Autonomic Networks –

Autonomic Communication

Wedge Greene & Barbara Lancaster

LTC International

Date: 08/15/2007

About LTC International Inc.

LTC International provides leading companies in the telecommunications and IT sectors with a unique level of service based on true subject matter expertise. Our Business Operations Architects® each have at least ten years of hands-on experience in service provider and IT intensive companies. Our consulting team has experience in all areas of business profit optimization, wireless and wireline communications, Internet services, as well as software and hardware planning, implementation and operations.

LTC has incorporated more than 1,000 years of first hand operating company and software application experience into our Business Management Toolkit. This comprehensive set of tools, guidelines, checklists, templates and training programs is designed to remove uncertainty and accelerate success for our clients.

http://www.ltcinternational.com

Expect Results®

AUTONOMIC NETWORKS – AUTONOMIC COMMUNICATION

Summary:

At its core, Autonomic Networks change our whole concept of 'what is a network', 'where are the corporate boundaries', and 'what is network and service management' - radically. Some of these principles are so compelling (such as discovery and self-configuration) that we are implementing them today even as our incomplete understanding of them improves year over year. But to truly understand and manage our existing and future networks, we need to see them as complex, organized, evolving systems and develop a 'top down' approach to controlling the transition of the existing network toward more autonomics. To make this practical concept a reality, we need (1) a well understood picture of how things actually work, (2) a set of clearly explicated goals, and (3) a set of plans and processes which provide for a practical transition from what we have to what we can see will be better – this is the epistle of the authors. This article also provides a summary of the key groups and initiatives currently addressing Autonomic Networks and Autonomic Communications.

Business Operations Architects ®
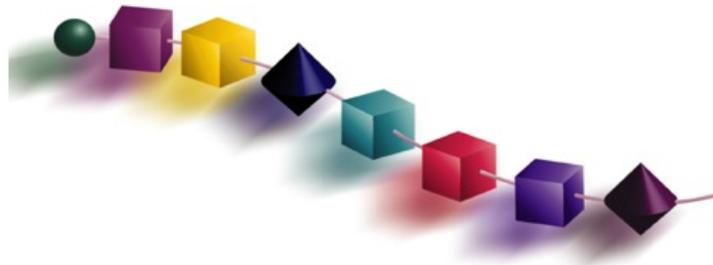
Blue Sky Investment

The EU in the last two years has made a series of significant multi-year research investments. All these investments center on designing, prototyping and validating the concept of autonomic networks. All involve multiple partners collaborating on a project of significant scope and extreme difficulty. That the EU has made multiple overlapping grants points out both the risks and the importance of this undertaking. And the EU is not alone, universities and corporations are also organizing groups to refine their thinking and align their approaches. Within North America, many corporations we canvassed admitted programs, but declined to discuss them, classifying them as being internal and strategic. If you do not have such a program or do not now participate in one, will it cost you more in the long run than you save sitting on the sidelines? Certainly there are many places to watch and a few where you can contribute:

- The Task Force on Autonomous and Autonomic Systems (TFAAS)

- Autonomic Communication Forum (ACForum)

- Autonomic Network Architecture (ANA)

- CASCADAS: (Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services)

- Serenity

- BIONETS (BIOlogically inspired NETwork and Services)

- HAGGLE - An innovative Paradigm for Autonomic Opportunistic Communication

- Open Grid Forum / Global Grid Forum

   In Part 2 Self-* Networks, we introduce these organizations and compare their particular emphasis and project goals. What we find is that they have much more in common with each other than any of them has with the way networks are architected and built today.

