Next Generation System and Software Architectures:

Challenges from future NASA exploration missions

Roy Sterritt[1], Christopher A. Rouff[2], Michael G. Hinchey[3], James L. Rash[3], Walt Truszkowski[3]

[1] School of Computing and Mathematics, University of Ulster, Jordanstown Campus, Shore Road, Newtownabbey, Co. Antrim, BT37 0QB, Northern Ireland r.sterritt@ulster.ac.uk

[2] SAIC, Advanced Concepts Business Unit, 1710 SAIC Drive, McLean, VA 22102, USA rouffc@saic.com

[3] Information Systems Division, Code 580, NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA Michael.G.Hinchey@nasa.gov, James.L.Rash@nasa.gov, Walt.Truszkowski@nasa.gov

Abstract

The four key objective properties of a system that are required of it in order for it to qualify as "autonomic" are now well-accepted—self-configuring, self-healing, self-protecting, and self-optimizing—together with the attribute properties—viz. self-aware, environment-aware, self-monitoring and self-adjusting. This paper describes the need for next generation system software architectures, where components are agents, rather than objects masquerading as agents, and where support is provided for self- properties (both existing self-chop and emerging self-* properties). These are discussed as exhibited in NASA missions, and in particular with reference to a NASA concept mission, ANTS, which is illustrative of future NASA exploration missions based on the technology of intelligent swarms.*

Keywords: Self-*, Selfware, Autonomous Systems, Autonomic Systems, Agent Architectures, Multi-Agent Technology, Intelligent Systems, Spacecraft.

1. Introduction

The advent of distributed object technologies such as CORBA removed the prior restriction on object-oriented implementations to have all objects residing on the same machine. Additionally, such approaches facilitate the use of multiple implementation languages as long as they can all use the same Interface Definition Language (IDL). The result has been very significant for system and software architectures, affording complex distributed architectures that were previously impossible.

Distributed object technologies are, however, severely limited in terms of the architectures they support. Notwithstanding their support for multiple platforms, multiple environments, multiple programming languages, and highly distributed implementations, they tend to result in a monolith of tightly-coupled objects, where method invocations are hand-coded, and where the names of methods must be known *a priori*, and where all objects are known at the outset (that is, there is no dynamic creation of objects).

Distributed agent technologies have, in many ways, overcome these problems. They allow for significantly greater flexibility, and in particular they offer autonomous behavior, whereby individual components can be self-directed, having their own agenda to pursue, which they do without human intervention. In technologies such as the Open Agent Architecture (OAA), a *Facilitator* enables agents to send requests to a single source, rather than knowing about other agents in the system. The result is a system that can adapt in ways that hard-coded distributed object systems cannot, and which can be extended dynamically through the addition of new agents, of which prior information need not be available. Add to this the ability to support *mobile agents*, whereby agents can move (themselves) to execute on other machines, and the result is a very powerful architecture.

Future NASA missions will push the limits of current system and software architectures. Planned and concept NASA missions represent some of the largest, most ambitious, most challenging, and expensive[1] software projects to date. These missions will exploit emerging concepts such as intelligent swarms, whereby large numbers of (relatively) heterogeneous and simple components will act in unison, analogous to swarms in nature, producing complex behaviors that could not be achieved without their emergent behavior. These missions must not only support fully autonomous behavior, in order to exploit the benefits of this emergence, they must be survivable in deep space, which requires that they also exhibit *autonomic* properties. Current system and software architectures fail to meet the needs of such missions.

The major point that is brought out in this paper is the need for next generation system and software architectures, in which architectural components are, for instance, autonomous agents, in contrast to the majority of the current assumptions that the behaviors of such components are relatively predictable, and that they communicate with one another using fixed protocols.

2. Self-* in NASA Missions

The term *selfware* has been coined to refer to the growing set of self-properties that are emerging in the Autonomic Computing and other related self-managing systems initiatives. The initial set of properties defined by [1]—namely self-configuration, self-healing, self-optimization and self-protection (objectives: what is to be achieved) through self-awareness, self-monitoring and self-

[1] The current estimates for the software component of the mission to Mars exceed US$48billion, which would make it the most expensive software project to be ever undertaken.

adjusting (attributes: how it is to be achieved)—has been expanded, and further properties are expected to be added to this ever-growing list. Additional monitoring constructs, such as pulse monitoring [13] and heart-beat monitoring, have also been proposed, along with biologically-inspired metaphors such as apoptosis and self-destruction [14],[16].

The autonomic computing initiative has firmly placed the goals of self-managing systems on the map through self-* properties, yet a lot of the objectives were already emerging or residing in the field of autonomous systems prior to the 2001 launch of the initiative. In fact, one definition of "autonomous" is "autonomic" [15]. What the initiative brings to the fore is that every system should exhibit these self-* properties in order to cope with the ever-rising complexity and total cost of ownership, and not just be a specialized autonomous domain. This is in addition to a focus on self-management (autonomicity) as opposed to self-governance (autonomy).

NASA, addressing the realities of increasing deep space exploration and the goal of more versatile and cheaper missions, has been addressing autonomy for some time now. This paper illustrates some of these self-* properties with reference to NASA missions. The challenge is to provide a suitable architecture for these in a cohesive, generic, and integrated fashion.

