

20 Years of Autonomic Computing

and the contribution to and from NASA

Roy Sterritt

r.sterritt@ulster.ac.uk


@RoysterUlster



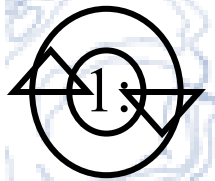
Google Scholar



Talk Outline

- 
- 1: 2001–2021: 20 years of Autonomic Computing in 20 mins
 - 2: Research with NASA in Autonomicity – *contribution to & from*
 - 3: The next 20 years for Autonomic Computing

Talk Outline



1: 2001–2021: 20 years of Autonomic Computing in 20 mins

2: Research with NASA in Autonomicity – *contribution to & from*

3: The next 20 years for Autonomic Computing

Where we are today (2001)?



Where we are today (2001)?

- MAD TV, “Antiques Roadshow, 3005 AD”

VALTREX:

“Ah ha. You paid 7 million Rubex too much. My suggestion: beam it directly into the disposal cube.

These pieces of crap crashed and froze so frequently that people became violent!

Hargh!”



“Worthless Piece of Crap: 0 Rubex”

Computer Software Systems are Not Self-Managing

NASA Ground Control ☺



Best Case



Worst Case



What is Autonomic Computing?

“Computer systems that can regulate themselves much in the same way as our autonomic nervous system regulates and protects our bodies”

*Paul Horn,
director of IBM research, 2001*



Times Higher Education (THE), 15th Aug 2003
Self-help is key to healthy computers



Autonomic Computing

Self-managing systems that ...

Increase Responsiveness

Adapt to dynamically changing environments

Business Resiliency

Discover, diagnose, and act to prevent disruptions

Operational Efficiency

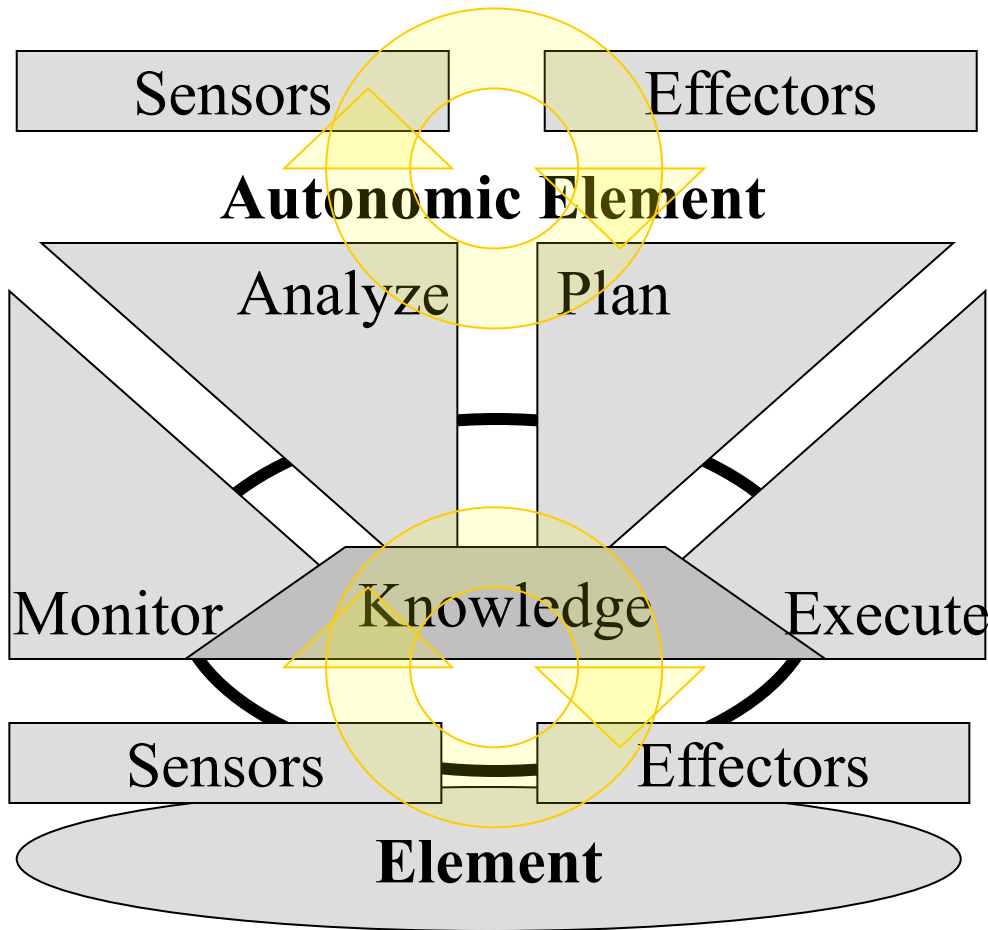
Tune resources and balance workloads to maximize use of IT resources

Secure Information and Resources

Anticipate, detect, identify, and protect against attacks



...achieving the correct balance between what is managed by a person versus the system



- An **autonomic element** contains a continuous control loop that monitors activities and takes actions to adjust the system to meet business objectives
- Autonomic elements learn from past experience to build action plans



Autonomic Maturity Levels

Basic Level 1

Manual analysis and
problem solving

Managed Level 2

Centralized tools,
manual actions

Predictive Level 3

Cross-reference
correlation and
guidance

Adaptive Level 4

System monitors,
correlates and
takes action

Autonomic Level 5

Dynamic business-
policy-based
management

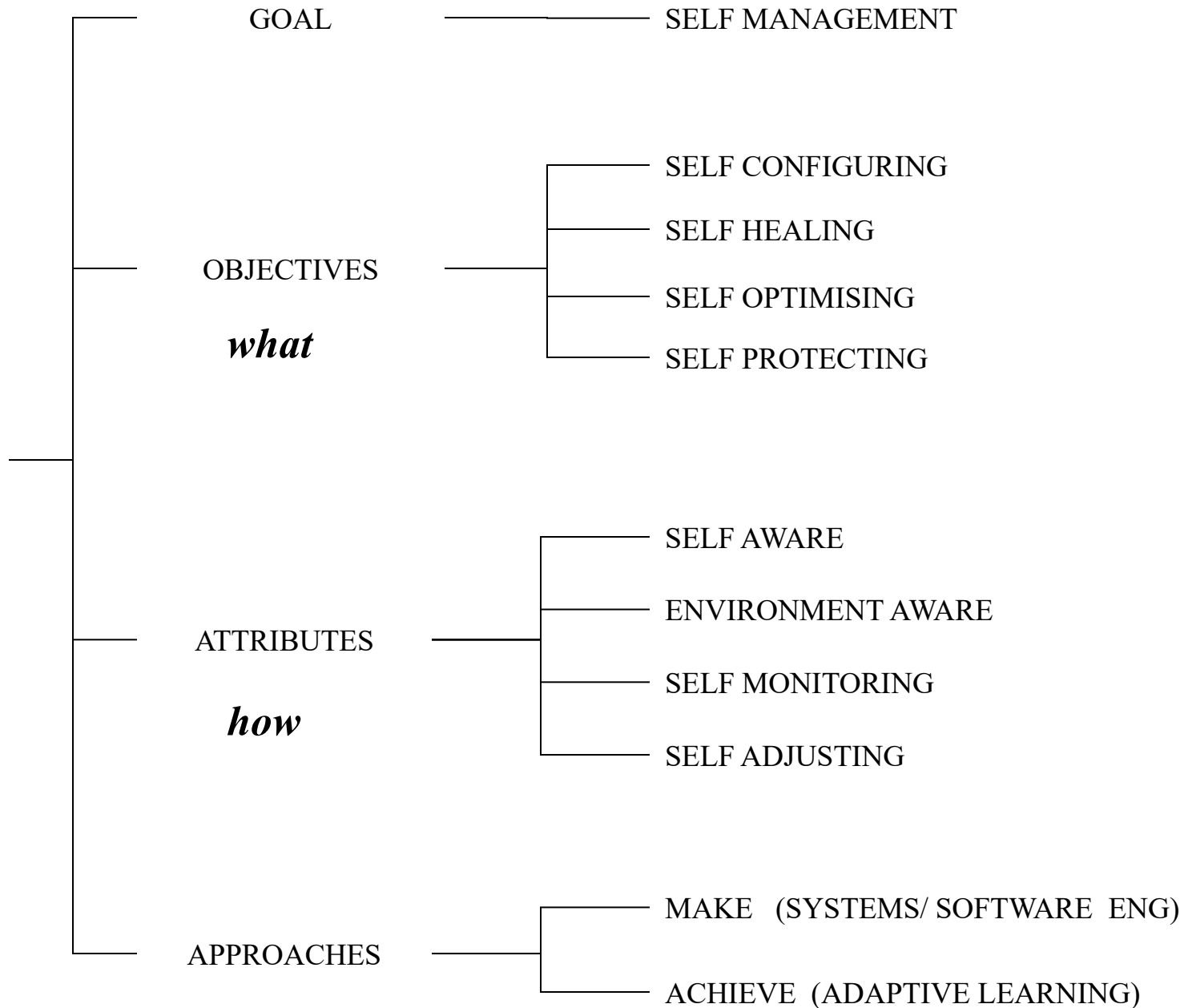


Autonomic Computing Characteristics

- Possess system identity - detailed knowledge of components
- Self configure & re-configure - adaptive algorithms
- Optimise operations - adaptive algorithms
- Recover - no impact on data or delay on processing
- Self protection
- Aware of environment and adapt
- Function in a heterogeneous world
- Hide complexity



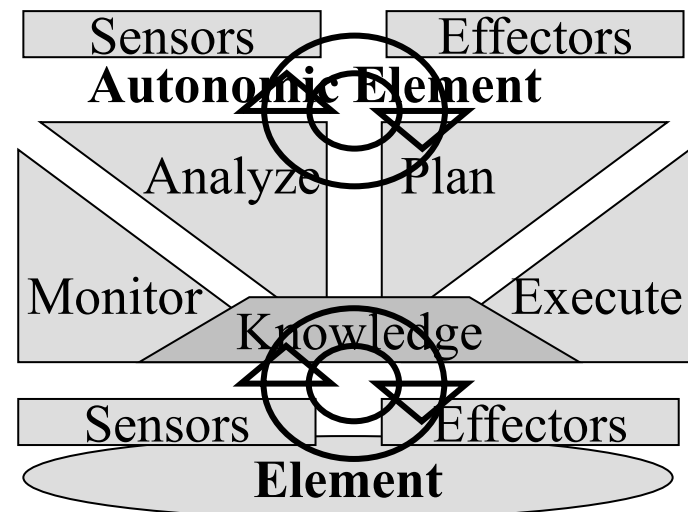
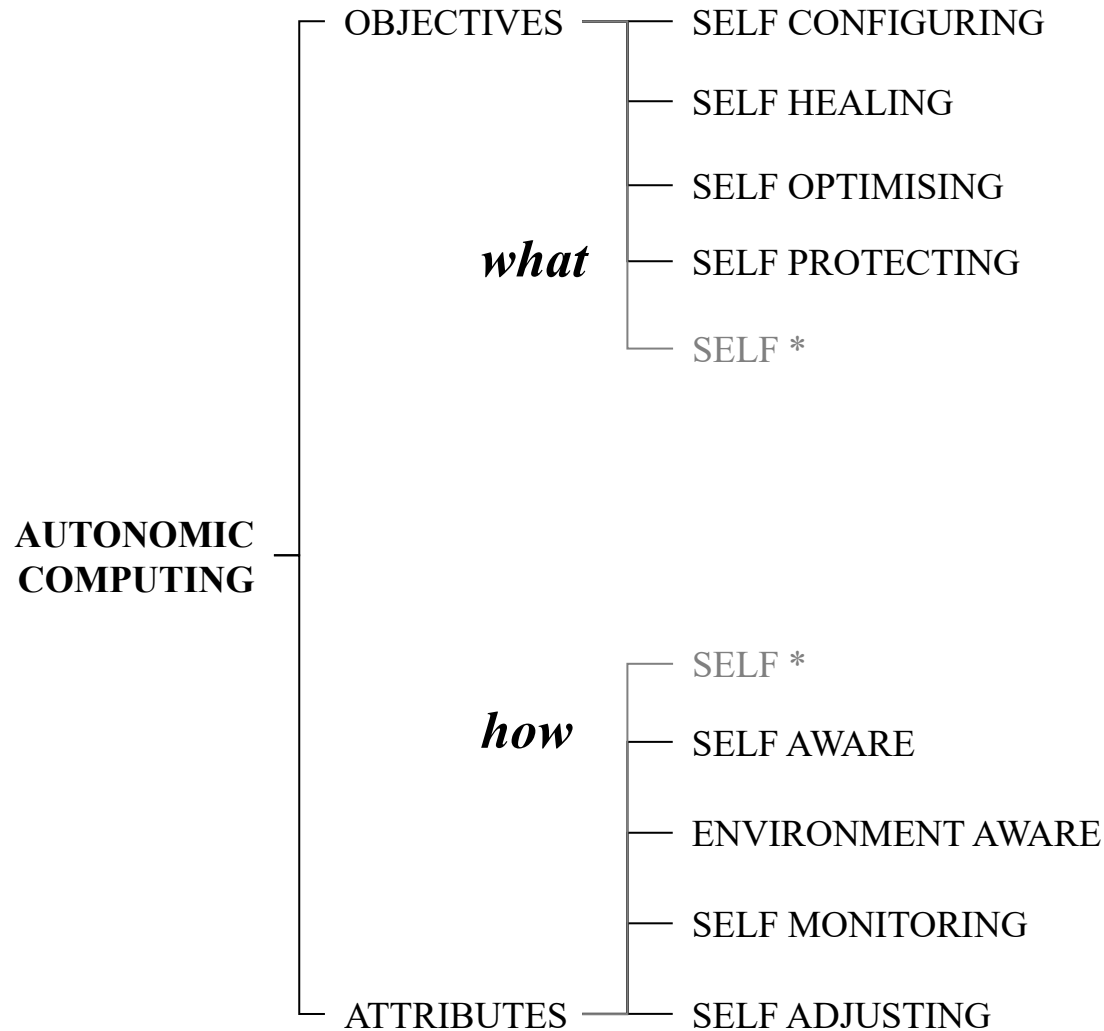
AUTONOMIC COMPUTING



The AC Tree



Autonomic Systems – *Selfware & self-* properties*





Autonomic Computing – *Other Emerging self-* properties*

- self-anticipating,
- self-adapting,
- self-critical,
- self-defining,
- self-diagnosis,
- self-governing,
- self-installing
- self-organized,
- self-recovery,
- self-reflecting,
- self-simulation
- self-stabilizing

Therefore ...

- self-* or self-ware

All positive?

- selfish ☺



**Self-managing :
(wo)man in the loop?**

Issue : Trust

Unexpected emergent behaviour

Race conditions etc.

Self- properties*



self-*

Self-managing properties.

self-anticipating

The ability to predict likely outcomes or simulate self-* actions.

self-assembling

Assembly of models, algorithms, agents, robots, etc.; self-assembly is often influenced by nature, such as nest construction in social insects. Also referred to as self-reconfigurable systems.

self-awareness

“Know thy self”; awareness of internal state; knowledge of past states and operating abilities.

self-chop

The initial four (and generic) self-properties (Self-Configuration, Self-Healing, Self-Optimisation and Self-Protection).

self-configuring

The ability to configure and re-configure in order to meet policies/goals.

Self- properties*



self-critical

The ability to consider if policies are being met or goals are being achieved (alternatively, self-reflect)

self-defining

In reference to autonomic event messages between Autonomic Managers: contains data and definition of that data–metadata (for instance using XML).

In reference to goals/policies: defining these (from self-reflection, etc.).

self-governing

As in autonomous: responsibility for achieving goals/tasks.

self-healing

Reactive (self-repair of faults) and Proactive (predicting and preventing faults).

self-installing

As in a specialized form of self-configuration – installing patches, new components, etc or re-installation of OS after major crash.

self-managing

Autonomous, along with responsibility for wider self-* management issues.

Self- properties*



self-optimizing

Optimization of tasks and nodes.

self-organized

Organization of effort/nodes. Particularly used in networks/communications.

self-protecting

The ability of a system to protect itself.

self-reflecting

The ability to consider if routine and reflex operations of self-* operations are as expected. May involve self-simulation to test scenarios.

self-similar

Self-managing components created from similar components that adapt to a specific task, for instance a self-managing agent.

self-simulation

The ability to generate and test scenarios, without affecting the live system.

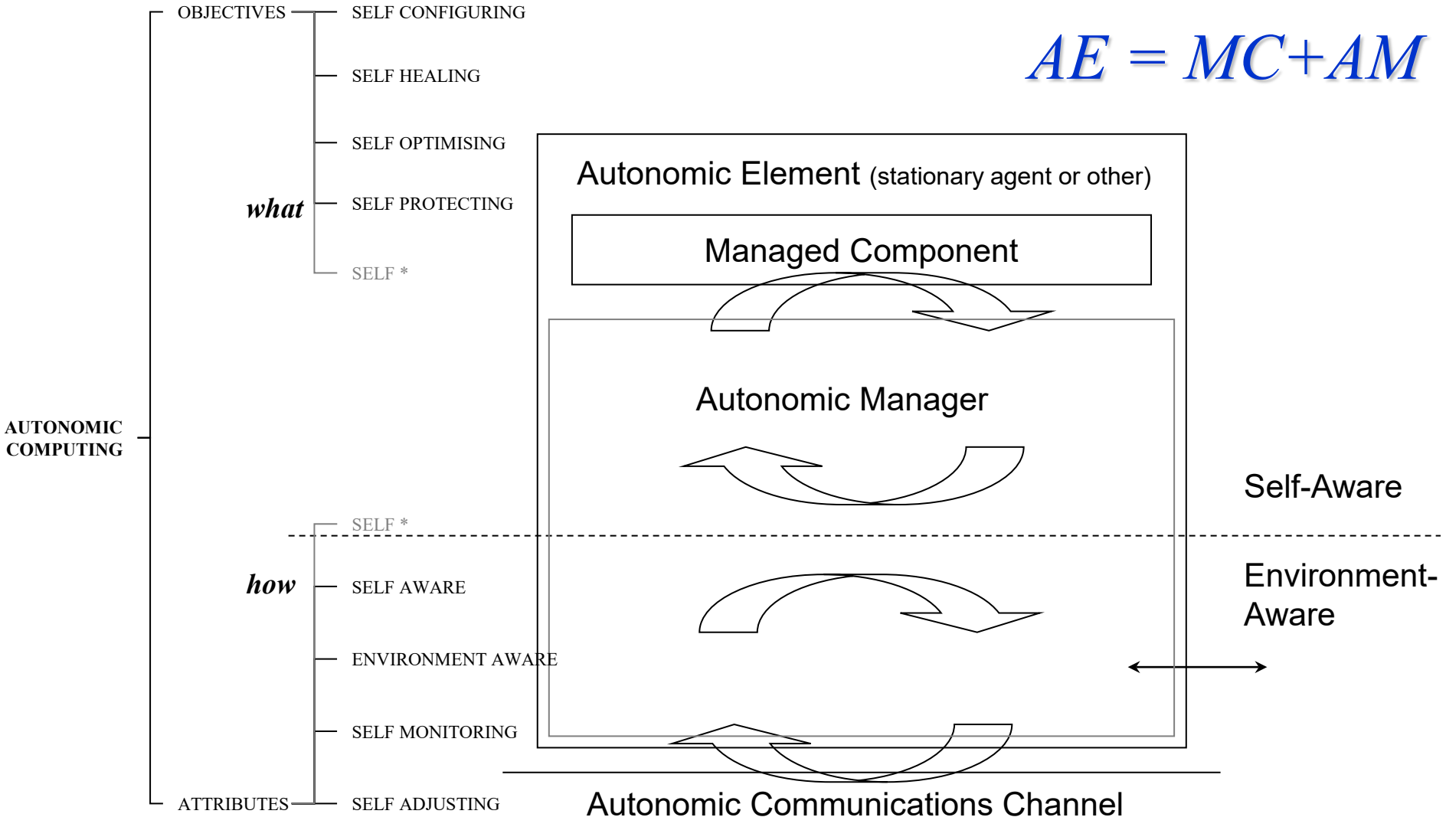
selfware

Self-managing software, firmware and hardware.



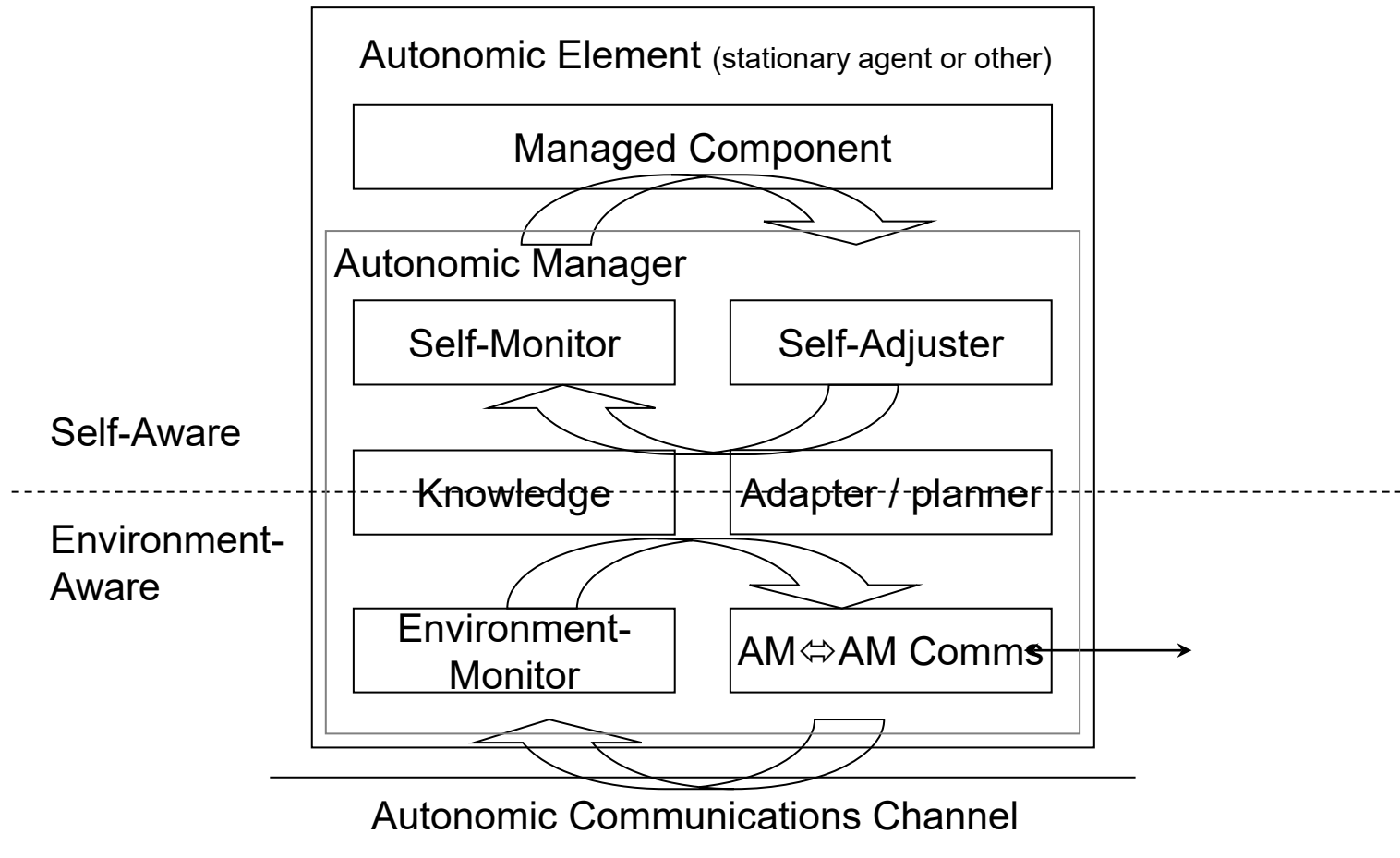
Autonomic Computing – *The Autonomic Element*

$$AE = MC + AM$$



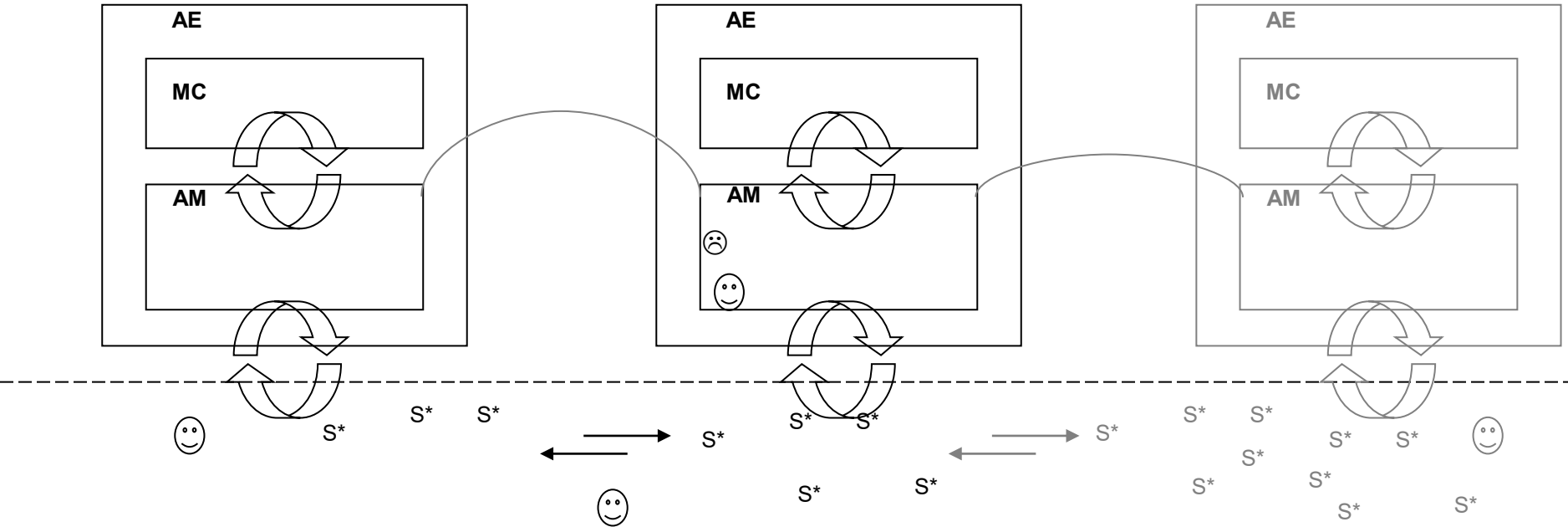


Autonomic Computing – *The Autonomic Manager*





Autonomic Computing – *The Autonomic Environment*



Autonomic Communications Channel

Key

S^* Self-* event messages

AE Autonomic Element (AM+MC)

MC Managed Component

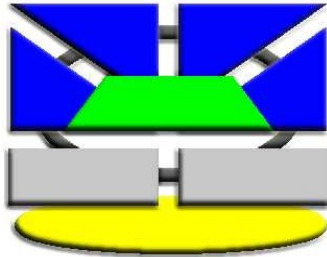
AM Autonomic Manager (Stationary agent)

☺ Autonomic Agent (Mobile agent)

How Do We Make Components Autonomic?

