Pipeline
Knowledge Is Power

www.pipelinepub.com Volume 5, Issue 4

The co-Evolution of Networks and Devices: Autonomics and Device Management

by Wedge Greene & Trevor Hayes

All the Fishes In the Seas

> *The moon came white and ghostly as we laid the treasure down,*
> *There was gear there'd make a beggarman as rich as Lima Town,*
> *Copper charms and silver trinkets from the chests of Spanish crews*
> iPhones, smart sensors, home gateways, game portals, just name a few
> [apologies to John Edward Masefield]

Our world of communications just keeps getting bigger. So too does our job of managing this burgeoning world. Once upon a time we managed big network switches. Then we managed connections and, of course, the CPE that terminated the connections. More and more different types of CPE were invented, all staying physically connected to the network. Then came a very different concept: intermittent connections - dial up communications to computers changed the notion that end devices were always connected to the network. Computers themselves radically changed the network, the devices and their interaction. Even more loosely tethered mobile phones followed and are rapidly gaining service intelligence, even outstripping the pace of advances in network technology. With each evolution of technology, the size and scope of the domain of devices needing to be managed increased. The complexity increased. Correspondingly, costs and opportunities also increased. Today we can see that our historical ability to manage millions of connections is going to have to somehow scale to manage tomorrow's trillions of devices.Often this is represented in drawing concentric rings where the network is in the center and the area of the rings increases as more kinds of devices are added over time.

Today's outermost ring is very large and includes such things as smart homes and sensor nets. Sometimes, when we draw this expanding trend of concentric rings, it illustrates the sheer number of things to manage, evoking images of scaling challenges and complexity – and ever expanding management and operational costs. Less often, this type of drawing is used to illustrate the expanding opportunity and richness available to today's service provider. But dangerously, the service provider and the network are always in the center. Dangerously, because we think this image may cause more harm than enlightenment. A fallacy of human

thinking is to equate a drawing with reality even when the drawing only loosely illustrates the situation, for example "the map is not the territory."

We frequently also see illustrations of devices placed on the end of a line drawn back to the network. In these pictures, as the multitude of devices increases, devices fixed by connection lines to the network become a bristle cone of spokes, each evoking the perception of permanent connectedness. This way of thinking narrows the possibilities of devices to only those associated with the network. But devices also interact with many other things such as other devices, other users, or power grids. Some of this is captured in "dandelion drawings" made of balls of branching spokes where closely routed device families are connected to common branches back to the network. This may represent network topology, but it does not represent device topology. Devices can of course connect to the network, but just as easily, today's devices can connect directly to each other. The startup company m2mi just released an application for the Apple iPhone called *icrowdsurf*. It uses the iPhone's built in WiFi to find and connect peer-to-peer to other iPhones within WiFi distance of each other – completely bypassing any network services. There is no service provider network in the center of things. Perhaps then there is a more useful way to conceptualize this trend of device proliferation.

Device management may well be the biggest future opportunity available to the Service Provider. The advent of Over The Top Services shows us that even as we move into a world where everything is connected to the network, the importance and the value of the network is under technical and financial attack. Many operators and the TM Forum have looked to media delivery as an answer. Yet they may have a much greater opportunity staring them in the face: the proliferation of network connected devices, often called *pervasive computing*. Yet the multiplication of connected devices is also the service providers' biggest opportunity to fail, as the same forces attacking traditional telecom services via OTT services, are embarking on substantial device OS and service development projects. Service Providers cannot meet this threat by applying "business as usual" thinking and current product strategy.

Some members of the TM Forum understand this. They launched the Device

Management initiative. They have made a good start and gathered some very forward thinking team members. The presentations at the Management World Nice Device Management session were quite strong. According to Chris Ballard, who heads this TMForum Device Management program, next on their agenda is a unifying **Framework for Device Management**, a parallel deliverable to the Service Delivery Framework which attempts to link the various Service Delivery Platforms of IMS and SIP vendors. (See Chris' article in this issue: "Simplifying Device Management.") But this team discovered it has bitten off a difficult task. The scope of the device world greatly exceeds the scope of SDP vendors. There is, however something, seductive about creating this grand vision. It is a rich and challenging intellectual task, like the search for unified field theorem. There is a haunting feeling that it can (and must) all be put together.

