GA Fink
PNNL-SA-57637

Who's the Boss?
Autonomics and New-Fangled Security Gizmos with Minds of Their Own

Glenn Fink

Glenn.Fink@pnl.gov

14 November 2007

USENIX LISA 2007

Battelle

Pacific Northwest National Laboratory
Operated by Battelle for the U.S. Department of Energy

Agenda

- Introduction: Me and AC
- State of the Art: Autonomics in use today
- Love/Hate Relationships with Autonomics
- Autonomics [Wish | Fear] List
- What about my job?
- Autonomics Future Directions
- Conclusion

Primary conclusions of my study

- There are no existing AC systems (yet)
  - We are at least 5-10 years away from anything like the IBM vision (Chess & Kephart)
- There are plenty of automated subsystems that have AC characteristics
  - But they don't interoperate
  - Level of autonomy varies greatly
  - Not based on open standards
- There will **always** be a need for human system administrators
  - But their duties will change substantially

Who am I?

- My background is information visualization for computer security (people.cs.vt.edu/~finkga)
- My orientation is human-centered computing:
  - Yes, but will it work reliably for unreliable humans?
  - Nice, but will it work under weird circumstances?
  - Very slick, but will people want it?
  - Cool, but will people actually be better off with it?
- My experience:
  - I worked 15 years as a computer scientist for the Navy and received my Ph.D. from Virginia Tech
  - Currently I am a senior research scientist at Pacific Northwest National Laboratory (PNNL, www.pnl.gov) in Richland, Washington
    - Adaptive Systems Focus Lead for PNNL's Information and Infrastructure Integrity Initiative
    - Researching how adaptive agent technologies can be used to implement autonomic defense across infrastructures

Where am I getting my information?

- Research into self-healing systems for large interdependent computer infrastructures
- Study of autonomic computing academic literature
- Interviews with colleagues:
  - System administrators
  - Autonomics researchers from various institutions
  - Computer security analysts
- Other information that rubbed off on me

Autonomic Computing Defined

- IBM was first to back a cogent, corporate vision for autonomic computing (AC):
  - 60% Self-Configuring: deployment of new components or changes with minimal human intervention
  - 25% Self-Healing: detect improper operations and initiate corrective action without disrupting system applications
  - 10% Self-Optimizing: automatically maximize resource allocation and utilization to meet end-users' needs
  - 40% Self-Protecting: detect hostile behavior and take autonomous actions to mitigate attacks and general failures
- Source: http://www.ibm.com/autonomic/pdfs/Autonomic_Computing_Overview.pdf

Levels of AC Maturity

www.ibm.com/autonomic/pdfs/Autonomic_Computing_Overview.pdf

- **Basic:** Manual analysis and problem solving
- **Managed:** Centralized tools, manual actions
- **Predictive:** Cross-resource correlation and guidance
- **Adaptive:** System monitors, correlates and takes action
- **Autonomic:** Dynamic business policy-based management

> *"Autonomic computing is not an overnight revolution in which system-wide, self-managing environments suddenly appear. Rather, it is a gradual evolution in which **new technologies, methodologies** and **best practices** are implemented using IT Infrastructure Library* (ITIL)*-aligned flows."*

Definitions of AC

- It's a continuum:
  - Automated systems can always be viewed as managing themselves in accordance with specifications, but it's a question of degree.
  - AC is a direction, not a goal
  - Hard to draw the line between Non-AC and AC
- But evolutionary changes can have revolutionary effects
  - As systems gradually prove themselves competent to take on certain tasks and behave more independently, the implications can end up being huge.
- Automation is lower-level, autonomics interacts with the environment
- Automation is handling a designed task well; autonomics is handling all the unexpected things that arise.
- AC is more contextual than simple rule-based reasoning
  - Implies some “social” awareness among systems