The origin of *Autonomic Networks* (AN) arises in the late 1990's. During that time, some companies, notably MCI, HP and IBM, committed to a 'vision of the future' that embraced early versions of what has become AN. As these efforts in background research became 'sexy' marketing differentiators, many names and definitions arose. This confusion of names continues today; there are still many names – but at heart, one vision; one goal. Some of the names include:

- Autonomic Computing / Autonomic Networks

- Autonomous Systems

- Ambient Intelligence (AmI) ecosystems

- Global Service Ecosystems

- Active Networks

AN design is really part of a long evolution in network and network device design - In solving for increased resiliency and rapid restoration, each new network platform technology gets more autonomic. But in this gradual evolution of increased resiliency, some designers saw a trend of diminishing returns in quality with marginal increase in the cost of each quality increment, set against increasing demands on network and service management. A new approach was needed.

Designing autonomic networks is a very big undertaking. At its core, our whole concept of:

- 'What is a network?'

- 'Where are the corporate boundaries?'

- 'What is network and service management?'

Radically changes. AN research is a very broad activity with enormous scope – many sub-areas require significant work. Why should you attempt this? Where do you turn for inspiration? Please read on.

Nature as the model

Nature is messy – very messy. Look out your window, even if it's a manicured corporate garden, there will be hundreds of different species performing different uncoordinated activities. Yet day after day, the view only changes in detail, not in substance. But the stresses on the environment just outside your window are severe: storms, gardening machines, invasive species, and human efforts to control what grows and where.   Against all this, the wood or garden continues to function and presents much the same view day after day. It is this high-level constancy, in the midst of constant low-level changes, which has been extensively studied in recent decades by Conservation Scientists, that intrigues the developers of next generation networks (NGN).

It is rare that any specific ecosystem in nature is broken, that the normal activities and interactions are so displaced that what we might observe in other similar climates and topologies is not evident. Small changes occur, but the overall ecosystem remains consistent. Can we build networks as self sufficient and steady state as nature?

> "Sometimes we technology specialists should leave our desks, forget everything we know about software architecture, computing systems and the complexity of distributed networks and take the time to observe the world around us and learn from Mother Nature. It's the best self healing system ever made," Dan Dura, OSS Integration Architect at 'Cingular now the new AT&T'.

Nature is complex. Way more complex than all the networks together. Yet nature continues to function and seemingly embrace this complexity. If we can understand basic principles of organization and behavior in nature, can we apply these to our simpler networks? Are these principles needed as networks continue to grow more complex?

Most of nature is in constant conflict. It is hard to imagine building a telecom network based on conflict of device against device. While nicely-ordered collaboration in nature is proportionately rare, where you find it, it is very successful: ants, fish schools, plains herds, wolf packs, and mankind. With the possible exception of the higher mammals, what participants of the new science of Complexity are finding, is that seemingly ordered, apparently well-designed collaborative behavior are found again and again, but the 'well-designed' is an illusion – in reality it is the result of sets of small rules interacting in complex systems generating *emergent* behavior.

Nature is not designed. All the interactions have been worked out by minute changes in the activities of the individual species acting over a long change. Ecosystems have no team with the job of network designer creating the pretty schematic of the integrated single network. But there are similarities in the way networks grow and ecosystems change. We seldom fork lift an elegant network design into place and fire it up from scratch as a complete working unit. Instead, in reality, networks are 'grown' part by part, from installation of this switch and those supporting devices and systems, to the overlaying of another new service on an existing network structure, and another, and another. Each new element must

be linked into the existing network. Seldom do we decommission devices, mostly we keep interconnecting them because they have economic life left or our customers are dependent on them. So maybe there is some useful information we can derive from observing nature and its 'organized complexity.'

NGOSS is probably the most advanced management and service integration style generally available today. Yet NGOSS is framework architecture. Its design was not modeled after nature. "Frameworks capture expertise in the form of reusable algorithms, extensible architectures, and component implementations. Application frameworks have emerged as a powerful technology for developing and reusing middleware and application software. Because current frameworks are application templates, they are not well suited to cope with scenarios with high degrees of heterogeneity, dynamism and unpredictability." (From Serenity, a project designing autonomic networks) "Patterns capture expertise in the form of reusable architecture design themes and styles, which can be reused even when algorithms, components implementations, or frameworks cannot." (Also from Serenity) Autonomic Network design depends on finding and creating patterns and reproducing them again and again. Three separate models from nature compete for use as patterns for the design of tomorrow's networks.