Why is this so big a job? Just look at the many classes of devices to be managed and the many different architectures to be assimilated and the many organizations who might contribute. The TMForum has made a good start on this with the "End-User Device Management - Industry Groups Positioning Document."

Too Much of a Good Thing

The TMF initiative initially is concerned with the major new devices just now being connected to the network. First, there are mobile phones. This is a well-studied association. Management standards for mobile phones exist in the OMA, OSGi, and the 3G standards, all of which gradually have been expanding the scope of management suggestions. With release 6 and 7, the 3G introduced on-board active management agents that, residing in the phone, could closely monitor and report on activity and health. The agent concept is expected to help in being able to provide QoS for advanced phones. But it is as yet only a suggestion and as an architecture, follows the classic manager (in the network) agent (in the device) model.

In addition, there is the Open Mobile Terminal Platform (OMTP), a group of companies acting like a forum. Rather than just thinking of the phone as a device,

this group concentrates on the system of user/owner and mobile phone, specifically during configuration, application download, and faults. They recognize passive monitoring will not cut it: "There will be significant cost savings in proactively monitoring the users' device and enabling capabilities which allow the operator to react quickly to customer problems (which may be complex and difficult to deal with in conversation with a user)." "To enable the next tranche of Connected Applications, user experience barriers and infrastructure limitations need to be addressed."

We would add other personal appliances to this mobile phone device group because of their developing mesh network relationship with the mobile phone. Problems might arise as Bluetooth devices attempt to connect, leading to a mobile phone customer calling a service provider trouble resolution call center. We must understand that there will be primary devices and satellite devices engaged in small, personal network clusters where only the primary device connects to the network, and to network management. Additionally, the close interaction of humans and devices is another aspect of the device domain that will generate unique management requirements.

Another broad group of devices under active consideration by the TMForum initiative are those participating in the smart home. Some are traditional network-attached equipment such as the optical termination points, the home gateways, and DSL-modems. These involve standards and organizational engagement from Cable Labs and the Broadband Forum (once DSL Forum). Some of these devices directly interact with human owner/operators, such as game consoles and IPTV equipment. Others interact with remote humans, such as home surveillance and alarm systems for burglaries and fire. Other systems have mostly automated local interaction, such as environmental controls for heating and cooling.

As the devices in the home become more capable, they will interconnect with multiple networks. A smart refrigerator will link both to the intelligent power grid and to the service provider telecom network to call out for new food items. Who should have the management responsibility for this refrigerator? Or how about in

this scenario: Sensors in the home connect in a smart mesh network. All of these devices interact with the smart home control center. This '"home brain" might be programmed to automatically adjust lights to a user-specific illumination profile as that sensor-identified user selects an IPTV movie to buffer and play from a service provider media center. Complexities of home management defy the traditional agent/manager device management model. They involve interactions of users, many networks, and many kinds of service providers.

Classical device management comes in two broad flavors: the service provider ITU/TMForum OmniPoint pyramid and the IETF derived enterprise device management market. Service providers follow a network element (E) that has an on-board active element agent (EA) that connects to element manager (EM) that connects to a network manager (NM) that connects to manager of managers (MoM), which, these days, connects to a service manager (SM), which often involves significant process automation. Each layer isolates the layer below, filtering information up and commands going down and providing some layer autonomy in its processes and data. Many communications protocols with devices are supported including CMIP, TL1, SNMP, CLI, & HTML; yet all of these simply pass data and commands via a 'hands off' management style. Modern mediation products allow fast transactions and scaling of information capture.