- **Autonomic elements have two management tasks**
 - **They manage themselves**
 - **They manage their relationships with other elements through negotiated agreements**

Autonomic Database

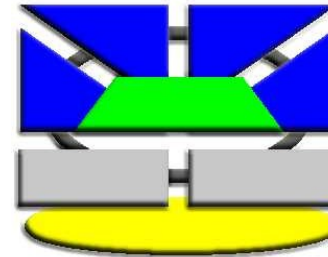


“I need to allocate
some additional table
space ”

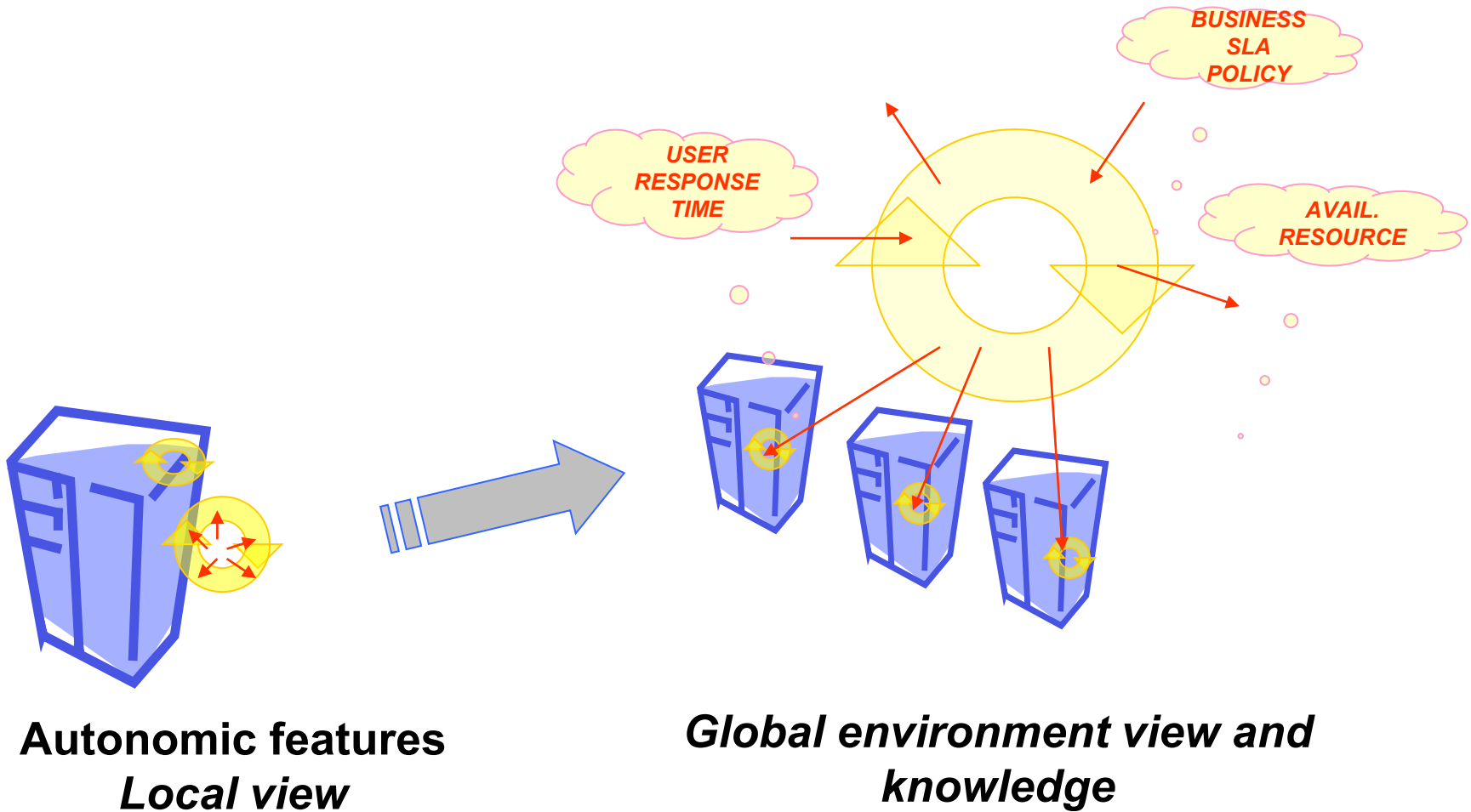


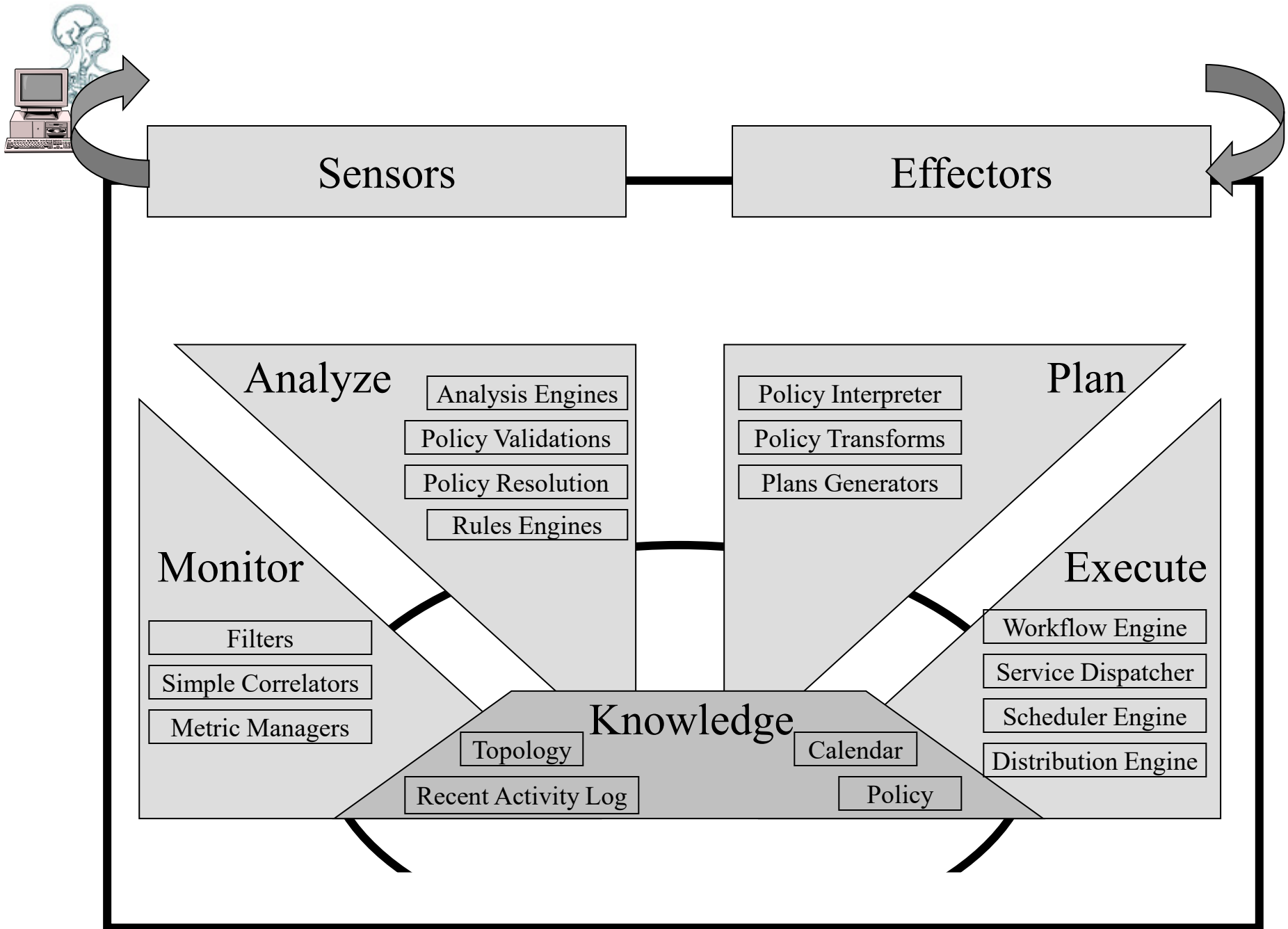
“I am reallocating
storage and
moving the
information”

Autonomic Storage Array



Autonomic control loops: next step evolution

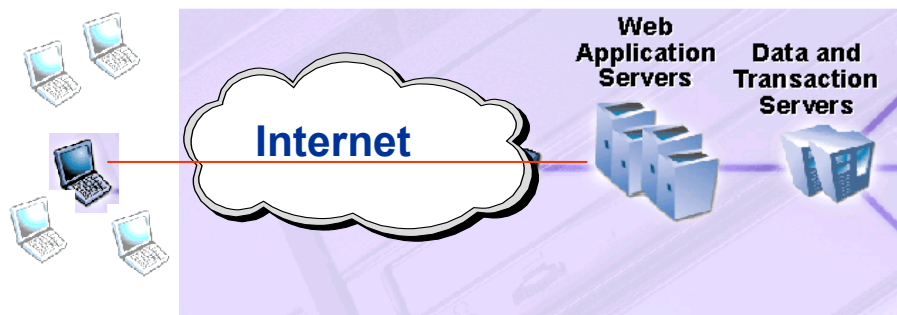






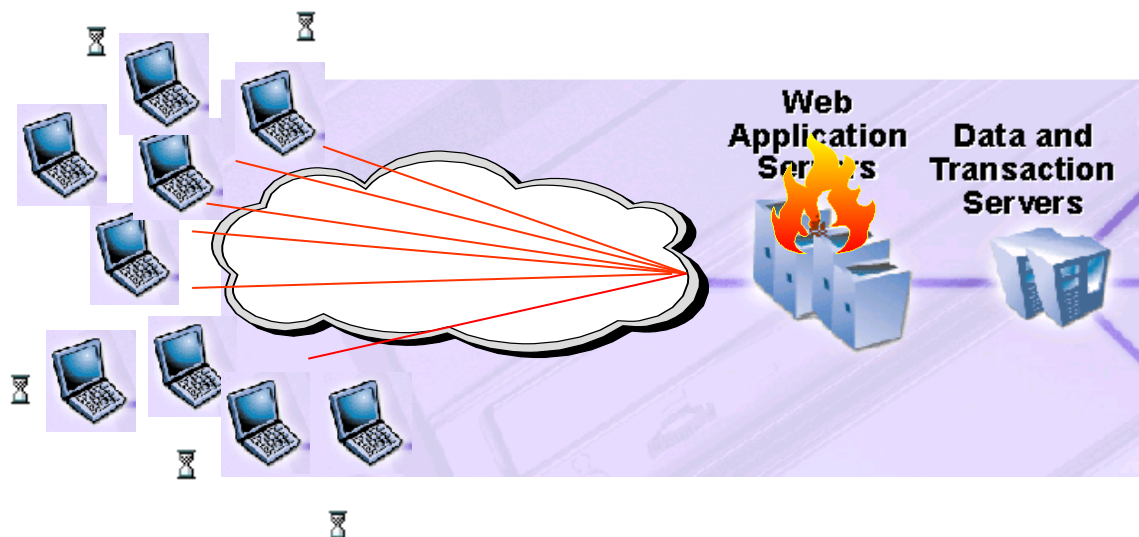
Putting It All Together – An Example

Systems can go from steady state ...

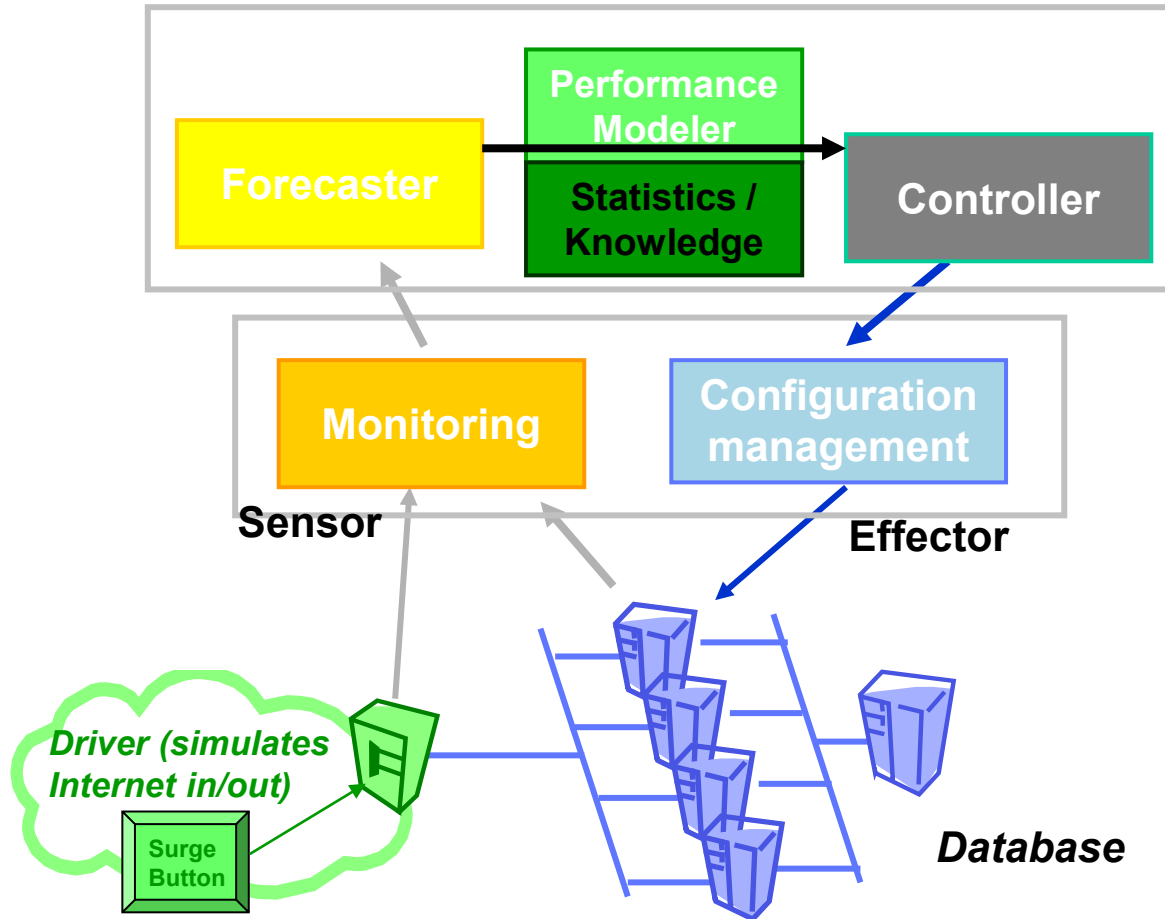


to overloaded without warning

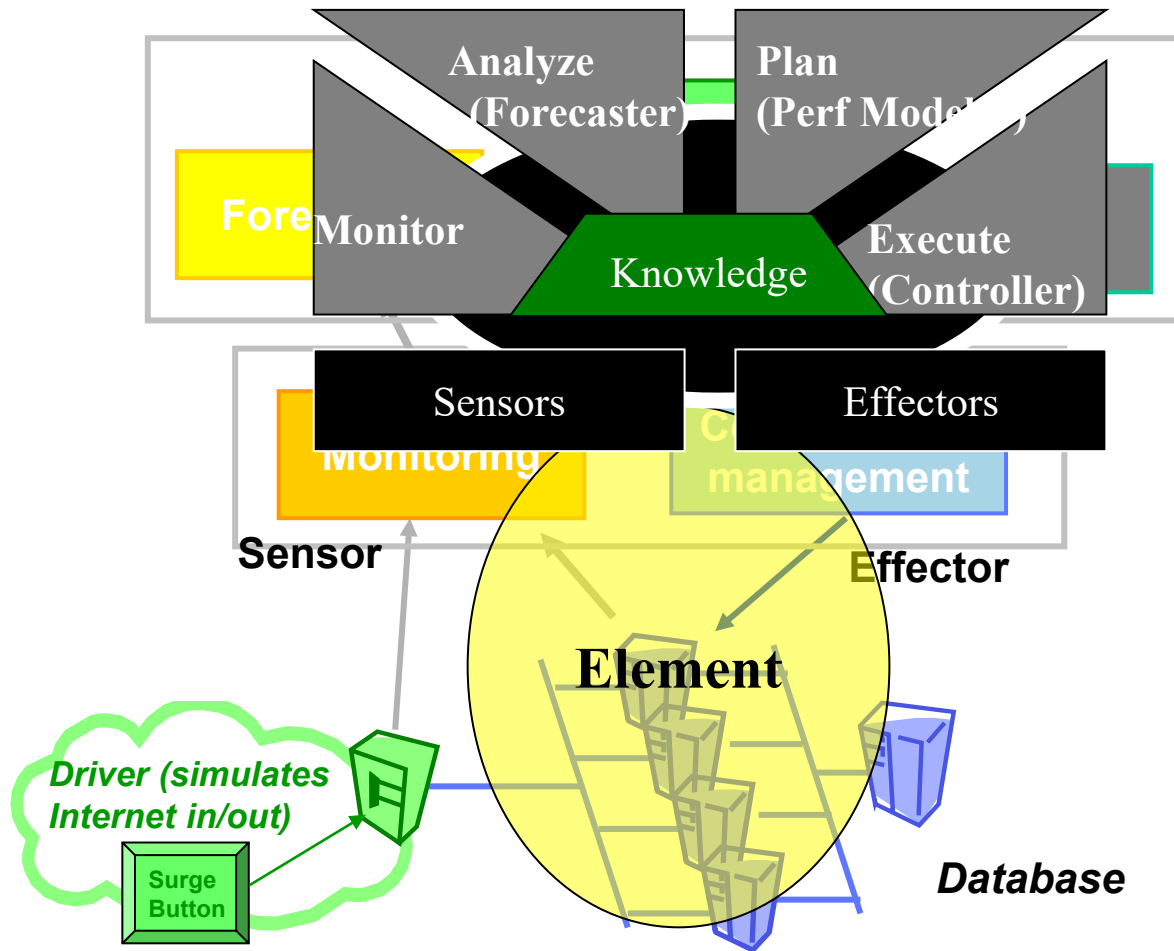
**Few
minutes
later...**



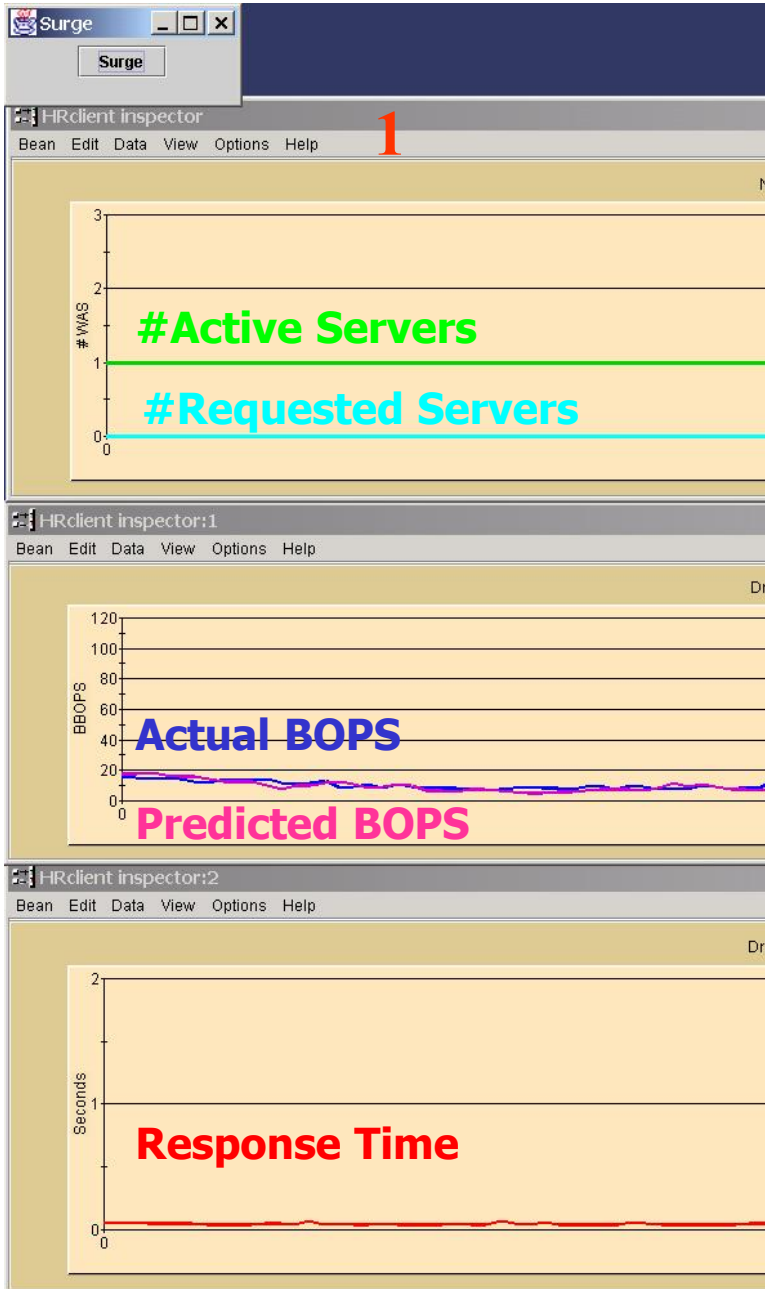
Autonomic Computing: Dynamic Surge Protection