2.1 The Challenge of NASA Missions

New paradigms in spacecraft design are leading to radical changes in the way NASA designs spacecraft operations [18],[19]. Increasing constraints on resources, and greater focus on the cost of operations, has led NASA to utilize adaptive operations and move towards almost total onboard autonomy in certain classes of mission operations [26],[21]. Moreover, the loss of human life in two notable Shuttle disasters has delayed human exploration [16], and caused greater focus on the use of automation and robotic technologies in circumstances where heretofore human effort would have been used (e.g. the now defunct Hubble Space Telescope Robotic Servicing Mission—HRSM).

Additionally, there are many missions where humans simply cannot be utilized, for a variety of reasons. These include, obviously, longevity of the mission due to the distances involved (cf. the Cassini mission taking 7 years to reach Titan, the most important of Saturn's moons, and DAWN, a multiyear mission to aid in determining the origins of our universe, which includes the use of an altimeter to map the surface of Ceres and Vesta, two of the oldest celestial bodies in our solar system).

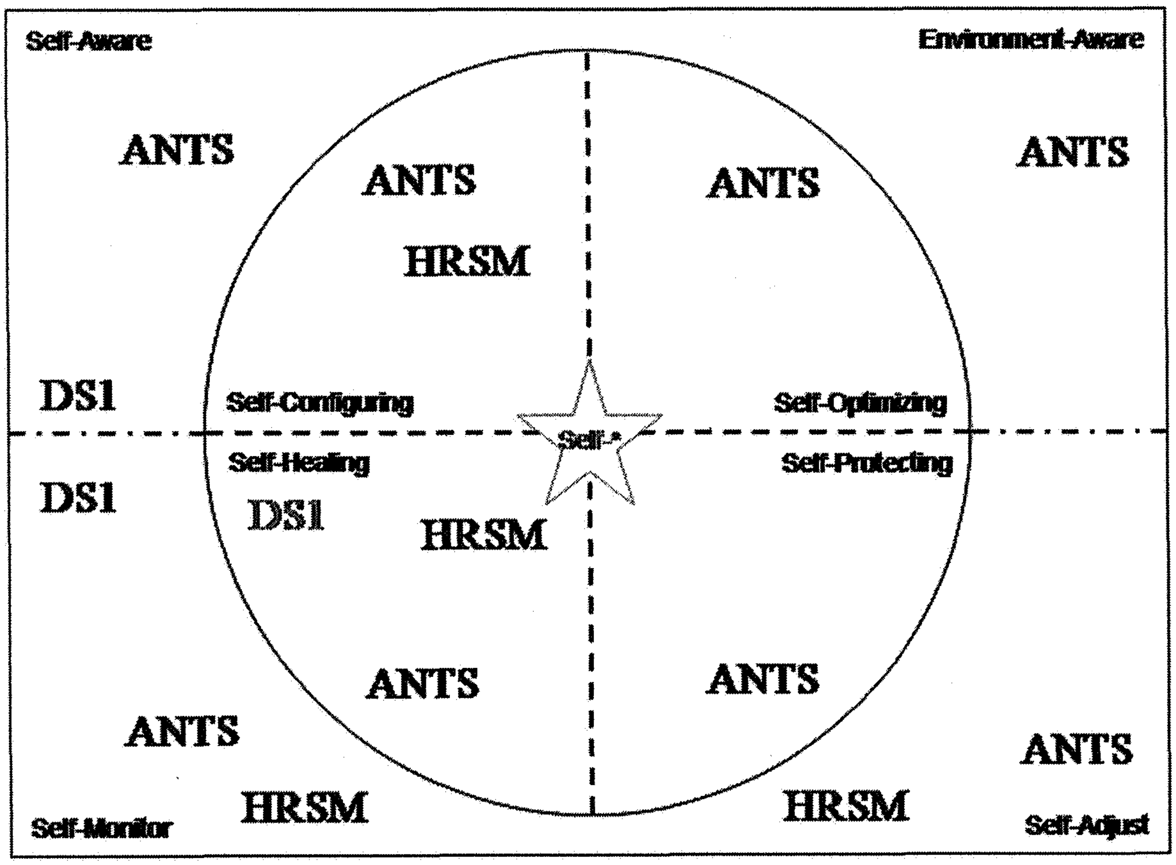

Figure 1 : Evolution of Self-* Properties in Missions

Risk is also a major factor pushing the use of unmanned craft (cf. HRSM, where lengthy space walks to perform the servicing would have entailed increased risks). There are also circumstances where it is just not safe to send humans (cf. the concept ANTS mission—discussed in more detail later—where miniature spacecraft will explore the asteroid belt, whereas a manned mission would be prohibitively expensive in terms of time and money, and would pose unacceptable risks to the crew, primarily due to the dangers of radiation).

More and more, these unmanned missions are being developed as autonomous systems, out of necessity. For example, almost entirely autonomous decision-making will be necessary to overcome the unacceptable time lag between a craft encountering new situations and the round-trip delay (of upwards of 40 (Earth) minutes) in obtaining responses and guidance from mission control.

More and more NASA missions will, and *must*, incorporate autonomicity as well as autonomy [22],[23],[24]. In short, as missions increasingly incorporate autonomy—being self-governing of their own goals—there is a strong case to be made that this needs to be extended to include autonomicity—that is, mission self-management [16].

One of the earliest systems to exhibit self-*, autonomy and some autonomicity (autonomic properties) was Deep Space 1 (DS1)—see Figure 1. In the DS1 mission [25], the responsibility of health monitoring was transferred from ground control to the spacecraft [12]. This marked a paradigm shift for NASA from its traditional routine telemetry downlink and ground analysis, to onboard health determination [25].