Enterprise Device management architecture and implementation is simpler. A passive agent with an onboard MIB connects via SNMP (or HTML) to a manager that connects to a console. To address scaling issues, satellite management services, such as data collectors, feed the manager. Over time, vendors blurred the implementation boundaries between the enterprise market and the service provider market. However, in every case, the network manager sits in the middle chatting via inter-mediators with all the devices. This worked somewhat with relatively small numbers (millions) of stupid devices.

While some members of the TMForum group are exploring new approaches, others continue to adhere to traditional management viewpoints. Both perspectives describe mesh networks and domains of managed devices, with everything eventually connecting upward to a network element management controller.

Swimming Up to Our Neck in Services

Traditional device management frameworks face a significant, current hurdle. In today's world, devices are smarter – they can download and host services. Once this was the realm of personal computers; unfortunately, in the many decades of computer use, no significant management standard was developed to manage smart, flexible devices with interchangeable services. Small pieces were standardized, such as with the DMTF information model, but basically this is a Wild West zone of proprietary solutions.

In the future of device management, more than physical devices must be managed; the services must also be managed. Sometimes the service is on the phone via manufacturer-installed software. Sometimes, as with the trend in smart phones, these services are downloaded by the user to become resident on the phone. Sometimes the services are in the network such as traditional call handling and

messaging; and other services are created and hosted by OTT service providers.

Today, the service/device interaction zone is the domain of mobile phones and computer. Tomorrow, many devices may have these capacities. So our concept of a unified device management framework must encompass devices and services.

Yet, services are managed by a different set of approaches, architectures, and products than are devices. Mostly today, the standards for service management are the work of W3C and web services. More pervasive so far, however, are the many strong proprietary approaches that exist, most notably from IBM and Microsoft. Sun Microsystems' Jini provided a mechanism for the MCI NewWave approach of a specific service management interface for services that complemented the functional service interface. Microsoft seems to be leaning toward this approach with perhaps the most modern service management approach today: CSF.

Another layer of complication is added when you consider the topology and grouping of service and device. Services can be vertical applications interlinked to traditional telco service providers tied to devices. Services can come from OTT service providers and link to edge hosting platforms, such as a web service on Google. Services can interact in meshed peer groups such as the aforementioned *icrowdsurf*. There are devices/services which interact with and cluster around a local laptop/server. The intelligent home is an example of a type of device clustering. Sensor networks (industrial, traffic, etc) are yet another type of clustering. Clusters of devices might even be associated with organizational and geographical regions, such as an office or academic campus. Electric, water, and other utility networks with their sensors, switches, controllers, and data gateways form a device cluster that again complicates device management, this time adding issues of national security to the device management problem.

Devices with service capability will interact with Data. Devices like sensors will capture environmental data. Meshed smart phones might see and even capture data about other user/owners. So device management must meet the regulatory requirements of personal data management. Configuration data and dynamically collected data may affect the health and behavior of devices and also influence their relationship to other devices in local and remote cluster groupings. Little standardization exists on how to manage data, yet data management must clearly also be part of a device management framework.

So Many Twists and Turns!

Current forms of device management will not cut it. It is not just a scaling issue, nor is it just a complexity issue.

Every headache that exists in connecting computers to wide area networks also exists with smart devices. A service provider engineer using the pseudonym John G Galt, complains that downloaded javascript services currently are severely impacting both the computer/smart device and the network, which must react to the device requests. He describes "babblers," which push continual, non meaningful noise through the boxes and up to the network, and "gobbers," which pull down data and consume local and network resources. Unsuspecting users download these

not knowing what impact these services will have, and then, once affected, not knowing where the impact originates. Few knew that when downloading the free Skype peer-to-peer phone service that they opened their computer to continually run data updates and routing computations for the Skype network; and that this would mean constant network usage, not just during calls. So any device management framework must include debugging tools. Debugging, either pre or post launch, is not part of the current OTT service market.

