 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 [1], as well as providing inbuilt redundancy and greater robustness [2], through to retrofitting legacy systems with autonomic capabilities that may benefit from an agent approach [3] to research depicting the autonomic manager as an agent itself, for instance, a self-managing cell (SMC) [4], containing functionality for measurement and event correlation and support for policy-based control.

Fig. 1 represents a view of an architecture for self-managing systems, where an autonomic element consists of the component required to be managed, and the autonomic manager [12]. It is assumed that an autonomic manager (AM) is responsible for a managed component (MC) within a self-contained autonomic element (AE). This autonomic manager may be designed as part of the component or provided externally to the component, as an agent, for instance. Interaction will occur with remote autonomic managers (cf. the autonomic communications channel shown in Fig. 1) through virtual, peer-to-peer, client-server or grid configurations. The figure depicts self-* event messages as well as mobile agents, which assist with self-managing activity, traveling along this channel.

Essentially, the aim of autonomic computing is to create robust dependable self-managing systems [5]. 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 [6, 7]. The notion behind the pulse monitor (PBM) is to provide an early warning of an undesirable 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 [8] – the introduction of multiple loops of control, some slow and precise, others fast and possibly imprecise, fitting with the biological metaphors of reflex and healing [6].

3.2 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 their breathing rate or how fast their heart is beating. Achieving the development of a computer system that can self-manage without the conscious effort of the user is the overarching vision of the Autonomic Computing initiative [9]. 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 no longer in danger (self-protection, self-configuration, and, if damage has occurred, self-healing) [10].

If you cut yourself and it starts bleeding, you will treat it and carry on with your tasks without any further conscious thought. Yet, often, the cut will have caused skin cells to be displaced down into muscle tissue [11]. If they survive and divide, they

have the potential to grow into a tumor. The body's solution to dealing with this situation is cell self-destruction. There is mounting evidence that some forms of cancer are the result of cells not dying fast enough, rather than multiplying out of control.

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 self-destruction is delayed due to the continuous receipt of biochemical reprieves. This process is referred to as apoptosis [12], meaning "drop out", and was 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" [13], where cells are prevented from putting an end to themselves due to constant receipt of biochemical "stay alive" signals.

Further investigations into the apoptosis process [14] have uncovered more details about this self-destruct predisposition. 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 tumor. However, simultaneously a cell must divide in order to build and maintain the body, and there is a constant conflict.

The suicide and reprieve controls have been compared to the dual-key on a nuclear missile [11]. 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 [11].

3.3 The Role of Apoptosis within Autonomic Agents

Agent destruction has been proposed for mobile agents to facilitate security measures [15]. 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 [16]. The mail had survived "pending" on Internet relays waiting for omega.univ.edu to come back up.

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

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

From an agent perspective, the first is through accidental or unintentional situations caused by that agent (race conditions and unexpected emergent behavior), the latter two through deliberate or accidental situations caused by external bodies acting upon the agent. The range of these situations and attacks have been categorized as:

damage, denial-of-service, breach-of-privacy, harassment, social engineering, event-triggered attacks, and compound attacks.

In the situation where portions of an agent's binary image (e.g., monetary certificates, keys, information, etc.) are vulnerable to being copied when visiting a host, this can be prevented by encryption. Yet there has to be decryption in order to execute, which provides a window of vulnerability [16]. This situation has similar overtones to our previous discussion on biological apoptosis, where the body is at its most vulnerable during cell division.

The principles of a Hearth-Beat Monitor (HBM) and Pulse(-Beat) Monitor (PBM) have been established. Heart-Beat Monitor (I am alive) is a fault-tolerant mechanism which may be used to safeguard the autonomic manager, and to ensure that it is still functioning by periodically sending 'I am alive' signals. The Pulse Monitor (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 to determine how healthy the patient is instead of merely detecting its existence (and the fact that the patient is alive).