Some longer-term drawbacks of the approach were discovered. As one of the primary goals was to reduce the amount of data sent to the ground (achieved by eliminating the download of telemetry data except under unhealthy circumstances), operators lost the ability to gain an intuitive feel for the performance and characteristics of the craft and its components, in addition to losing the ability to run the data through simulations [19].

To resolve this, engineering data summarization was introduced to facilitate ground study of the long-term behavior of the spacecraft [11]. This now represented a *fast loop* of real-time health assessment, supplemented by a *slow loop* to study the long-term behavior of the spacecraft. Specifically, the *engineering data summarization* is a set of abstractions regarding the sensor telemetry, which is then sent back to ground to provide the missing context for operators. This dual approach has conceptually much in common with the biological reflex and healing approach [12],[13].

2.2. Self-* in NASA's future

The Exploration Initiative (EI) augurs great opportunities for learning more about our universe. Simultaneously it poses great challenges for developing complex autonomous systems that will make the goals of the EI achievable.

We have argued elsewhere that all autonomous systems ought to be autonomic [22],[15]. Future NASA missions will increasingly exhibit autonomicity. This is particularly true of intelligent swarms, a paradigm that seems to offer great potential for future space exploration. The intent is that roles previously performed by a single large spacecraft, will now be performed by a swarm of smaller, less expensive, spacecraft operating autonomously. This permits exploration where single-spacecraft missions simply could not achieve the same goals (e.g., multiple simultaneous observations from different locations); it also offers greater redundancy and protection of valuable space assets. Future swarm missions will include armies of tetrahedral walkers exploring the lunar surface, swarms of miniature spacecraft exploring the Martian surface, in just minutes covering the same amount of ground that the now-famous rovers covered in months. The US Department of Defense is exploring similar technologies for the investigation of extreme environments on Earth, and for under-water exploration.

Along with all the benefits that these intelligent swarms offer, there are also significant difficulties. In particular, since these swarms are intended to learn, as are many other autonomous and autonomic systems, traditional testing approaches are of limited value, yet we must be able to be assured of the correct operation of such a highly-complex mission. Formal methods offer a solution in this respect [7], and the NASA FAST project (Formal Approaches to Swarm Technologies) is researching a suitably tailored formal specification notation [7],[8],[9].

3. Self-* Properties of ANTS

3.1 ANTS

The Autonomous Nano-Technology Swarm (ANTS) mission [3],[4],[5] is a concept NASA mission that illustrates the issues involved in swarm-based systems. The concept involves the launch of a swarm of autonomous pico-class (approximately 1kg) spacecraft that will explore the asteroid belt for asteroids with certain characteristics.

Figure 2 gives an overview of the ANTS mission [21]. In this mission, a transport ship, launched from Earth, will travel to a point in space where gravitational forces on small objects (such as pico-class spacecraft) are all but negligible. From this point, termed a Lagrangian, 1000 spacecraft, that have been assembled *en route* from Earth, will be launched into the asteroid belt. Since each spacecraft has only solar sails to provide thrust for maneuvering, collisions between spacecraft or with asteroids during operations are likely, and 60 to 70% of them are expected to be lost over the duration of the mission. Because of their small size, each spacecraft will carry just one specialized instrument for collecting a specific type of data from asteroids in the belt. As a result, spacecraft must cooperate and coordinate using a hierarchical social behavior analogous to colonies or swarms of insects, with some spacecraft directing others.

To implement this mission, a heuristic approach is being considered that provides for a social structure to the spacecraft based on the above hierarchy. Artificial intelligence technologies, such as genetic algorithms, neural nets, fuzzy logic, and on-board planners, are being investigated to assist the mission to maintain a high level of autonomy. Crucial to the mission will be the ability to modify its operations autonomously to reflect the changing goals of the mission and the large delay and low bandwidth of communications links with Earth. Approximately 80 percent of the spacecraft will be workers that will carry the specialized instruments (e.g., a magnetometer or an x-ray, gamma-ray, visible/IR, or neutral mass spectrometer) and will obtain specific types of data. Some will be coordinators (called leaders) that have rules that decide the types of asteroids and data the mission is interested in and that will coordinate the efforts of the workers. The third type of spacecraft are messengers that will coordinate communication between the rulers and workers, and communications with the Earth ground station.

The swarm will form sub-swarms under the control of a ruler, which contains models of the types of science that it wants to perform. The ruler will coordinate workers each of which uses its individual instrument to collect data on specific asteroids and feed this information back to the ruler, which will determine which asteroids are worth examining further. If the data matches the profile of a type of asteroid that is of interest, an imaging spacecraft will be sent to the asteroid to ascertain the exact location and to create a rough model to be used by other spacecraft for maneuvering around the asteroid. Other teams of spacecraft will then coordinate to finish the mapping of the asteroid to form a complete model [4].