Autonomic Computing: Dynamic Surge Protection

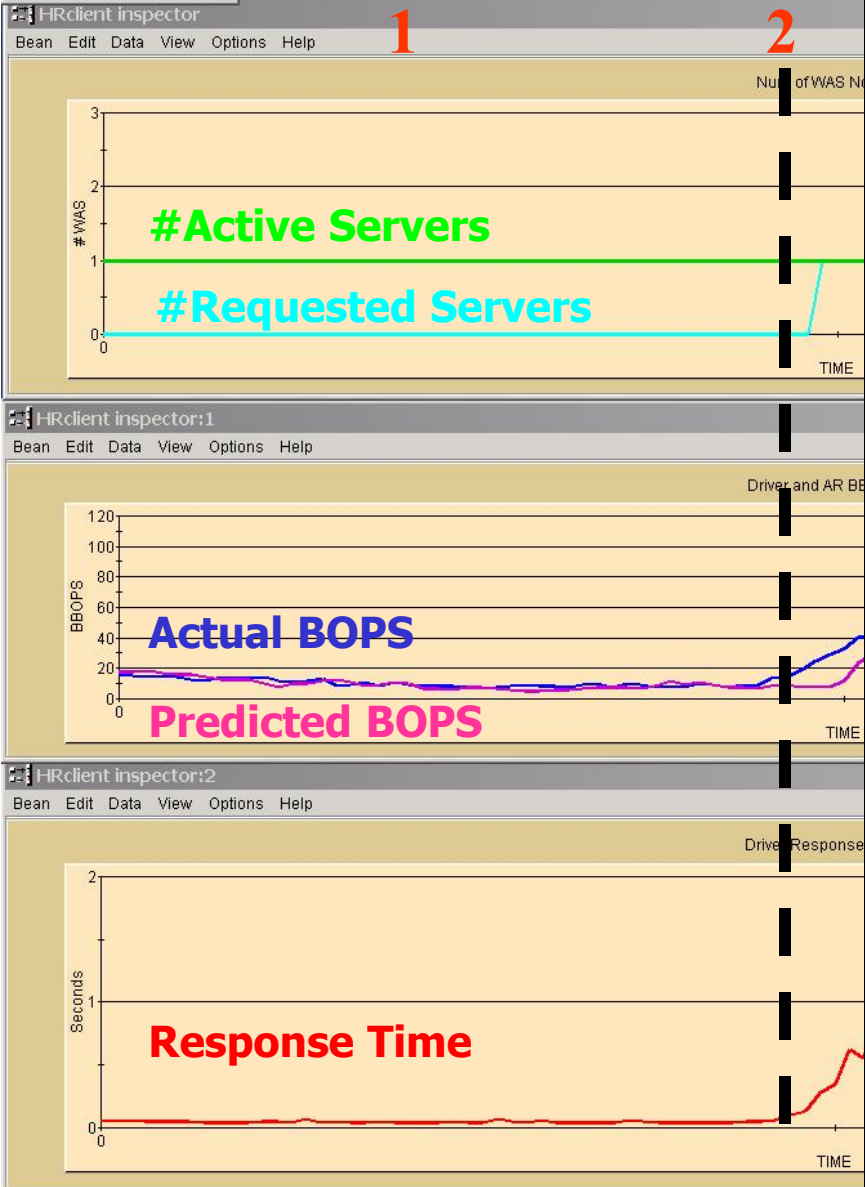


1. Steady State



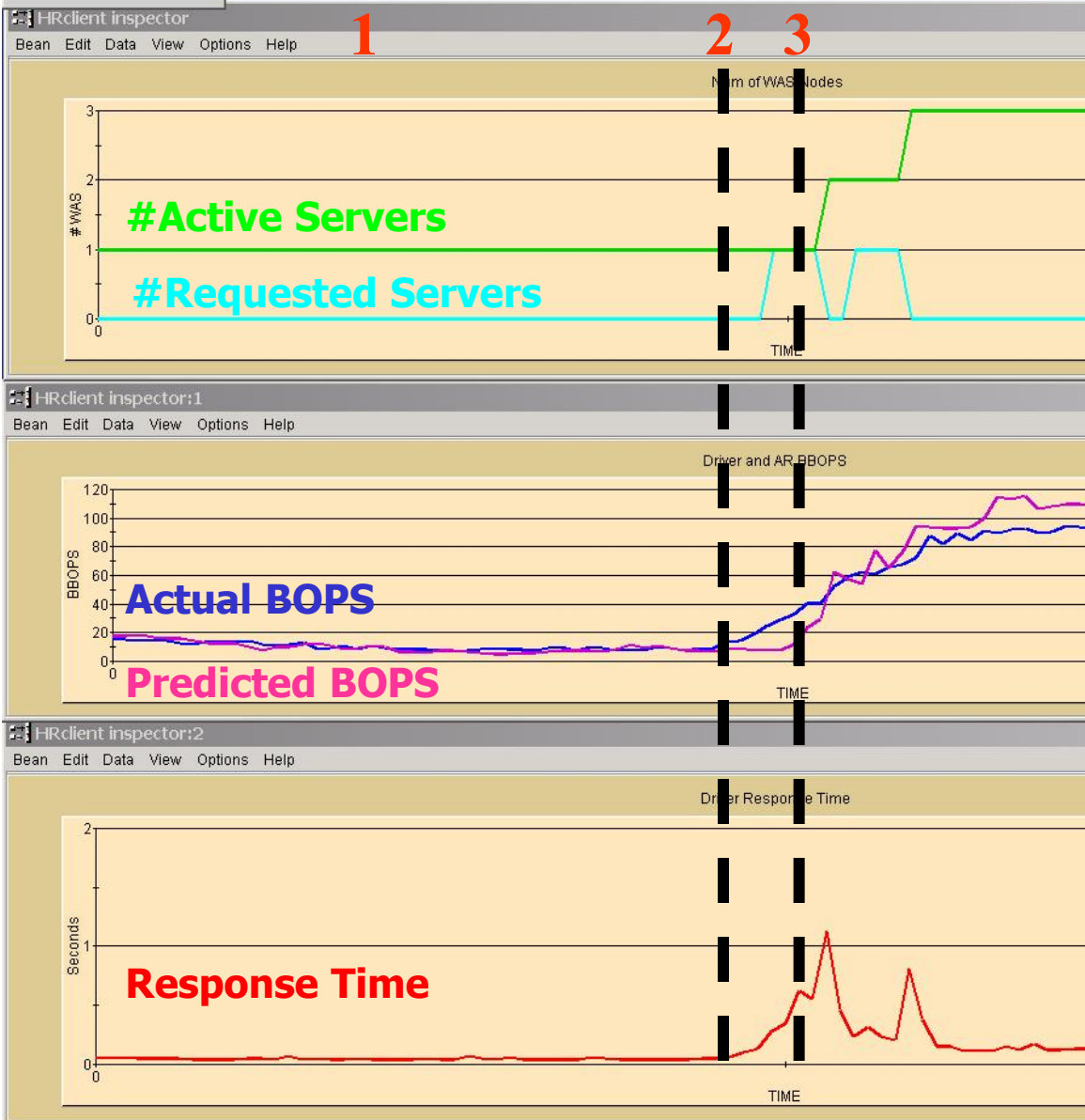


2. Monitor, Detect Surge

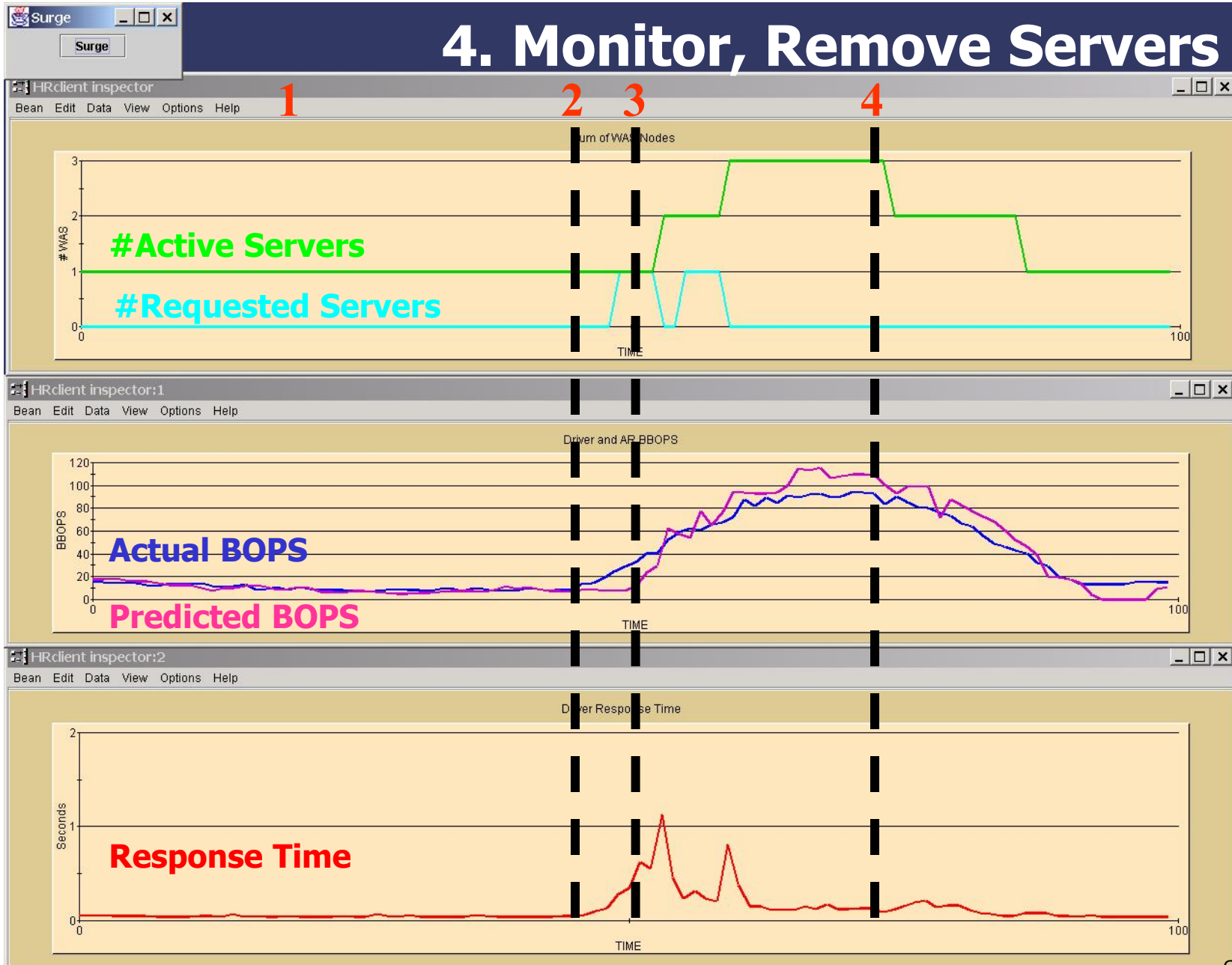




3. Forecast, Provision Servers



4. Monitor, Remove Servers



Talk Outline

1: 2001–2021: 20 years of Autonomic Computing in 20 mins

 2: Research with NASA in Autonomicity – *contribution to & from*

3: The next 20 years for Autonomic Computing



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1012128

L
I
M
I
T
E
D

FN-LA

A
C
C
E
S
S



ROY STERRITT
UK

VISITING: HINCHEY, MICHAEL

EXPIRES:

12/03/04



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1013195



Goddard Space Flight Center
Greenbelt, Maryland 20771

ID# 10118077

L
I
M
I
T
E
D

FN-LA

A
C
C
E
S
S



ROY STERRITT
UK

VISITING: HINCHEY, MICHAEL

EXPIRES:

10/23/10



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1014456



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1014666

L
I
M
I
T
E
D

FN-LA

A
C
C
E
S
S



ROY STERRITT
UK

VISITING: HINCHEY, MICHAEL

EXPIRES:

10/24/05



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1016160

L
I
M
I
T
E
D

FN-LA

A
C
C
E
S
S



ROY STERRITT
UK

VISITING: HINCHEY, MICHAEL

EXPIRES:

04/30/06



Goddard Space Flight Center
Greenbelt, Maryland 20771
ID# 1017819

L
I
M
I
T
E
D

FN-LA

A
C
C
E
S
S



ROY STERRITT
UK

VISITING: HINCHEY, MICHAEL

EXPIRES:

12/14/06

Definitions (1)



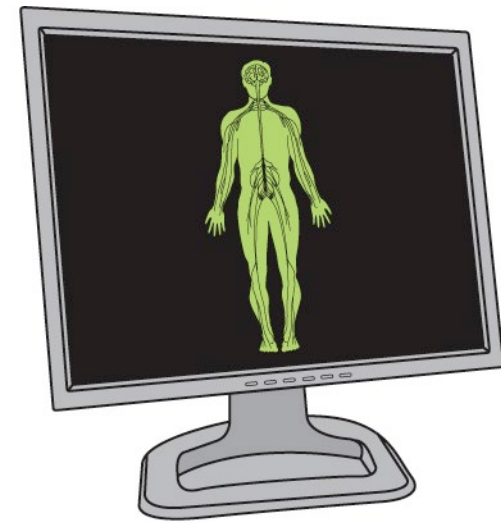
au·ton·o·mous (aw tónnəməs)

adj.

- Not controlled by others or by outside forces; independent: *an autonomous judiciary; an autonomous division of a corporate conglomerate.*
- Independent in mind or judgment; self-directed.
 - Independent of the laws of another state or government; self-governing.
 - Of or relating to a self-governing entity: *an autonomous legislature.*
 - Self-governing with respect to local or internal affairs: *an autonomous region of a country.*
- Autonomic.

[From Greek autonomos : auto-, *auto-* + nomos, *law*]

Definitions (2)



au·to·nom·ic (àwtə nómmik)
adj.

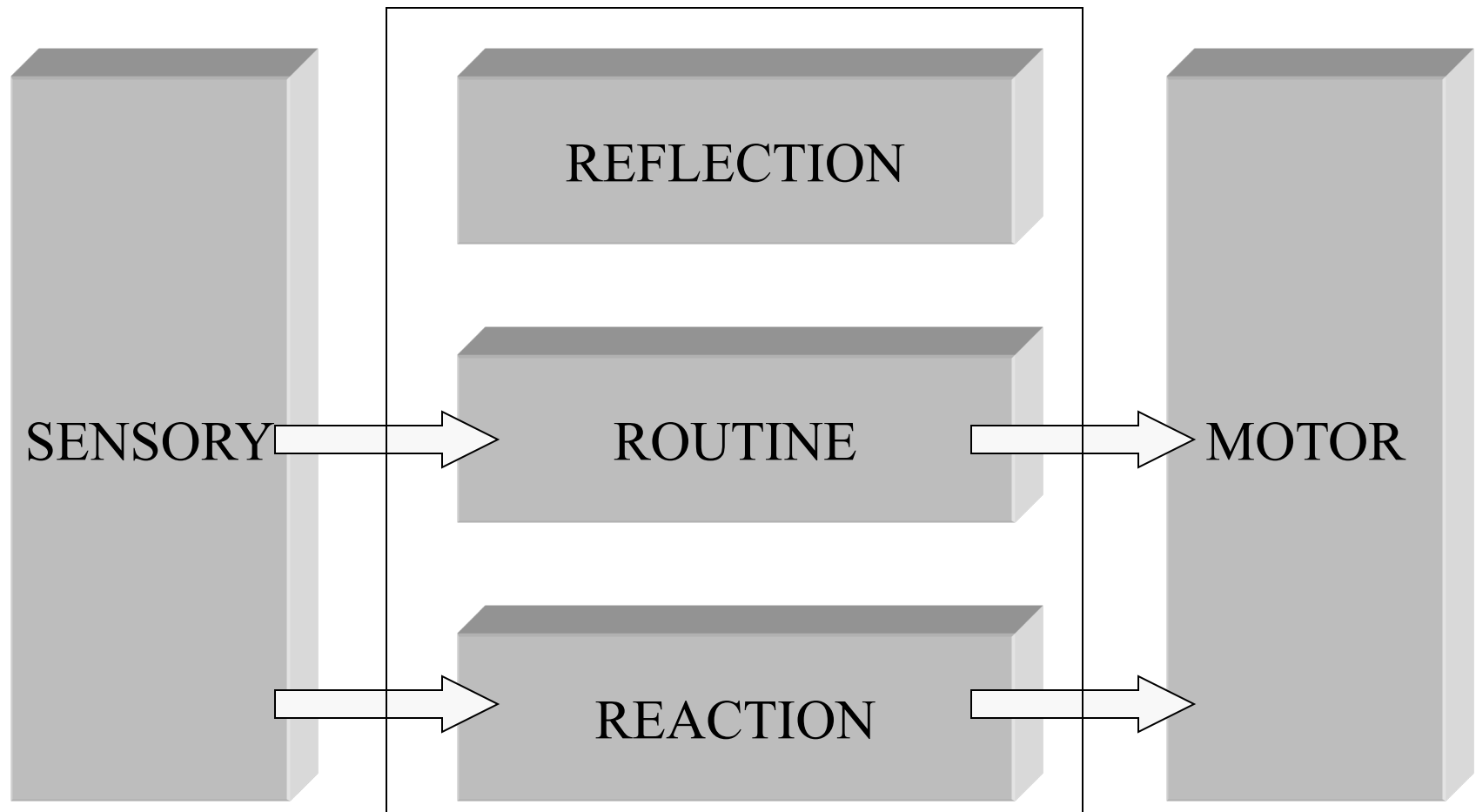
Physiology.

- Of, relating to, or controlled by the autonomic nervous system.
- Occurring involuntarily; automatic: *an autonomic reflex*.
- Resulting from internal stimuli; spontaneous.

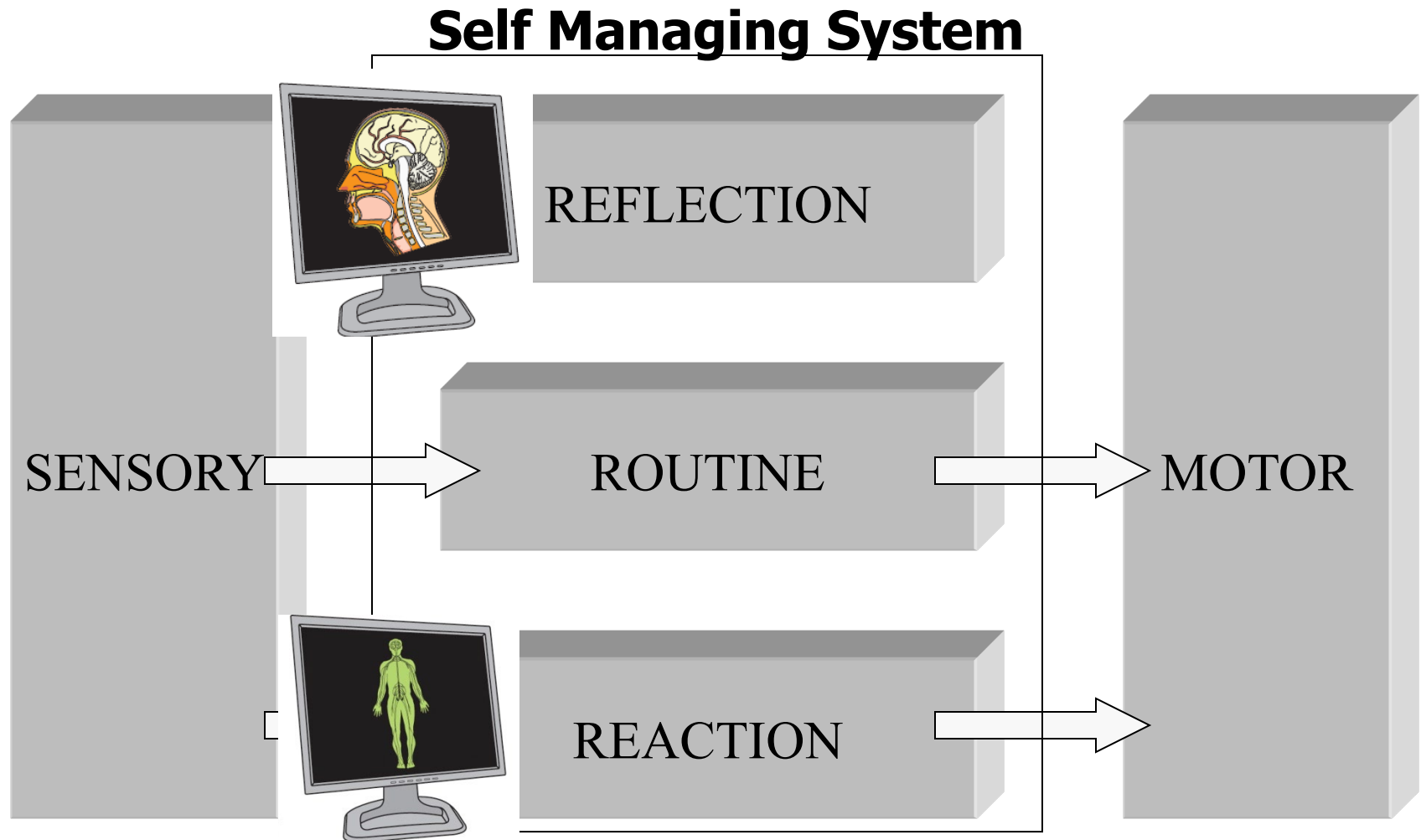
au·ton·o·mic·i·ty (àwtə nóm i sítte)
n.

- The state of being autonomic.

One View of the Relationship ?



One View of the Relationship ?



Another View of the Relationship ?

Self Managing System – DARPA/ISO's Autonomic Information Assurance (AIA) program



MISSION PLANE



CYBER
PLANE



HARDWARE PLANE

Another View of the Relationship ?

Self Managing System – Communications / (EU) Autonomic Communications



KNOWLEDGE PLANE

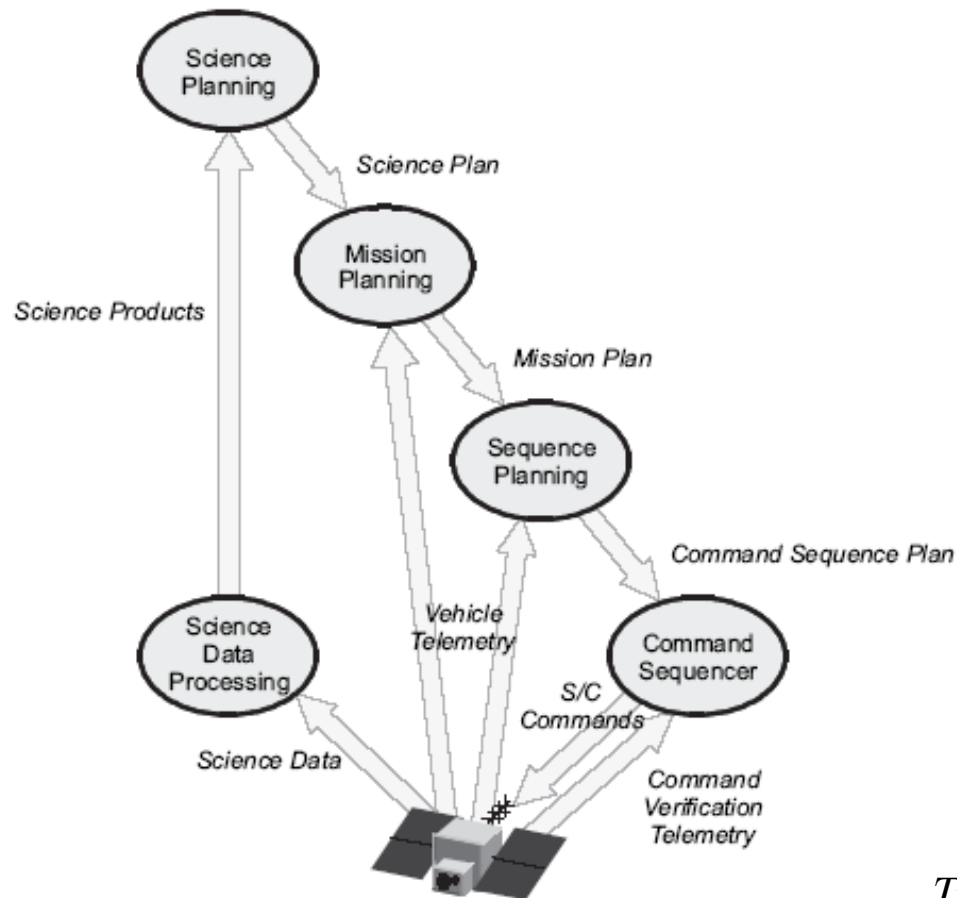
MGT / CTL
PLANE



DATA PLANE

Another View of the Relationship ?

Self Managing System – NASA Cooperative Autonomy



... the Relationship ?

Self Directed & Managing System



AUTONOMOUS

Self-governing
Mission layer
High level policies
Global view



SELFWARE



AUTONOMIC

Self-managing
Reflex layer
low level policies
Situating view



Autonomic Computing – *Biological Inspiration*

Autonomic Computing, launched by IBM in 2001, is emerging as a valuable approach to the design of effective computing systems.

The autonomic concept is inspired by the human body's autonomic nervous system.