(1) Most often cited as the source archetype is the *autonomic nervous system* of the higher mammals – hence the most common name – Autonomic Networks. This model assumes a distributed hierarchal control system of sensors and patterned responses. It is the most comfortable pattern of those we derive from nature because it is already familiar. Most networks today are hierarchal in their organization and their management. The notion of sensor/responder is somewhat parallel to our existing network management pattern of agent/manager.

(2) *Social insects* also provide a pattern. Here swarms and colonies of often related individuals produce quite complex and seemingly intelligent decisions based on the interaction of a much simpler set of behaviors. This pattern is used by humans seeking to build policy-controlled systems that are comprised on mostly homogeneous elements.

(3) *Predator-prey, producer-consumer ecosystems* are another biological example from which patterns can be abstracted and applied to network and service designs. The give and take, up and down flows of these systems are well matched to peer-to-peer systems; and many people think they can provide insight into the governing mechanisms for P2P systems - allowing the rules of engagement in the environmental matrix to provide indirect controls for the autonomous rules of interacting agents. This type of pattern bears a relationship to today's economic systems and may model the interaction of commercial networks. It is the most complex pattern and relies heavily upon the identification and reinforcement of often unexpected emergent behaviors. Herding behavior of cows and sheep, or schooling of fish, behavior driven by predation, are such example patterns.

Whatever pattern you pick from nature's examples, it can be applied to the organization and architecture that a group uses in the design of an autonomic network. Each pattern has strengths and weaknesses. Each is a better or worse fit in specific circumstances - for after all, networks are not natural systems. Fundamentally, natural systems reach stability via the interaction of evolutionary principles over long time spans. Despite our overloading of the meaning of the word 'evolution', there is no actual Evolution going on in networks and devices. They do not have or propagate DNA. Still, with caution and remembering that "the map is not the territory":

(1) The autonomic nervous system is a good pattern for active management and control systems.

(2) The social insects are a good pattern for designing stable, well mannered, homogeneous networks of similar devices, for example, optical networks or routed IP networks.

(3) Producer-consumer ecosystems can give us insights into how the larger, heterogeneous commercial network environments behave.

But the sheer complexity of producer-consumer ecosystems systems do not readily allow for 'design' of behavior. Instead behavior stabilities and abnormalities are observed and cataloged, using methods developed via Complexity Science, for use in predicting future behavior and indirectly influencing behavior by the tweaking of parameters in the environment.

Do autonomic networks exist today?

There are three voices to this. Most today say that architecting autonomic networks is a fledgling and mostly academic field of study that will someday, but not anytime particularity soon, lead to autonomic networks. There certainly is much theoretical work left to do, easily years of it. Read the journal articles produced by some of the example organizations below and you might quickly succumb to the viewpoint that we are decades away from a complete understanding of AN.

On the other hand, entrepreneurial companies are already trying to implement the early principles and findings of autonomics into their products. Their take is that 'the characteristics of autonomic networks are so compelling that competitive advantage can be found in achieving even a few, partial steps in the direction of autonomics.'

Finally, there are also those people (including the authors), that believe that if you step back and look at the totality [man, system, network, and economics] as the complete boundary of the system, that this larger system today looks and behaves like a complex ecosystem – this is the leverage point for network complexity science. We have to understand the complex dynamics which are present in complex networks to find the simple trigger points and tipping points for effective management.

It is likely that each of these separate views is partially correct. We can study principles of autonomics and begin to design autonomic networks from the bottom up by assembling components and services within a governance framework. Some of these principles are so compelling (such as discovery and self-configuration) that we are implementing them today even as our understanding of them improves year over year. But to truly understand and manage our existing and future networks, we need to see them as complex, organized, evolving systems and develop a 'top down' approach to controlling the transition of the existing network toward more autonomics. To achieve this practical concept, we need:

(1) A well understood picture of how things actually work,

(2) A set of clearly explicated goals, and

(3) A set of plans and processes which provide for a practical transition from what we have to what we can see will be better

– This is the epistle of the authors.