But should debugging requirements be part of the service provider market? Is this the responsibility of the service provider? When you envision the network in the middle and all devices connected slavishly to the network, device debugging requirements would clearly fall to the service provider (think big labs). We argue that certainly the effects must be monitored, but we are uncertain that the ownership of the problem lies with the network provider. One can just as easily invert the framework model and put the smart phone in the middle. Then the problem should be owned by the manufacturer of the device. Certainly Apple believes in the opportunity of the vendor to provide the configuration and service installation (and the sales of such). But there do not seem to be, for example, onboard debugging tools in the iPhone. Currently, that is the domain of the hacker identifying a problem and the blogger reporting it.

John G. Galt, "I predict gadget downloads to your desktop are going to be the next hackers heaven." Soon smart phones, like unsecured operating systems on computers, will be platforms attacked by worms becoming participants in denial of service attacks. Galt also is on the lookout for installations of billing malware. The threats from doing nothing are significant. But it is not clear who owns this problem and must react. Complications arise from ownership of the device. Once network connected devices were owned by the service provider, then they were owned by the individual or the enterprise. Once we have answered the question of ownership, we must then consider responsibility. If the user's smart phone participates in a denial of service attack, is the user legally responsible? Is the network carrying the traffic responsible? Is the web hosting center with inadequate threshold security responsible? Or is the US government who licenses the DNS system responsible? Figuring this out should be part of the determinations and suggestions of the TMForum Device Management Initiative.

Of course, these are only a few of the many ways devices need to automatically respond to stimulators and conditions. The only technical solution we see is that devices must incorporate policy enforcers that get policy from policy servers. Fortunately, we do have existing policy management standards. But to date, only network elements and some advanced CPE enact these. In the future, smart devices would enforce policy received from network or from vendor-resident policy control points with a device-resident Policy Enforcement Point (PEP). In some cases the Policy Decision Point and the PEP would both reside on a smart device such as a mobile phone.

An unavoidable implication of devices getting more powerful is an increase in the sophistication of services that run on the device and that engage resources inside networks. What is to stop these from being management services? After all, on a power limited budget, it's 1000x cheaper to process locally than communicate via

radio with the network. Evolution of devices into participants in P2P autonomous networks is occurring now. What is to stop these smart devices from brokering services into that mesh network? Smart devices will also begin to control local mesh networks composed of slaved or otherwise servicing devices. It may become both feasible and efficient to move service control points from their traditional location in the network down into smart devices.

It may also be efficient to have these smart devices proxy management functions for the local mesh. This would involve downloading a manager into the smart device. Even if the mesh group to which the device is a member is a logical mesh using remote connections, a single device could be nominated to broker management for some or all of the mesh. Obviously, we are way beyond the old manager/agent model. Still, this could be leveraged by providers to both solve device management issues and derive revenue. Someone will step up to provide these services as soon as others are ready to pay for solutions to management or security needs.

It is unlikely that just inventing another protocol, as the ITU and the ITEF do, will address any of these problems. Nor will creating new eTOM processes do it; not even creating additional new services such as with the OMTP – Open Services Initiative. A new vision leading to a new architecture is needed.

The Ship of Our Dreams

The TMForum is searching for a framework for device management. Any framework must address the requirements and goals of existing approaches. It must also encompass the plurality of existing solution approaches, the standards and architectures of the historically different device classes. But as these different device classes come to interact with each other (for example the user, mobile phone, and ticket kiosk interacting in a purchase), the framework must subsume all the pluralities not as isolated domains, but as open systems. When we re-use existing approaches, we must avoid the traditional myopia of the SP-in-the-middle.

Picking the right reference image will be important. It cannot be a network centric framework like the expanding rings of device classes. All the existing approaches from the ITU, DFTF, OSGi etc. are network centric. Yet, it would be costly and time consuming to throw these out and start over, so we need to live with and reconcile many different models. This requires using a collaborative architectural approach, built into the Framework. It probably means giving up on the demand for *rigorously consistent*, *all encompassing* information data models like the TMForum SID or the DMTF CIM. Using logical groupings of devices to segment the problem space might work if there were a good way to resolve the intra group interactions, provided the network centric viewpoints are purged from them all.