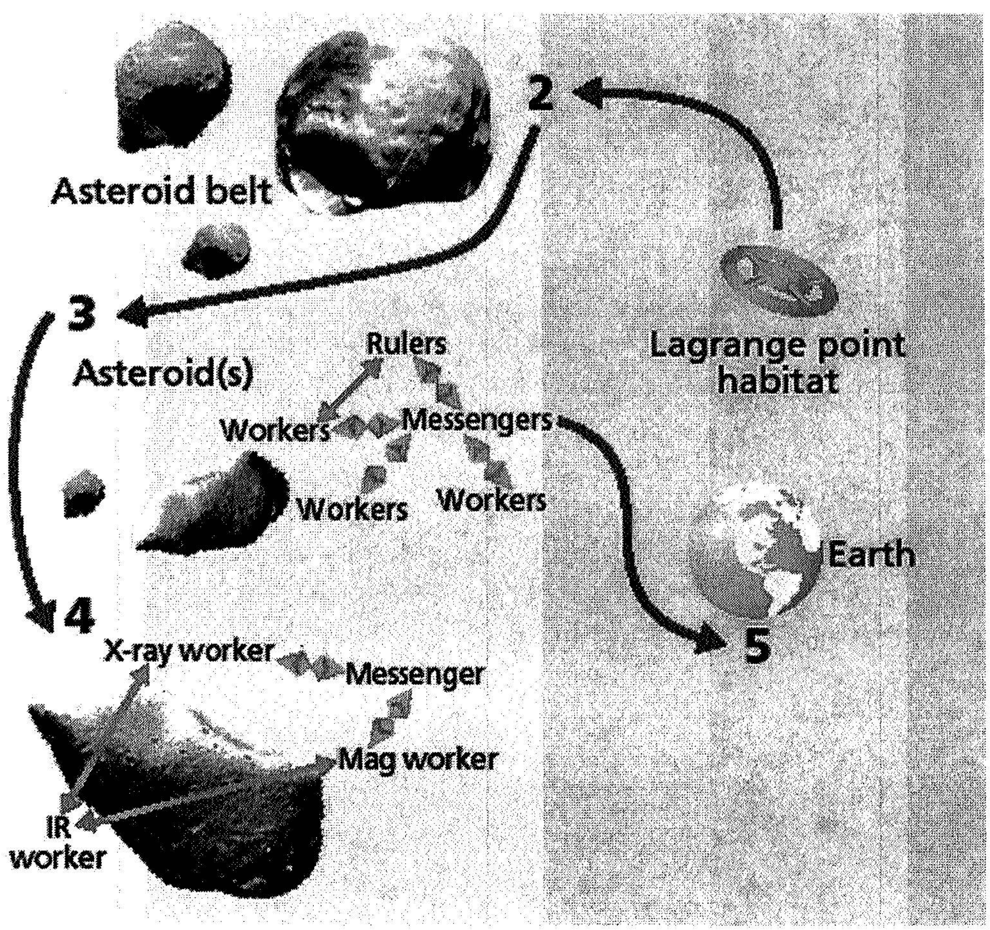

Figure 2 : ANTS mission concept

3.2 Self-CHOP

In terms of self-* properties, the ANTS mission will exhibit almost total autonomy, and will also exhibit many of the properties required of an *autonomic system* [23] (see Figure 1). This section presents some examples of these properties, utilized independently and in cooperation, to achieve selfware [15].

Self-Configuring: The resources of ANTS must be fully configurable to support concurrent exploration and examination of hundreds of asteroids. Resources must be configured at both the swarm and team (sub-swarm) levels, in order to coordinate science operations while simultaneously maximizing resource utilization.

Self-Optimizing: Rulers self-optimize primarily through learning, and improving their ability to identify asteroids that will be of interest. Messengers self-optimize through positioning themselves appropriately. Workers self-optimize through learning and experience. Self-optimization at the system level propagates up from the self-optimization of individuals.

Self-Healing: ANTS must self-heal to recover from damage due either to solar storms or (possibly) to collision with asteroids or with other ANTS spacecraft. Loss of a ruler or messenger may involve a worker's being "upgraded" to fulfill that role. Additionally, loss of power may require a worker to be killed off.

Self-Protecting: In addition to protection from collision with asteroids and other spacecraft, ANTS teams must protect themselves from solar storms, where charged particles can degrade sensors and electronic components, and destroy solar sails (the ANTS spacecrafts' sole source of power). ANTS teams must re-plan their trajectories, or, in worst-case scenarios, must go into "sleep" mode to protect their sails.

ANTS *must* have these properties. As previously discussed, spacecraft large enough for humans would be destroyed immediately, so the mission must be unmanned. If the individual ANTS spacecraft were to be controlled by mission control, there would constantly have to be a human-in-the-loop, which would increase costs and limit the scope of future missions. Moreover, there is a twenty minute delay between sending messages from mission control and their receipt by the spacecraft. Therefore, instructions from Earth could not be received in time to avoid collisions with asteroids or other spacecraft. Analogous constraints are exhibited by real-time mission critical systems in the commercial world.

3.3 Self-protecting / Self-healing

Self-protection and self-healing are inextricably linked. Consider the human body: self-protection necessitates being self-healing. When the skin is cut, cells are displaced, and the body reacts (in time) using cell division to cause scabbing to protect the area until it fully heals via the growth of new cells [16].

Self-protection in ANTS occurs at both the macro (swarm) level and micro (individual spacecraft) level. A great problem for ANTS is potential damage to solar sails or instruments or other subsystems caused by solar storms. Fortunately, advance warning of such activity can be given either by mission control, or by a spacecraft in the mission that constantly watches the solar disc for signs of an impending storm. Upon receiving such a warning, individual spacecraft will take appropriate action —selectively powering down subsystems and reorienting solar panels (which supply power to recharge batteries)—in an attempt to avoid damage.

