GLENN FINK AND DEB FRINCKE

autonomic computing: freedom or threat?

Glenn Fink is a Senior Research Scientist at Pacific Northwest National Laboratory (PNNL) in the Cyber Security group. His research interests include the effects of technology on the humans who use it.

glenn.fink@pnl.gov

Deb Frincke is a Chief Scientist at PNNL and the founder of the Intrinsically Secure Computing initiative there. Before coming to PNNL she was a faculty member at the University of Idaho, where she specialized in forensic research.

deborah.frincke@pnl.gov

NO LONGER IS THE QUESTION *WHETHER* autonomic computing will gain general acceptance, but *when*. Experts such as Alva Couch expect autonomic computing to be widely used within 10 years [1]. When it does become mainstream, how will autonomics change system administration and corporations, and will the change be for better or worse? The answer depends on how well we anticipate the limitations of what autonomic systems are suited to do, whether we can collectively address the vulnerabilities of autonomic approaches, and whether administrators, companies, partners, and users are prepared for the transition. In this article, we present some design considerations to address the first two issues, and we suggest some survival techniques for the third.

What Is Driving Autonomic Computing?

Computing systems used to have reasonably well-defined borders, both geographically and logically. Today, corporate, educational, and even government computer-based systems coexist in an open mesh of overlapping infrastructures. The complexity of these networks and of the systems that comprise them have given rise to the need for more automation in their configuration and management. Companies rely on a growing number of increasingly complex systems. Increased interconnectivity has led to increased exposure to attacks from around the world. Speed and frequency of attacks have continued to increase exponentially, with malware capable of saturating the Internet in minutes. The rise in numbers, increase in complexity, and need for quicker protective actions all point to a need for additional automation.

At the same time, because companies cannot afford to hire and appropriately compensate the required number of skilled workers to handle this complexity, they have increasingly resorted to offshore outsourcing and commoditization of system administration in recent years to save labor costs. All of this explains why self-managing, intrinsically secure computing is an attractive notion. The idea of systems that can take care of themselves can be either a dream come true or a nightmare, depending on your perspective. There are many stakeholders, including (1) owners of the systems looking for savings in time or money,

(2) system administrators who face the tension between enjoying the freedom from drudgery that autonomic systems promise and the worry that this freedom will ultimately make their jobs unnecessary, (3) users who are looking for increased work efficiency from the systems, (4) business partners who share the networks, resources, benefits, and risks of autonomic systems, (5) legal counsel who will need to sort out responsibility and liability in the new world, and, last but not least, (6) attackers who will view autonomic systems as either an effective barrier to their access or as a great way to bend the systems to their will automatically and invisibly.

Autonomic Computing: Freedom?

Autonomic computing derives its name metaphorically from the operation of the human autonomic nervous system. Autonomic systems are intended to be self-managing, keeping mundane details of operations hidden from the operator while increasing predictability, speed of response, and reliability. The idea of pervasive computing, where tiny networked computers embedded in the environment will constantly adjust to our needs, requires autonomic computing. IBM has defined the crucial elements of autonomic systems in their autonomic computing manifesto [2]. In a following paper, Kephart [3] describes an inspiring vision of what autonomic computing will do for technology and society.

Autonomic computing promises a more natural boundary around the complexity of the systems we live with today. People don't generally consciously interact on the cellular or atomic level with others; we interact at the natural boundary that separates one person from another. As the internal complexity of computational systems increases, a new boundary between computers and human administrators becomes necessary. People shouldn't have to fiddle with the vagaries of configuration files any more often than they should have to modify their kernel source code. Looked at this way, autonomic computing is a natural and necessary way to internalize and compartmentalize complexity.

If the vision of autonomic computing becomes reality, systems will be self-managing so that the administrator won't have to be summoned on an emergency basis nearly as often. Kephart's autonomic systems may not only patch themselves but also automatically seek updates, new software, and better configurations that will give them better performance (Figure 1). They will find workarounds when services they depend on break. Autonomic systems will negotiate service agreements with external systems, using and providing services whenever it is consistent with system goals, and they will protect themselves when they sense that they are under attack.