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

Is there a role for the apoptosis metaphor in the development of autonomic agents? [17, 18]

With many security issues, the lack of an agreed standard approach to agent-based systems prohibits, for now, further practical development of the use of apoptosis for agent security in a generic fashion within autonomic systems. Later, in a subsequent section, we propose a certification means between agents and hosts to work around this.

3.4 Autonomic Reflex Signal – Lub-Dub Pulse Emission

The autonomic environment requires that autonomic elements and, in particular, autonomic managers communicate with one another concerning self-* activities, in order to ensure the robustness of the environment. Fig. 1 illustrates that the autonomic manager communications (AM⇔AM) also includes a reflex signal. This may be facilitated through the additional concept of a pulse monitor—PBM (an extension of the embedded system's heart-beat monitor, or HBM, which safeguards vital processes through the emission of a regular 'I am alive' signal to another process, as previously described) with the capability to encode health and urgency signals as a pulse [12]. Together with the standard event messages on the autonomic communications channel, this provides dynamics within autonomic responses and multiple loops of control, such as reflex reactions among the autonomic managers [9].

This reflex component may be used to safeguard the autonomic element by communicating its health to another AE [13]. The component may also be utilized to communicate environmental health information [12]. For instance, in the situation where each PC in a LAN is equipped with an autonomic manager, rather than each of the individual PCs monitoring the same environment, a few PCs (likely the least busy

machines) may take on this role and alert the others through a change in pulse rate to indicate changing circumstances.

An important aspect concerning the reflex reaction and the pulse monitor is the minimization of data sent—essentially only a "signal" is transmitted. Strictly speaking, this is not mandatory; more information may be sent, yet the additional information must not compromise the reflex reaction. For instance, in the absence of bandwidth concerns, information that can be acted upon quickly and not incur processing delays could be sent. The important aspect is that the information must be in a form that can be acted upon immediately and not involve processing delays (such as is the case of event correlation).

Just as the beat of the heart has a double beat (lub-dub) the autonomic element's (Fig. 1) pulse monitor may have a double beat encoded – as described above, a *self* health/urgency measure and an *environment* health/urgency measure. These match directly with the two control loops within the AE, and the self-awareness and environment awareness properties.

3.5 The ALice Signal

An aspect to this research is that Anonymous Autonomous/Autonomic Agents need to work within the Autonomic System to facilitate self-management; as such the agents and their hosts need to be able to identify each other's credentials through such means as an ALice (Autonomic License) signal [19]. This would allow a set of communications to ensure the visiting mobile agent has valid and justified reasons for being there as well as providing security to the visiting agent in interaction with other agents and host. An unsatisfactory ALice exchange may lead to self-destruction for self-protection.