In addition, individual spacecraft protect themselves by attempting to avoid collisions with other spacecraft and with asteroids in the asteroid belt. Clearly this will not always be possible, and it is anticipated that spacecraft will regularly be damaged and even completely destroyed while engaged in asteroid observations.

Self-protection at the swarm level is required precisely because of this damage. Protecting the entire swarm, and ensuring that the mission continues, necessitates a form of self-healing. When spacecraft are damaged or lost, other spacecraft may take over their roles. For example, if a messenger, used for intra-swarm communication and for communication with mission control back on Earth, is destroyed, another spacecraft may take over that role. If a leader is destroyed, another spacecraft may be promoted to fulfill its role.

Collection of science data requires the use of specialized instruments (x-ray, mass spectrometer, magnetometer, etc.), as shown in Figure 3. Clearly if these are no longer available in the swarm, another spacecraft cannot simply be promoted to cover the loss. . Autonomic behavior at the swarm level would involve self-healing to avoid failure of the mission when losses of particular capabilities reaches a danger point. Swarm-level intelligence and self-awareness and environmental awareness will result in planning and action to cause the factory ship to assemble new spacecraft in a timely manner to compensate for losses.

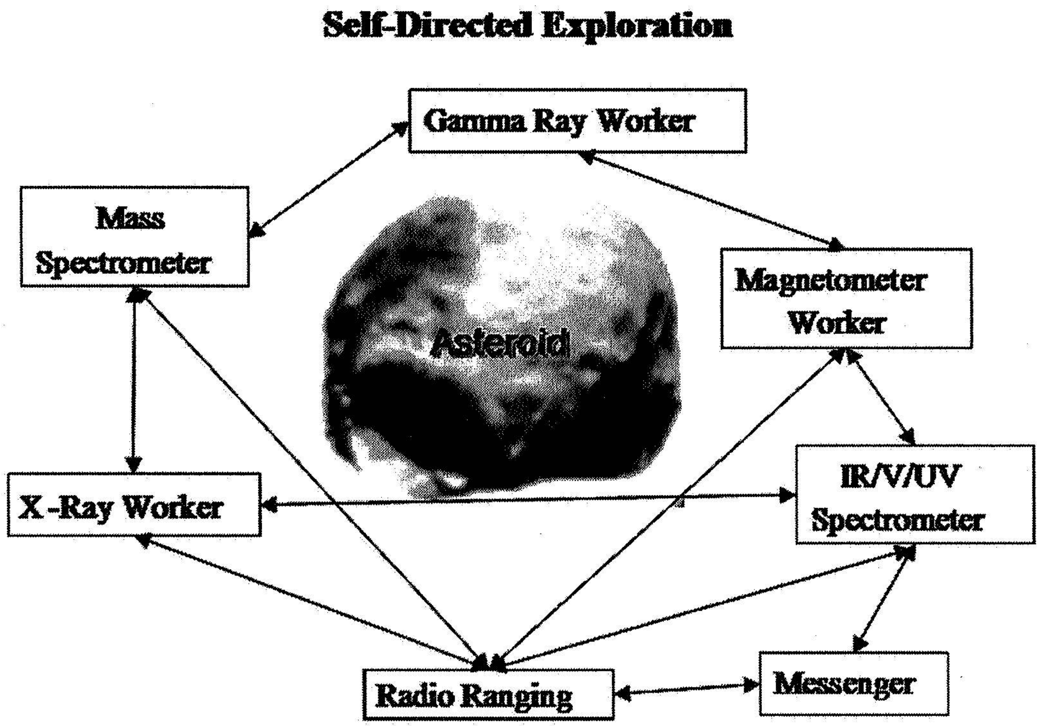

Figure 3 : ANTS coordination and cooperation.

3.4 Self-healing / Self-configuring

Healing, too, requires a certain degree of self-configuration or self-reconfiguration. Again, consider the human body, and damage from a cut. As the skin heals, there is a certain degree of reconfiguration of the various layers, and in particular of the surface layers. Depending on how well this is performed by the body (and on how significant the damage was to start with), a certain degree of scarring may result.

Healing in ANTS, as we have described, may necessitate a spacecraft taking over the role of another. This clearly requires a certain degree of self-configuration of the spacecraft, as it must alter the way it behaves and the work that it must do. For example, a worker that was supposed to assist in making x-ray observations of various asteroids of interest, may be required to take over the role of a messenger. Now its role changes from orbiting a particular asteroid in coordination with other spacecraft equipped with the necessary instruments, to just positioning itself close to the sub-swarm and relaying appropriate messages between the workers and leader, and between the leader and mission control.

Self-reconfiguration may also be required when a spacecraft has been replaced by another. This could be either another from the swarm, which would have to self-reconfigure as described above, or a new craft. In the latter case, the entire sub-swarm would have to self-reconfigure in order to make allowance for the new spacecraft in its orbit, and to allow for it in the leader's decision making.

3.5 Self-configuring / Self-optimizing

After self-configuration, or self-reconfiguration, a certain degree of self-optimization is not strictly essential, but is certainly desirable.

Once again, consider the human body. Damage to skin can result in its becoming rougher and tougher following healing. This is a form of primitive self-optimization that follows the self-configuration of the surface layers. This optimization makes the skin more resilient in anticipation of future possible damage. Similarly, following a hair cut, or trim, the hair eventually grows back longer, and both stronger and thicker. The latter is self-optimization.