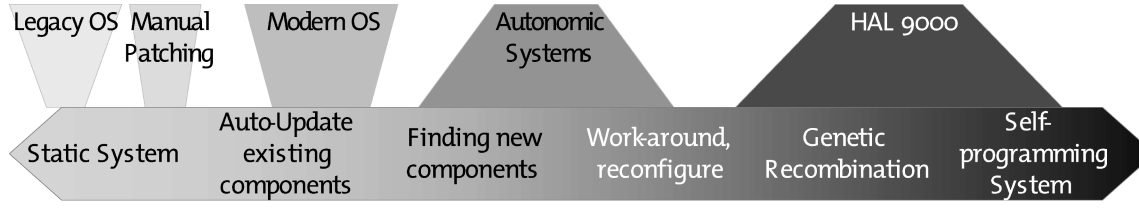

FIGURE 1: THE SELF-MAINTENANCE CONTINUUM

But here we begin to see that the idea of autonomic computing has gone far beyond the metaphor offered by the human autonomic nervous system. There seems to be a subtle difference between what we mean by "autonomic" and the meaning of "autonomous." "Autonomic" traditionally means

that normal operations will continue without interruption or need for intervention. But "autonomous" goes beyond normal operations, implying a sense of self, needs that must be met, and freedom to act. In humans, autonomy implies individual actions springing from intelligence and a will free from conscious external coercion. Kephart's vision for autonomic computing reaches beyond the accepted definition of "autonomic" into territory much better described by the word "autonomous." Is this more freedom than we want our systems to have? Or is it the freedom they *must* have if they are to be what we need them to be?

Autonomic Computing: Threat?

The very freedoms that we expect to gain from autonomous technology may also be cause for concern. An autonomic system providing services to humans or other systems is different from an autonomous system working as an agent on behalf of a human or organization. They are different in the degree of independence they are afforded and in the way accountability and control are administered. Millions daily give eBay's auction software the permission to commit to spending funds within preset limits. This requires eBay users to seriously consider beforehand whether they are willing to buy the products they bid on. Once the user's software agent wins the auction, the user is obligated to pay. Similarly, when allowing systems to autonomously seek new software, enter service agreements, and protect themselves, we must be sure we are able to bound the consequences and that we are willing to pay the potential price.

If autonomic systems are given enough freedom to act and interact, the overall infrastructures may behave in emergent ways we cannot predict. Autonomy means that decisions will be made locally, but the emergent qualities of complex systems demonstrate that local decisions can have far-reaching and unpredictable global results. As Kevin Kelly so poignantly puts it, "Wherever the word 'emergent' appears, there disappears human control" [4]. Automation is one thing, but autonomy is quite another. Once systems become autonomous, by definition they will have a "mind of their own." We will be asking software to make decisions for us that have traditionally been entrusted to humans alone. This is monumental in stand-alone systems, but in the world of overlapping corporate infrastructure boundaries and numerous (and potentially conflicting) stakeholders, the implications are astounding.

Consider a scenario where a self-managing system is empowered to negotiate with an ISP about the amount of bandwidth the owner's commercial storefront Web servers require. The policy enforced by the self-managed system may include "spend the least amount of money that supports peak anticipated access rates based on trends over the previous seven days." If there is a surge in accesses, the system will make the decision to commit corporate dollars to expand bandwidth. If, instead, the access demands have reduced, it may decide to "save money" by reducing the amount of ISP service based on lower access rates and slow sales. Notice that the policy as written fails to take into consideration peak demands such as holidays or the sudden popularity of an item.

The previous situation was fairly straightforward, but weightier agreements between suppliers and purchasers could have more far-reaching economic effects. When autonomous software agents are making the deals, who is legally bound to meet the terms of the agreement, and how can all parties

be made aware of their obligations? How do the humans renegotiate a promise that their automated systems made, and what are the legal, societal, and logistical implications of such intervention? Who decides how much authority to delegate to these systems, and how much can a partner trust the authority delegated? Service-level agreements made by autonomic systems will have economic impacts that imply risks for multiple stakeholders. Before an organization decides to trust a promise made by an autonomic system, it will have to be able to verify that the system's word is worth the risk [5]. And when things *do* go wrong, how will organizations debug the policy language that allowed the problem to occur?

Another aspect of self-management that must be considered is the continuum of potential response to threat—sometimes referred to as active defense or active response [6]. Different organizations have different approaches to defending their infrastructure. Thus, it is reasonable to assume that different defensive policies will be enacted. The impact of executing divergent, autonomously enforced policies within a mixed cyber infrastructure is difficult to express [7]. Consider a virtual community such as the open science grid computing community. Suppose one organization has a policy that it disconnects from the larger network when it detects wormlike activity. What does this do to another organization that may have real-time dependencies on the first organization's services? How about an organization whose policy is to avoid negative effects on others? Given the overlapping nature of dependencies in cyber infrastructures, it will be difficult or impossible to automatically (or any other way) verify that there is an acceptably low level of negative consequences stemming from a policy change. Will this cause autonomous systems to be paralyzed into making no decision at all? Clearly, policies must not be conceived in a vacuum without a consideration of their wider effects on other organizations.

