# Biologically-Inspired Concepts for Self-Management of Complexity
Extended Abstract

Roy Sterritt[1] and Michael G. Hinchey[2]
[1] *University of Ulster, Jordanstown, Northern Ireland*
[2] *NASA Goddard Space Flight Center, Greenbelt, MD, USA*

## Abstract

*Inherent complexity in large-scale applications may be impossible to eliminate or even ameliorate despite a number of promising advances. In such cases, the complexity must be tolerated and managed. Such management may be beyond the abilities of humans, or require such overhead as to make management by humans unrealistic. A number of initiatives inspired by concepts in biology have arisen for self-management of complex systems. We present some ideas and techniques we have been experimenting with, inspired by lesser-known concepts in biology that show promise in protecting complex systems and represent a step towards self-management of complexity.*

## 1. Introduction

Many aspects have been cited as the cause or source of complexity: size or scale, lack of understanding of the domain, the number of interacting components, hard real-time requirements, the implementation environment, existence of undocumented and badly-structured legacy code, to name just a few. These, and many other aspects of system development, do indeed contribute to complexity. One certain contributing factor relates to the interconnection of our systems in the global environment we now operate in. Essentially, we have vast systems-of-systems that operate reasonably well when all is considered, but whose behavior can appear non-deterministic to the human operator under certain conditions simply due to the sheer scale and cascading effect in the system.

A number of research areas have developed to address these and many other issues in an attempt to reduce complexity, and to find solutions that would control or even eliminate complexity. Many promising advances have been made in a variety of subject areas towards this goal.

However, sometimes complexity just “is”. The complexity is inherent in the application domain or in the application itself. While a number of techniques and research results may indeed help to constrain that complexity to an extent, and possibly even reduce it somewhat, it is clear that there are many cases where complexity simply cannot be eliminated.

Regardless of the cause of that inherent complexity, sometimes we must simply resign ourselves to its existence, and try to *manage* it.

Future pervasive and ubiquitous systems (being distributed, dynamic, and open, without prescribed modes of interaction) will require new mechanisms for management and regulation, otherwise the emergent complexity may become overwhelming.

Various biologically-inspired concepts and related paradigms of system development have arisen in an attempt to facilitate this management [15]. Their common goal is to enable this management and to take the task of managing complexity away from users, administrators, and developers, and create *self-managing systems* [1].

## 2. Biologically-Inspired Computing

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.

In complex systems, 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.

### 2.1 Autonomic Computing

The Autonomic Computing (AC) initiative focuses on managing complexity with self-managing systems, taking inspiration from the human autonomic nervous system (ANS) [2, 4].

The ANS is that part of the nervous system that controls the vegetative functions of the body, such as circulation of the blood, intestinal activity, and secretion and production of chemical "messengers" (hormones) that circulate in the blood. The *sympathetic nervous system (SyNS)* supports "fight or flight", providing various protection mechanisms to ensure the safety and wellbeing of the body. The *parasympathetic nervous system (PaNS)* supports "rest and digest", ensuring that the body performs necessary functions for long-term health.

The general properties of an autonomic (self-managing) system can be summarised by four objectives: being self-configuring, self-healing, self-optimizing and self-protecting, and four attributes: self-awareness, self-situated, self-monitoring and self-adjusting. Essentially, the objectives represent broad system requirements, while the attributes identify basic implementation mechanisms.

Self-configuring represents a system's ability to re-adjust itself automatically; this may simply be in support of changing circumstances, or to assist in self-healing, self-optimization or self-protection.

Self-healing, in reactive mode, is a mechanism concerned with ensuring effective recovery when a fault occurs, identifying the fault, and then, where possible, repairing it. In proactive mode, it monitors vital signs in an attempt to predict and avoid "health" problems (reaching undesirable situations).

Self-optimization means that a system is aware of its ideal performance, can measure its current performance against that ideal, and has defined policies for attempting improvements. It may also react to policy changes within the system as indicated by the users. A self-protecting system will defend itself from accidental or malicious external attack. This necessitates awareness of potential threats and a means of handling those threats.