It is the part of the nervous system that controls the vegetative functions of the body such as circulation of the blood, intestinal activity and secretion and the production of chemical ‘messengers’, hormones, that circulate in the blood

Fight or Flight



sympathetic (SyNS)

Rest and Digest



parasympathetic (PaNS)

REFLECTION

ROUTINE

REACTION

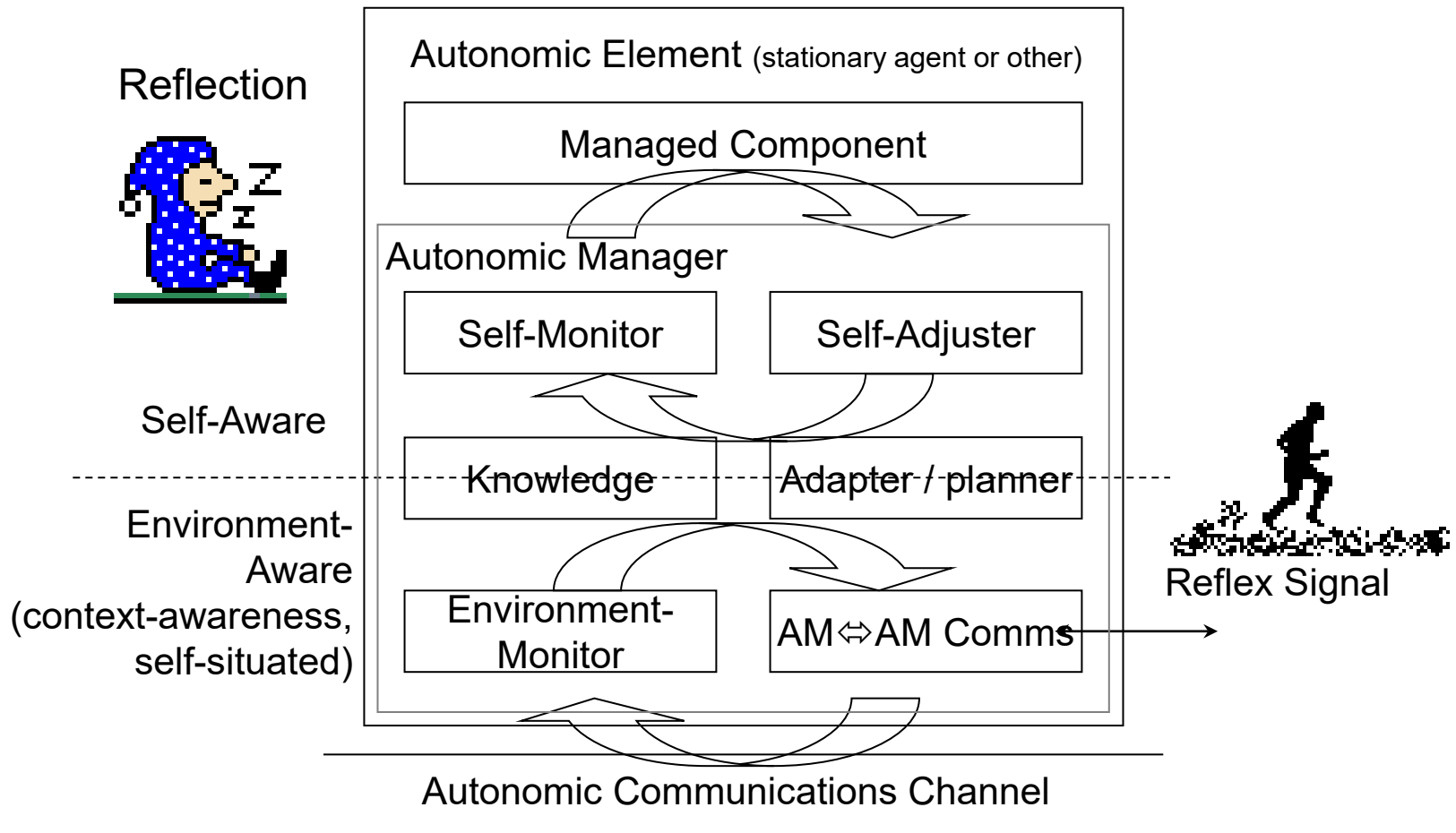
AUTONOMOUS

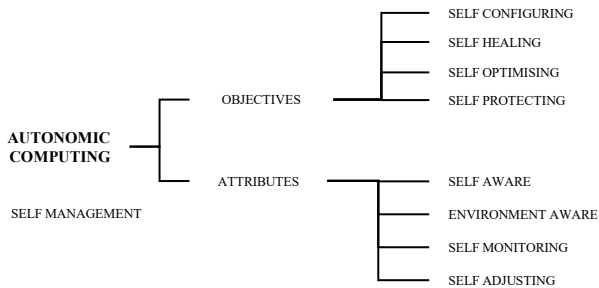
SELFWARE

AUTONOMIC

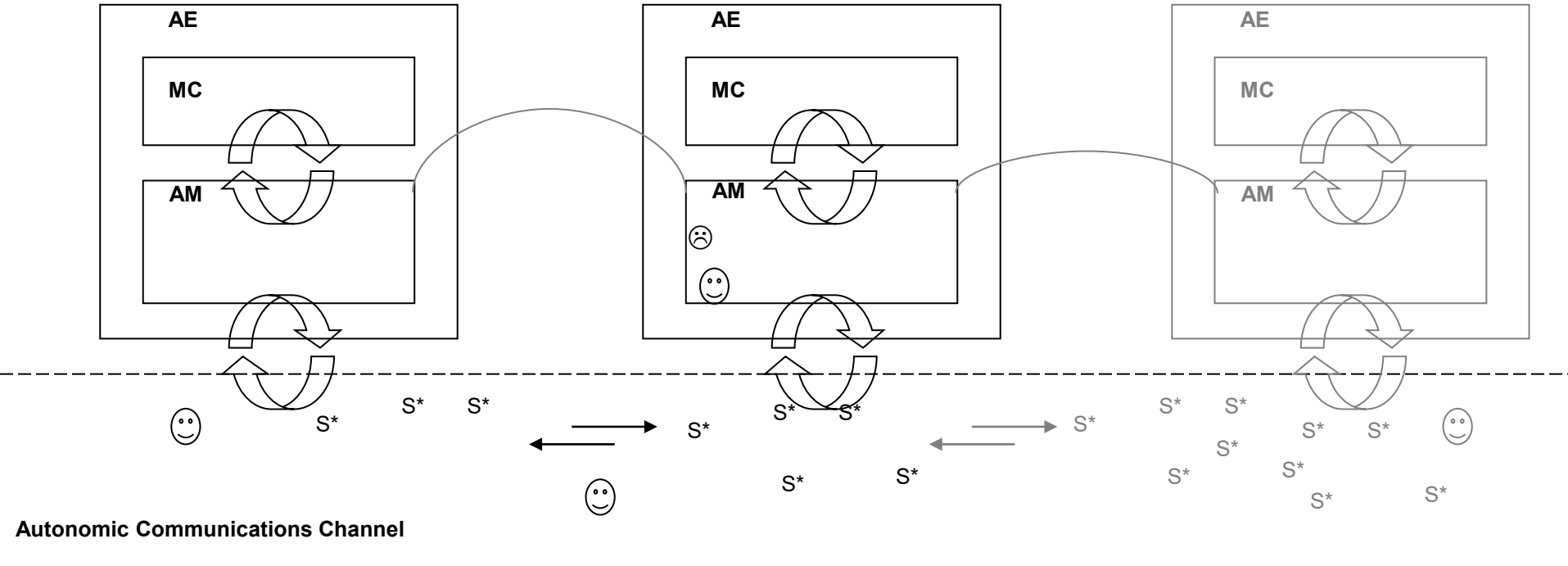
Autonomic Computing – *The Autonomic Element*

Adding Reflection and Reaction to the Routine





Autonomic Computing – *The Autonomic Environment*



Key

S* **Self-* event messages**

AE **Autonomic Element (AM+MC)**

MC **Managed Component**

AM **Autonomic Manager (Stationary agent)**

😊 **Autonomic Agent (Mobile agent)**



Reflex and Healing

A concept inspired by biological systems is the dual approach of *reflexes* and *healing*. Animals have a reflex system, where the nerve pathways enable rapid response to pain.

Reflexes cause a rapid, involuntary motion, such as when a hot surface is touched.

The effect is that the system reconfigures itself, moving away from the danger to keep the component functioning.

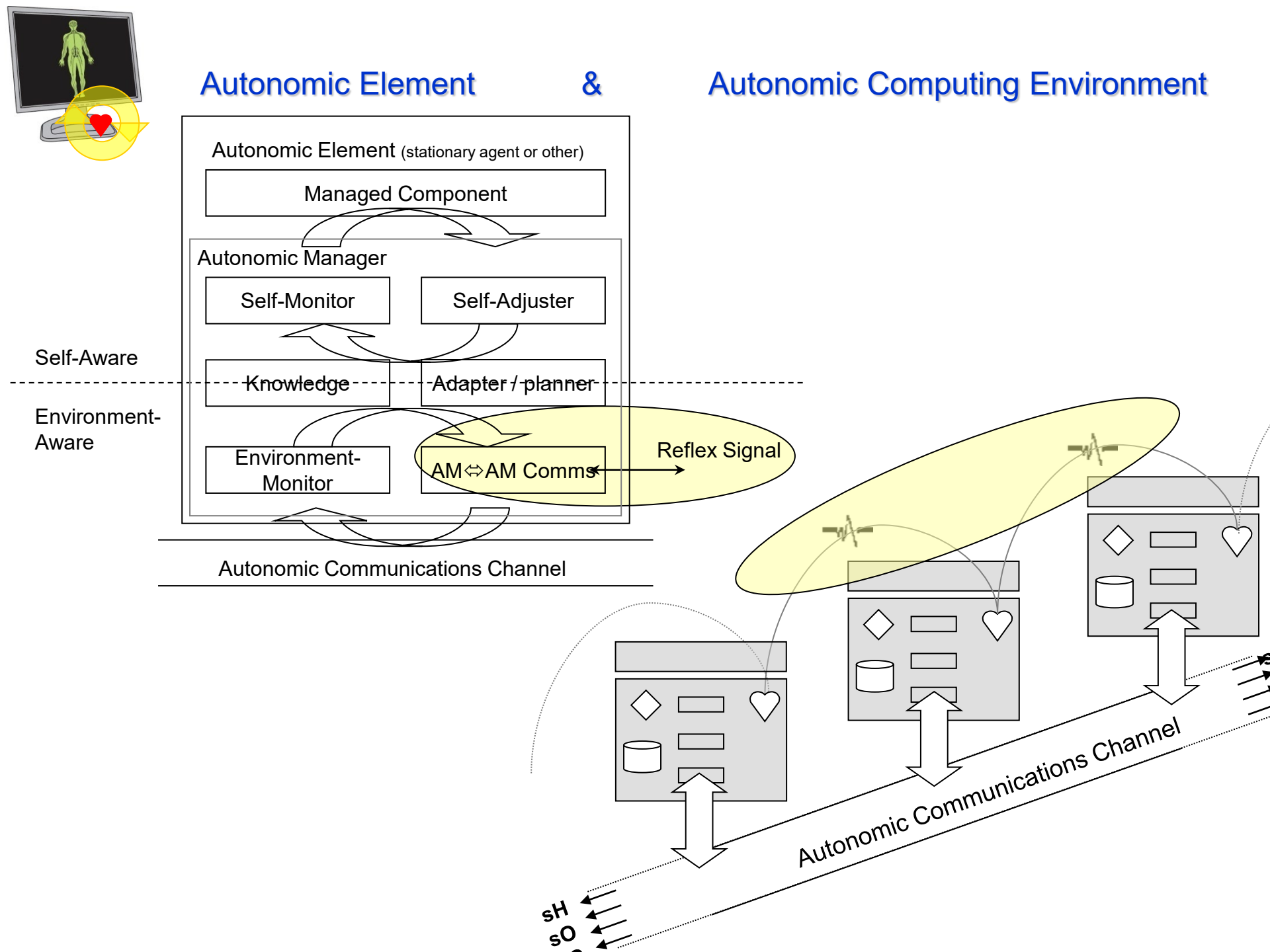
On a much longer timescale, the body will heal itself.

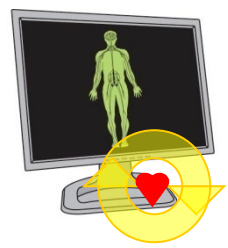
Resources from one part of the system are redirected to rebuild the injured body part, including repair of the reflex response network.

Autonomic Element

&

Autonomic Computing Environment





Embedded Systems : Heart-Beat Monitoring (HBM)

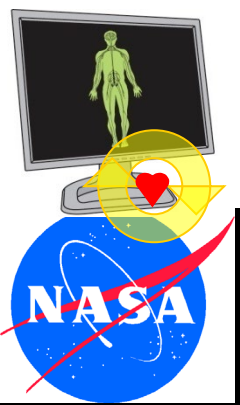
The typical approach for system management is based on events which are generated and sent under fault or problem conditions.

In the embedded system space the opposite is typically the case.

A system management action occurs when something does not occur.

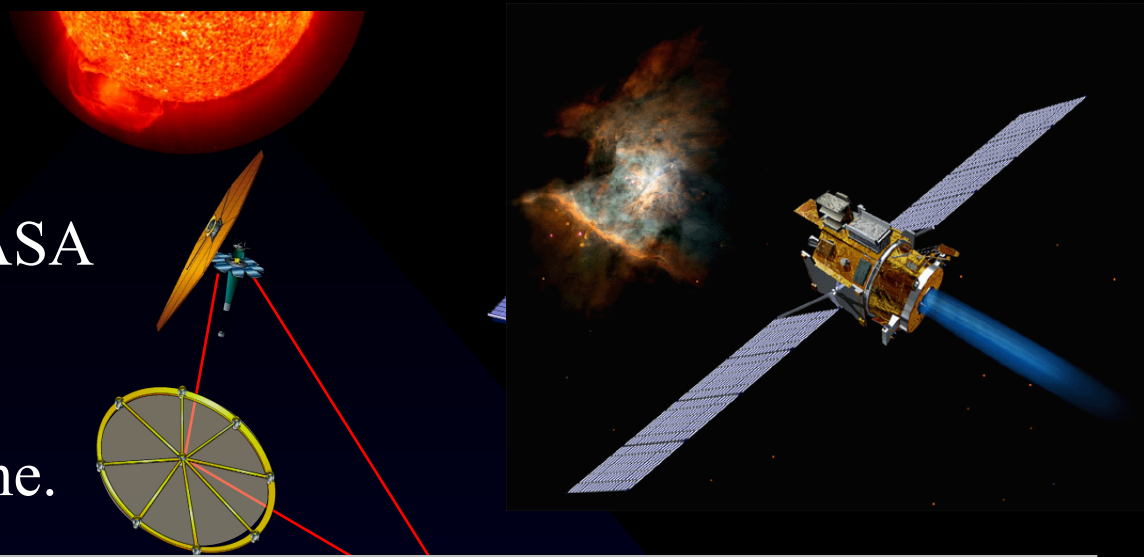
An example is the fault tolerant mechanism of a heartbeat monitor (HBM), through a combination of the hardware (the timer) and software (the heartbeat generator) an '**I am alive**' signal is generated periodically to indicate all is well. The absence of this signal indicates a fault or problem. Some embedded processors have a hardware timer which, if not periodically reset by software, causes a reset/restart. This allows a particularly blunt, though effective, recovery from a software hang.

This approach offers the advantage that through continuous monitoring problem determination becomes a proactive rather than a reactive process.

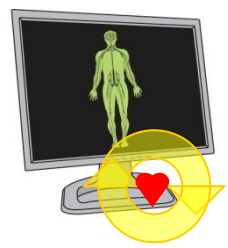


NASA's Beacon Monitoring

A similar concept used by NASA is the Beacon monitor, deep space craft send back a signal containing a urgency level tone.



Nominal	All functions as expected no need to downlink.
Interesting	Interesting – non-urgent event. Establish comms when convenient.
Important	Comms need to take place within timeframe or else state could deteriorate.
Urgent	Emergency. A critical component has failed. Cannot recover autonomously and intervention is necessary immediately.
No Tone	Beacon mode is not operating.



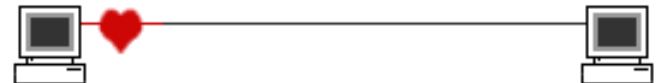
Pulse Monitoring (PBM)

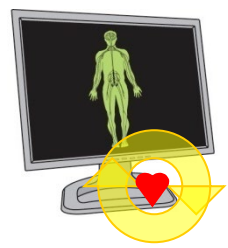
The HBM only informs if a process is alive or dead
(assuming comms working)
– not the processes actual health or state of being.

PBM = HBM + Beacon urgency concept

With a *pulse monitor* instead of just checking the presence of a beat, the rate or tone within the beat would also be measured.

This can give a quick measurement as to the state of health of a process.

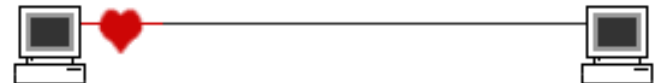


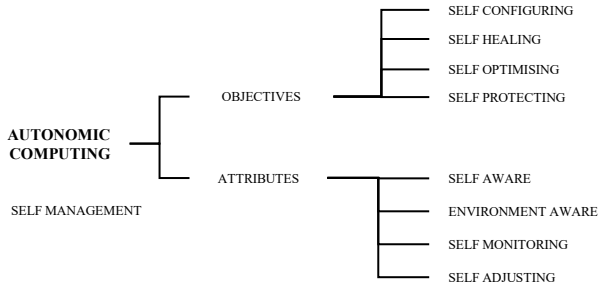


PBM – a mechanism for autonomicity

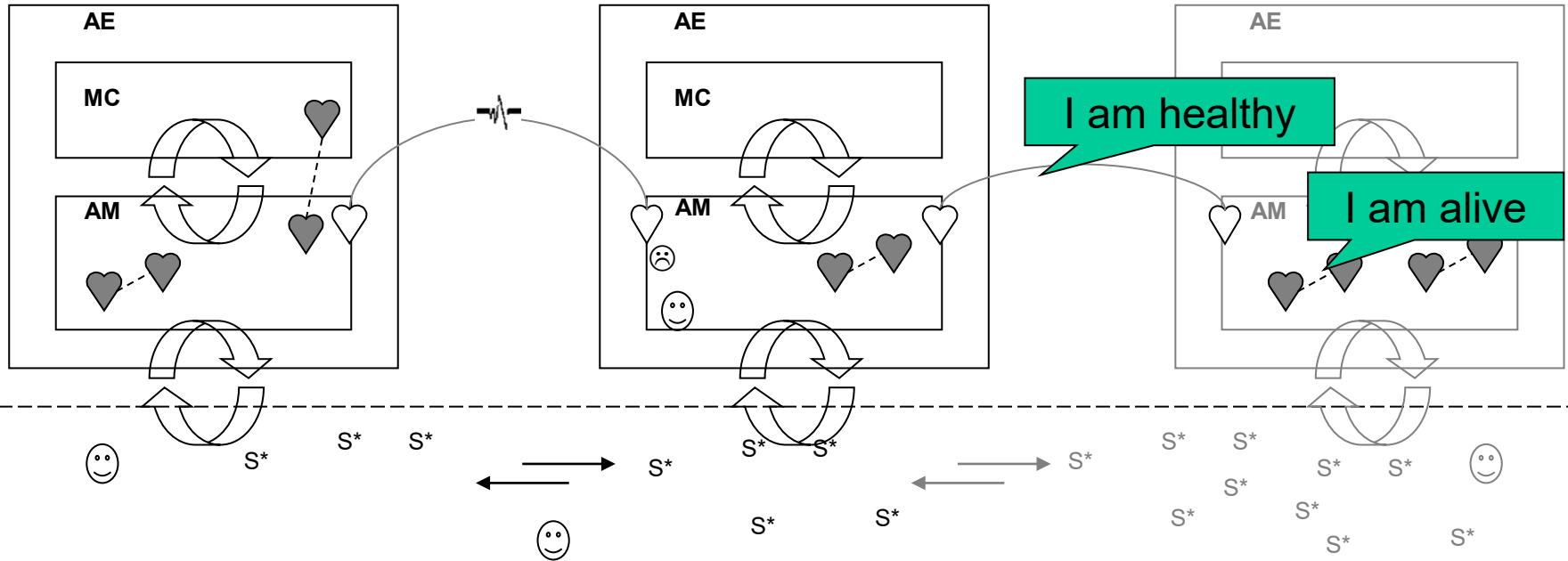
The concept of incorporating a pulse monitor to provide

- a reflex reaction for indicating the ‘health’ based on vital signs
- providing dynamics within autonomic responses and multiple loops of control;
 - some slow and precise (normal event message channel)
 - others fast and possibly imprecise (PBM)





Autonomic Computing – *The Autonomic Environment*



Autonomic Communications Channel

Key

S*	Self-* event messages	AE	Autonomic Element (AM+MC)
♡ - -	Pulse Monitor (PBM)	MC	Managed Component
♥	Heart-Beat Monitor (HBM)	AM	Autonomic Manager (Stationary agent)
😊	Autonomic Agent (Mobile agent)		



Personal Autonomic Computing Reflex Reactions and Self-Healing

Roy Sterritt, Member, IEEE, and David F. Bantz, Member, IEEE

Abstract—The overall goal of this research is to improve the self-awareness and environment-awareness aspect of personal autonomic computing (PAC) to facilitate self-managing capabilities such as self-healing. Personal computing offers unique challenges for self-management due to its multiequipment, multisituation, and multuser nature. The aim is to develop a support architecture for multipatform working, based on autonomic computing concepts and techniques. Of particular interest is collaboration among personal systems to take a shared responsibility for self-awareness and environment awareness. Concepts mirroring human mechanisms, such as reflex reactions and the use of *vital signs* to assess operational health, are used in designing and implementing the PAC architecture. As proof of concept, this was implemented as a self-healing tool utilizing a pulse monitor and a vital signs health monitor within the autonomic manager. This type of functionality opens new opportunities to provide self-configuring, self-optimizing, and self-protecting, as well as self-healing autonomic capabilities to personal computing.

Index Terms—Autonomic computing (AC), environment aware, personal autonomic computing (PAC), personal computing, self-aware, self-healing, self-managing systems.

I. INTRODUCTION

PERSONAL autonomic computing (PAC) is autonomic computing (AC) [1] in a personal computing environment [2]. Personal computing has evolved substantially into a consumer product. Its scope now extends from end user computing in the office to home PCs, wireless laptops, palmtops and is evolving into applications of personal embedded computing, for instance with next-generation mobile/cell phones and iPods. In the near future, these will be leaf nodes in the self-managing ubiquitous and pervasive computing environments incorporating next-generation internet.

Manuscript received October 15, 2004; revised April 25, 2005. This work was supported in part by the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative. This work was presented in part at the Proceedings of the IEEE Workshop on the Engineering of Autonomic Systems (EASe 2004), 11th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2004), Brno, Czech Republic, May 24–27, 2004; DEXA 2004 Workshops, 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04), Zaragoza, Spain, Aug. 30–Sep. 3, 2004; Workshop on the Engineering of Autonomic Systems (EASe 2005), 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005), Greenbelt, MD, Apr. 3–8, 2005. Part of this work was carried out while D. F. Bantz was at IBM TJ Watson Center.

R. Sterritt is with the School of Computing and Mathematics, Faculty of Engineering, University of Ulster, Jordanstown Campus, Newtownabbey, U.K. (e-mail: rsterritt@ulster.ac.uk).

D. F. Bantz is with the Computer Science Department, School of Applied Science, Engineering and Technology, University of Southern Maine, Portland, ME 04104 USA (e-mail: bantz@cs.usm.maine.edu).

Digital Object Identifier 10.1109/TSMCC.2006.871592

Personal computing is an area that can benefit substantially from autonomic principles. Examples of current difficult experiences that can be overcome by such an approach include [2]: 1) trouble connecting to a wired or a wireless network at a conference, hotel, or other work location; 2) switching between home and work; 3) losing a working connection (and shouting across the office to see if anyone else has had the same problem!); 4) going into the IP settings area in Windows and being unsure about the correct values to use; 5) having a PC which stops booting and needs major repair or reinstallation of the operating system; 6) recovering from a hard-disk crash; and 7) migrating efficiently to a new PC. Coping with these situations should be routine and straightforward, but in practice such incidents are typically stressful and often waste a considerable amount of productive time.

PAC shares the goals of personal computing—responsiveness, ease of use, and flexibility—with those of AC—simplicity, availability, and security [3].

Personal computing also creates some problems for the implementation of autonomic principles. In particular [2], personal computing users are often, of necessity, system administrators for the equipment they use. Most are amateurs without formal training, who perform system operations infrequently. This reduces their effectiveness and typically requires them to consult with others to resolve difficulties.

This paper presents relevant background and related work before proceeding to discuss the PAC architecture in Section III. These concepts are explored in a proof-of-concept tool that embodies reflex self-healing. We detail these results in Sections IV–VI and conclude with some observations and suggestions for future work.

II. BACKGROUND AND RELATED WORK

A. Autonomic Computing (AC)

IBM introduced the AC initiative in 2001, with the aim to develop self-managing systems [4]. With the growth of the computer industry, with notable examples being highly efficient networking hardware and powerful CPUs, AC is an evolution to cope with rapidly growing complexity of integrating, managing, and operating computing-based systems. Computing systems should be effective [5], should serve a useful purpose when they are first launched, and continue to be useful as conditions change. The realization of AC will result in a significant reduction in system management complexity [6].

The autonomic concept is inspired by the human body's autonomic nervous system [6]. The autonomic nervous system monitors heartbeat, checks blood sugar levels, and keeps the

Background Reading

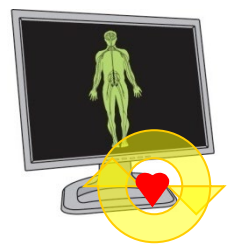
R. Sterritt and D. F. Bantz,
"Personal autonomic computing reflex reactions and self-healing," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 3, pp. 304–314, May 2006, doi: 10.1109/TSMCC.2006.871592.