The Future is Certain, But the Path is Unclear

- Demand for IT professionals outstrips supply 18:1
  - Implication: More jobs and higher salaries?
  - See: “If there's an IT skills shortage, where's my job?” http://www.itworld.com/Career/1827/070904job/pfindex.html
- Growth of IT infrastructure is exponential
  - Implication: Market demand drives unsustainable rates of increase in computing power and complexity
    - Software crisis: Over budget, beyond schedule, buggy, unmaintainable
    - Hardware crisis: Volume overtakes reliability: Death by Moore’s Law (http://www.scidac.gov/Conference2007/presentations/gibson_pres.pdf)
    - Education crisis: Few qualified people for high-tech jobs; overseas workers are disproportionately well-educated
- Cost of IT personnel is prohibitive
  - Implication: Automate, outsource, or die
    - Thousands of able-minded Asians want your job!
    - And they’ll do it cheaper (see automotive industry)

State of the Art: Autonomics in use today

- Autonomics is currently in the research stage. Current work falls primarily into two categories:
  - Vertical systems that are autonomous but narrow
    - Port Scan Attack Detection (PSAD)
    - Automatic software updates
    - Linux-HA

VERTICAL

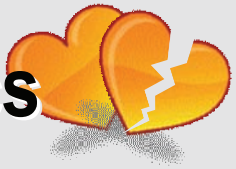
HORIZONTAL

  - Horizontal systems that provide broad automation without real autonomy
    - IBM Tivoli Intelligent Orchestrator (TIO)—Tivoli is an actuator for AC
    - cfEngine, Puppet, etc.—Automation for system administration

Purist vs. Pragmatist

- Purist
  - Maintenance is the dominant cost in the long run
  - Autonomic policies should be centrally defined and best practices pushed or pulled to clients
  - Constrain administrator activities to avoid conflicting with the autonomic processes
  - Seems to flourish in environments with lots of similar machines
- Pragmatist
  - Downtime is the dominant source of cost (both money and jobs)
  - Decentralized rather than client-server
  - Allow humans to cobble a quick-fix now and fix it properly later
  - Flourishes in heterogeneous, high-pressure environments
- We need a third way
  - Ensure that pragmatic fixes feed back into an established, inspected, trusted library of practices
  - "Open source" autonomics with "social" learning
  - Human supervision of autonomic computing

Love/Hate Relationships with Autonomics



Love	Hate
Saves system administrator labor	Hides information
Accomplishes individual tasks well	Fails to account for interactions between systems (story time)
Can handle tasks previously only humans could do	Doesn't know when to ask for help
Anticipates user action (e.g., web prefetching)	May mask human activity (or serve as an excuse)
Handles known errors well	Can't handle completely new problems
Could result in huge efficiency gains	Could violate unstated constraints
Faithfully does what humans might neglect (backups?)	May waste or duplicate effort

Autonomics Wish List

- Make it like a junior sysadmin
  - Investigate and report
  - Open-ended tasks
  - But don’t hide information; let me in
- Robust handling of real-world situations without instruction/supervision
- Communicate like a human
  - Just enough (and not too much) detail in reports
  - Natural language processing for instructions and reports

Autonomics Fear List

- Will AC systems know when to ask for help?
- How do you verify self-configuration is good?
- Can I trust that self-update is getting clean execs from a trustworthy source?
  - Malware is getting more slick all the time. Wouldn't it be easy for mal-folk to bend an AC to get its updates from them?
  - Are "legit" companies any better (shades of Sony rootkits)?
- If we can't get something as simple as automatic spell checking right, what business do we have designing autonomics?

Autonomics Fear List (2)

- Could autonomic systems be "flipped" to do harm rather than good?
- Will AC dumb-down new generations of admins so they won't know how to fix anything?
  - Has this already happened???
  - Does it matter?
- Will AC hide so much information that investigation will be impossible?
- Will AC systems be OS agnostic, or will they force new levels of vendor lock-in?

Autonomics Fear List (3)

Multi-organizational AC Issues

- AC systems become nodes in a grid of cooperating, mutually defending organizations
- Agreements:
  - Similar to trust relationships in grid computing, but dynamic and more far-reaching
  - What kind of agreements should be emplaced regarding mutual upgrades/patching? Legacy applications could be trouble
- Can AC systems negotiate new agreements?
  - What happens when my AC system agrees to something that costs me money and I change my mind?
  - What happens when my AC system does something that costs you money and you want to sue me?