In achieving such self-managing objectives, a system must be aware of its internal state (self-aware) and current external operating conditions (self-situated). Changing circumstances are detected through self-monitoring and adaptations are made accordingly (self-adjusting). As such, a system must have knowledge of its available resources, its components, their desired performance characteristics, their current status, and the status of inter-connections with other systems, along with rules and policies of how these may be adjusted. Such ability to operate in a heterogeneous environment will require the use of open standards to enable global understanding and communication with other systems.

These mechanisms are not independent entities. For instance, if an attack is successful, this will necessitate self-healing actions, and a mix of self-configuration and self-optimisation, in the first instance to ensure dependability and continued operation of the system, and later to increase the self-protection against similar future attacks. Finally, these self-mechanisms should ensure there is minimal disruption to users, avoiding significant delays in processing.

Figure 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, 16]. 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 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.

## 3. Some Biological Inspiration

The Autonomic Computing initial set of self-* properties, commonly known as self-CHOP (CHOP: configuring, healing, optimizing, protecting), represented the general goal of the Autonomic Computing initiative. They were not intended to be mutually exclusive, nor definitive. In fact, a whole plethora of further self-* properties are being proposed and developed, leading to the coining of the term *selfware* [1, 15].

Just as the Autonomic initiative itself was inspired by the biological autonomic nervous system, we are seeking other biological inspiration, no matter how obscure, from (obviously, pre-existing) biological mechanisms, in order to establish a set of self-managing properties and functions to assist in dealing with the complexity inherent in today's and tomorrow's systems.

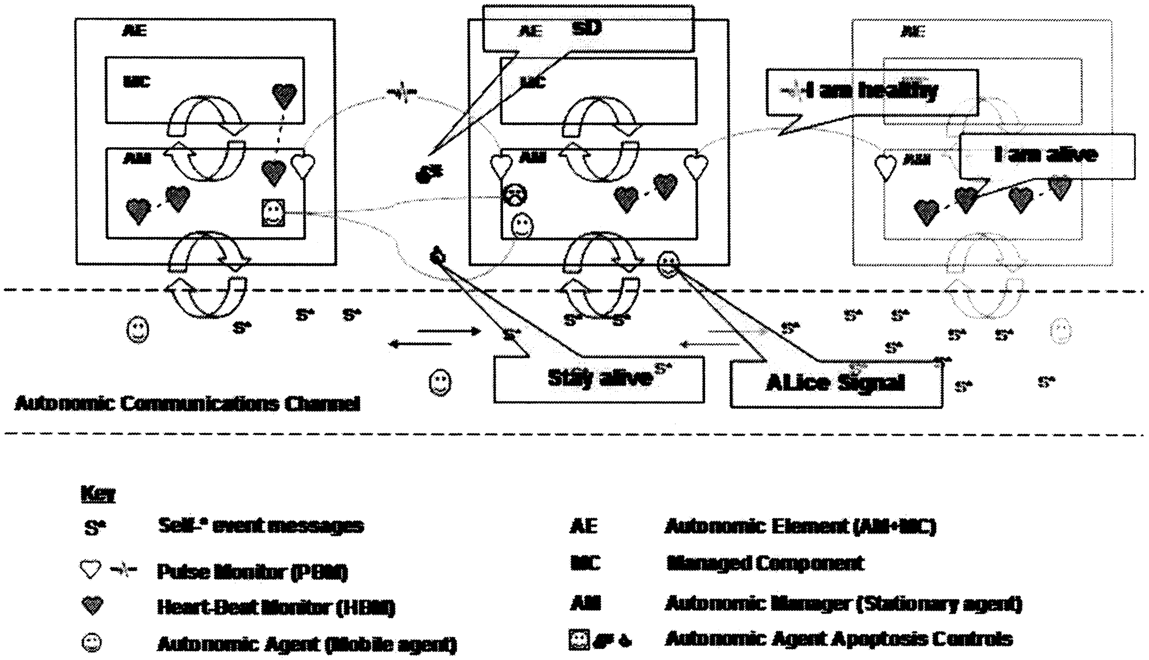
AE
MC
AM
AE SD
I am healthy
I am alive
Stay alive S*
ALice Signal
S*
Autonomic Communications Channel
Key
S* Self-* event messages
Pulse Monitor (PBM)
Heart-Beat Monitor (HBM)
Autonomic Agent (Mobile agent)
AE Autonomic Element (AM+MC)
MC Managed Component
AM Autonomic Manager (Stationary agent)
Autonomic Agent Apoptosis Controls

Figure 1: Autonomic Computing Environment.