Many companies accept that the economics of self configuration, self management, and all the other "**self-***" (pronounced self-star) will lead to significantly cheaper networks and this has become a reason for closed door, gradual research on AN. This seems to be the policy of most North American vendors and service providers we approach.

But we maintain that the principle driver for research and prototyping of AN is not simply a marginal economic incentive. Future networks and communication services are expected to be much richer and

more complex than today's networks and services. Despite all the efforts at 'network rationalization' (consolidating networks and decommissioning older technologies), networks are growing more complex, not simpler. New is layered over old. Services are getting richer in features, more pervasive, of enhanced scope and applicability, more numerous, and detached from the network technology on which they are delivered. Any service, any network, any where, instantly is the mantra of the future consumer.

To achieve this, our future selves will be deploying *Grids*, *Pervasive Computing*, and *Ubiquitous Networks*. *Grids* are massive accumulations of interconnected/networked computers on which services deploy as needed. *Pervasive Computing* is the notion that every device will eventually be connected to the network or be within range of a monitoring sensor that is networked (in archetype, Vint Cerf's smart toaster.) *Ubiquitous Networks* refers to the interconnection of every network on the planet, such that every person is always connected at any location and any service is instantly available via trans-network communications. It is this complex future - which demands autonomic networks - for this rapidly arriving future is unmanageable by traditional approaches. The EU seems to understand this and is the largest investor into public AN research.

Co-evolution of Computer and Network

While we tend to separate computers and networks as different areas of study during college, and then partition them as careers later on, the design and function of computers and networks are more similar than different. This separation of network and computer is based on the historical separation of voice and data, and perhaps even more so on 'large and small' in scientific inquiry – a classic approach that partitions fields of study and activity. However, both computers and networks are systems and as such are governed by many of the same principles of systems engineering, information theory, and electronics. Computers and networks have co-evolved from the start. Sun Microsystems summed it up with the classic corporate slogan: "The network is the computer." The scientific revolution that is following on the growth of Complexity Science involves a different way of seeing the world. Instead of separating large from small, Complexity Science shows us how all layers follow similar organizational principles: that each layer reflects the same structure of behavior as the layer above and below. Today, in the study of autonomics, the boundaries of computer and network are particularly indistinct and the academic communities work in tandem. AN networks are designed so that each layer reflects those above and below.

Science Publications (www.scipub.org), New York, in 2006 planned a Special Issue for The Journal of Computer Science (ISSN: 1549-3636) on "Reliability and Autonomic Management". The charter of the issues captures the broad structure of current thinking on autonomic systems.

> "Most computer systems become increasingly large and complex, thereby compounding many reliability problems. Too often computer systems fail, become compromised, or perform poorly. To improve the system reliability, one of the most interesting methods is the Autonomic Management which offers a potential solution to these challenging research problems. It is inspired by nature and biological systems (such as the autonomic nervous system) that have evolved to cope with the challenges of scale, complexity, heterogeneity and unpredictability by being decentralized, context aware, adaptive and resilient. This new era of computing is driven by the convergence of biological and digital computing systems and is characterized by being self defining, self-configuring, self-optimizing, self-protecting, self-healing, context aware and anticipatory."

This quote could be speaking of autonomic networks instead of autonomic computing. These same self-* characteristics are considered the principle characteristics of AN. Indeed, the more we study them, there

becomes very little separation between autonomic computing and AN. The service provider of the future will not be a telecom company, a software company, or a computer company – but a fusion of the three.

When grids become the platform of choice for deploying services, and open networks can access grid based services, the re-identification of computer and network will be complete.