The existence of intelligent attackers brings to light the need to verify that the autonomic system hasn't been subverted and is still acting within the intentions of its owners. Autonomic computing may lead to true complex-adaptive systems whose ultimate behavior is very difficult to predict from initial conditions [8]. One thing about attackers is certain: They will learn to adapt to autonomic systems and to bend them to their will. Another certainty is that attackers will want to keep their activities secret. Autonomic computing brings to attackers the promise that no human will be watching. There must be a way to allow human administrators the ability to inspect the operation of the system and verify that it is still on their side.

Computer programs don't go to jail. They aren't afraid of losing their jobs. But when things go wrong and the cost of a bad decision is estimated, it is certain that some human(s) will pay the price [9]. If system administrators are to be held responsible for the decisions of autonomous systems, then both the responsible persons and their employers will want to ensure that there is the possibility of some human awareness and intervention in these decisions. At the same time, humans will not want to be involved in every decision—that would make autonomous systems pointless. We will need to rethink the meaning and limitations of trust in the new world of autonomous systems.

Design Considerations for Autonomic Systems

The open issues for self-managing systems go well beyond the scope of a single article, and each issue has implications for design. There are, however, a few considerations that we can suggest to manage at least some of the

risks this article has broached. We see five broad areas where design guidelines would help:

- Awareness: Allowing human insight into system activities via cyber analytics.
- Management: Enabling human influence over distributed autonomous systems via hierarchical design.
- Attribution: Certifying the correctness of independent actions of the system.
- Integrity: Ensuring that the system has not been subverted.
- Limits: Stating clearly what the autonomous system may and may not do.

AWARENESS

The emerging discipline of *cyber analytics* (the application of visual and predictive analytics to understanding the workings of computer infrastructures) holds great promise for humans to gain and maintain awareness in the face of overwhelming amounts of data. But awareness alone is not sufficient to control large distributed systems. Coupling autonomous control with the situational awareness of cyber analytics will allow humans the ability to manage an unprecedented number of computer systems. We have shown that human oversight is essential, even if the systems work flawlessly. Thus we propose that designers of autonomic systems keep human awareness and management in mind even as they design their systems not to require human intervention.

MANAGEMENT

Humans need a single point of influence to have multiple points of effect within their systems. We only have at best ten fingers and one brain, but we need to be able to exert consistent influence over large numbers of heterogeneously configured systems simultaneously. This is the point of policy (and, by extension, autonomic computing). But centralized control is not the answer. Large, centralized artificial intelligence becomes brittle and computationally intractable for distributed, highly constrained problems. Other solutions, such as swarming intelligence, are useful for such problems [10], but they are very difficult to understand and control. We suggest that hierarchical deployment of a variety of intelligent agents [11] will provide both the single point of influence and the multiple points of effect needed. The highest-level agents can translate the activities of a swarm of "digital ants" to the human and can implement the user's policy via lower-level agents. We believe that a hierarchy of varied intelligent agents will increase human influence while reducing the need for human intervention.

AUTHORIZATION

Actions that involve agreements across organizational boundaries require some way to distinguish the activities of the autonomic systems from those of the humans who are ultimately responsible for them. Successful delegation of high-level duties will require separate digital identities for the human supervisor and the autonomic systems and digital reputation accounts for the autonomic systems. Systems could be "punished" or "rewarded" via feedback from other systems and from their owners, enabling machines to learn from their mistakes. Similarly, if the system acts outside its authority, its own signature would be on the agreement, allow-

ing the responsibility to be properly allocated. Attribution of responsibility is also a key to debugging these complex systems. These mechanisms certainly don't solve the technical, legal, and social problems raised by an interacting society of autonomous systems, but they do lay some groundwork that may make such problems solvable.

INTEGRITY

Autonomic computing will spare humans many details they simply have no time for, but attackers can turn this information hiding to their advantage. Additionally, adaptive systems will be able to change the way they behave, and possibly even their own behavioral parameters. Autonomic systems must act in a predictable manner, even when portions of their systems are actually subverted by attackers. Thus, it is important that system designers provide an ability to make sure that autonomous systems are acting in accordance with stated policy and the intent of their owners. Cryptographic methods may be employed in a number of ways to check the integrity of autonomous agents, while static code verification may help assure adherence to policy.