### 3.1 Reflexes

Essentially, the aim of autonomic computing is to create robust, dependable self-managing systems [8] in an aim to deal with complexity.

At the heart of the architecture of any autonomic system are sensors and effectors. A control loop is created by monitoring behavior through sensors, comparing this with expectations (knowledge, as in historical and current data, rules and beliefs), planning what action is necessary (if any), and then executing that action through effectors. The closed loop of feedback control provides the basic backbone structure for each system component. Figure 2 highlights that there are two conceptual control loops in an Autonomic Element – one for self-awareness and another for self-situation (environmental awareness and context-awareness).

IBM represents this self-monitor/self-adjuster control loop as the monitor, analyze, plan and execute (MAPE) control loop. The monitor-and-analyze parts of the structure process information from the sensors to provide both self-awareness and an awareness of the external environment. The plan-and-execute parts decide on the necessary self-management behavior that will be executed through the effectors. The MAPE components use the correlations, rules, beliefs, expectations, histories, and other information known to the autonomic element, or available to it through the knowledge repository within the AM.

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. Figure 2 depicts that the autonomic manager communications (AM⇔AM) also includes a reflex signal.

To facilitate this, 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 [9, 10].

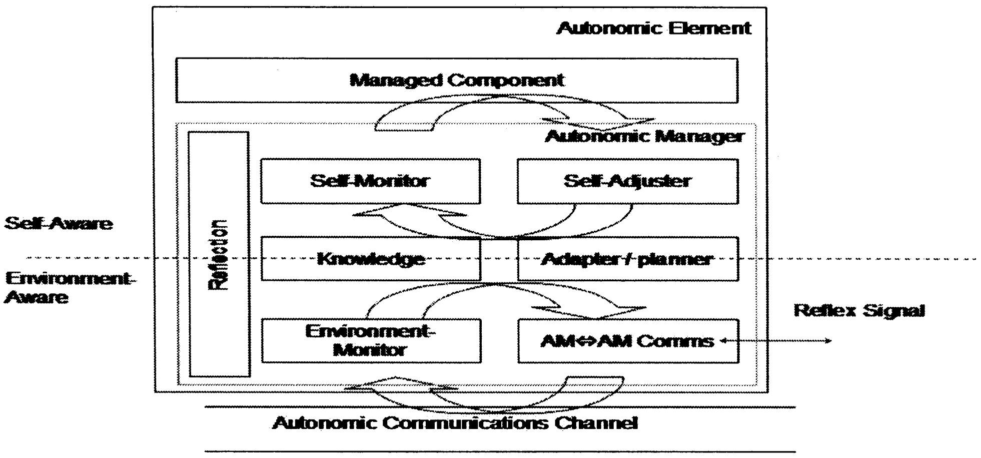

Figure 2: Autonomic Element, with reflection and reflex layers.

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 [11] – 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 [9].

This reflex component may be used to safe-guard the autonomic element by communicating its health to another AE. The component may also be utilized to communicate environmental health information. 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 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 (Figure 2) 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.

### 3.2 Biological Apoptosis

It is believed that a cell knows when to commit suicide because cells are programmed to do so—self-destruction (sD) is an intrinsic property. This sD is delayed due to the continuous receipt of biochemical retrieves. This process is referred to as *apoptosis*, 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 [5, 6, 7]. The process has also been nicknamed "death by default," where cells are prevented from putting an end to themselves due to constant receipt of biochemical "stay alive" signals [3].

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

The suicide and reprieve controls may be compared to the dual-key on a nuclear missile. 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.

The need for component self-destruct is just one self-* property being investigated to provide an intrinsic safety mechanism against non-desirable emergent behavior [13, 14, 15]. Apoptosis (stay alive) is a proposed additional construct used to safeguard both the system and agent; a "stay alive" signal indicates that the agent is still operating within the correct context and behavior, and should not self-destruct.

Agents and their hosts need to be able to identify each other's credentials through such means as an ALice (Autonomic License) signal [15]. 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.

### 3.3 Vaccination and Immunization

The human body is quite amazing in the lengths it goes to in order to protect itself. Apoptosis, described above, whereby cells are pre-programmed to die is an extreme form of self-protection.