In ANTS, self-optimization after self-configuration is desirable, both at the individual level and the (sub-)swarm level. At the individual level, this may involve optimizing use of an instrument, or optimizing the spacecraft's orbit around an asteroid of interest and in relation to other workers in the swarm. At the sub-swarm level, optimization may involve a redistribution of duties, as appropriate, or realignment of workers to ensure sufficient coverage of instrument roles.

3.6 Self-* Summary

We see that self-healing actions can be the result of self-protecting actions. The self-healing actions then may cause self-configuration, which in turn may trigger self-optimization.

From the analysis of ANTS in terms of the four properties of autonomic systems, we see significant overlap in the scenarios. In particular, self-healing is often likely to require self-configuration. Clearly, this will not always be the case—a system where one component is replaced by a homogeneous component will likely not need to be re-configured. In another example, where a worker loses so many of its sensors that it can no longer make science observations, the ruler may give it the goal to take the role of a communications node (messenger agent), and this would entail a degree of self-reconfiguration (and possibly self-re-optimization) by the ANTS team.

Similarly, self-protection may require the addition of components (whether or not they are identical to other components in the system) or replacement of components with others that have better protection mechanisms. This will likely require some degree of re-configuration (in the case of an autonomic system, this will be performed autonomously by the system itself), and possibly some degree of optimization to take advantage of the new components and to ensure that all resources are being used effectively. In the class of systems we are discussing, these actions would represent self-configuration and self-optimization. This may be considered to confirm a view that the Autonomic Systems initiative is concerned with the interaction and interfacing between such self-* properties in contrast to distinct emerging fields such as self-healing systems.

4. ANTS Software Architecture

The early ANTS concept, as described in the previous sections, was envisioned as a (possibly very large) set of agent-based spacecraft such that the agents had the general architecture shown in Figure 4. This architecture illustrates the desire to have a three-level software layout: high-level software to support deliberative reasoning, which involves using software for heuristic reasoning and evolutionary

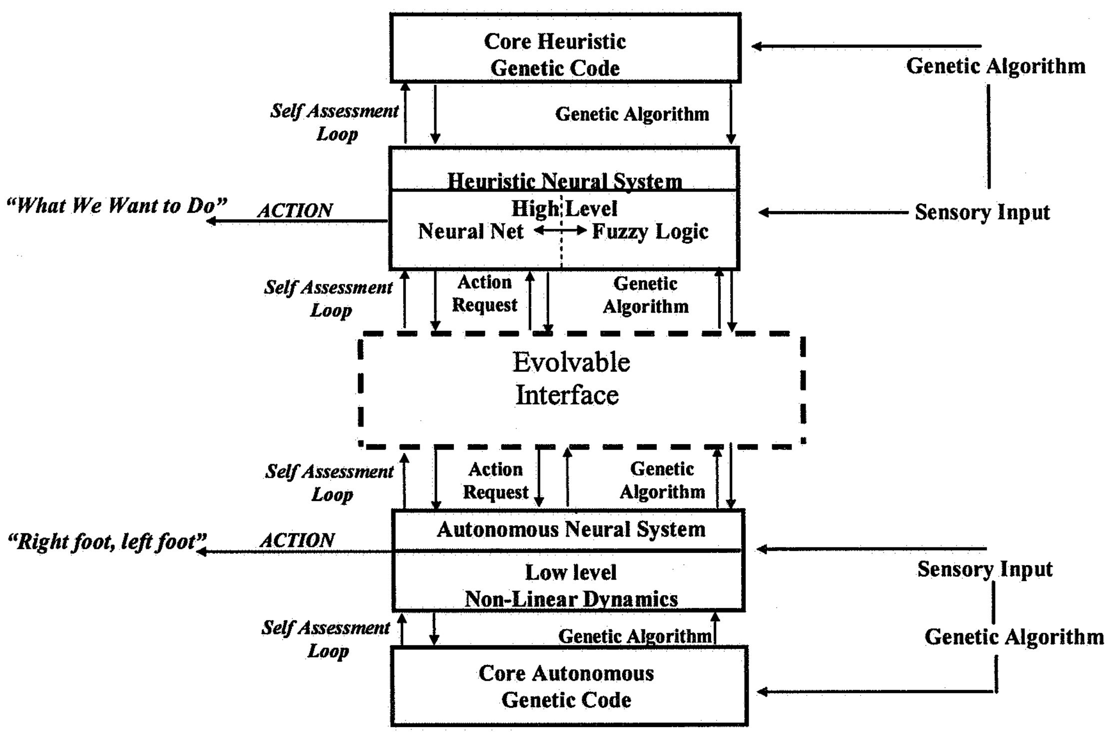

Figure 4: Architecture of a single ANT spacecraft.