@IEEE

[Personal autonomic computing reflex reactions and self-healing - IEEE Journals & Magazine](#)

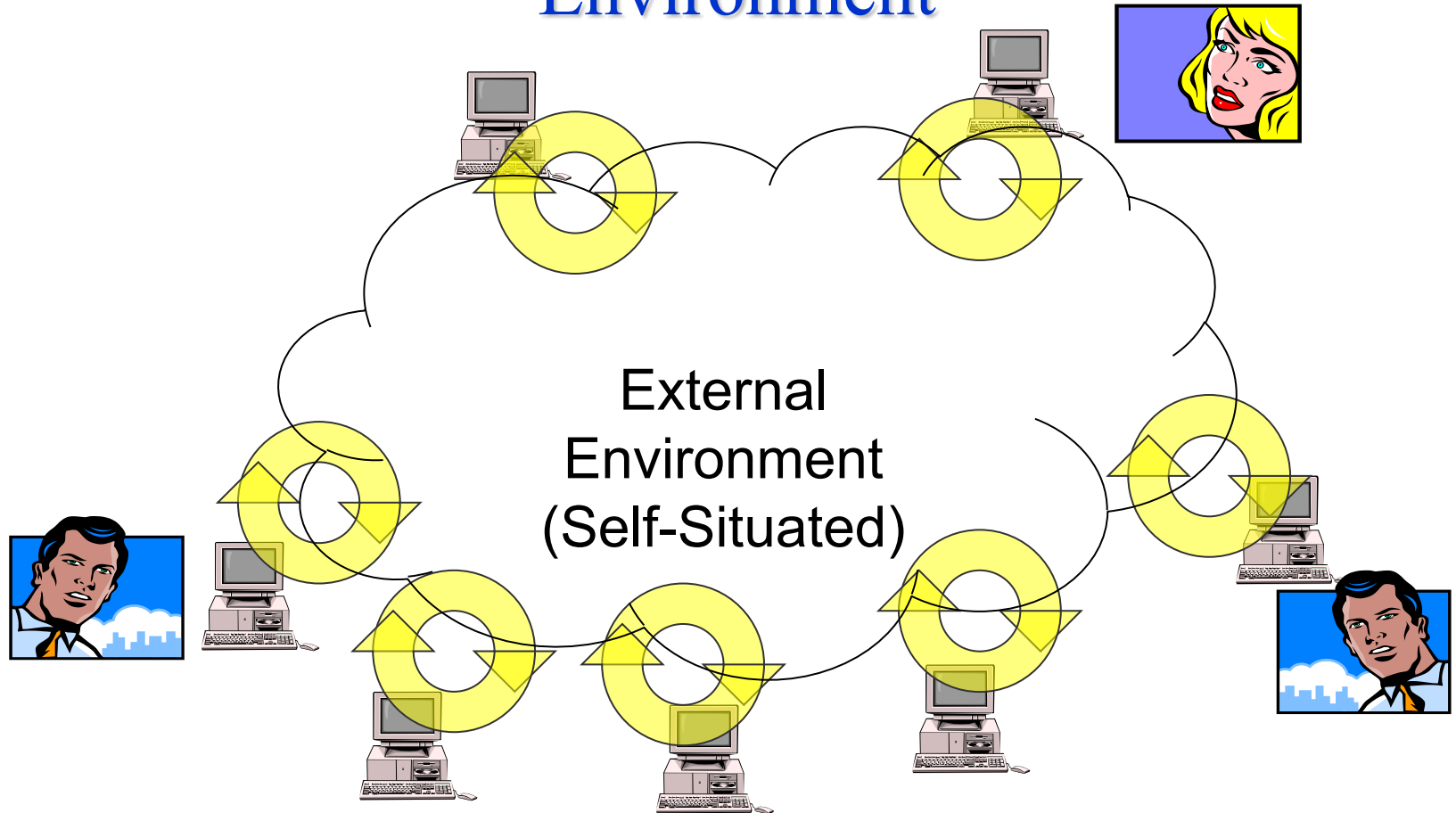
@Pure

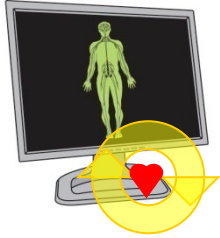
[Personal Autonomic Computing Reflex Reactions and Self-Healing — Ulster University](#)



PAC-MEN

Personal Autonomic Computing Monitoring Environment



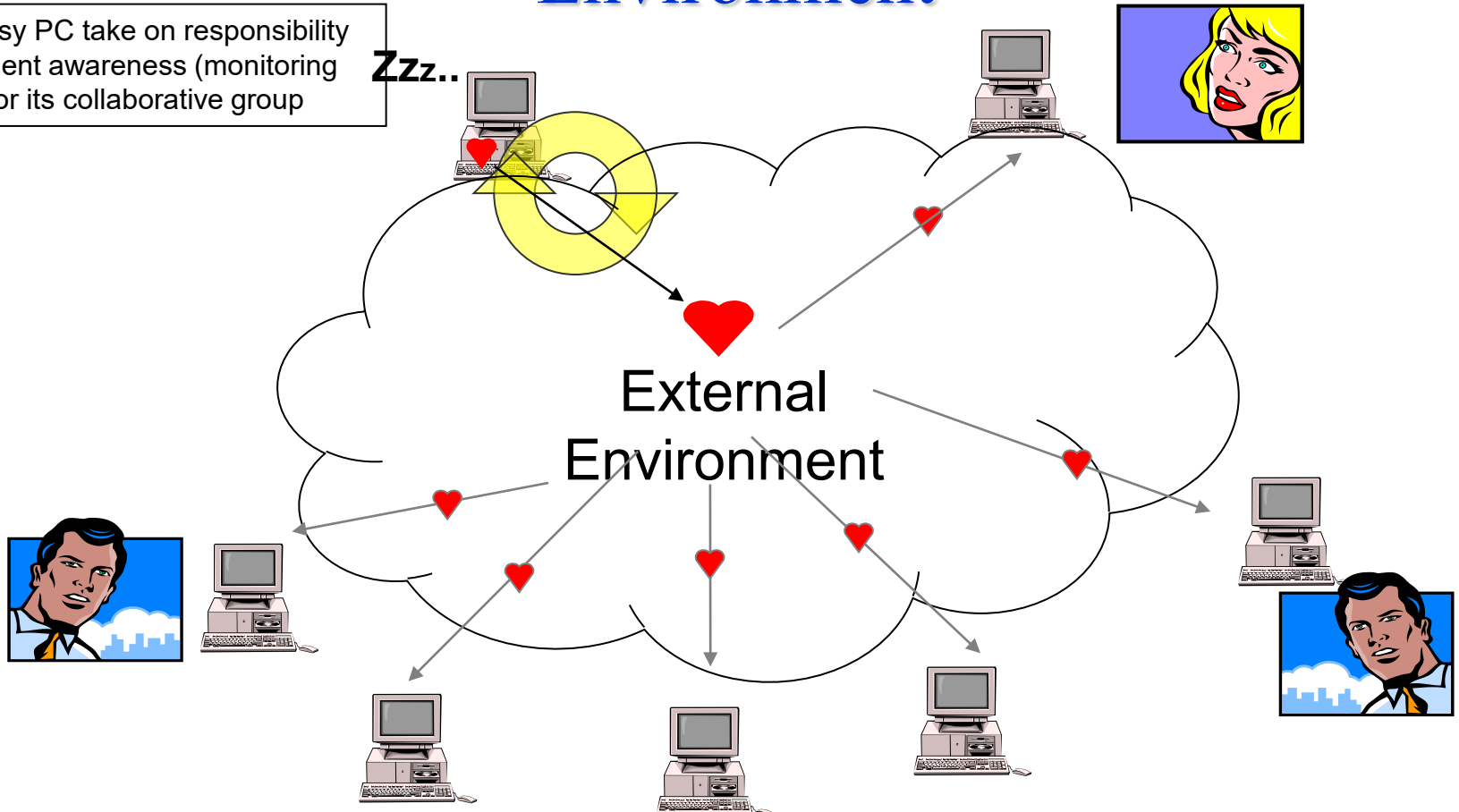


PAC-MEN

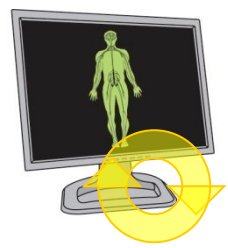
Personal Autonomic Computing Monitoring Environment

e.g. least busy PC take on responsibility for environment awareness (monitoring vital signs) for its collaborative group

Zzz...



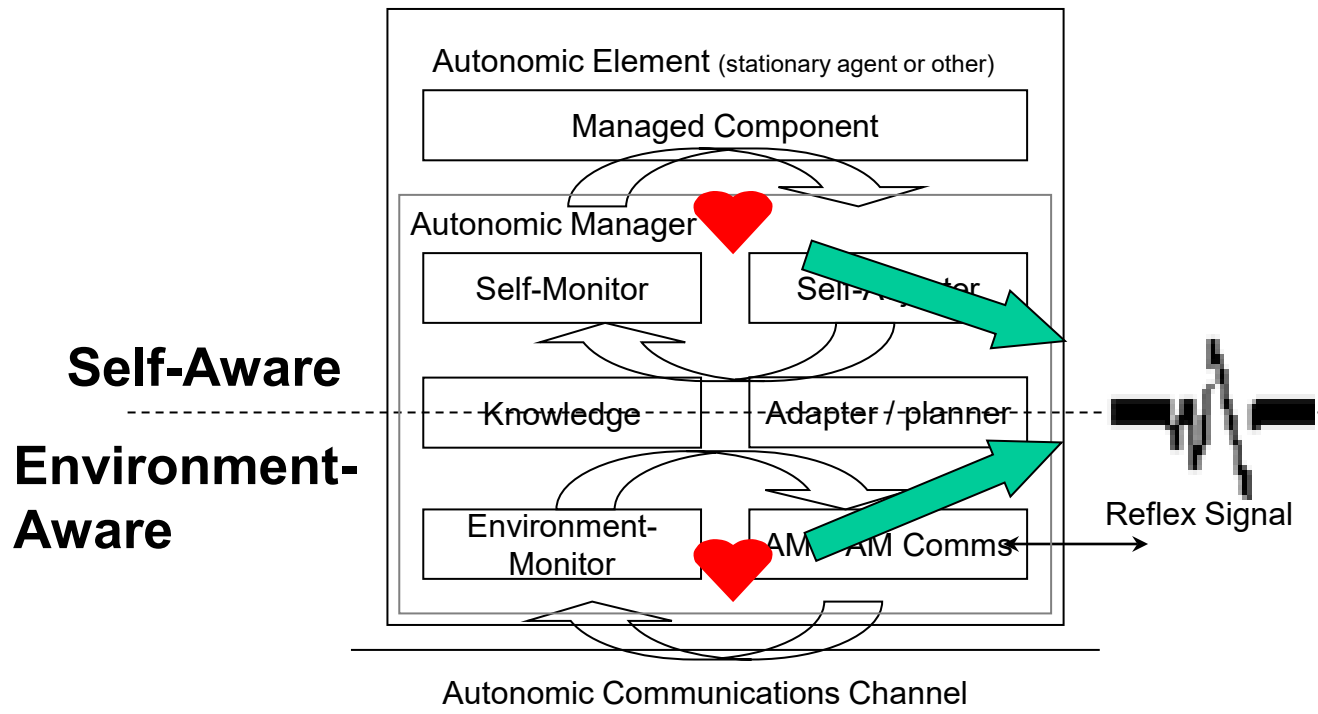
e.g. through changes in pulse (*reflex reaction*) other PCs become alerted to change in take on responsibility for environment awareness (monitoring) for its collaborative group



Lub-Dub

Two signals encoded in the heart-beat

The PBM utilized for environment awareness should be considered in addition to the self-awareness pulse mechanism i.e. utilized to create a shared dynamic group environment awareness as well as individual self-protection



Story so far ...

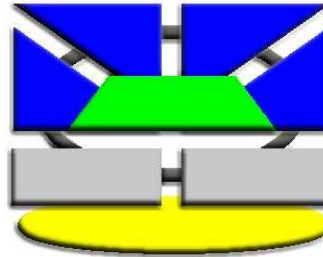
Autonomic Computing Tools & Techniques



+



+



MAPE/MAPE-K

AUTONOMOUS

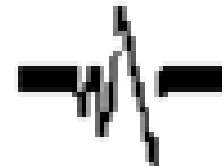
SELFWARE

AUTONOMIC

+



+





NASA Space Exploration Missions

- Future space missions will require cooperation between multiple satellites/rovers/craft for e.g.

ANTS (Autonomous Nano-Technology Swarm)

- Developers are proposing intelligent, autonomous swarms to do new science
- Swarm-based systems are highly parallel and nondeterministic
- Testing these systems using current techniques will be difficult to impossible
- This raises issues for self-protection of system (mission) goals,

Curtis et al 2000; Clark et al 2001, ...

<https://attic.gsfc.nasa.gov/ants/>

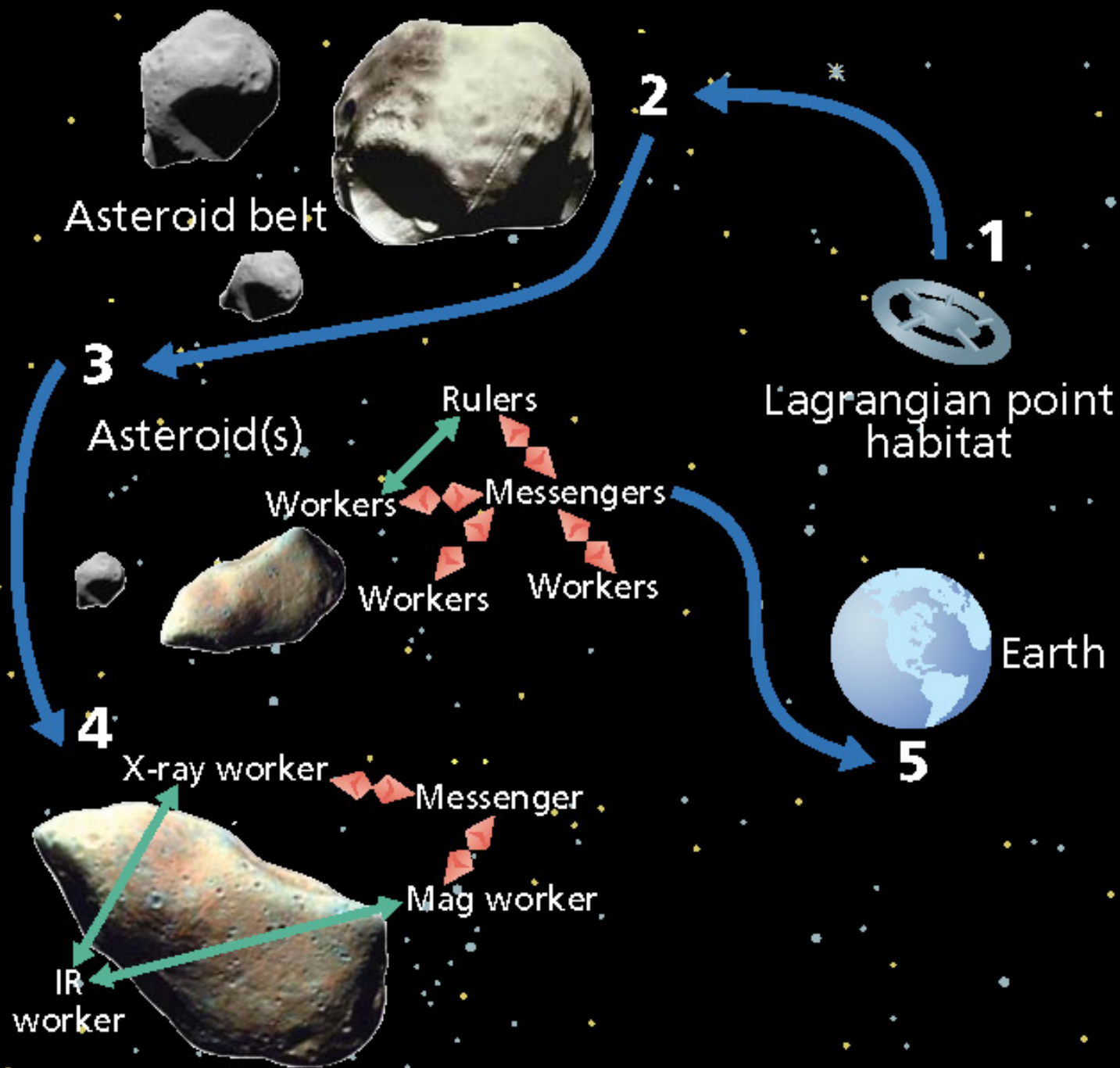


ANTS Space Exploration Missions

- The ANTS architecture is itself inspired by biological low level social insect colonies with their success in the division of labor.
- Within their specialties, individual specialists generally outperform generalists, and with sufficiently efficient social interaction and coordination, the group of specialists generally outperforms the group of generalists.
- Thus systems designed as ANTS are built from potentially very large numbers of highly autonomous, yet socially interactive, elements.
- The architecture is self-similar in that elements and sub-elements of the system may also be recursively structured as ANTS

ANTS Space Exploration Missions (*cont.*)

- Targets for ANTS-like missions include surveys of extreme environments on the
 - Earth,
 - Moon,
 - Mars, as well as
 - asteroid,
 - comet,
 - or dust populations.
- The ANTS paradigm makes the achievement of such goals possible through the use of many small, autonomous, reconfigurable, redundant element craft acting as independent or collective agents



1. Assembly and Release

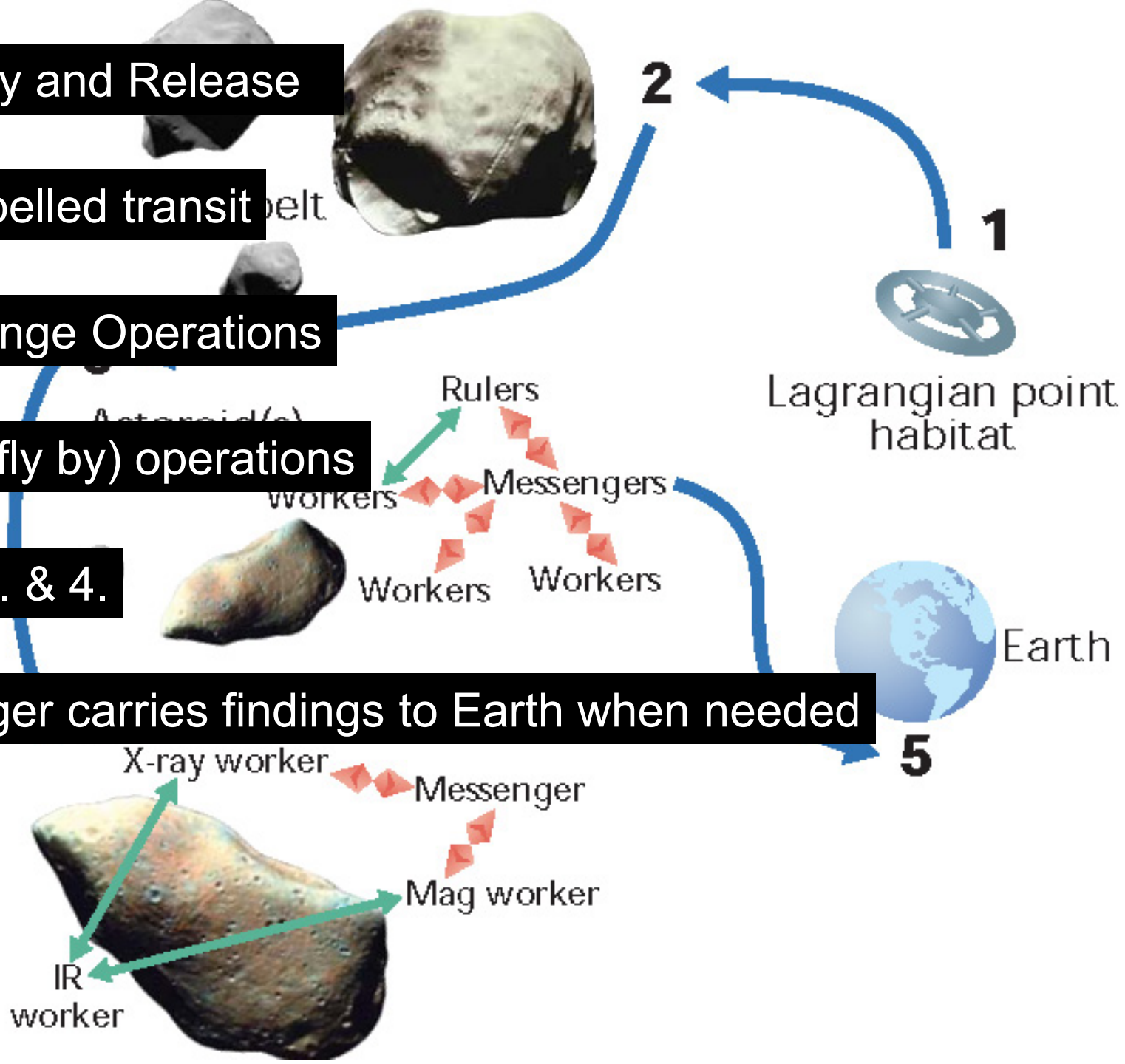
2. Self propelled transit belt

3. Long Range Operations

4. Swarm (fly by) operations

Repeat 3. & 4.

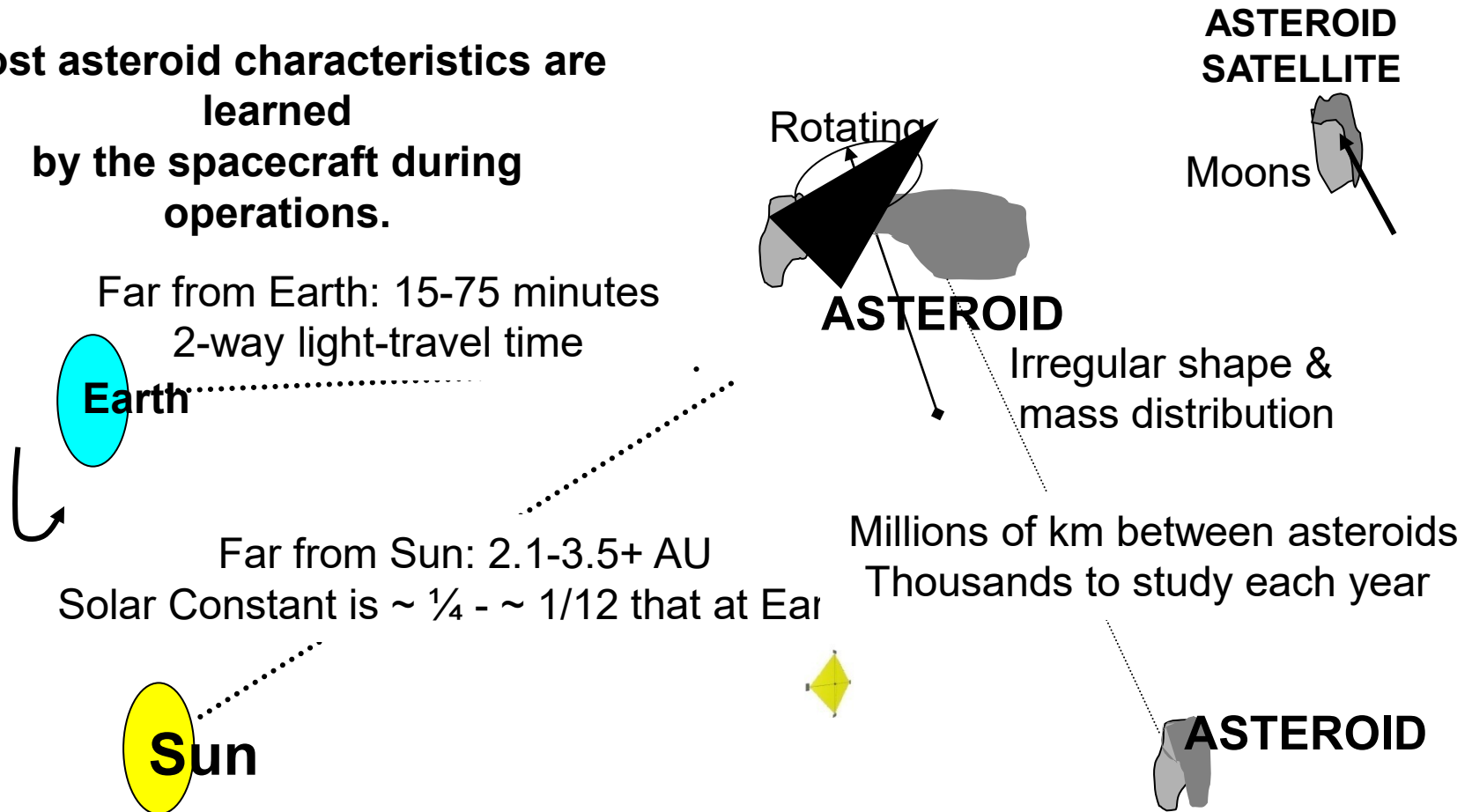
5. Messenger carries findings to Earth when needed



ANTS MISSION

PAM Challenges for low thrust Solar Sail spacecraft

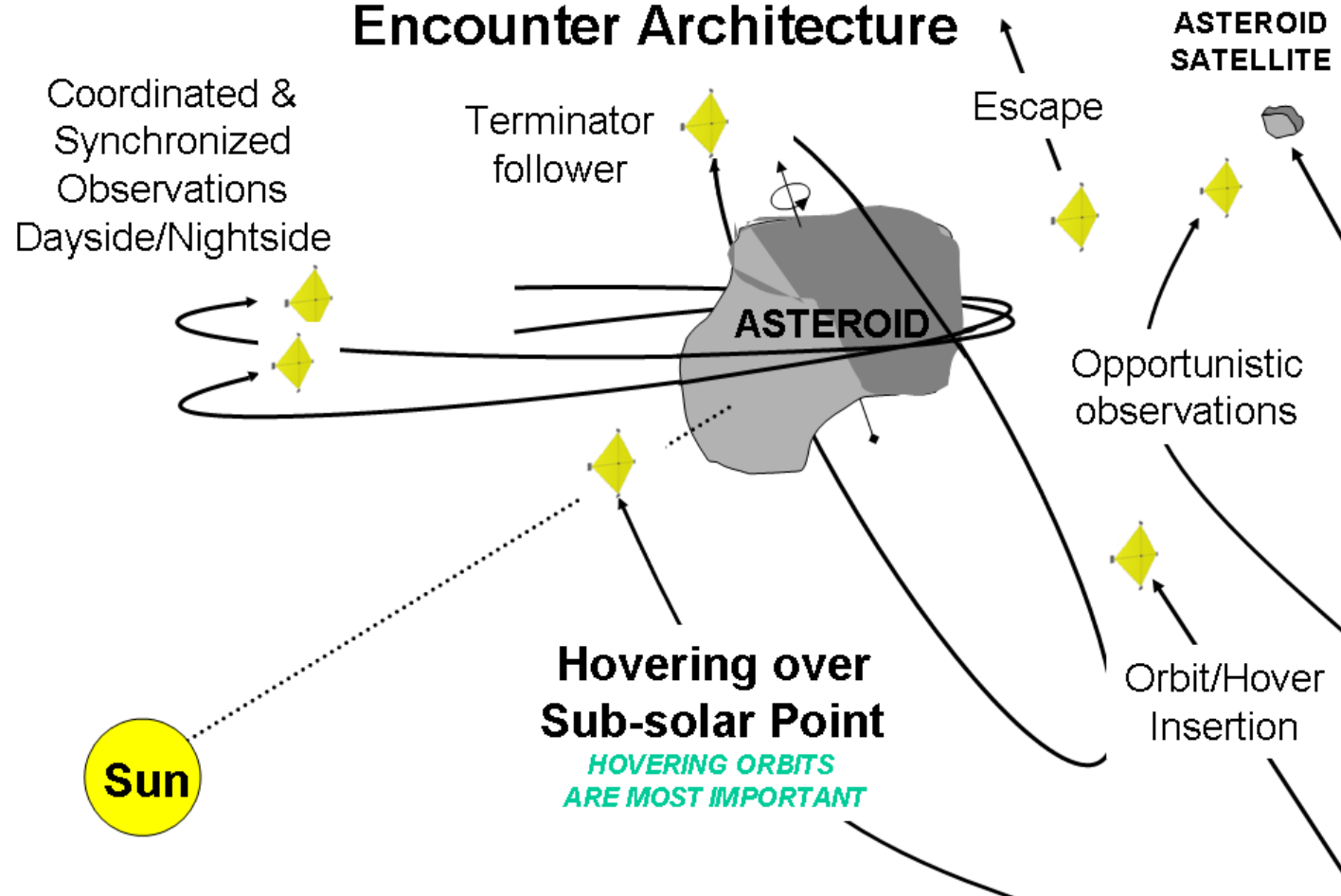
Most asteroid characteristics are learned by the spacecraft during operations.