We see the start of this today in two landmark companies and it is no coincidence they are both models of success for the internet economy. Akamai is a grid of tens of thousands of distributed servers that spool content to edge users – Akamai provides parallel downloads at points that are 'network close' to the end user. Google provides an ever growing number of consumer services hosted in parallel on what many believe to be the world's largest grid – on the way to hundreds of thousands of servers. These companies are unified service, computer, and network platforms. Already, the financial powerhouses, based on the success of their use of grids as massive computation engines, are studying moving all their future services to grids. Where goes the financial service industry, soon goes the rest of industry. Grids, Pervasive Computing, and Ubiquitous Networks are likely to be common in one decade. This places pressure on turning the study of autonomic networks into the applied deployment of autonomic network infrastructures as quickly as possible.

Since we are re-inventing not just networks, but services and organizations, a broader name for the activity is needed - one both comprehensive and immediately clear.

> *"Autonomic Communication* (AC) - a new communication paradigm to assist the evolution of communication networks towards functional adaptability, extensibility and resilience to a wide range of possible faults and attacks. Special emphasis is given on the grounding principles to achieve purposeful behavior on top of self-organization (including self-management, self-healing, self-awareness, etc.) … in its interaction with numerous often-dynamic network groups and communities. The goals are to understand how autonomic behaviors are learned, influenced or changed, and how, in turn, these affect other elements, groups and the network."
> [http://www.autonomic-communication.org/Annales/]

This term, Autonomic Communication (AC), seems to us to best summarize the field of activity. Hereafter we refer to the paradigm shift and research as Autonomic Communication (AC) and the product that will be deployed as the Autonomic Network (AN).

Architecture of Autonomic Networks

Not everything is new in transitioning to Autonomic Networks. The current thought on network structure remains. For example, autonomic communications holds the layers as:

- User
- Applications
- Services / Components
- Middleware
- Network & Computers
- Device
- Agent (soft) or Sensor (hard)

Notice however that this layering includes network, computers & software, transmissions, and information. So while these are familiar, what was once several distinct groups are now combined.

From the requirement of supporting Ubiquitous Networks, a strong domain model must be incorporated in Autonomic Communication architectures. Domains replace layers as the primary organizing unit. Domains will provide for semi-permeable boundaries, much like the edge & gateway routers do today in IP networks. But domains will need to automatically communicate large amounts of information about what they will and will not accept; what they are capable of, and what current capacity and QoS is available. Intra-network control-communications-paths will need strong security structures, gauged to how open or how restricted is the information context of the services they accept. As we described before, here is a fractal-like repeating pattern. Domains will contain domains will contain domains - Each with similar computational control systems but with different locations, contexts, and communities. Pan-domain structures will link and provide shortcuts across these numerous networks. Indeed, something patterned like the old I-PNNI model may return.

A distributed computing substrate (including tools and platforms) will be an embedded part of the network. While routers and switches will still pass data, they will outsource all but the simplest and quickest of decisions to the embedded distributed computing substrate; only things like lookups will stay. Indeed, the concept of a management interface fundamentally changes from a 'passive responder to queries' or a 'broadcaster of alarms' into an 'active interface' which links to control systems and network/service models. In effect, every device will have a service interface with published methods. In some architectural systems, the device-embedded service will actively register itself with service modelers and discovery components – principle of which is the agent-service providing self-restoration. This probably means that every device will incorporate Java or .NET and provide service *and management* interfaces accessible by soft services.

The dichotomy of Network and Services still exists, yet AC provides for collaborative automation of services in conjunction with networks. The goals and activities of each layer still are distinct; it is the patterns which repeat across layers. At some extremes of design, visionaries see networks as physically self assembling. These distant future networks would look strange to us. But most effort today is directed at *logical* self assembly and management. What is radically different is a profound *situational awareness* in each layer of: (a) the activities and (b) the structure of other layers and (c) the environment. In fact, the amount of information available when testing the truth of a firing rule or in determining behavior in near-real-time may be prohibitively large – negatively impacting performance. Context filtering will go hand in hand with context awareness.