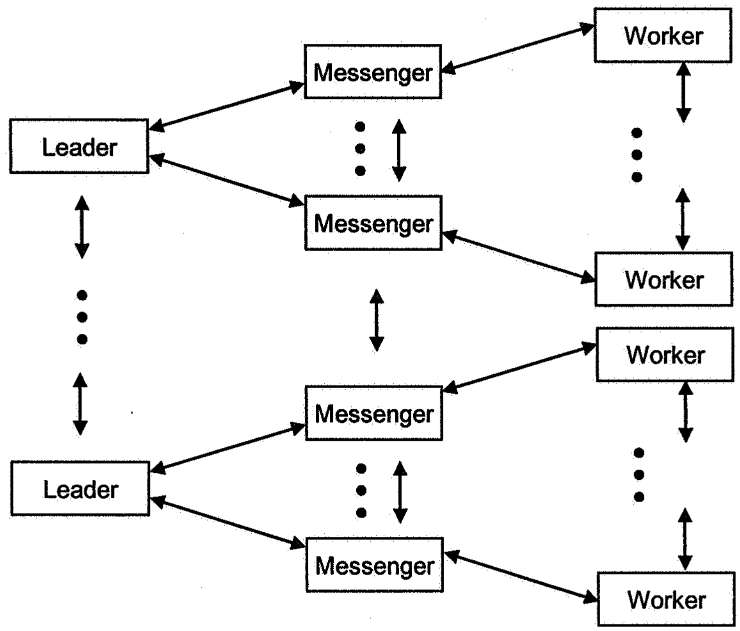

Figure 5: Communications between ANTS Spacecraft

programming constructs; low-level software to handle command/control of spacecraft dynamics; and an interface layer to relate the deliberative reasoning to the low-level control. This three level architecture allows for separation of the high-level and low-level reasoning functions of the spacecraft while allowing interaction between the two levels.

Both the high-level and low-level reasoning will be involved in implementing autonomic properties. The high-level will implement the mission (including reflective) autonomic policies and the low-level will implement the reactive autonomic properties. Mission autonomic policies would include such things as performing optimally and increasing performance (self-optimizing); determining if a spacecraft will come too close to an asteroid if it stays on the current trajectory (self-protecting); given the fact that some of its systems has failed, determining what can be done to become useful to the team again (self-configuring), etc.

Figure 5 shows the communications between the various ANTS spacecraft and how they will exchange information within and between teams. Workers can communicate with each other directly or via messengers. This allows them to coordinate with each other so they don't duplicate exploration of the same asteroids, and also so that they can form new teams to investigate asteroids of interest. In most cases, leaders will communicate with workers via messengers since the workers will be distributed across a large area while prospecting for asteroids. Finally, workers on the same team can communicate between each other. This is needed so they can coordinate or cooperate when jointly examining an asteroid.

5. Related Work

As has been highlighted, traditional approaches to computer-based systems management are often centralized and hierarchical yet today's large-scale systems-of-systems are highly distributed with ever-growing complexity and connectivity resulting in centralized and hierarchical schemes being impractical and unfeasible. As such, MAS (multi-agent systems) have been proposed as a way forward, resulting in much of the work in the MAS community; MAS architectures, incorporating group self-formation, emergent behavior, multi-agent adaptation and coordination, are very relevant [20]. For instance, [2] proposes a software architecture approach, including an architecture description language based on calculus for describing the structure and behavior, a development methodology for evolution, and mechanisms for feedback and change.

6. Conclusions

NASA missions represent some of the most extreme examples of the need for survivable systems that cannot rely on support and direction from humans while accomplishing complex objectives under dynamic and difficult environmental conditions. Future missions will embody greater needs for longevity in the face of significant constraints, in terms of cost and the safety of human life. Future missions also will have increasing needs for autonomous behavior not only to reduce operations costs and overcome practical communications limitations (signal propagation delays and low data rates), but also to overcome the inability of humans to perform long-term missions in space. There is an increasing realization that future missions must be not only autonomous, but also exhibit the properties of autonomic systems for the survivability of both individuals and systems.

This paper has illustrated the need for self-* properties with reference to ANTS, a NASA concept mission. These illustrations with relevance to ANTS highlight that these properties are, in general, interrelated. The paper also illustrated the high level software architecture for such futuristic swarm based self-directed and self-managing systems.

Acknowledgements

This work was supported in part by the NASA Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP) and managed by the NASA Independent Verification and Validation (IV&V) Facility, and by NASA Headquarters Code R. The development of this paper was supported at the 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. This paper is based substantially on [18].

References

[1] R. Murch, *Autonomic Computing,* IBM Press, 2004.

[2] D. Balasubramaniam, R. Morrison, G. Kirby, K. Mickan, B. Warboys, I. Robertson, B. Snowdon, R.M. Greenwood, and W. Seet, A software architecture approach for structuring

autonomic systems. In *Proceedings of the 2005 Workshop on Design and Evolution of Autonomic Application Software* (St. Louis, Missouri, May 21 - 21, 2005). DEAS '05. ACM Press, New York, NY, pp1-7, 2005.

[3] P. E. Clark, S. A. Curtis, and M. L. Rilee, ANTS: Applying a New Paradigm to Lunar and Planetary Exploration. In *Proceedings Solar System Remote Sensing Symposium*, Pittsburg, 2002.

[4] S. A. Curtis, J. Mica, J. Nuth, G. Marr, M. Rilee, and M. Bhat, ANTS (Autonomous Nano-Technology Swarm): An Artificial Intelligence Approach to Asteroid Belt Resource Exploration. In *Proceedings International Astronautical Federation, 51st Congress*, October 2000.

[5] S. Curtis, W. Truszkowski, M. Rilee, and P. Clark, ANTS for the Human Exploration and Development of Space. In *Proceedings IEEE Aerospace Conference*, 2003.

[6] R. Murch, *Autonomic Computing,* IBM Press, 2004.