Adequate agility and acceleration imply large sails and pico-spacecraft.
Mechanical stability, etc, of such spacecraft is an open issue.

ANTS MISSION

Prospecting Asteroid Mission Encounter Architecture

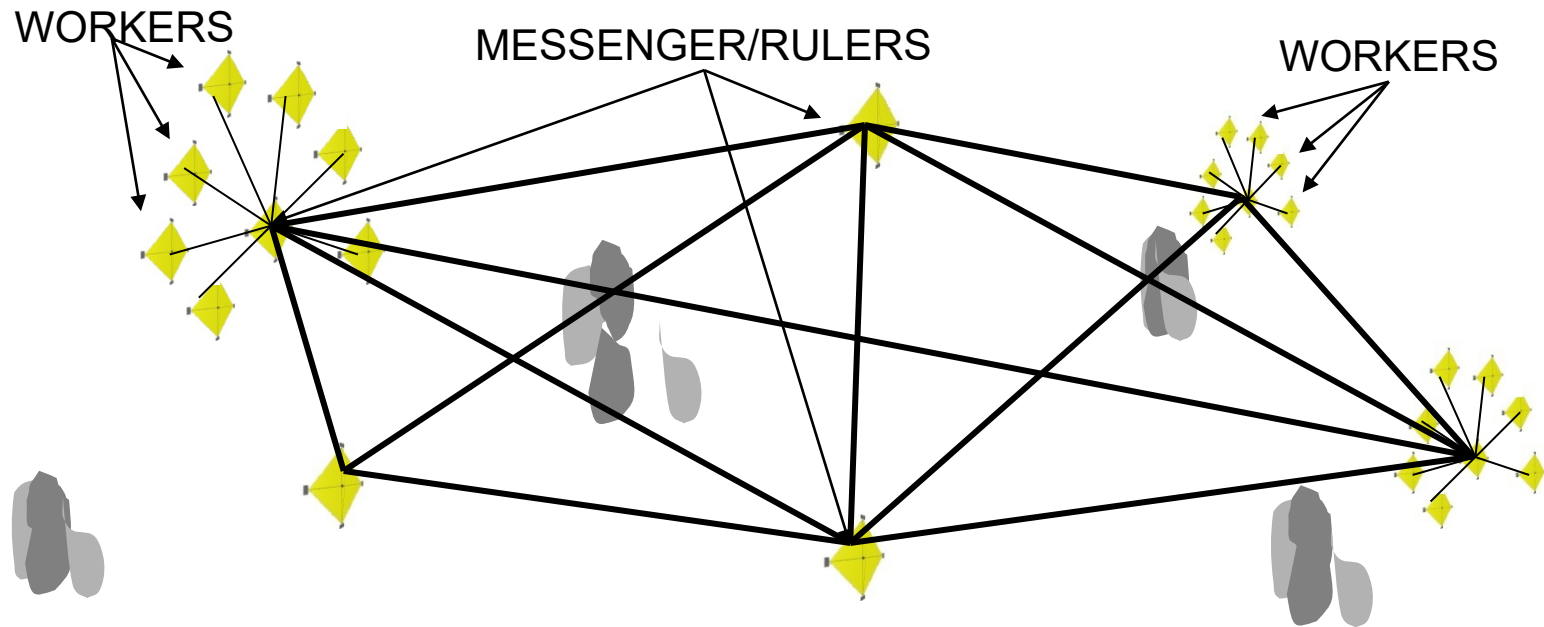




ANTS - Self-CHOP

Self-
Configuring

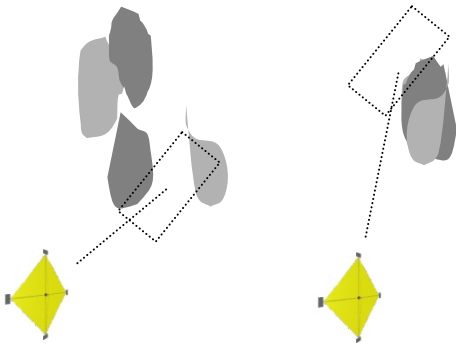
Swarm/Constellation Communications, Control, & Cohesion



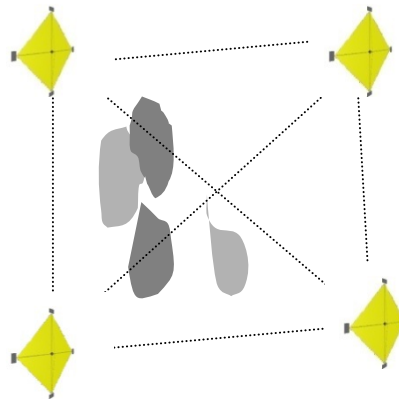


Autonomous, Optimized Science Operations

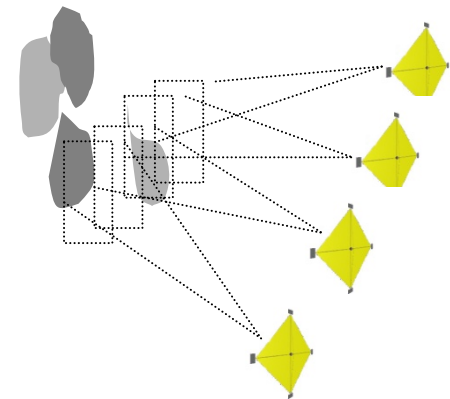
Single S/C, Local Scope
e.g. X-Ray Spectrometry
& Long-range imaging



Multi-S/C, Global Scope
e.g. Radio Science Gravimetry
“Ad hoc GPS for Asteroids”



Multi-S/C, Local Scope
e.g. Imaging, Sounding,
Mapping

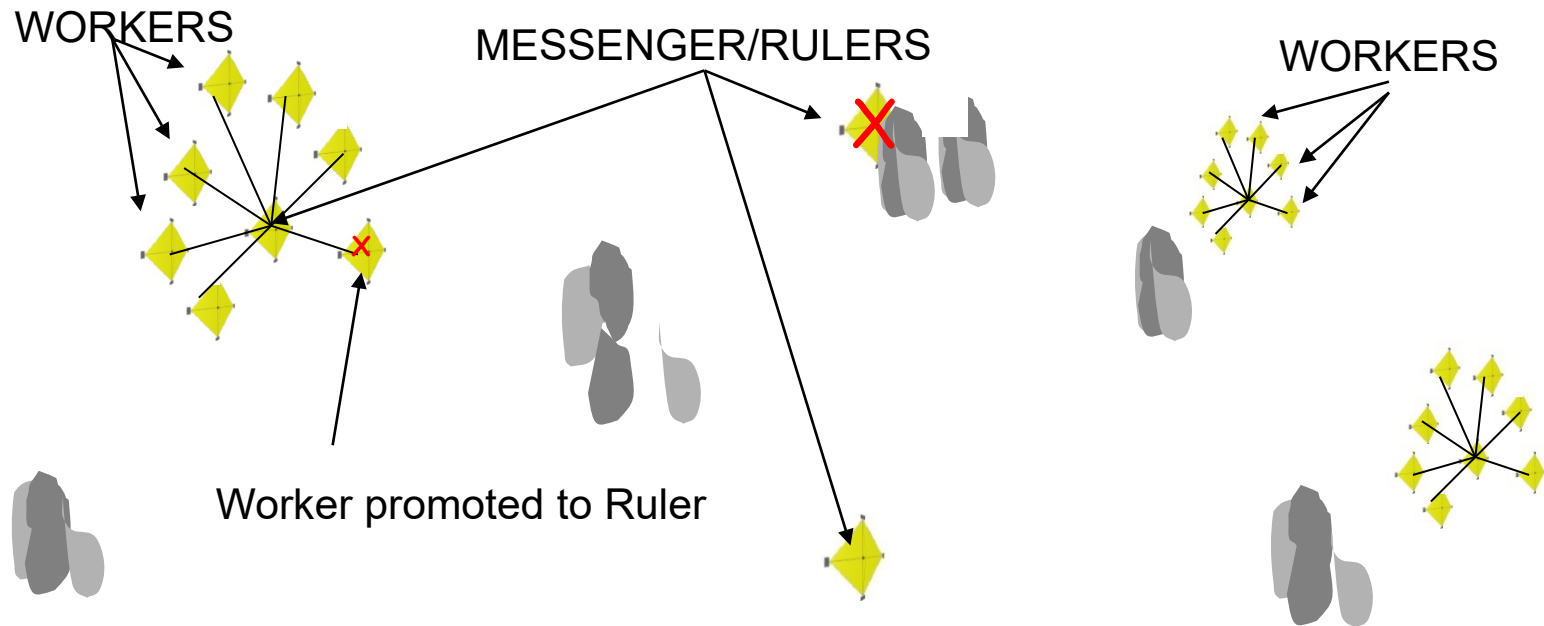




ANTS Self-CHOP

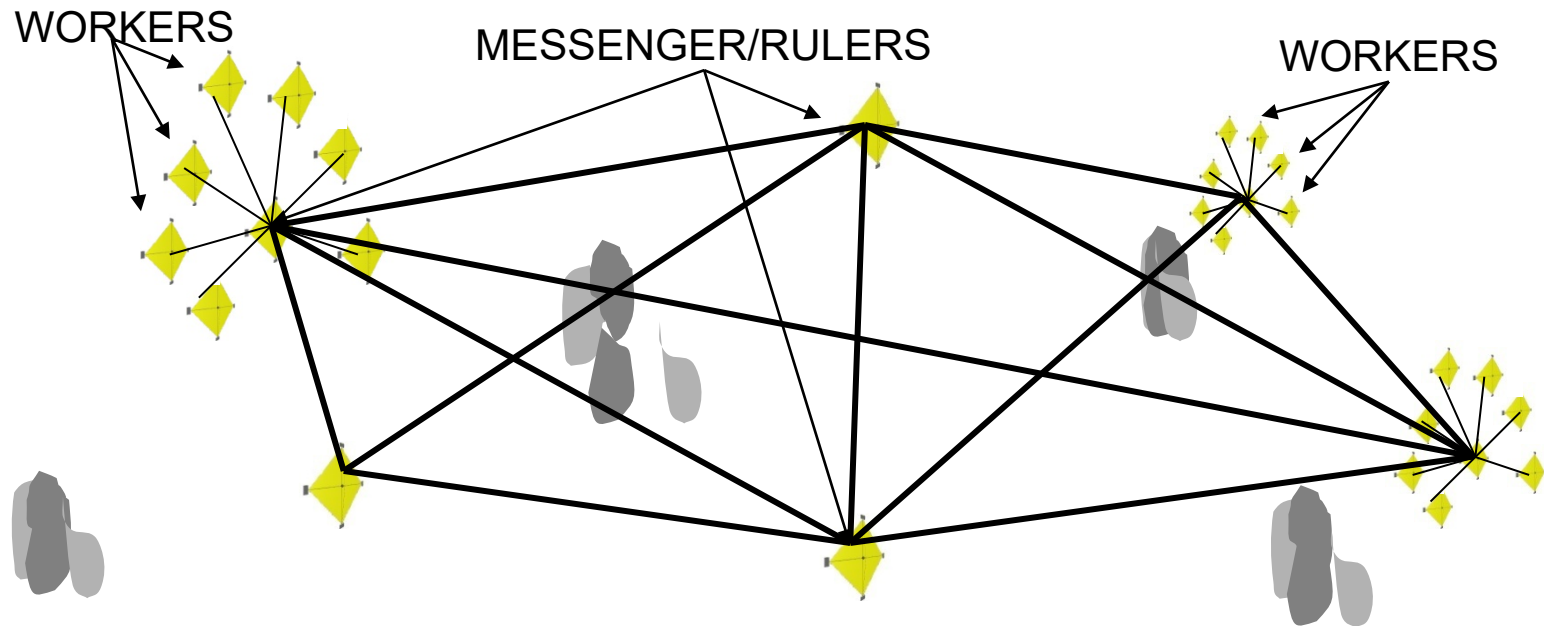
Self-Healing

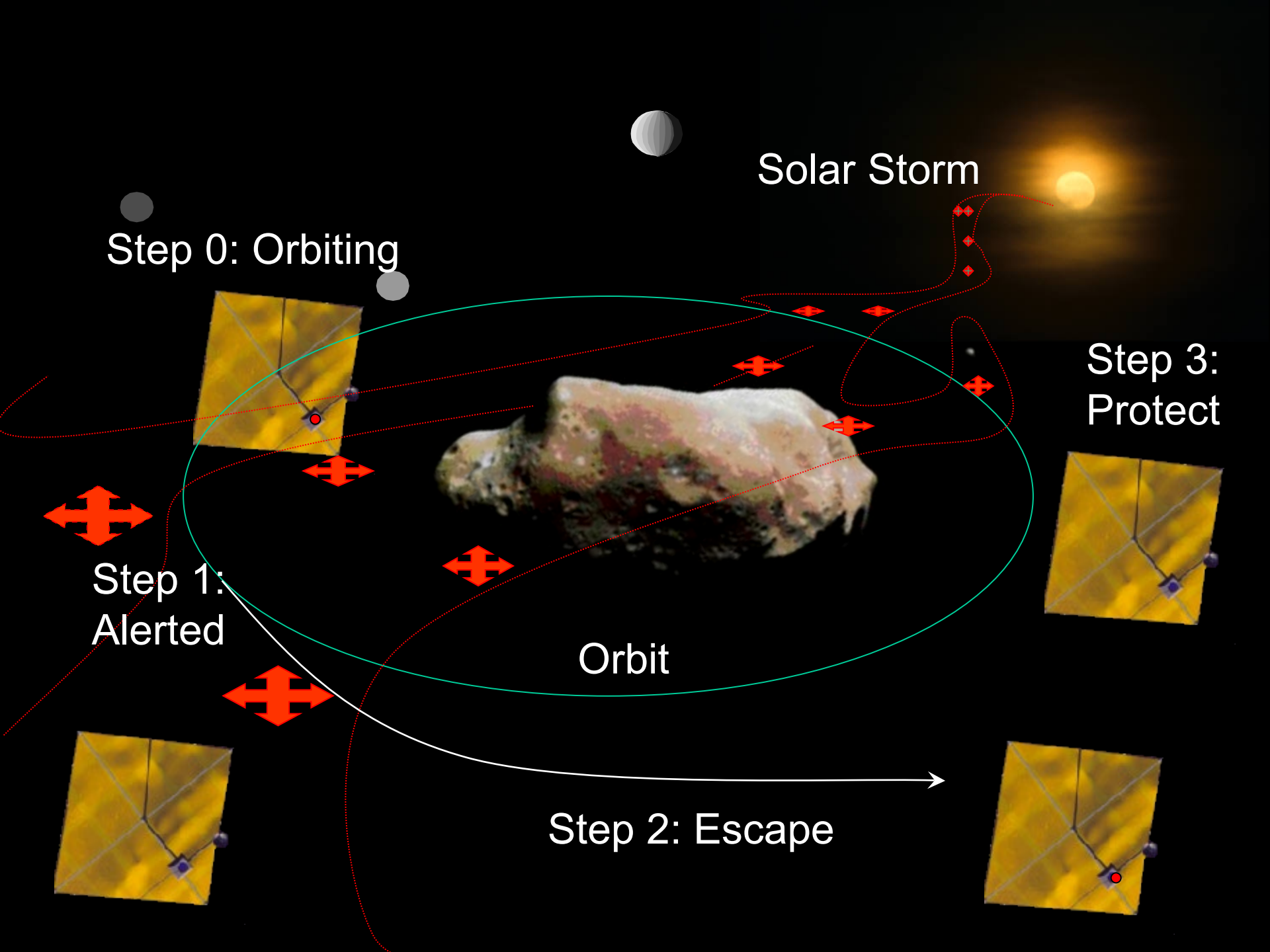
**Worker instrument damaged?
Available to replace a lost Ruler or Messenger**





Solar Storm –Replan trajorectories or even sleep mode







Autonomic Management - Agent Software

α Configuration Agent Primary

β Configuration Agent Secondary

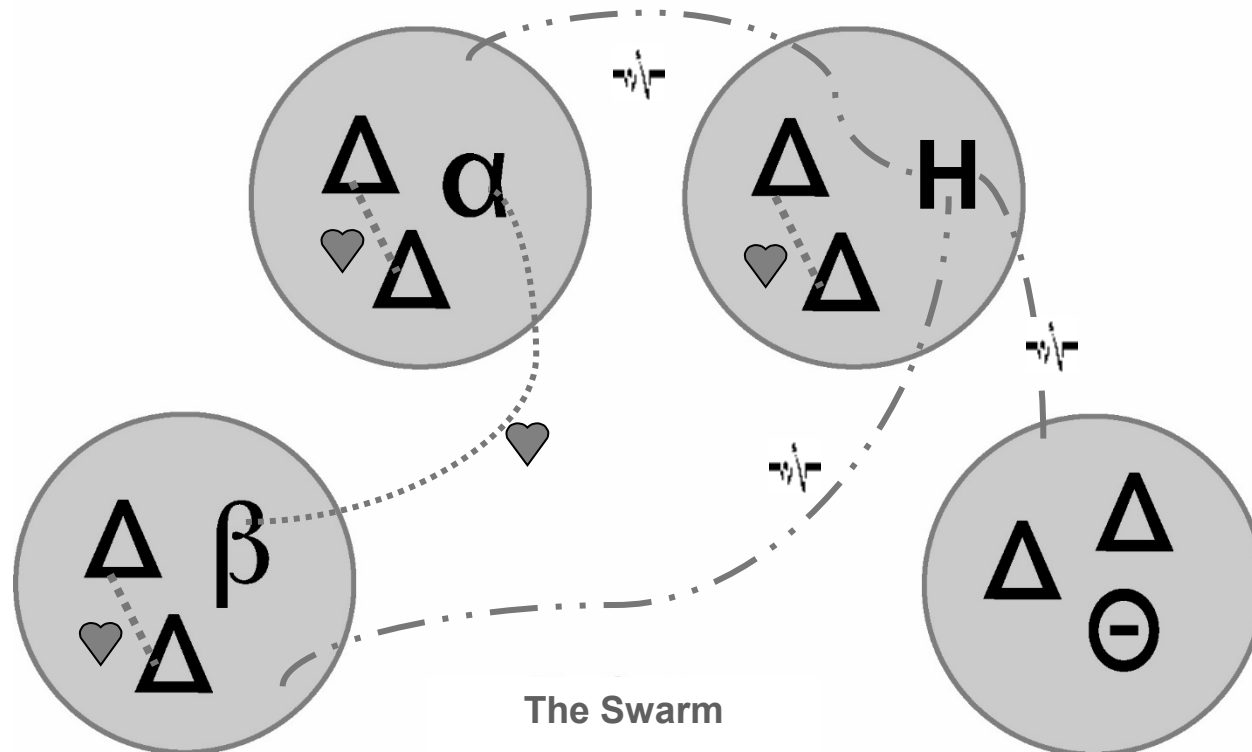
\ominus Optimization Agent

Δ General Agent

H Health Agent

 Heartbeats

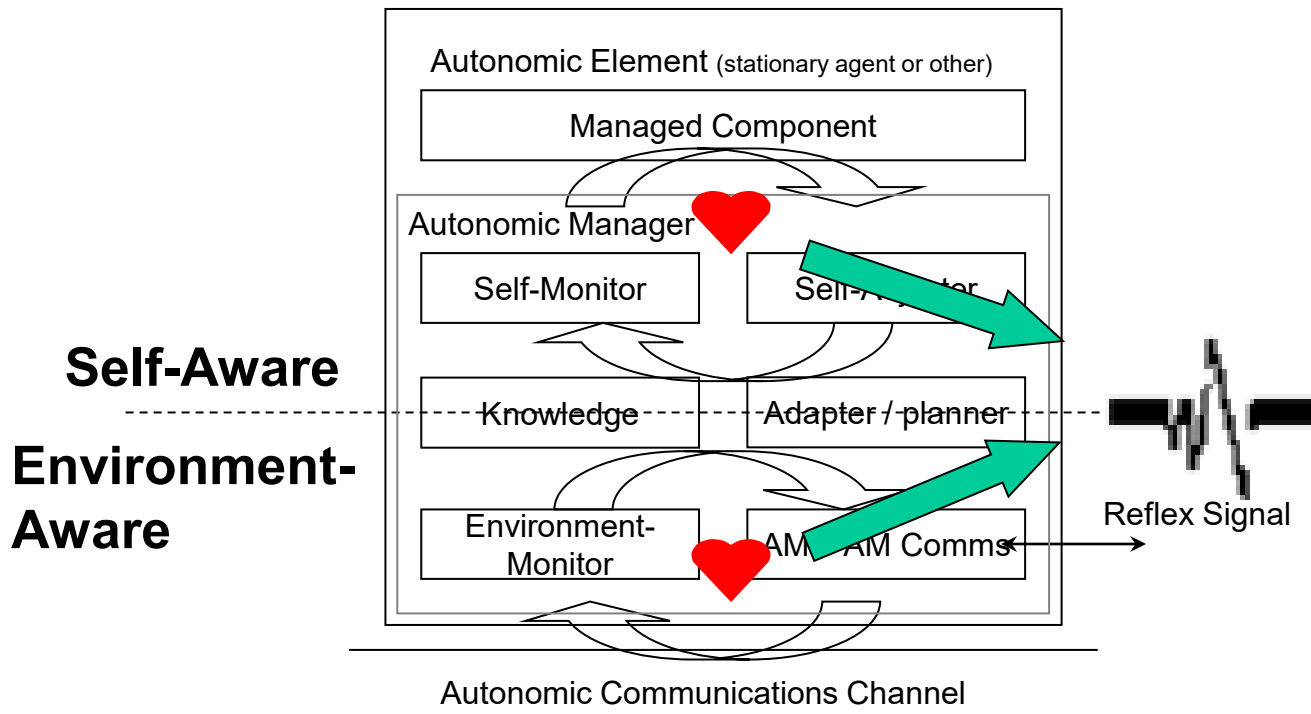
 Pulses



Lub-Dub

Two signals encoded in the heart-beat

The PBM utilized for environment awareness should be considered in addition to the self-awareness pulse mechanism i.e. utilized to create a shared dynamic group environment awareness as well as individual self-protection