Apple iTunes currently manages iPhone configurations and the Apple store exclusively sells loadable services. Even if the technical need did not exist, the market is driving this direction. On-board agents are also in evidence. In an autonomous function, the iPhone checks Apple for authorization before an application is allowed to load. The phone software enforces Apple policy of "Apple as only service provider."

Devices are evolving in capabilities and any framework must be flexible and adaptable in order to adjust. While Android will start as a mobile OS, Google intends to enhance it into a universal operating system for smart homes (controllers and set-top boxes for televisions) and personal devices (cameras and mp3 players). Indeed, speculation by Internet co-creator Vint Cerf is that Android will subsume other communication, media devices, and services. "*In an internet enabled world, there is no reason that a projector could not be online and downloading images, maybe using the Blackberry as a control device. Surrounded by networked equipment that is reachable anywhere, devices harnessed on a temporary basis to do something for you and then released. I am predicting that during this decade, we will see more systems interacting with other systems like this.*" VentureBeat is sure Google wants to create an ecosystem that autonomously supports communication between any two devices. Similarly, Microsoft desires to connect all your devices with its Live Mesh platform.

Any new Device Management Framework must provide an overall vision to replace the network centric, single network attached illustrations of the past. It must provide an information model that links devices & services; fortunately, a good start to this exists in the TMForum SID. Still, the DMTF Broadband Forum TD series, the IETF, and the OSGA must all be reconciled into the SID or be brokered in translations by a semantic mediation product like Progress Software's DataExtend. Any model must provide for ownership and for identity management (of devices) and allow for multi-network, multi-provider, and inter-device recognition. It will need to future proof with RDF and w3C approaches, including work on the semantic web. In the meantime, the framework needs to operate in a technology specific environment of ESB and other SOA approaches.

A critical distinction can be made between devices that have people attached to them and those that just react with the network and the physical world. We can model the behaviors of both. Both must respond to external actions with conscribed and pre-determined reactions, but when a human is pushing the buttons, the potential of complex aggregations of behaviors greatly increases. And, of course, service of the device is more difficult, because the expectations of the human drive the satisfaction of service and along with the human capabilities, the utility of any specific fix.

Management processes in this environment are going to be incredibly complex. Cisco has attempted to define all the process needed to configure and support the devices used in their Enterprise Class Teleworker (ECT) product. These were shared in a presentation at the TeleManagement World conference in Nice this year. But this product manages only a subset of devices that will interact and many of the Teleworker participation devices will have lives and roles outside of telecommunicating. Likely, the entire process approach just fails under the weight of this complexity. Some of this can be absorbed into substituting a policy model for process via mechanisms the authors have discussed in earlier papers. But the only full answer must include Autonomics.

We have looked for some out-of-the-box-thinking reference models and toyed with either cars in a system of highways or schools of fish. Both involve autonomous

control within the element (fish brain or car driver). One must conform to policy as embodied in traffic law and tradition. The other conforms to policy reactions derived from evolution. Both systems allow the device to learn new behaviors. Both are examples of *complexity systems*. Small changes in parameters can have large system consequences, or large system perturbations can be damped by systematic minute adjustments in existing policies.

Systems' Models and Autonomic Behavior

The presentations at the Management World's Device Management track put stakes in the ground in our collective search for an encompassing architecture; one that will be the center of a future-proof Device Management Framework. Accenture proposed the *New "4Ps"*: (1) Pods – Digital Networked Devices; (2) Panels – User Interface & Controls; (3) Plexes – Servers for the Data, Content & Services; and (4) Pipes – The Network that links them together. Certainly all these need to be included, but clearly this is just too simple an approach. HP proposed the use of the Resource Description Framework (RDF) of the W3C with its description of devices via Composite Capabilities/preference profiles (CC/PP). Valuable, but just one facet of the jewel we seek. Of those contributed presentations, the future likely lies with the suggestions of the Siemens team to start with and follow on from the work of the European Celtic-Madeira project.