What about my job?

- There will always be a need for human system administrators because:
  - The complexity of systems is growing faster than the complexity of software solutions to manage them
  - With autonomics to take care of the well-defined problems, only the difficult ones remain
    - There will always be ill-defined technical problems that require human intervention
  - Autonomics save work but cannot handle every case
  - More automation will be needed, implying probably no net job loss
  - Someone will still have to verify that the system is working correctly

What about my job?

But AC will change the profession

- System administration is tied to ever-changing technology—change is the only constant
  - Evolutionary changes can cause revolutionary tipping points
  - Computers will be trusted with more kinds of work
- Overall effects of AC:
  - Fewer tedious jobs (+)
  - More time to help human users (+/-)
  - More complexity per case requires greater specialization (-)
    - **Generalists** might work for AC consumers (Nurse Practitioner model)
    - **Specialists** would work for AC vendors (MD Specialist model)
    - **Super-generalists** might be independent contractors (MD General Practitioner model)
  - AC will impact IT specialists (DB, storage, etc.) more than system or network admins (+/-)

Autonomics Future Directions

- AC is not a commodity like the cell phone. AC is currently a "nice to have."
- A year or so before <Company> will be ready to release an industrial-based prototype that might be considered true AC.
- AC will require serious new OS primitives
  - E.g., utility functions as opposed to simple job priorities
  - Reliability, modularity, standardized interfaces, security, etc.
- AC will have to be better than "Moon Launch" reliable before it can be seriously adopted
- My opinion as an HCC researcher: Much more ethnographic research needs to be done before AC is ready for prime time

Conclusions

- AC is coming, but slower than you might think
- Outsourcing is probably a greater job threat
- Prepare for AC by:
  - Staying informed
  - Embracing change
  - Delivering great value to your employer
    - Use autonomics to improve your job performance
    - Be part of the revolution—use and develop new tools
- Don’t panic! ☺

Contact Info:

Glenn.Fink@pnl.gov, 509-375-3994

Please contact me if you would like to participate in my AC survey!

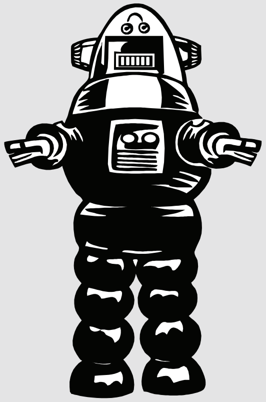
http://surveyext.pnl.gov/cgi-bin/autonomic/ezs.exe?database=autonomic

Answers to Frequently Asked Questions

- Regarding doubts about my accuracy:
  - AC doesn't exist yet, so nobody knows. These are just the musings of my interviewees.
  - I am not a fortune teller.
  - This is ongoing research, contact me to participate
- Operational issues:
  - I'm a researcher, not an operations guy, what do you expect?
  - Are you asking to learn something or just to show off how smart you are? ☺
- All other issues:
  - Read my paper first, then come talk to me (April 2007 ;login: http://www.usenix.org/publications/login/2007-04/openpdfs/fink.pdf)
  - I don't know, but I'm open to your opinion

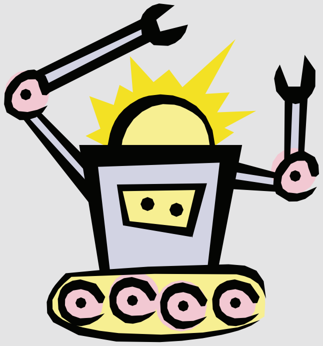
Human-supervised hierarchy of intelligent software agents from 30,000 feet

Supervisors: Humans responsible for high-level guidance

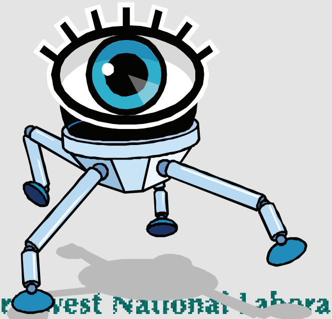


Sergeants: Heavyweight, autonomous software agents responsible for an organizational unit