OBJECTIVES

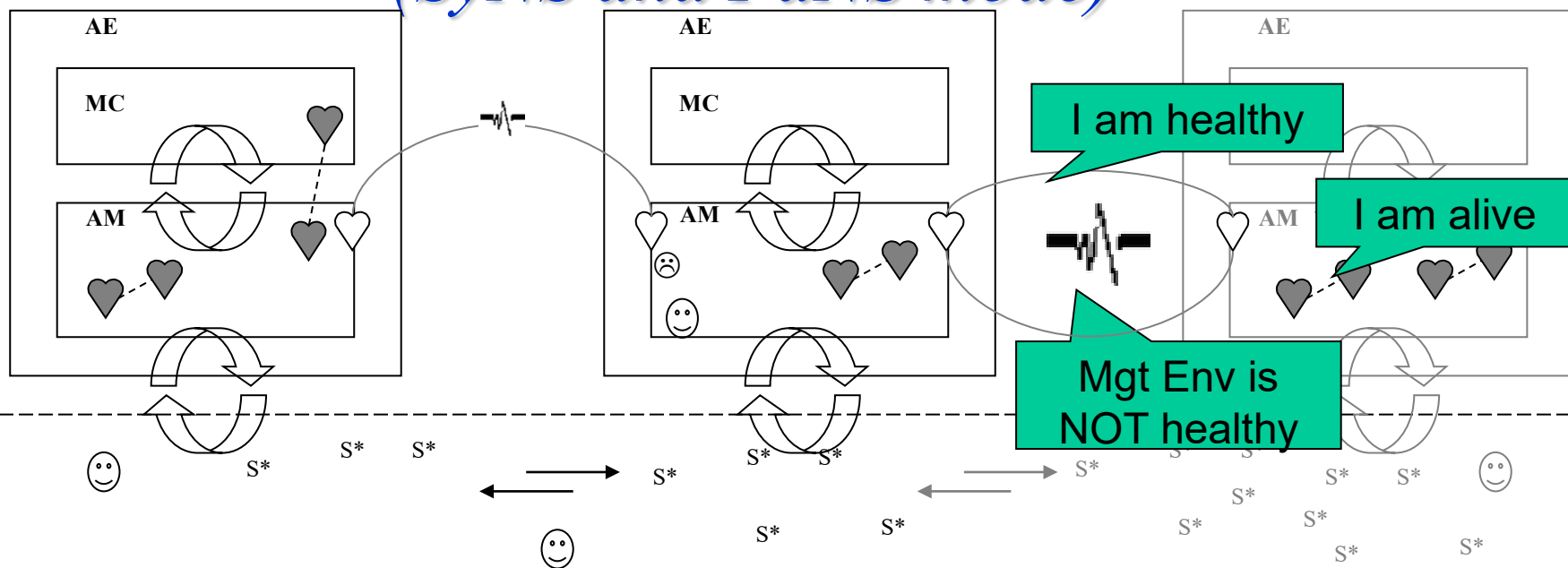
ATTRIBUTES

SELF CONFIGURING
 SELF HEALING
 SELF OPTIMISING
 SELF PROTECTING
 SELF AWARE
 ENVIRONMENT AWARE
 SELF MONITORING
 SELF ADJUSTING

AC – The Autonomic Environment

inc Adaptive pulse monitoring

(SyNS and PaNS mode)



Autonomic Communications Channel

Key

S* Self-* event messages

Pulse Monitor (PBM)

Heart-Beat Monitor (HBM)

Autonomic Agent (Mobile agent)

AE Autonomic Element (AM+MC)

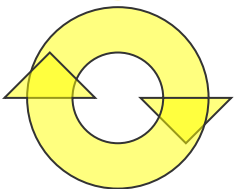
MC Managed Component

AM Autonomic Manager (Stationary agent)

Key:



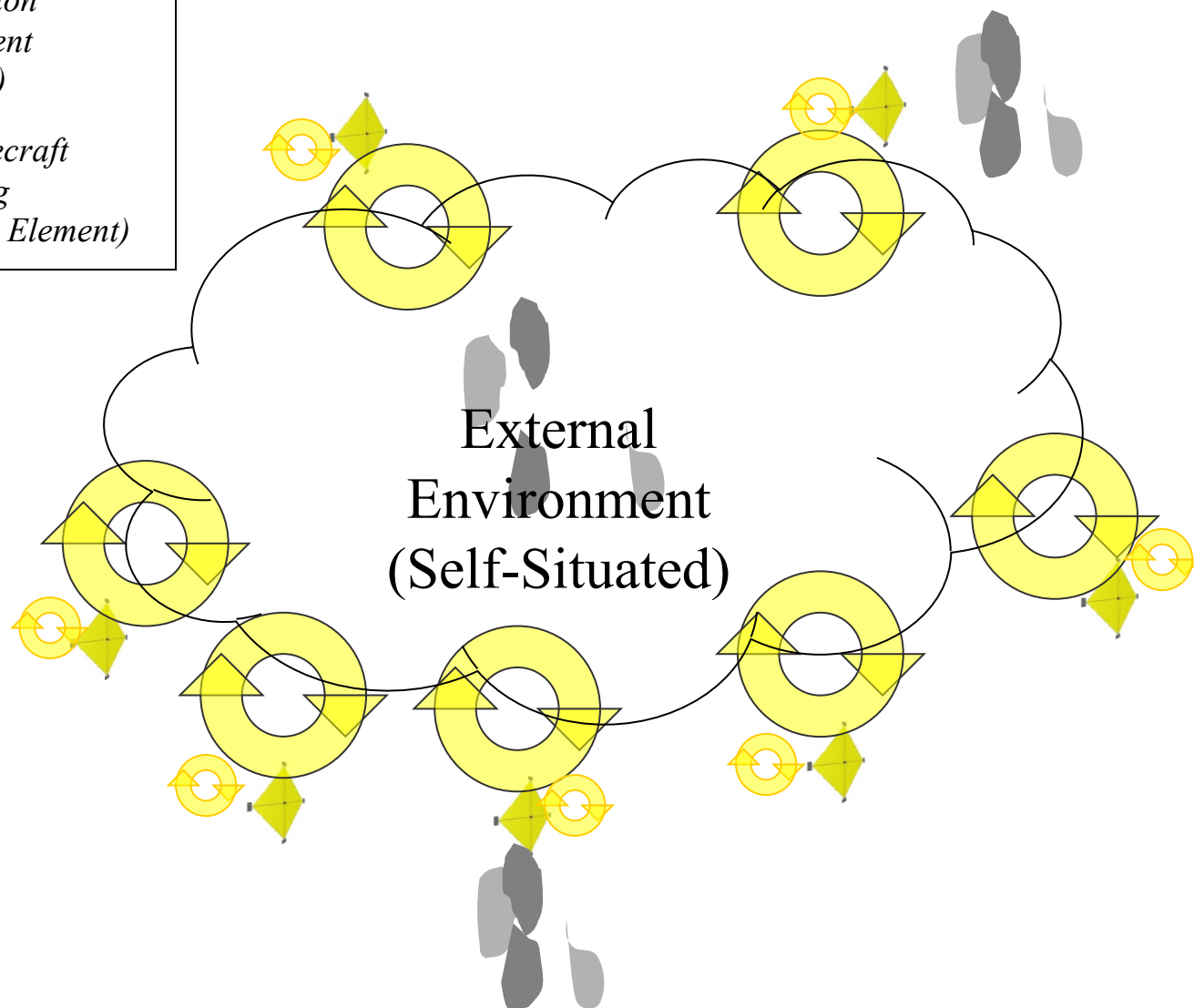
Self-Awareness



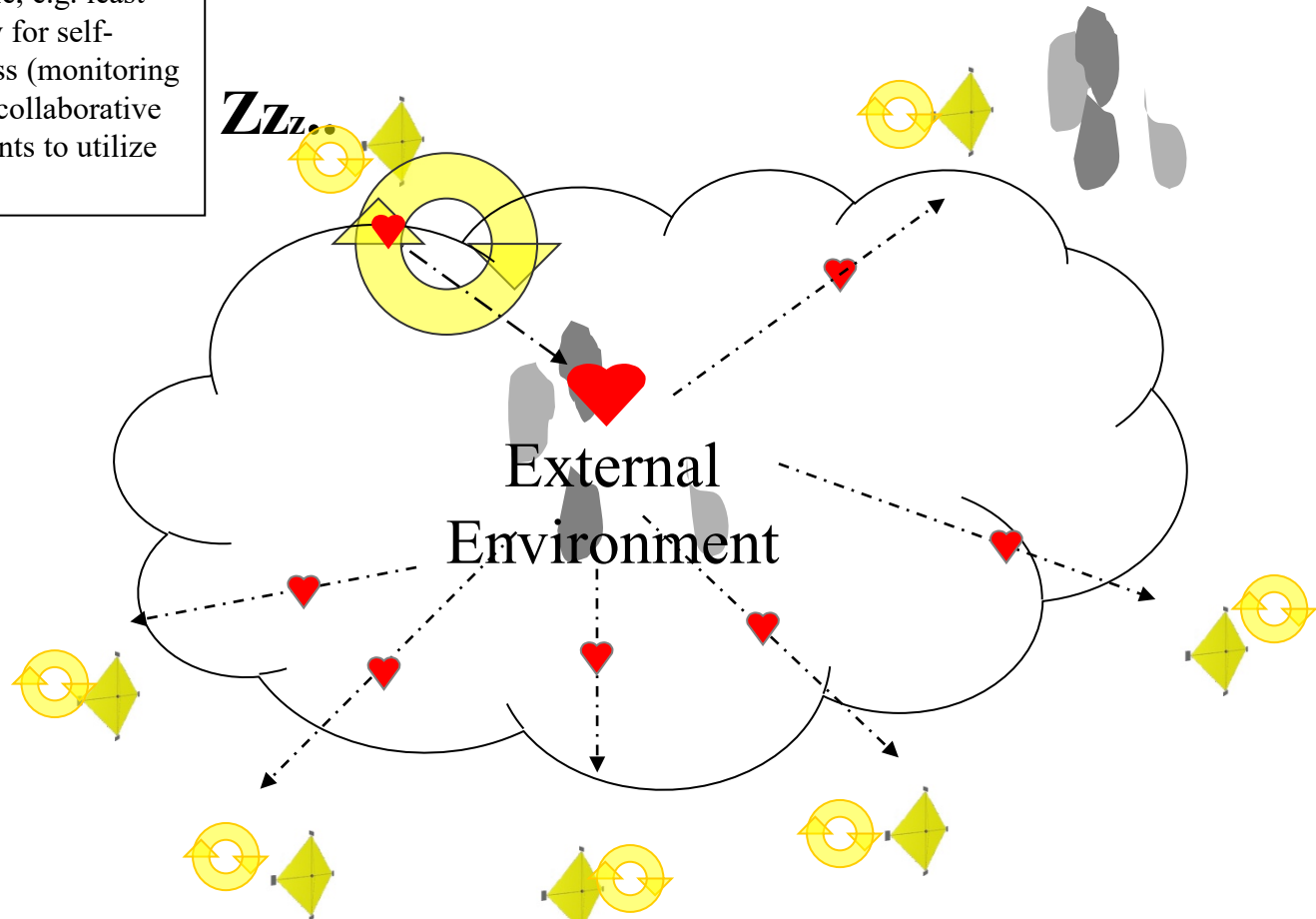
*Self-Situation
(environment
awareness)*



*ANT Spacecraft
(containing
Autonomic Element)*



Instead of every craft in the sub swarm monitoring the same local environment, one or two elements take on this role, e.g. least busy craft, taking responsibility for self-situation/environment awareness (monitoring vital signs) for its peer-to-peer collaborative group allowing the other elements to utilize their processing elsewhere.



When an environmental incident occurs, through changes in pulse (*reflex reaction*), the other crafts autonomic elements become alerted to emerging change in situation and switch on their external monitoring.

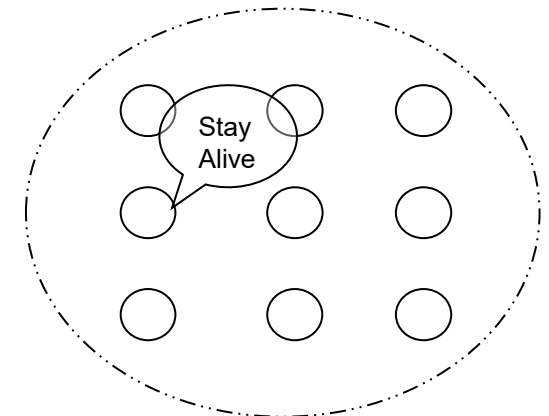
Key: ❤️ *Pulse Monitor (providing autonomic reflex reaction)*

Difficulty of Testing Swarms

- Emergent properties that may not be known
- Highly distributed and parallel
- Large number of interacting entities
- Worse than exponential growth in interactions
- Intelligent entities (capabilities increase over time)
- Total or near total autonomy
- What if things do go wrong?

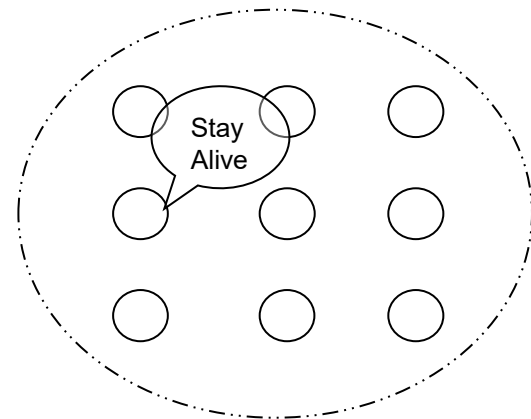
Apoptosis – *returning to Biological metaphors*

- if you cut yourself and it starts bleeding...
- often, the cut will have caused skin cells to be displaced down into muscle tissue
- if they survive and divide, they have the potential to grow into a tumour
- the body's solution to dealing with this is cell self-destruction
- with mounting evidence that certain types 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
- i.e. self-destruct (sD) is an intrinsic property.



Apoptosis – *returning to Biological metaphors (cont.)*

- this sD is delayed due to the continuous receipt of biochemical reprieves.
- 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.
- 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



Self-protection & Trust

Security issues with Agents

Greenburg (1998) 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.
- the mail had survived 'pending' on Internet relays waiting for *omega.univ.edu* to come back up.

The same situation could arise for mobile agents;
these would not be rogue mobile agents –
they would be carrying proper authenticated credentials.

The mobile autonomic agent could cause substantial damage,
e.g., deliver an archaic upgrade (self-configuration)
to part of the network operating system resulting in bringing down the entire network



Self-protection & Trust

self-destruct property a solution?

Generally the security concerns with agents

- misuse of hosts by agents (accidental)
 - accidental or unintentional situations caused by that agent (race conditions and unexpected emergent behavior)
- misuse of agents by hosts (accidental or deliberate)
- misuse of agents by other agents (accidental or deliberate)
 - 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.



Self-protection & Trust

self-destruct property a solution? (cont.)

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.

This situation has similar overtones to biological apoptosis,
where the body is at its most vulnerable during cell division.

AUTONOMIC
COMPUTING

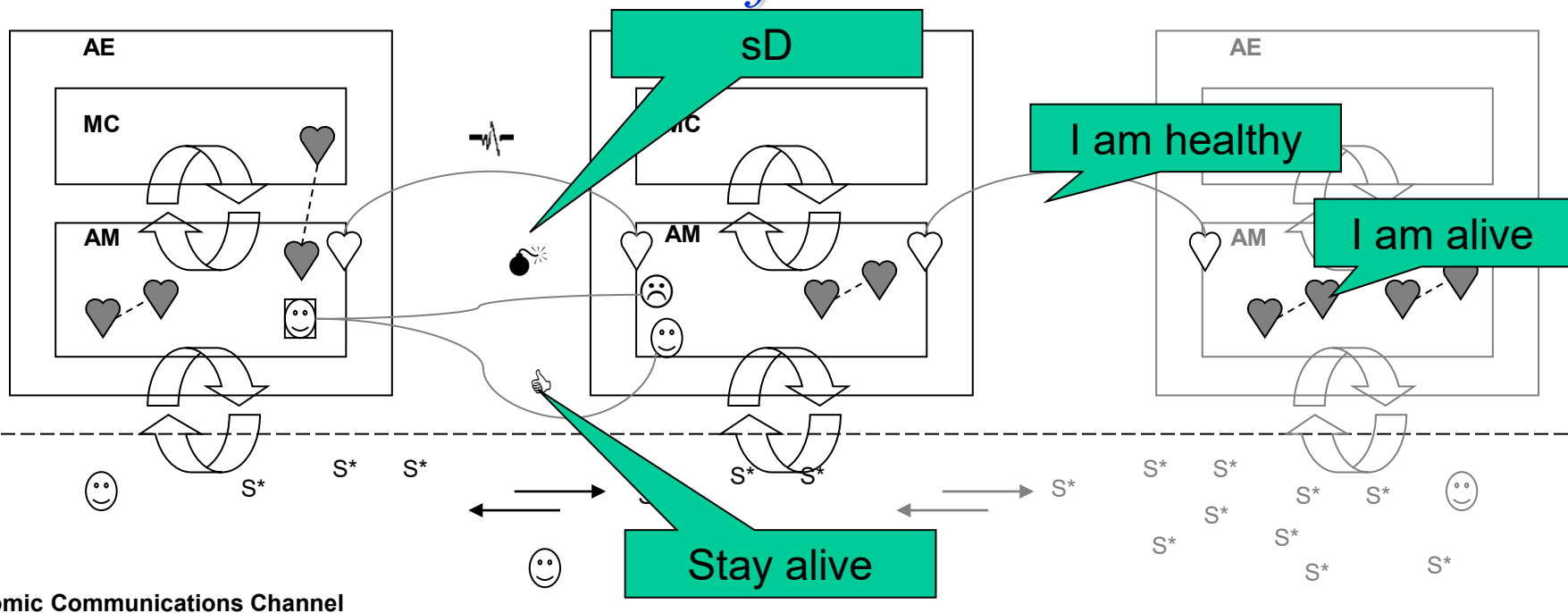
SELF MANAGEMENT

OBJECTIVES

ATTRIBUTES

SELF CONFIGURING
SELF HEALING
SELF OPTIMISING
SELF PROTECTING
SELF AWARE
ENVIRONMENT AWARE
SELF MONITORING
SELF ADJUSTING

Autonomic Computing – *The Autonomic Environment inc self-destruct*



Key

S* Self-* event messages

♥ PBM Pulse Monitor (PBM)

♥ HBM Heart-Beat Monitor (HBM)

😊 AA Autonomic Agent (Mobile agent)

AE Autonomic Element (AM+MC)

MC Managed Component

AM Autonomic Manager (Stationary agent)

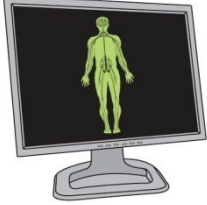
😊💣👍 AA Autonomic Agent Apoptosis Controls

Background Reading

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

@IEEE
[Apoptotic computing: Programmed death by default for computer-based systems - IEEE Journals & Magazine](#)

@Pure
[Apoptotic Computing: Programmed Death by Default for Computer-Based Systems — Ulster University](#)



COVER FEATURE

Apoptotic Computing: Programmed Death by Default for Computer-Based Systems

Roy Sterritt, University of Ulster

Inspired by the cellular self-destruct mechanisms in biological apoptosis, apoptotic computing offers a promising means to develop self-managing computer-based systems.

At the 2009 International Joint Conference on Artificial Intelligence, researchers warned that the nightmare scenarios depicted in sci-fi films such as *2001: A Space Odyssey*, the *Terminator* and *Matrix* series, *Minority Report*, and *I, Robot* could come true. "Scientists fear a revolt by killer robots" proclaimed the UK's *Sunday Times*,¹ which highlighted alarming findings at the conference that mankind might lose control of computer-based systems that carry out a growing share of society's workload, from chatting on the phone to waging war, and have already reached a level of indestructibility comparable with the cockroach. For instance, unmanned predator drones, which can seek out and kill human targets, have already moved out of the movie theatre and into the theatre of war in Afghanistan and Iraq. While presently controlled by human operators, these drones are moving toward more autonomous control. Similar devices may also soon appear above city streets to carry out domestic surveillance. Samsung, the South Korean electronics giant, has developed autonomous sentry robots with "shoot to kill" capability to serve as armed border guards.²

To provide for this future, the Apoptotic Computing project has been working since 2002 toward the long-term goal of *programmed death by default* for computer-based systems.³⁻⁶ Motivated by the apoptosis mechanisms in multicellular organisms, apoptotic computing can be considered a subarea of bio-inspired computing, natural computing, or autonomic systems. Two example applications are autonomic agent-based environments and swarm space exploration systems.

BIOLOGICAL APOPTOSIS

Developing a self-managing computer system is the vision of autonomic computing.^{7,8} As the "Autonomic System Properties" sidebar explains, an autonomic computing system is analogous to the biological nervous system, which automatically maintains homeostasis (metabolic equilibrium) and controls responsiveness to external stimuli. For example, most of the time you are not consciously aware of your breathing rate or how fast your heart is beating, while touching a sharp knife with your finger results in a reflex reaction to move the finger out of danger.¹⁰

If you cut yourself and start bleeding, you treat the wound and carry on without thinking about it, although pain receptors will induce self-protection and self-configuration to use the other hand. Yet, often the cut will have caused skin cells to be displaced down into muscle tissue.¹¹ If the cells survive and divide, they have the potential to grow into a tumor. The body's solution to this situation is cell self-destruction (with mounting evidence

Anonymous Autonomous/Autonomic Agents & Trust *ALice Signal?*

Assumptions so far,
autonomic elements work together to achieve self-managing system

Heterogeneous and potentially open systems -
Anonymous Autonomic Agents

As such the agents and their hosts need to be able to identify each others
credentials through for instance,
such means as an Autonomic License (ALice) signal.

AUTONOMIC
COMPUTING

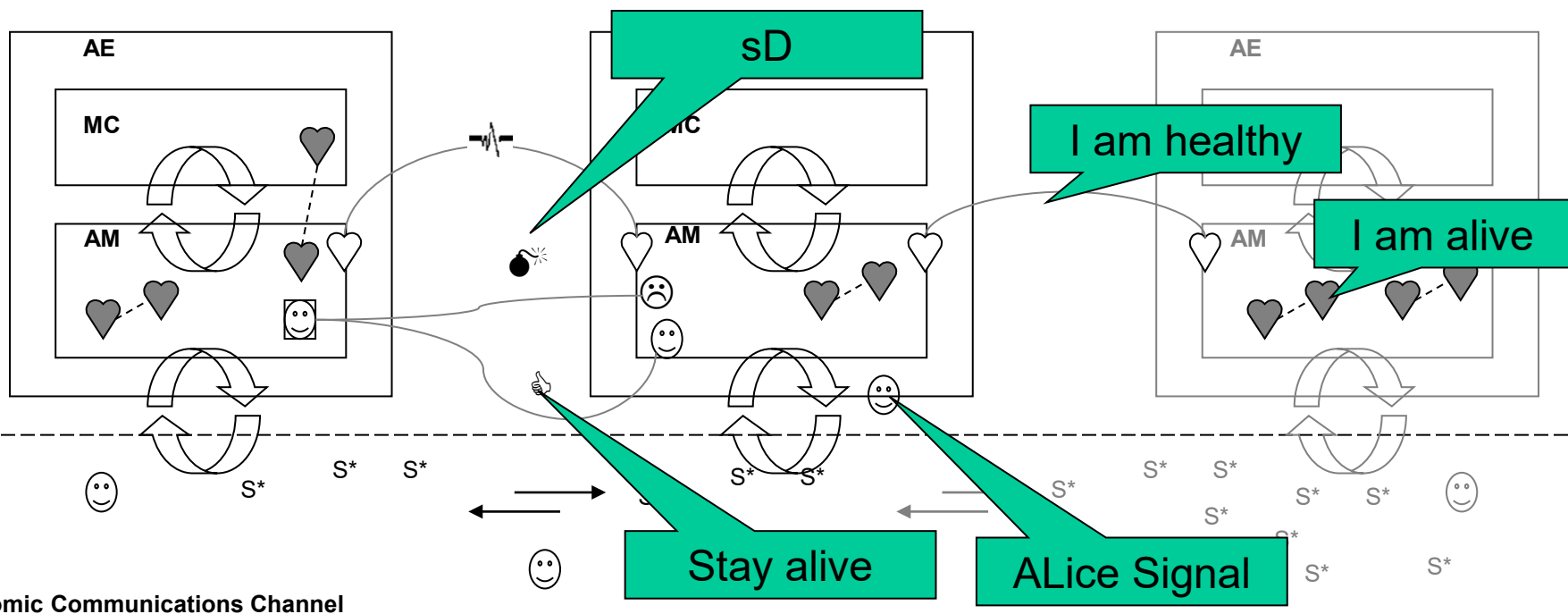
SELF MANAGEMENT

OBJECTIVES

ATTRIBUTES

SELF CONFIGURING
SELF HEALING
SELF OPTIMISING
SELF PROTECTING
SELF AWARE
ENVIRONMENT AWARE
SELF MONITORING
SELF ADJUSTING

Autonomic Computing – *The Autonomic Environment inc ALice*



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

AUTONOMIC
COMPUTING

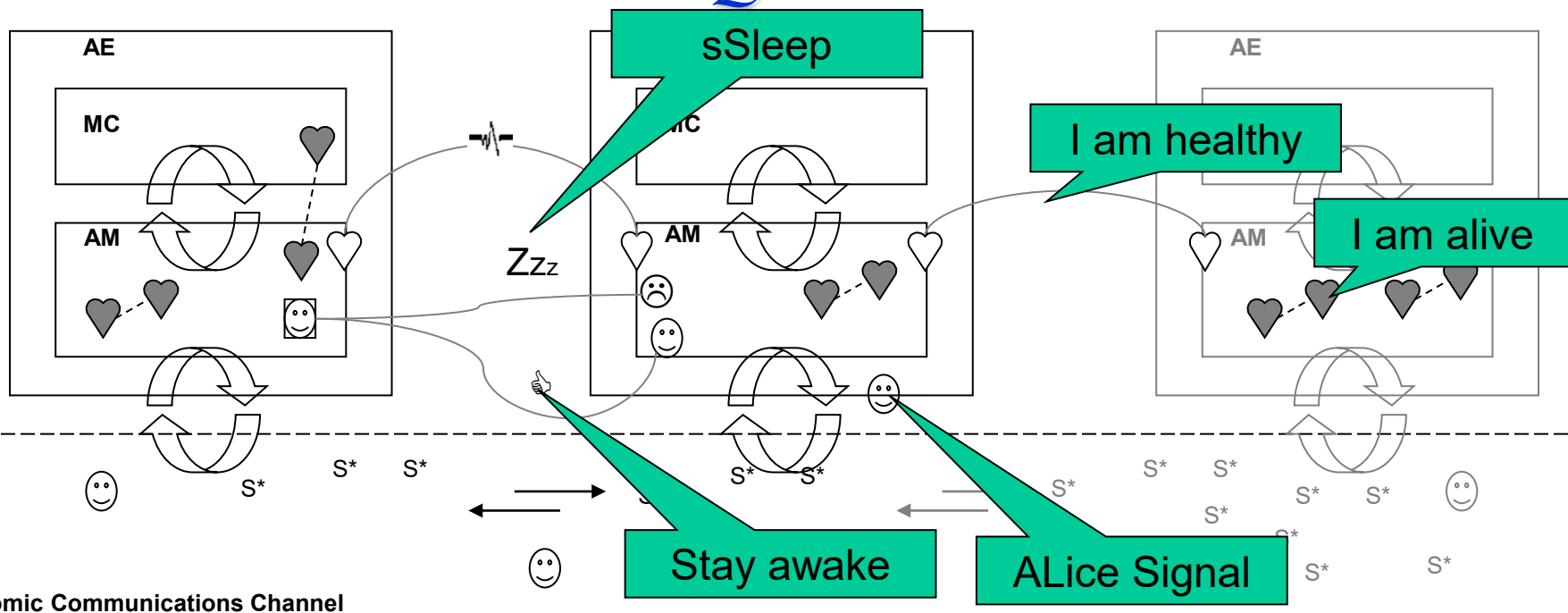
SELF MANAGEMENT

OBJECTIVES

ATTRIBUTES

SELF CONFIGURING
SELF HEALING
SELF OPTIMISING
SELF PROTECTING
SELF AWARE
ENVIRONMENT AWARE
SELF MONITORING
SELF ADJUSTING

Autonomic Computing – *The Autonomic Environment inc Quiescent*



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

Real Time Considerations?