Nevertheless, we expect many past lessons of network design and management must be encoded in the new architectures. For instance, not everything needs to be dynamic all the time. When covered by automatic restoration, lower levels can be "nailed up" for longish time periods, letting the higher levels adjust and adapt in shorter periods. Also, it is not always necessary to provision a new VPN if a data flow can identify and then hitch a ride on an existing virtual circuit that provides the necessary QoS and Security. But in AC, no longer would the layers compete at network restoration: a collaborative agent would invoke restoration according to the fit of pattern at that specific event circumstance – for example, 'speed-of-reaction still overriding other rules' for extreme outages.

Fundamental to the design of all autonomic networks is an embedded security model. In traditional networks, security was applied after the fact - mostly using external devices, applications and tools. The world would be a much more organized and safer place if TCP/IP had been designed with embedded security – and even this is being looked at again. Autonomic networks must be self configuring, self deploying, and self assembling. Frequently either mobile code or at least re-locatable services are used to achieve self-deployment. Without iron-clad AAA (Authentication, Authorization, & Accounting) built into this structure, these re-locatable service building blocks are potential openings for intrusion and abuse. While absolute security cannot be guaranteed and security does add cost, most current designs

have produced workable compromises where the security of a network is well known and generally better than that provided with current approaches. The best designs can adjust cost and security as properties of network domains, controlled by policy and bounded on the borders with a AAA service that filters user and service admittance.

Several more characteristics of AC must become common features of devices and services. These are novel behaviors not generally present in today's networks; except for a few in some specific instances such as Cisco's proposed Application Aware VPN (AAVPN) and a commercially available edge controller from Ipanema.

- *Situational awareness* covers the capability of services to autonomously adapt to the context from which they are requested and in which they execute. It demands that technologies capture situational data and effectively exploit it.

- *Self-organization* uses control services that implement patterns often derived from nature and identified in the extensive examples and principles of Complexity Science.

- *Fractal/self similarity* concentrates on patterns of organization that reproduce nearly identical structures over multiple scales. Very similar to the concept of recursion found in the definition of, for example the Pascal programming language, we see this embodied in the fractal graphs of complex systems. In AC, networks should organize using similar patterns at every layer and in every sub-domain. Only the mix of components and services and the goals change from sub-system to subsystem and layer to layer, not the structure of organization.

- *Self-healing/Self-preservation/survivability* which provides a way for every device, path, and service to recreate itself upon failures, ameliorating the impact of the failure on overall service delivery. [more in the next issue]

Traditional network architectures have been largely concerned with structure – what is where, what connects to what, what protocols are present. Architecture in AC is more concerned with behavior and process than with structure. Since the network is expected to have situational and context awareness, the architecture looks like a central nervous system performing its reactions to stimuli. Four basic stages in this reaction (per Dobson's 'A Survey of Autonomic Communications') are designed into AN:

- Collect: gather information from environmental and device sensors and instrumentations; determine application or service requirements; look up user context and security credentials.

- Analyze: use the most modern of inference tools to process this collected information including: Fuzzy logic engines, bounds and envelopes, economic models, rules and polices, and game theory.

- Decide: process with the best available computation engines including decision theory; risk analysis, hypothesis generation, genetic algorithms, and neural networks; scope actions to situations and needs.

- Act: inside transactional contexts: invoke services; assemble components; manage collaboration; configure managed elements; signal devices; inform users and administrators; log actions and the strategic analysis that determined them.

Elements, Components & Services

There is some strong consistency of viewpoint on the major component elements of the AC. Today, many of these components are present in the standardized architecture of grids. The fundamental components of AN are:

- Network/devices/sensors: the physical & electronic world

- Modeler: a logical image of the network and communication paths linked via management control interfaces to the physical and configuration information of the network and maintaining a near-real-time display of state. Multiple models can exist with different purposes – for instance as load processors and predictive models of the future state. *FineGrain Networks* architecture has the modeler as a 'living service' always present in memory providing a near-real-time, active computational model.