3.6 Biological Quiescence

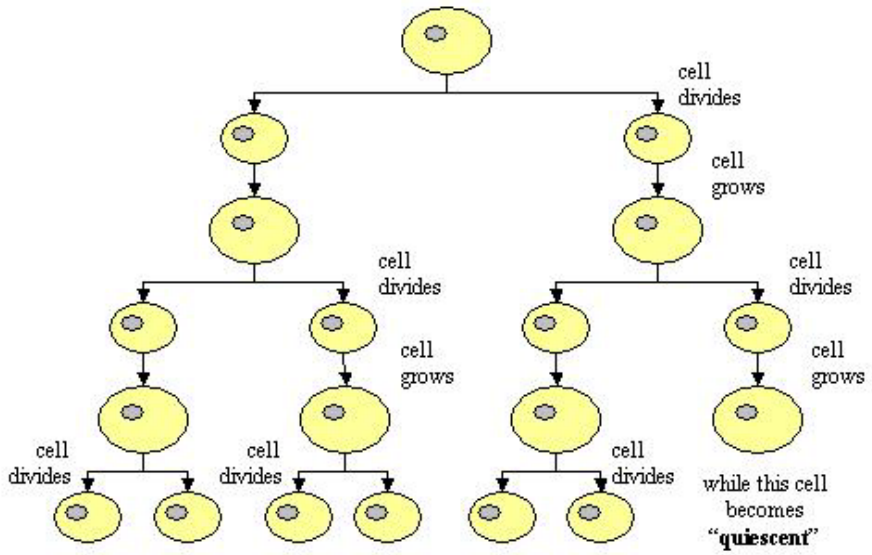

Fig. 2. Biological Cell Cycle - including Quiescent Cell

The biological cell cycle is often described as a circle of cell life and division. A cell divides into two "daughter cells" and both of these cells live, "eat", grow, copy their genetic material and divide again producing two more daughter cells. Since each daughter cell has a copy of the same genes in its nucleus, daughter cells are "clones" of each other. This "twinning" goes on and on with each cell cycle. This is a natural process.

Very fast cell cycles occur during development causing a single cell to make many copies of itself as it grows and differentiates into an embryo. Some very fast cell cycles also occur in adult animals. Hair, skin and gut cells have very fast cell cycles to replace cells that die naturally. While, as was highlighted earlier, some forms of cancer may be caused by cells cycling out of control (as well as not dying quickly enough).

But there is a kind of "parking spot" in the cell cycle, called "quiescence". A **quiescent** cell has left the cell cycle, it has stopped dividing. Quiescent cells may re-enter the cell cycle at some later time, or they may not; it depends on the type of cell. Most nerve cells stay quiescent forever. On the other hand, some quiescent cells may later re-enter the cell cycle in order to create more cells (for example, during pubescent development) [20].

3.7 The Role of Quiescence within Autonomic Agents

The agent self-destruction proposed earlier (Autonomic Apoptosis) to facilitate security measures may be considered an extreme or ultimate self-protection measure – for cases when the agent's security has been breached or the agent is endangering the system (for instance demonstrating undesirable emergent behavior) [17, 18]. Yet, not all cases may require this extreme reaction. Self-sleep (Quiescent state) instead of self-destruct (Apoptosis) may be all that is required for certain circumstances. As the situation emerges and is clarified, the agent may resume its activity or be put into an apoptotic state.

In the case of Greenberg's authenticated mobile agent carrying an archaic upgrade, as described in Section 3.3, since this is a about to perform an activity that poses a security risk, its intrinsic nature could be such that it enters a quiescent state until its behavior is confirmed and before it proceeds with its activity. As was highlighted earlier, these situations have similar overtones to where the body is at its most vulnerable during cell division. High-risk security self-managing activity can be protected by apoptosis and quiescence used to act as intrinsic mechanisms for self-destruct or self-sleep.

4 Biologically-Inspired Concepts and Autonomicity for future NASA Missions

These concepts may assist in the new radical paradigms for spacecraft design to facilitate adaptive operations and the move towards almost total onboard autonomy in certain classes of mission operations [21, 22].

A concept mission, ANTS, Autonomous Nano-Technology Swarm, planned for sometime between 2020 and 2030 is viewed as a prototype for how many future unmanned missions will be developed and how future space exploration will exploit autonomous and autonomic behavior.

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.

Section 2 clearly highlights the general problem of agent security, whether from the agent's or host's perspective. In terms of generic contribution to autonomic agent development, with many security issues the lack of an agreed standard approach to

agent-based systems prohibits immediate further practical development of apoptosis and quiescent states for generic autonomic systems.

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 behavior within acceptable parameters, where upon *apoptosis* and *quiescence* 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 behavior, and was failing to self-heal correctly. That emergent behavior (depending on what it was) may put the scientific mission in danger. The agent may be put to sleep or 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. During a solar storm the *workers* could be put into a quiescent state to protect themselves from damage.

All of the spacecraft are to be powered by batteries that are recharged by the sun using solar sails [21, 22]. 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.