LIMITS

Another important facet of autonomous system assurance is making policy limitations expressible in terms that are human-understandable, complete, and translatable down to the machine instruction level and back. Natural language is ambiguous and hard to parse. XML is "human-readable" only in the sense that it is expressed in printable ASCII. Much research in policy languages remains to be done to achieve assurance that policy is expressed correctly and can be executed as expected.

At Pacific Northwest National Laboratory (PNNL), we are designing autonomic systems for computer security. Our Intrinsically Secure Computing [12] initiative embodies three core principles: Trustworthy Engineering, Adaptive Defense, and Appropriate Response. We believe that at least some of what we are learning in the security arena is applicable to autonomic systems in general. As part of this initiative, we are building a human-agent defense grid that will enable greater levels of autonomous behavior in system defense without losing sight of the fact that humans are ultimately responsible for the activities of their systems. We intend to apply our findings broadly to autonomic systems of other sorts in the hope that these systems will be a blessing and not a curse to the administrators who use them.

Conclusions: Thoughts on Life in an Autonomic World

In conclusion, let us consider what the advancement of autonomic computing will mean for system administrators as a profession. Do self-managing systems pose a threat to system administrator job security or a promise of increased job satisfaction? This question has been asked in many forms. In a panel discussion at ICAC 2006 [13], Kumar Goswami stated that barriers to autonomic system administration might arise from concern over lack of trust in the system, loss of hands-on control, and fear that automation would eliminate the system administrator's job.

In the 1820s, during the Industrial Revolution, debate on the "Machinery Question" was even hotter than it is now. People feared that machinery

would replace human labor and result in widespread unemployment and poverty. This fear proved to be short-sighted because, after automation, a skilled workforce was needed to make repairs and manage machinery [14]. Factory jobs that were already highly mechanical were taken over by machines, but new jobs that required human capabilities were created. Industry expanded, with machines doing more than humans ever did before. Human workers had to develop new skills and gain education to remain competitive (see Figure 2), because the rates of job automation and job creation were not tightly tied. While ours is a different age, many similar forces are in play, so the comparison has merit.

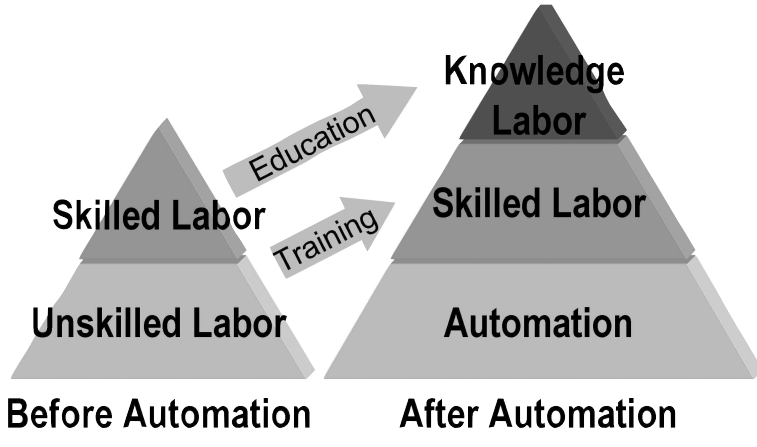

FIGURE 2: EFFECTS OF AUTOMATION ON THE HUMAN WORKFORCE

With the advent of autonomic computing, human system administrators will definitely not become obsolete. However, autonomics will significantly change the way administrators work. Some of the push toward autonomic computing comes from corporations that are unable to hire enough qualified system administrators right now at rates they can afford. We expect that autonomic systems can increase productivity and produce the corporate capital needed to alleviate the existing problem, not put people out of jobs. But this result will require time for society to find a new equilibrium.

During rapid changes in technology, "[p]erhaps the best skill . . . is how to learn (and unlearn) quickly" [14]. Historically, technological leadership has always been hard to sustain, and this will be as true for the technocrats of today as it always has been. Arguably, Britain lost the technological leadership it enjoyed during the first part of the Industrial Revolution because it clung to the products and processes that had made it great rather than adapting to new ways. This is a lesson that all technologists would do well to note: Adaptability is the best defense in a changing world.