- Registry: contains accessible information that stays relatively constant including all the physical maps, locations, and configuration baselines of all devices and services. Also contains user profiles and security credentials. Domain security contexts and service availability features are stored. (The 3GPP/IMS HSS is a step in this direction, as is the VeriSign security registry.)

- Policy Engines: contain the rules and procedures for actions and activities.

- Control systems: Invoke actions based on stimulus and maintain the overall health and stability of the autonomic network.

This architecture is radically different from today's management structure. Device MIBs, agents, element managers, managers, manager of managers, service managers, etc. are no longer the principal management machine. Neither is EAI tunneled messages controlled by work flow present. Instead, management is contained in the localized event-invoked actions of the control systems and is governed by state, policy, and situational context.

But we do not expect to throw out everything we have and have learned. Today, IMS is a policy-controlled architecture, and when it becomes incorporated with grids, will form bridging steps in the direction towards autonomic networks. What we expect is that the structure of management and the current instantiation of big, external management applications will become a marginal activity or disappear. When automation fails, there will still be a NOC with engineers who can respond to the unexpected or overly complex; who can resolve policy conflict. Contact Centers will still deal with issues and maintain CRM systems – for some time.

Our recommendation is that the current best thought and practice be encoded in agents and policies, becoming the brain of the initial AN central nervous system. For example, eTOM can be patterned into policy engines and control system logic.

But 'small is better' will win the day. Most decisions are much localized in scope, even today. A single device has a small configuration change. A bit of user profile data is updated. A message is sent. These kinds of actions can be realized with simple and local agents – they do not need the overhead of massive OSS applications. When small, when the decision is made in close proximity to the scope of its influence, massive parallelism becomes "built in" to the autonomic network. Scaling which would choke today's applications is possible.

Similarly, as is the case with NGOSS, Framework services and Business Services will be present. Repeatable, consumable, commonly used services will be available as resources in the autonomic

network. These framework services will be commodity items used at commodity prices. Business Services will be localized and individualized by company with more value-like pricing. But these will be organized in patterns (not workflow).

Finally tools for analysis, testing and validation will be framework services invoked as a normal part of service provisioning or restoration. When a device is installed and turned on it will discover the control network, register itself, seek and download its most recent software and configuration files. Then it will perform self-diagnostics, test all outside links and circuits, and flag itself as available. Similarly, software services and components will dynamically load into the business or control grid, instantiate themselves, register with control and discovery, download their configuration files, and automatically discover and link to all the resources they need. The specifics of this vary, the patterns alter, the business goals slightly differ, but the main thrust of this activity is common:

- CASCADAS: "Components dynamically self-organize as needed with each other and with the already deployed ones, and will start interacting so as to provide the desired functionality in a situation-aware way without (or with very limited) configuration efforts."

- MCI's NewWave: Services dynamically deploy, find and connect to other needed business and framework services or else dynamically launch those needed services. Once launched, component services maintain themselves and their existence through all network perturbations and failures unless specifically requested to terminate.

- FineGrain Networks theory: Software automatically deploys and then self-creates its own logical network. Soft services dynamically deploy, identify from situational and location information the component services they need and then dynamically interact with network and VPN configuration services to either find & attach to an existing VPN or to dynamically deploy a new VPN with the required QoS.

Putting it all together

Rick Thau of 'The Thaught Process' is a skeptic: "The biggest problem with some of the newer systems is that they try to "boil the ocean" by doing too many different things in an automated fashion. … chaos automated is just a disaster happening faster."

Whereas *Serenity* sees: "The concepts of system and application as we know them today will disappear. Static architectures with well-defined pieces of hardware, software, communication links, limits and owners will be replaced by architectures that will be sensitive, adaptive, context-aware and responsive to users' needs and habits."