[7] C.A. Rouff, M.G. Hinchey, J.L. Rash, W.F. Truszkowski and D. Spears (eds.), *Agent Technology from a Formal Perspective,* NASA Monographs in Systems and Software Engineering, Springer, London, 2005.

[8] C.A. Rouff, M.G. Hinchey, J.L. Rash, W.F. Truszkowski, *A Survey of Formal Methods for Intelligent Swarms*, Technical Report, NASA Goddard Space Flight Center, 2005.

[9] C.A. Rouff, M.G. Hinchey, W.F., Truszkowski, and J.L. Rash, Experiences applying formal approaches in the development of swarm-based space exploration systems. *Software Tools for Technology Transfer,* Springer, 2006, to appear.

[10] C.A. Rouff, A. Vanderbilt, M.G. Hinchey, W.F. Truszkowski, and J.L. Rash. Properties of a Formal Method for Prediction of Emergent Behaviors in Swarm-based Systems. 2nd IEEE

International Conference on Software Engineering and Formal Methods. Beijing, China, 26-30 Sept., 2004.

[11] R. Sherwood, J. Wyatt, H. Hotz, A. Schlutsmeyer, M. Sue, Lessons Learned During Implementation and Early Operations of the DS1 Beacon Monitor Experiment, In *Proceedings of the Third International Symposium on Reducing the Cost of Ground Systems and Spacecraft Operations*, Tainan, Taiwan, 1999.

[12] R. Sterritt, Towards Autonomic Computing: Effective Event Management, In *Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW-27),* Greenbelt, MD, USA, December 3-5 2002, IEEE Computer Society Press, pp 40-47.

[13] 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, Alberta, Canada, 22-23 August 2003, pp 433-440.

[14] R. Sterritt and M. Hinchey, Apoptosis and Self-Destruct: A Contribution to Autonomic Agents? In *Proceedings FAABS-III, 3rd NASA/IEEE International Workshop on Formal Approaches to Agent-Based Systems*, Springer-Verlag LNAI **3228**, January 2005.

[15] R. Sterritt and M. Hinchey, Why Computer-Based Systems *Should* be Autonomic. In *Proceedings ECBS 2005, 12th IEEE International Conference on Engineering of Computer-Based Systems*, Greenbelt, MD, 4-7 April 2005, IEEE Computer Society Press, pp 406-412.

[16] R. Sterritt and M. Hinchey, Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions, In *Proceedings EASe 2005, 2nd IEEE Workshop on Engineering Autonomic Systems, at ECBS 2005, 12th IEEE International Conference on Engineering of Computer Based Systems*, Greenbelt, MD, 4-7 April 2005, IEEE Computer Society Press, pp 506-511.

[17] R. Sterritt and M. Hinchey, Autonomic Computing - Panacea or Poppycock? In *Proceedings EASe 2005, 2nd IEEE Workshop on Engineering Autonomic Systems, at ECBS 2005, 12th IEEE International Conference on Engineering of Computer Based Systems*, Greenbelt, MD, 4-7 April 2005, IEEE Computer Society Press, pp 535-539.

[18] R. Sterritt, C.A. Rouff, J.L. Rash, J.L., W.F. Truszkowski, and M.G. Hinchey, Self-* Properties in NASA Missions. In Proc. 4th International Workshop on System/Software Architectures (IWSSA'05) in Proc. 2005 International Conference on Software Engineering Research and Practice (SERP'05), Las Vegas, NV, 27 June 2005, CSREA Press, pp 66-72.

[19] M.A. Swartwout, *Engineering Data Summaries for Space Missions*, SSDL, 1998.

[20] G. Tesauro, D. M. Chess, W.E. Walsh, R. Das, A. Segal, I. Whalley, J.O. Kephart, and S.R. White, A multi-agent systems approach to autonomic computing, *Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pp 464-471, 2004.

[21] W.F. Truszkowski, M.G. Hinchey, J.L. Rash and C.A. Rouff, NASA's Swarm Missions: The Challenge of Building Autonomous Software, *IEEE IT Professional*, September/October 2004, pp 51-56.

[22] W. Truszkowski, M. Hinchey, C. Rouff and J. Rash, Autonomous and Autonomic Systems: A Paradigm for Future Space Exploration Missions, *IEEE Trans. on Systems, Man and Cybernetics, Part C*, May 2006, to appear.

[23] W.F. Truszkowski, J.L. Rash, C.A. Rouff and M.G. Hinchey, Asteroid Exploration with Autonomic Systems, In Proceedings *of IEEE Workshop on the Engineering of Autonomic Systems (EASe 2004) at the 11th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2004),* Brno, Czech Republic, 24-27 May 2004, pp 484-490.

[24] W. Truszkowski, C.A. Rouff, H.L. Hallock, J. Karlin, J.L. Rash, M.G. Hinchey and R. Sterritt, *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, NASA Monographs in Systems and Software Engineering, Springer, London, 2006, to appear.

[25] J.Wyatt, H. Hotz, R. Sherwood, J. Szijjaro, M. Sue, Beacon Monitor Operations on the Deep Space One Mission, In *Proceedings 5th Int. Sym. AI, Robotics and Automation in Space*, Tokyo, Japan, 1998.

[26] J. Wyatt, R. Sherwood, M. Sue, J. Szijjarto, Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment, In *Proceedings 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99),* ESTEC, Noordwijk, The Netherlands, 1-3 June, 1999.