5 Related Work

Forrest *et al*. in their classic work described the problem of protecting computer systems as a general problem of learning to distinguish *self* (legitimate users, corrupted data, etc.) from *other* (unauthorized users, viruses, etc.); their solution was a method for change detection inspired by the generation of T cells in the immune system [26, 26].

In relation to the Autonomic Initiative, the autonomic manager may take on this function of self-non-self discrimination as part of its self-awareness to facilitate self-protection. Yet to achieve the envisaged Autonomic Initiative long-term vision of system-level self-direction and self-management requires a high level of interaction among AMs, and since AMs at the local level will view their world as *self*, activity from the external environment may be perceived from a local AM view as *others/non-selfs.* (In the greater scheme of things, all these legitimate self-management activity will actually be *self* as opposed to *other/non-self* but the vastness of the systems of systems could result in a local AM perception/classification that these legitimate

activities are of *other/non-self*). As such, the work described in this paper is complementary to Forrest *et al*'s research; ALice used to identify and distinguish agents from the external environment as indeed part of the greater *self* as opposed to *other/non-self* as well as complementary biological inspiration from apoptosis and quiescence for intrinsic mechanisms to facilitate correct operation by *self* (for instance avoiding undesirable emergent behavior) and not just to distinguish *self* from *non-self/other*.

6 Conclusions

Autonomic agents have been gaining ground as a significant approach 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 Systems initiative, agent technologies have the potential to become an intrinsic approach within the initiative [23], not only as an enabler (e.g., ABLE agent toolkit [24]), but also in terms of creating autonomic agent environments.

Apoptosis was introduced and previously discussed in [17, 19]. We have extended this here with Autonomic Quiescence—self-sleep, a less drastic form of self-protection.

We have briefly described research into biologically-inspired concepts to be used together as intrinsic mechanisms within agents to provide inherent safety and security both at the agent and system level. We briefly discussed this in terms of the NASA concept mission, ANTS. More detailed accounts of the ANTS mission are given in [22] and [25]. We continue to seek inspiration for modeling and developing computer-based systems from ideas that are observed in nature.

Acknowledgments. This work was undertaken while Mike Hinchey was with NASA Goddard Space Flight Center, Greenbelt, MD. This research is partly supported at University of Ulster by the Computer Science Research Institute and the Centre for Software Process Technologies (CSPT) which is funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

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).

Some of the technologies described in this article are patent pending and assigned to the United States government.

References

1. N.R. Jennings and M. Wooldridge, "Agent-oriented Software Engineering," in J. Bradshaw (ed.), *Handbook of Agent Technology*, AAAI/MIT Press, 2000.