Autonomic systems will take over the "plumbing," enabling humans to work at the higher, policy level. As long as humans are responsible for information systems, administrators will be needed to translate operating policies and business practices into clear, complete, and consistent machine instructions. The only difference is that machines are learning to understand language closer to the way humans are accustomed to expressing it. Machines will also learn to find inconsistencies in policy and ask intelligent questions about them. But there is still a place for human system administrators to act as go-betweens for management and machines. And there will always be a place for highly skilled individuals who can "look under the hood" when things *do* go wrong.

Management is good at comprehending business objectives. Machines are good at executing programs. System administrators must learn to translate

business logic into policy logic for machines. System administrators will have to interface well with both humans and machines. The first generation of autonomic systems is already being put into use, and wise system administrators will learn how they work early on. There are two reasons for this: (1) because the technology will be demanded by their employers sooner or later, and (2) so that administrators can join the nascent autonomic computing design dialogue.

Some system administrators see autonomic systems as a threat to their employment rather than liberation from drudgery. We believe this is an unfounded fear if administrators are willing to make some adjustments: Let go of the need to control the details of low-level configuration, trust but verify, and learn to understand the business needs that will drive policy.

System administrators should consider autonomic computing to be a promotion to a position of more responsibility and respect. Administrators will become management consultants rather than technicians. No technology can live without highly skilled troubleshooters, but the numbers of these professions will likely dwindle as autonomic systems improve. We believe the overall number of system administrators will probably not decrease, because the number of machines being fielded and placed on the Internet is increasing exponentially. No matter how smart the machines get, there will always be a place for intelligent, adaptable, human system administrators.

REFERENCES

[1] A.L. Couch, "The Future of System Administration: How to Stop Worrying and Love Autonomic Computing," Presentation at LISA '06. Available at http://www.usenix.org/events/lisa06/tech/slides/couch.pdf.

[2] IBM, "Autonomic Computing: IBM's Perspective on the State of Information Technology," 2001. Available from http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf.

[3] J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," *IEEE Computer*, pp. 41–50 (January 2003). Available at http://www.research.ibm.com/autonomic/research/papers/AC_Vision_Computer_Jan_2003.pdf.

[4] K. Kelly, *Out of Control: The New Biology of Machines, Social Systems and the Economic World* (Perseus Books, 1994). Available at http://www.kk.org/outofcontrol.

[5] E.A.R. Dahiyat, "Intelligent Agents and Intentionality: Should We Begin to Think Outside the Box?" *Computer Law & Security Report*, 22(6): 472–480 (2006). Available at http://dx.doi.org/10.1016/j.clsr.2006.09.001.

[6] S. Caltagirone and D.A. Frincke, "The Response Continuum," *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 15–17 June 2005* (IEEE Press, 2005). Available at http://www.classstudio.com/scaltagi/papers/professional_papers/westpoint05v2_2.pdf.

[7] D.A. Frincke, A. Wespi, and D. Zamboni, "From Intrusion Detection to Self Protection," Computer Networks, 2007.

[8] J.H. Holland, *Hidden Order: How Adaptation Builds Complexity* (Addison Wesley Longman, 1995).

[9] M. Scher, "On Doing 'Being Reasonable,' " *;login:* 31(6): 40–47 (2006). Available at http://www.usenix.org/publications/login/2006-12/pdfs/scher.pdf.

[10] H.V.D. Parunak, " *'Go to the Ant':* Engineering Principles from Natural Multi-Agent Systems," *Annals of Operations Research* (Special Issue on Artificial Intelligence and Management Science), 75: 69–101 (1997).

[11] H.V.D. Parunak et al., "Hybrid Multi-Agent Systems," *Proceedings of the Fourth International Workshop on Engineering Self-Organizing Systems (ESOA '06)* (Hakodate, Japan: Springer, 2006).

[12] Pacific Northwest National Laboratory, Computational & Information Sciences Directorate, "Intrinsically Secure Computing" (2006). Available at http://cisd.pnl.gov/secure_computing.stm.

[13] Panel Report: O.F. Rana and J.O. Kephart, "Building Effective Multivendor Autonomic Computing Systems," *IEEE Distributed Systems Online* 7(9) (2006).

[14] J. Mokyr, "Are We Living in the Middle of an Industrial Revolution?" *Federal Reserve Bank of Kansas City Economic Review*, pp. 31–43 (1997). Available at http://12.154.48.181/PUBLICAT/ECONREV/pdf/2q97mokr.pdf.