In perhaps the best survey article on the field of AN, Konstantinou describes how difficult it is currently to build autonomic networks:

"Attempts at automating network operations have so far met with limited success due to the design of existing management architectures. Current architectures assume a manager-agent (client-server) model in which element performance and status information is presented to human managers. Managers must collect and interpret this information in relation to network policy. Policy enforcement requires manual change management over distributed, heterogeneous element configuration repositories. Managers are further required to manually log and coordinate configuration updates across multiple elements due to lack of transactional configuration access mechanisms. These architectural limitations create significant safety, scalability, and reliability

challenges to automation." [Towards Autonomic Networks; Alexander V. Konstantinou, Ph.D. Thesis; Columbia University, New York, NY, USA; October, 2003]

Dan Druta of AT&T remains a qualified optimist: "The reality is that in some areas there's significant progress being made. A lot has to do with the proliferation of Service Oriented Architecture concepts and the mandatory requirements to treat components as services. This forced some mentality shift within the vendor community as well as providers and IT organizations in general. Then another big thing came in the form of *Software As A Service* (SaaS) where SLA's where required and as a consequence the functions mentioned earlier had to be there. Also this combined with the Web 2.0 buzz around mesh applications, emphasized the need to treat any endpoint as a system and any system as a subsystem. Technology wise there's progress being made in the "spaces" area. Jini is a perfect example where there's now a programmers toolkit available to create peer to peer applications."

Herein, Dan is referencing two groups that built AC on a Jini model: MCI's New Wave and the TMF FineGrain NGOSS team. In 2001 & 2002, the FineGrain NGOSS Catalyst demonstrated "live" autonomic management on real equipment (Sun servers, Hatteras, Juniper, Alcatel) for modern layered services (Ethernet access, GigE switching, VPNs, MPLS, SONET):

> "Fine Grain NGOSS is an approach realized through specific infrastructural services and business policies that enable the cooperation of a multitude of technologies and business functions… The aim of the Catalyst project is to define a *Service-Oriented Architecture* with *Real-Time, Run-Time Plug-and-Play* capability.

> "Policy based provisioning and management is demonstrated in an *e*business model. As [line] cards are plugged in, resources are discovered, services automatically are established, target customers identified, host servers connected and up selling occurs. When under stress, the network automatically reconfigures to match SLA contracts."[Fine Grain NGOSS Phase II Overview]

We therefore have known, and demonstrated for nearly a decade now, how to build autonomic, policy driven management services that greatly exceed typical current capabilities. Still, where in use, these advances are declared "only experimental" or "trial deployments" – at best "specialized vendors". It is the demonstrated inertia of IT departments, and the lack of buying demand from service provider engineering departments for actively managed devices – that is slowing AC, not lack of knowledge. Recently, one of the star developers on both the teams for MCI's NewWave and the TMF FineGrain NGOSS Catalyst, lacking a project to join that utilizes their skills in AC, was placed in an "SOA Immersion course"; their candid response, "I wasn't terribly impressed, but I guess I need to go with the flow and follow the crowd." We fear that as those developers who built the early success stories are scattered to the wind, we are on the verge of losing what experience the industry has gained.

Nevertheless, we'll apply the qualified optimist label to ourselves. The major components have been identified, groups of talented people are focused on task, and there are new revenue models waiting in the wings. While we suspect that humans will slow up progress given our immense capability to resist change, the laws of nature, and of economics, will out: Autonomic Networks will emerge to make Autonomic Communications possible. And they are just too neat to deny.

- *Almost End…*

- *See part 2: "Self*Networks: Helping Networks Help Themselves" -*

Addendum: Want more? Background reading with personal accents…

You can read Wedge's past writings in this general area at LTC International's Inside Out blog: a five segment series. In the context of personal examples of experience with data networks, Wedge covers many of the technical design aspects and knotty problems of autonomic communications.

- The New Telecom Ecosystem 1
  [http://www.ltcinternational.com/inside-out/2006/the-new-telecom-ecosystem-1-preparing-for-the-new-ecosystem.php]
- The New Telecom Ecosystem 2
  [http://www.ltcinternational.com/inside-out/2006/the-new-telecom-ecosystem-2-complex-systems-autonomic-networks.php]