2. M..N. Huhns, V.T. Holderfield and R.L.Z. Gutierrez, "Robust software via agent-based redundancy," In *Proc. Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003*, 14-18 July 2003, Melbourne, Victoria, Australia, pp. 1018-1019.
3. G. Kaiser, J. Parekh, P. Gross, G. Valetto, "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems," *In Proc. Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop*, Seattle, WA, USA, June 2003.
4. E. Lupu *et al*., EPSRC AMUSE: Autonomic Management of Ubiquitous Systems for e-Health, 2003.
5. R. Sterritt, D.W. Bustard, "Autonomic Computing: a Means of Achieving Dependability?" In *Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS'03)*, Huntsville, AL, USA, 7-11 April 2003, pp. 247-251.
6. R. Sterritt, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID," In *Proceedings of IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003*, Banff, AB, Canada, 22-23 August 2003, pp. 433-440.
7. R. Sterritt, "Towards Autonomic Computing: Effective Event Management," In *Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW)*, Maryland, USA, 3-5 December 2002, IEEE Computer Society Press, pp. 40-47.
8. R. Sterritt, D.F. Bantz, "PAC-MEN: Personal Autonomic Computing Monitoring Environments," *In Proceedings of IEEE DEXA 2004 Workshops - 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04)*, Zaragoza, Spain, 30 August – 3 September, 2003.
9. J. O. Kephart and D. M. Chess. "The vision of autonomic computing". *Computer*, 36(1):41–52, 2003.
10. R. Sterritt, D.W. Bustard, "Towards an Autonomic Computing Environment," In *Proceedings of IEEE DEXA 2003 Workshops - 1st International Workshop on Autonomic Computing Systems*, Prague, Czech Republic, September 1-5, 2003, pp. 694-698.
11. J. Newell, "Dying to live: why our cells self-destruct," Focus, Dec. 1994.
12. R. Lockshin, Z. Zakeri, "Programmed cell death and apoptosis: origins of the theory," *Nature Reviews Molecular Cell Biology*, 2:542-550, 2001.
13. 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.
14. 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.
15. J.D. Hartline, *Mobile Agents: A Survey of Fault Tolerance and Security*, University of Washington, 1998.
16. M.S. Greenberg, J.C. Byington, T. Holding, D.G. Harper, "Mobile Agents and Security," *IEEE Communications*, July 1998.
17. R. Sterritt and M.G. Hinchey, "Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?" In *Proceedings FAABS-III, 3rd NASA/IEEE Workshop on Formal Approaches to Agent-Based Systems*, 26-27 April 2004, Greenbelt, MD, Springer Verlag LNCS **3228**, 2005.
18. R. Sterritt and M.G. Hinchey, "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions," In *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, IEEE Computer Society Press, pp. 506-511.
19. R. Sterritt and M.G. Hinchey, "Radical Concepts for Self-Managing Ubiquitous and Pervasive Computing Environments" In M.G. Hinchey, P. Rago, J.L. Rash, C.A. Rouff, R. Sterritt and W. Truszkowski (eds.) "Innovative Concepts for Agent and Autonomic Based

Systems" Proceedings of Second International Workshop on Radical Agent Concepts (WRAC 2005) LNAI **3825**, Springer Verlag, Heidelberg, 2007, DOI: 10.1007/11964995_33.
20. J. Love, *Science Explained*, 1999.
21. M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff and R. Sterritt, "Challenges of Developing New Classes of NASA Self-Managing Missions," In *Proceedings of Workshop on Reliability and Autonomic Management in Parallel and Distributed Systems (RAMPDS-05) at ICPADS-2005*, Fukuoka, Japan, 20-22 July 2005, pp. 463-467.
22. M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff and R. Sterritt, "Autonomous and Autonomic Swarms," *In Proceedings of Autonomic & Autonomous Space Exploration Systems (A&A-SES-1) at 2005 International Conference on Software Engineering Research and Practice (SERP'05)*, Las Vegas, NV, 27-30 June 2005, CREA Press, pp. 36-42.
23. J. McCann and M. Huebscher, "Evaluation issues in Autonomic Computing," In H. Jin, Y. Pan, N. Xiao, and J. Sun (Eds.): GCC 2004 Workshops, LNCS **3252**, pp. 597–608, 2004.
24. J.P. Bigus *et.al*., "ABLE: a toolkit for building multiagent autonomic systems," *IBM Systems J*., **41**(3):350-371, 2002.
25. C.A. Rouff, M.G. Hinchey, W.F Truszkowski, and J.L. Rash, "Experiences Applying Formal Approaches in the Development of Swarm-Based Space Exploration Missions," *International Journal on Software Tools for Technology Transfer*, 8(6):587-603, November 2006.
26. S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri, "Self-nonself discrimination in a computer," In *Proc. IEEE Symposium on Research in Security and Privacy 1994*, 16-18 May 1994, pp. 202-212.
27. P. D'haeseleer, S. Forrest, P. Helman, "An immunological approach to change detection: algorithms, analysis and implications," In *Proc. IEEE Symposium on Security and Privacy, 1996*, 6-8 May 1996, pp. 110-119.