An equally extreme form of protection occurs where the body (or more specifically the Sympathetic Nervous System) determines that the body is in danger, and protects itself by pre-empting a threat and causing, and subsequently dealing with, a similar but lesser threat.

This idea has been adapted by the medical profession to derive the processes of inoculation and, later, vaccination. Inoculation involves giving a patient a small dose of a virus with the intention that the body will build up an immunity to that virus. It does, however, run the risk that the patient will not build up immunity and may die, in the case of serious viruses. The idea of vaccination was first suggested by Edward Jennings, who's experiments determined that infection with the relatively innocuous cowpox virus could result in immunity to the more serious but related disease, smallpox. Louis Pasteur later perfected this idea by to use attenuated (weakened) infectious agents to produce active immunity and termed the weakened agent a "vaccine" (from "vacca," the Latin for "cow").

The body itself does something similar when in danger of exposure to extreme sounds that may damage the ear drum. The *acoustic reflex*, or stapedius reflex, is an involuntary muscle contraction in the middle ear of mammals in response to high-intensity sound stimuli.

More interestingly, a mechanism (only discovered in the late 1970s) called *otoacoustic emission*, involves the generation of sound from within the inner ear in response to over-activity of the cochlear amplifier. That is, when the body is presented with a sound that is potentially damaging, the inner ear generates a counter-sound, which is benign, and protects the inner ear from hearing the damaging sound. A similar concept is used in noise-cancelling headphones—sound is sampled and a sound wave that is out of phase and exactly the opposite shape is generated to counteract and effectively eliminate the undesirable noise.

We have been looking at an analogy of this as a security mechanism in self-managing systems. In this context, the "noise" is spurious signals or signals generated by a rogue agent that has failed to engage in a satisfactory ALice exchange (as described in Section 3.2) and requires the system to self-protect. The idea is to protect the system by counteracting the noise by generating a signal that would stop autonomic managers from receiving (or at least from reacting to) these unwanted signals, effectively having an overriding effect on any reflex signal received by the autonomic managers.

## 4. Conclusion

The Autonomic Computing initiative, and other similar initiatives based on inspiration from concepts exhibited in biological systems, have the goal of developing self-managing systems. This goal in turn is all about managing complexity and ensuring that complex systems operate correctly without human intervention, often because management by humans is simply not realistic or even feasible.

We have presented a number of biologically-inspired concepts which build upon phenomena observed in nature, and are adapted to apply to computer-based systems. This approach augurs much promise for self-management of complexity.

## Acknowledgements

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 Software Engineering Laboratory, Goddard Space Flight Center (Code 581).

This research is partly supported at University of Ulster by the Computer Science Research Institute (CSRI) 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.

Several of the technologies described in this paper are patent-pending and assigned to the United States Government.

## References

1. M. G. Hinchey and R. Sterritt, "Self-Managing Software," *Computer*, 39(2):107-109, February 2006.
2. P. Horn, "Autonomic computing: IBM perspective on the state of information technology," IBM T.J. Watson Labs, NY, 15th October 2001
3. 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.
4. J. O. Kephart and D. M. Chess. "The vision of autonomic computing". *Computer*, 36(1):41–52, 2003.
5. 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.
6. R. Lockshin, Z. Zakeri, "Programmed cell death and apoptosis: origins of the theory," *Nature Reviews Molecular Cell Biology*, 2:542-550, 2001.
7. J. Newell, "Dying to live: why our cells self-destruct," Focus, Dec. 1994.
8. 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.
9. 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.
10. 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.
11. 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.
12. R. Sterritt and 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.
13. 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.
14. 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.
15. R. Sterritt and M.G. Hinchey, "Biological Inspired Concepts for Self-Managing Ubiquitous and Pervasive Computing Environments" In *Proceedings WRAC-II, 2nd NASA/IEEE Workshop on Radical Agent Concepts*, Sept. 2005, Greenbelt, MD, Springer Verlag LNCS, 2006.
16. R. Sterritt and M.G. Hinchey, "From Here to Autonomicity: Self-Managing Agents and the Biological Metaphors that Inspire Them," Proceedings of Integrated Design & Process Technology Symposium (IDPT 2005), Beijing, China, 13-17 June, pp 143-150.