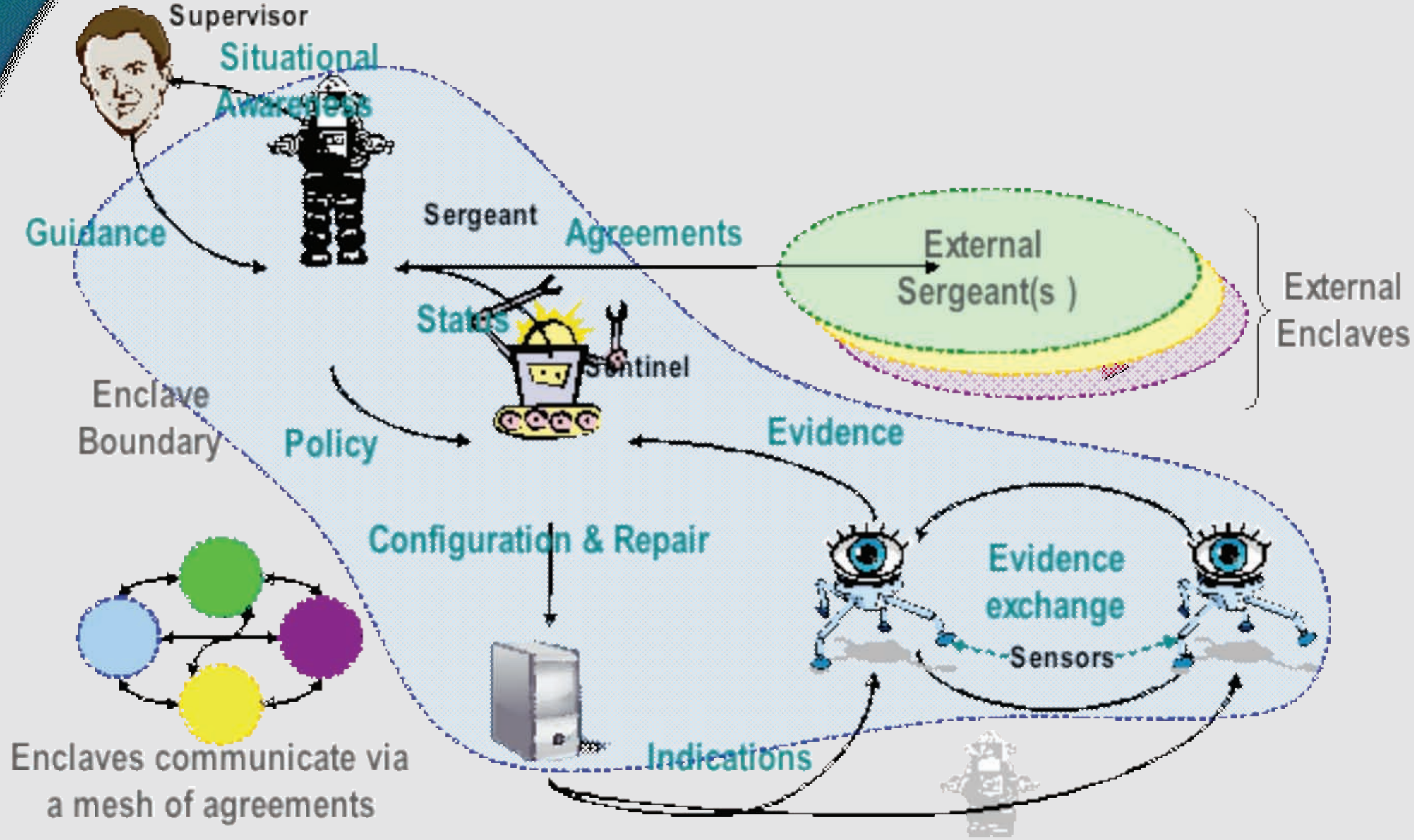
Sentinels: Middle-weight software agents responsible for individual machines



Sensors: Lightweight, swarming, mobile software agents that roam share information and detect problems



Cooperative Infrastructure Defense



Enclaves communicate via a mesh of agreements