

MARK BURGESS

## autonomic computing—the music of the cubes



Mark Burgess is professor of network and system administration at Oslo University College, Norway. He is the author of *cfengine* and many books and research papers on system administration.

*Mark.Burgess@iu.hio.no*

**AUTONOMIC COMPUTING IS A PHRASE** invented by IBM to sell mostly existing technologies for automation in the marketplace. IBM has led a corporate procession away from centralized management technologies toward self-maintenance. They were among the first companies to see light at the end of the system administration tunnel.

Over the past few years, “network management” (i.e., system administration) research has been taking giant strides in a random walk of hunting and gathering, taking and improving upon ideas about automation, some of which have been known for thirty years or more. Researchers have been putting these ideas into some kind of context or practice, and the activity has been substantial. Some of those ideas have been developed and used in the USENIX community through research in configuration management and policy-based automation. Some of this development genuinely comes from the big corporations, including IBM, HP, Motorola, and now the academic EMANICS Network of Excellence in Europe. We’re all good friends, ignoring the hype and working on the real issues, while the marketing departments justify the work with colorful banners.

So, what does autonomic mean? The name gives us a clue: it is formed from two Greek roots: *autos*, meaning “self,” and *nomos*, meaning “the law.” In other words, it is about self-governance—or, as it is sometimes paradoxically expressed, self-management. As such, it brings together three ideas: automation, decentralization, and autonomy of decision. If you like, it is a self-help program for computers.

While the corporate autonomic computing campaign has been, for end users, more of a triumph of XML style over substance, the underlying idea has been taken very seriously by multitudinous corporate and academic researchers, anxious to see their work realized in the marketplace.

But if we have known about these ideas for a long time, why the big song and dance now? To understand the whys and wherefores one needs to take a step back from the servers to look at corporate politics and produce, because the bandwagon is a commercial development, not a research development.

The traditional view of system management, in many large organizations, has been dominated by

the monopolistic telecom dream of world domination. Everyone has to do as the service providers say; don't they? So the telecoms have believed they could simply create the usual management position to monitor systems and issue instructions to make magic happen. Everything would then be "managed" and hence hunky-dory. The IETF and DMTF, formed mainly from the genes of telco leviathans, gave birth to TMN, SNMP, NETCONF, etc., based upon this belief in the power of central authority.

But the thing that none of them could ignore is that there are, in reality, now more than five computers in the world. In fact, there's one in every pocket and on every wrist, not to mention desk, car, aircraft, etc. ad nauseum. Like it or not, computer *ownership* is now utterly decentralized, and information privacy rests on the lips of every global citizen. There is a feeling that one should not surrender to authority. Neither these personal computing devices nor their owners are looking to open their hearts and minds to a just and angry trunk provider. The spirit of the times is rather to be found in the rustling of the leaf nodes.

Today the computing industry is still having a hard time letting go of this central-command mentality. Companies such as IBM and HP are studying "control theory," or self-regulation, as it is known from electrical engineering, in the same breath as they speak of autonomics. No one quite believes it all yet—their corporate consciences are whispering to them that making machines run themselves is going just a little too far. Who will be pushing the buttons?

It is somehow reminiscent of firewalls and intrusion detection. Do you centralize security or try to avoid bottlenecks?

Let's isolate systems to keep them safe!

What do you mean, you need to talk to them?

All right, we'll introduce a manager in the network to take care of security: a firewall will keep us safe!

What do you mean, the host opened up a back door through a wireless LAN?

Immediately people think management means a centralized point of responsibility. Does it? Of course it doesn't. Try explaining the schools and shoals and swarms of fish as they swim in perfect coordination, or colonies of ants. Do birds fly around just dying for promotion into a magnificent managerial role? Let me lead the flocking way! No, they get along fine without having to invent the leash.

Don't you need the management privilege to violate private boundaries to get that all-important deep knowledge and perspective?

Ants do not rely on satellite communications or radar to navigate. They manage vastly complex tasks from a low-cost local perspective, by interacting cheaply with immediate and local information through smell, not through gigabytes of collected data. Hurrah for the war against tera! We can be economical if we don't try to overmanage or micromanage.

Managers think they can command excellence, but systems never achieve crystalline perfection. They are not quite ecological slime, but they are getting close! Have you seen the pictures of the Internet lately? Maintaining software and systems is something like trying to solve hundreds of Rubik's cubes—just when you think you've fixed a single face, you push all the other faces out of synch.

So is autonomics important? Yes, it is, and especially one aspect of it that truly is new. It is not automation nor yet distribution. I would say that

*autonomy* is the key challenge to the future of computer management or system administration. We have to unlearn what we are used to and ask: What is a system? Where are its boundaries? Who has political control over it? How can we get users to behave nicely when their private devices come together?

My own work, around cfengine, has always placed autonomy as an important principle, more out of a sense of belief in local adaptability than of an awareness of the onset of the future. But it seems clear that it was a lucky guess as far as the proliferation of personal computing is concerned.

You might not really want to speak of *management* when it's just one mobile phone we're talking about, but *maintenance* is a reasonable word. In an autonomous world, the process of fitting in with our neighbors will borrow more from symbiosis and swarming than from command and strike force. In the end the swarms of unfinished Rubik's cubes might fall into similar patterns simply autonomically. And that curiously imperfect state of consistency might be the best we can hope for.

## Thanks to USENIX & SAGE Supporting Members

Addison-Wesley Professional/ Prentice Hall Professional	Interhack
Ajava Systems, Inc.	The Measurement Factory
AMD	Microsoft Research
Asian Development Bank	MSB Associates
Cambridge Computer Services, Inc.	NetApp
EAGLE Software, Inc.	Oracle
Electronic Frontier Foundation	OSDL
Eli Research	Raytheon
FOTO SEARCH Stock Footage and Stock Photography	Ripe NCC
GroundWork Open Source Solutions	Sendmail, Inc.
Hewlett-Packard	Splunk
IBM	Sun Microsystems, Inc.
Infosys	Taos
Intel	Tellme Networks
	UUNET Technologies, Inc.

It is with the generous financial support of our supporting members that USENIX is able to fulfill its mission to:

- Foster technical excellence and innovation
- Support and disseminate research with a practical bias
- Provide a neutral forum for discussion of technical issues
- Encourage computing outreach into the community at large

We encourage your organization to become a supporting member. Send email to Catherine Allman, Sales Director, [sales@usenix.org](mailto:sales@usenix.org), or phone her at 510-528-8649 extension 32. For more information about memberships, see <http://www.usenix.org/membership/classes.html>.