Under normal conditions the pulse would act like the HBM, sending at regular intervals, yet on encountering circumstances affecting the system the pulse rate will increase to warn of the problem.

This dynamic pulse rate is consistent with the biological metaphor, but it is also desirable to ensure that information is reported more frequently when operating conditions become difficult (flight or fight, SyNS).

To achieve the reflex reaction a signal should be sent immediately, implying a change in the pulse rate, which should then stay high, reporting state information, until the situation is resolved.

Real Time Considerations?

Yet in some scenarios, under fault conditions a major fault can cause such a cascade of alarm event messages that it affects the real-time operation of the system and appears non-deterministic; under such management event message flooding it can be difficult to even provide adequate service .

Therefore, another extension, the adaptive pulse monitor, whereby the rate of the pulse will adapt to take into consideration bandwidth concerns and the congestion on the network.

In effect, after the initial reflex reaction, the pulse and other self-*event messages would actually decrease, (in effect putting the self-managing system into rest and digest, parasympathetic (PaNS) mode).

The key concept is that we must actively reduce alerting so that achieving autonomicity is not actually making the situation and response worse.

Autonomic Apoptosis in NASA Missions?

- With NASA missions we are not considering a generic situation.
- Mission control and operations is a trusted private environment.
- This eliminates many of the wide range of agent security issues
- Leaving the particular concerns;
 - is the agent operating in the correct context
 - and showing emergent behavior within acceptable parameters?

Autonomic Apoptosis in NASA Missions?

- 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.
- 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.
- Another worker could self-configure to take on the role of the lost worker; i.e., the ANTS adapt to ensure an optimal and balanced coverage of tasks to meet the scientific goals.
- 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.



Points to Consider at this stage ...



Self-Management

Human out of the low-level loop

Trust an issue that warrants building in self-protection mechanism such as self-destruct and self-sleep?

Aka pre-programmed default responses

Think Asimov's 3 laws of Robotics

Talk Outline

- 1: 2001–2021: 20 years of Autonomic Computing in 20 mins
- 2: Research with NASA in Autonomicity – *contribution to & from*
- 3: The next 20 years for Autonomic Computing

Background Reading

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

@IEEE

[Fulfilling the Vision of Autonomic Computing - IEEE Journals & Magazine](#)

@Pure

[Fulfilling the Vision of Autonomic Computing — Ulster University](#)



COVER FEATURE



FULFILLING THE VISION OF AUTONOMIC COMPUTING

Simon Dobson, *University of St. Andrews, UK*
Roy Sterritt, *University of Ulster, Northern Ireland*
Paddy Nixon and Mike Hinchey, *Lero—the Irish Software Engineering Research Centre*

Efforts since 2001 to design self-managing systems have yielded many impressive achievements, yet the original vision of autonomic computing remains unfulfilled. Researchers must develop a comprehensive systems engineering approach to create effective solutions for next-generation enterprise and sensor systems.

In 2001, IBM researchers predicted that by the end of the decade the IT industry would need up to 200 million workers, equivalent to the entire US labor force, to manage a billion people and millions of businesses using a trillion devices connected via the Internet.^{1,2} Only if computer-based systems became more autonomic—that is, to a large extent self-managing—could we deal with this growing complexity, and they accordingly issued a formal challenge to researchers.

We have reached 2010, and, much like the Y2K problem, the situation clearly is not as extreme as anticipated. So was it all hype, or has the IT industry had a very productive decade? Have we met IBM's challenge, or have we simply performed another heroic effort without solving the underlying problem?

BACK TO THE FUTURE

In its autonomic computing call to arms, IBM compared what the IT industry faced in 2001 to what occurred in the US telephony industry in the 1920s. At that time, the rapid expansion and infiltration into daily life of the phone aroused serious concern that there would not be enough trained operators to work the manual switchboards. Analysts predicted that by the 1980s, half the country's population would have to become telephone operators to meet demand. AT&T/Bell System's implementation of the automated switching protocol and other technological innovations averted this crisis.

In 2001, unfilled IT jobs in the US alone numbered in the hundreds of thousands, even in uncertain economic conditions, and global demand for IT workers was expected to increase by more than 100 percent in the next five years. Today's actual employment numbers are hard to determine, as government statistics do not explicitly capture system administration, IT maintenance, and other related functions. However, crude data from the Bureau of Labor Statistics suggests that there are approximately 260,000 IT workers in the US, with employment in the industry declining slightly but steadily over the past decade³ despite the enormous increase in computing power available. This trend suggests that consolidation of computing power, which will increase alongside the use of cloud computing

Still Fulfilling the Vision of AC...

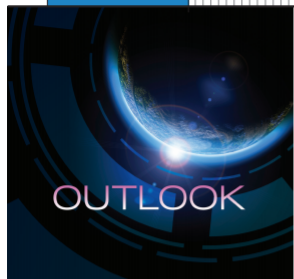


COVER FEATURE

The Vision of Autonomic Computing



COVER FEATURE



FULFILLING THE VISION OF AUTONOMIC COMPUTING

Simon Dobson, University of St. Andrews, UK
Roy Sterritt, University of Ulster, Northern Ireland
Paddy Nixon and Mike Hinchey, Lero—the Irish Software Engineering Research Centre

Efforts since 2001 to design self-managing systems have yielded many impressive achievements, yet the original vision of autonomic computing remains unfulfilled. Researchers must develop a comprehensive systems engineering approach to create effective solutions for next-generation enterprise and sensor systems.

BACK TO THE FUTURE

In its autonomic computing call to arms, IBM compared what the IT industry faced in 2001 to what occurred in the US telephony industry in the 1920s. At that time, the rapid expansion and infiltration into daily life of the phone aroused serious concern that there would not be enough trained operators to work the manual switchboards. Analysis predicted that by the 1980s, half the country's population would have to become telephone operators to meet demand. AT&T/Bell System's implementation of the automated switching protocol and other technological innovations averted this crisis.

In 2001, unfilled IT jobs in the US alone numbered in the hundreds of thousands, even in uncertain economic conditions, and global demand for IT workers was expected to increase by more than 100 percent in the next five years. Today's actual employment numbers are hard to determine, as government statistics do not explicitly capture system administration, IT maintenance, and other related functions. However, crude data from the Bureau of Labor Statistics suggests that there are approximately 260,000 IT workers in the US, with employment in the industry declining slightly but steadily over the past decade despite the enormous increase in computing power available. This trend suggests that consolidation of computing power, which will increase alongside the use of cloud computing

In 2001, IBM researchers predicted that by the end of the decade the IT industry would need up to 200 million workers, equivalent to the entire US labor force, to manage a billion people and millions of businesses using a trillion devices connected via the Internet.¹ Only if computer-based systems became more autonomic—that is, to a large extent self-managing—could we deal with this growing complexity, and they accordingly issued a formal challenge to researchers. We have reached 2010, and, much like the Y2K problem, the situation clearly is not as extreme as anticipated. So was it all hype, or has the IT industry had a very productive decade? Have we met IBM's challenge, or have we simply performed another heroic effort without solving the underlying problem?

0018-9162/10/\$26.00 © 2010 IEEE

Published by the IEEE Computer Society

JANUARY 2010 35

In 2010, as we approached a decade, we published an intentional follow up to Kephart & Chess seminal article in Computer – with a lot less impact! “Fulfilling the vision of Autonomic Computing”.

A key point made: “Over the years, the AI field has fallen victim to unrealistic expectations, and we see similar warning signs in the autonomic computing field. Yet from the beginning there has been a successful focus on evolutionary research, tightly linked to applied industrial problems. Additional funding and industrial collaboration are crucial to future success, but something more is required: Researchers must develop a long-range, overarching strategy to realize the vision propounded by Kephart and Chess.”

“Yet in some ways that success is deceptive. Researchers have devised innovative autonomic solutions to individual problems, but the larger, more difficult task of combining more difficult task of combining these point solutions into wider autonomous systems remains. More consideration must be given to integrating solutions, and to choosing solutions from the range of possibilities—to trustworthy and assured autonomous and autonomic systems engineering, in other words.”

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



The 2010s of AC...

At the end of the second decade of the Autonomic Computing much of this is still the case, although great success has occurred with Cloud Computing & A.C.

And A.I. is now starting to live up to expectations...

2009 world-wide economic crash took much funding & sponsorship away from the field, so the next decade had a lot less hype.

The top cited paper in the field is Kephart & Chess' AC Vision paper from 2003 (with over 8000 citations), the 2nd highest cited paper is from 2012 "Addressing cloud computing security issues" (with over 2300 citations), note Autonomic isn't even in the title of the paper! This reflects the 2010s for AC, that AC effectively converged with the new star of the research block – Cloud Computing – so much so, that although Autonomicity is key to the success of Cloud Computing – many within the field don't know it as Autonomics ...

Within the Space sector, Autonomics has ticked along much slower – with changes in operation at NASA and more outsourcing to the private sector etc..

Personally, EASe @ SMC-IT every two years, NASA journal ISSE & PhD programmes, for instance on Autonomic Computing with Swarm computing, AC for preventing Space Junk, etc..





The 2020s and beyond for AC...

With GSaaS (Ground Station as a Service) including offerings from Amazon ([AWS Ground Station](#)) and Microsoft ([Azure Orbital GSaaS Support Services \(microsoft.com\)](#)) now a thing, Cloud has come to the Space sector...

with Autonomicity a key aspect of Cloud Computing, its future is assured.

We've been pushing for all computer-based systems to be autonomic for 20 years, with personal research interest in Space, with GSaaS – autonomicity is going to be needed at the other end aka satellites and craft. This should lead to a renewed interest in the Autonomic field.

The 4th wave of AI has been on the Gartner Hype cycle (curve) these last few years ...

Seeing real industrial application of AI. AI has been highlighted as a necessary contributor from the birth of AC although the Engineering side took off first ... and ventures into AI community via IJCAI in [2003](#) & [2005](#) didn't have the AI researchers rushing to join in ...

But is now the time for properly addressing the learning side of AC?

This renewed interest in AI offers the potential for renewed interest in Autonomic Computing beyond the existing researchers.

Autonomicity can be presented as the separation of concerns i.e. (self-) management from that of the task/mission in **Autonomy**. Both will benefit from new developments and funding in AI.



The 2020s and beyond for AC...

Concern about Trustworthy in Autonomous Systems (TAS) and Assured Autonomous Systems is of current interest, as expressed by UKRI funding ([for instance](#)), and an argument can be made that it can be (partially) achieved [through Autonomic Computing](#).

A fantastic amount has been achieved in the last 20 years within **Autonomic Computing**, including for the Space industry (although not always explicitly under the Autonomic banner). For instance, the research briefly recapped here lead to [16 patents with NASA](#), and too many publications!

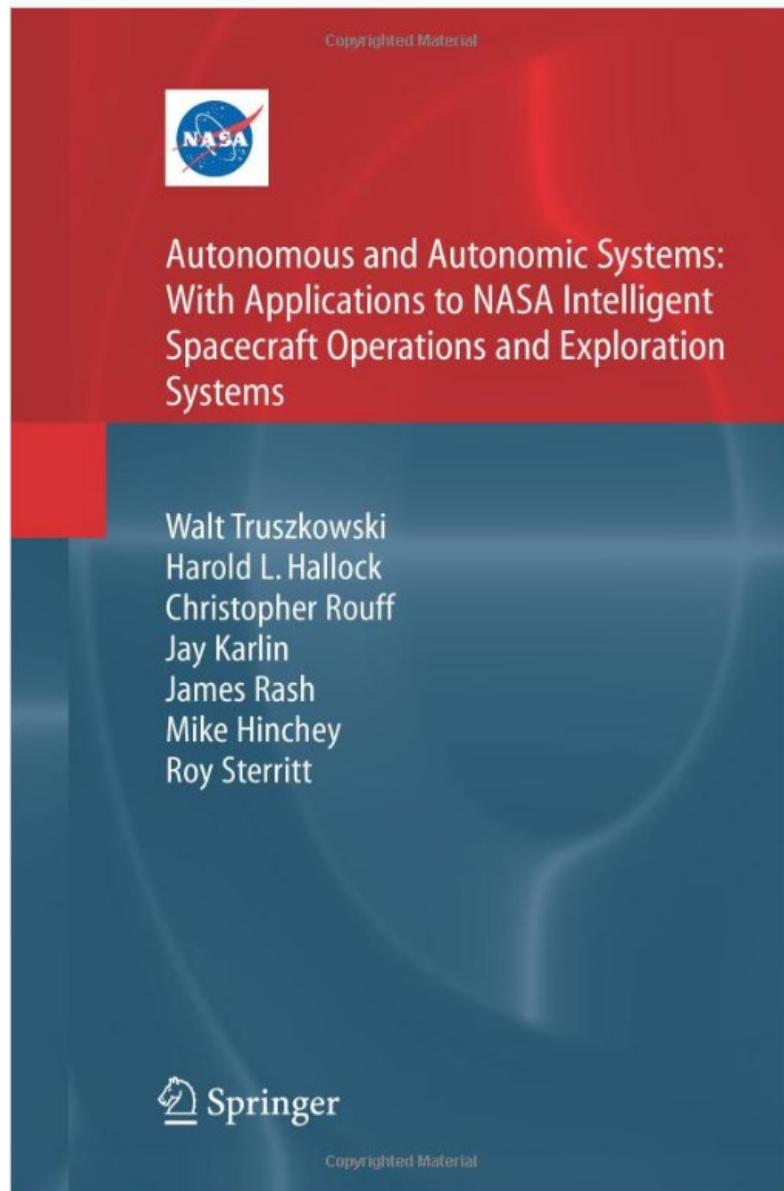
NASA
GSFC
Patents
Awards
2012



The next decade or two should see that fulfilment of the Autonomic Computing Vision.



Acknowledging my former NASA colleagues & Background Reading



Truskowski, W., Hallock, H. L., Rouff, C., Karlin, J., Rash, J., Hinchey, M. G., & Sterritt, R. (2009). *Autonomous and Autonomic Systems With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*. Springer. <http://springer.com/978-1-84628-232-4>

@Springer

[Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems | Walt Truskowski | Springer](http://springer.com/978-1-84628-232-4)

@Pure

[Autonomous and Autonomic Systems With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems — Ulster University](http://pure.ulster.ac.uk/publication/autonomous-and-autonomic-systems-with-applications-to-nasa-intelligent-spacecraft-operations-and-exploration-systems)

2001-2021 ...
we could only achieve so much...



Thank you.
Questions?
Comments?

2001-2021 ...
we could only achieve so much...



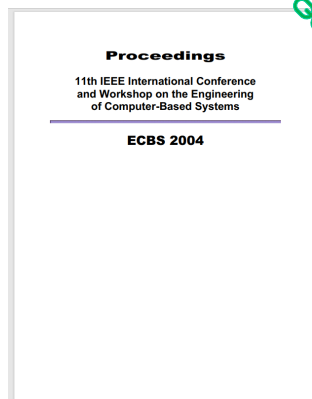
Some background on
EASe follows for
reference ...

IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASE)

Copy Persistent Link

Browse Title List

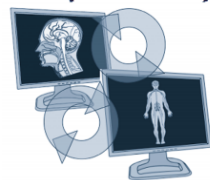
Sign up for Conference Alerts



1st EASe (2004) in ECBS

Third IEEE International Workshop on Engineering of Autonomic & Autonomous Systems

Proceedings of the Third IEEE International Workshop on **ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2006)**



28-30 March 2006
Potsdam, Germany

24-28 April 2006
Columbia, MD, USA

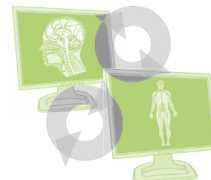


Edited by:
Roy Sterritt
Michael G. Hinchey
Ted Bapty

Proceedings of the Fifth IEEE International Workshop on
ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2008)

Proceedings of the
Fifth IEEE International Workshop on

ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2008)



Making Systems More Effective

31 March – 4 April 2008
Belfast, Northern Ireland

Edited by:
Roy Sterritt, David W. Bustard, David Lewis, Joel J. Fleck II

Sponsored by
IEEE Computer Society Technical Committee on Autonomous and Autonomic Systems (TC-AAS)



Proceedings of the Seventh IEEE International Conference and Workshop on
ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2010)

Proceedings of the
Seventh IEEE International Conference and Workshop on

ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2010)



22-26 March 2010 • Oxford, England
Edited by: Roy Sterritt and Julie McCann



Incorporating Post-Proceedings of AA-IEEE for Autonomic & Autonomous Special Exploration Systems (AA-IEEE), Potomac, July 2009
Sponsored by IEEE Computer Society Technical Committee on Autonomous and Autonomic Systems (TC-AAS)



2nd EASe (2005) in ECBS

Fourth IEEE International Workshop on Engineering of Autonomic & Autonomous Systems

Proceedings of the Fourth IEEE International Workshop on **ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2007)**



March 26-29 2007
Tucson, AZ, USA

Including
EASe Special Session on
Autonomous and Autonomic
Space Exploration Systems
(AA-SES-3)
March 6-8 2007
Baltimore, MD, USA

Edited by:
Roy Sterritt, Michael G. Hinchey, Ted Bapty

Incorporating Selected Letters of the
IEEE Task Force on Autonomous and Autonomic Systems (2006)



Sponsored by
IEEE Computer Society Task Force on Autonomous and Autonomic Systems

Proceedings of the Sixth IEEE International Conference and Workshop on
ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2009)

Proceedings of the
Sixth IEEE International Conference and Workshop on

ENGINEERING OF AUTONOMIC & AUTONOMOUS SYSTEMS (EASe 2009)

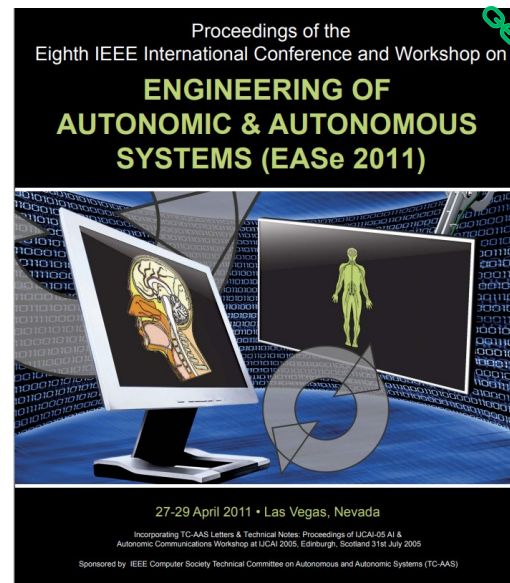


14 - 16 April, 2009
San Francisco, CA, USA

Edited by:
Roy Sterritt



Sponsored by
IEEE Computer Society Technical Committee on Autonomous and Autonomic Systems (TC-AAS)



Incorporating TC-AAS Letters & Technical Notes, Proceedings of LCAI-05 AI & Autonomic Communications Workshop at LCAI 2005, Edinburgh, Scotland 31st July 2005
Sponsored by IEEE Computer Society Technical Committee on Autonomous and Autonomic Systems (TC-AAS)

IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASE)

Copy Persistent Link

Browse Title List

Sign up for Conference Alerts

The 1st EASE workshop was held at the 11th ECBS (Engineering of Computer-Based Systems – a more modern term would be CPS!) in 2004 (after establishing it at ECBS 2003 following the ECBS paper “[Autonomic Computing-a Means of Achieving Dependability?](#)”) as a strategy to bring in a new hot topic which was very relevant to the IEEE ECBS Technical Committee (System & Software Engineers).

My personal plan for EASe was it would remain a supporting workshop at ECBS ... but other things took over. Once a TF then TC for AAS was established metrics become important, and EASE was needed as a TC-AAS sponsored event. For its active (published proceedings) years it co-located with ECBS and often ICECCS. In those early years it was often also run with the IEEE/NASA [SMC-IT](#) (Space Mission Challenges for I.T.), which was held tri-annually then bi-annually, as this linked with some of my personal research Autonomic Computing for Space, a collaboration with NASA GSFC – but as a sub-workshop of EASe known as *AA-SES: AUTONOMIC & AUTONOMOUS SPACE EXPLORATION SYSTEMS*. EASe also found a role as the ‘concepts’ conference on AC, as often ICAC would mainly accept papers with proven findings – and who would blame them with many IBMers with working systems keen to publish in those early days. This role suited the research with NASA, especially on concept missions (30 years hence), sometimes described at that stage as more Science Fiction than Science...

With the sunseting of the TC-AAS (TC-ECBS & TC-CX were also sunset) EASe became a workshop solely at SMC-IT (AA-SES title being dropped) with the aim to keep Autonomic Computing on the agenda within Space Software research (for instance [EASe@SMCIT-2021](#))



- Home
- Author Information
- Call for Papers
- Manuscripts
- Call for Participation
- Committee
- Contact Us
- Exhibits
- Night Out
- Program
- Registration
- Schedule
- Travel and Local Area Information
- Workshops
- Archives
- SMC-IT 2006
- SMC-IT 2003
- Advertising Materials



MINI-WORKSHOP ON AA-SES-IV: AUTONOMIC & AUTONOMOUS SPACE EXPLORATION SYSTEMS

<http://www.infj.ulst.ac.uk/~autonomic/AA-SES/>

CALL FOR PAPERS

Held in conjunction with the Third International Conference on Space Mission Challenges for Information Technology (SMC-IT 2009), July 19-23, 2009, in Pasadena, California.

<http://smc-it.org>

Abstract

Space missions require the use of complex hardware, software and embedded systems, often with hard realtime requirements. Most missions involve significant degrees of autonomous behavior, often over significant periods of time. While missions typically have human monitors, many missions involve very little human intervention, and then often only in extreme circumstances. It has been argued that Space Exploration Systems should be autonomic (self-managing) as well as autonomous (self-governing), and that all autonomous systems should be autonomic by necessity. Indeed, the trend is in that direction in forthcoming Space missions. This technical session will present papers on research into Autonomicity and Autonomy for Space exploration missions. With Autonomy and Autonomicity becoming mainstream within computing through such initiatives as Autonomic Computing, this session aims to be of general interest to both the Space and Information and Communication Technologies (ICT) communities.

Index Terms — Autonomy, Autonomicity, Self-Managing Systems, Space Exploration Systems

Organizers

Roy Sterritt, *Member, IEEE*; Michael G. Hinchey, *Senior Member, IEEE*

Principal Organizer

Roy Sterritt,
University of Ulster
Northern Ireland
+44 (0)2890368198
Fax: +44 (0)2890366068
r.sterritt@ulster.ac.uk



*For instance,
AA-SES-IV @ SMC-IT
2009 post-proceedings
included in EASe 2010
proceedings ...*