> "The goal of Madeira is to provide novel technologies for Network Management Systems, enabling them to facilitate the provision of self managed services and networks in a world of increased network scale, heterogeneity and transience… to provide an innovative architectural framework, requisite interface protocols and/or standards and a reference software platform with prototypical implementations for a distributed network management system based on a non-hierarchical peer-to-peer paradigm… [in order] to scale to huge mesh networks, in terms of both the network topology and number of network elements managed."

The Madeira Platform was an *architecture of component services* that correspond to those usually included in autonomic self-* systems: containers, life-cycle, code distribution, security, notification, directory, persistence, naming, specification, monitoring/health, & policy. They add key additional services similar to those suggested by the 2001 TMForum FineGrain catalyst project implementation: visualization, connectivity and grouping. Central to their architecture was an Adaptive Management Component on board the device, locally serviced by utilities and information repositories.

Unfortunately there have been no postings to Celtic-Madeira web site since 2006 and their program was left incomplete. Hopefully these teams are continuing their work in another venue.

Where we depart from Celtic-Madeira is the insistence that all these services be on-board the autonomic device. Autonomic behavior can also be reached by having the device seek and find these utility services in the network group in which it participates. This is the approach first realized by the now open source Jini Network

Architecture. It is also found in the concept of the *service registry* and realized in Microsoft's CSF.

These services are architectural building blocks, but they themselves do not guarantee an autonomic technical implementation. They must serve a broad concept that all devices interwork in multiple overlapping and/or orthogonal systems. Devices need to become semiautonomous artifacts, controlled by policy to react with adaptive behaviors, including and using well known features and functions of service control points in order to behave in a predictable and secure fashion, even faced with the potential chaos of complex networks.

Sometimes a service which is resident on a device will be owned or well known to the networks and companion devices, and at other times it will be novel. The networks need some way of trusting the device as it connects: that the device is who and what it claims to be and that the services it offers or loads are secure.

We have a lot of tools. For example, 3GPP optional onboard element agents are a start towards self management. RDF and the semantic network is a strong start on identity. But for a unified Device Management Framework, autonomic agents will be necessary. And to resolve the different management network groups to which devices can belong during their service life, collaborative interconnect architectures, such as the service aggregator, the rendezvous service, and persistent queue pipes are required.

The scope of what we mean by management will continue to expand....

The number of devices will continue to grow...

The capability of devices will continue to evolve in many different ways...

Device management as traditionally conceived seems inappropriate for the scale and diversity of the problem.

We must stop thinking of (i) product services, (ii) network software, (iii) network & service management, and (iv) device management as separate disciplines with differently skilled practitioners using very different reference architectures and their "only-invented-here" approaches. Convergence is occurring and must accelerate. The TMForum Device Management team understands this and needs help reaching a collective solution.

Yahoo's *Fire Eagle* is a harbinger of what is happening in the OTT service providers.

"Fire Eagle is a network service that gives users a place to store and manage information about their location, and offers developers protocols for updating or accessing that information. Yahoo! said its goal with Fire Eagle is to help developers create Internet experiences that are geo-aware. For users, Fire Eagle acts as a simple interface for managing location information and deciding how -- and with whom -- to share it." [Yahoo release reported in Techmeme]. Today's natural advantage of the network service provider is fast eroding. With it might go most opportunities for revenue from providing new services on devices. As Apple shows with the iPhone, it is certainly clear that management of devices will allow for advantage in selling services to device owners.

Service providers have a natural advantage, a bigger context to put it all together. We propose a composite **Service and Device Management Framework** that provides for (A) autonomic devices which connect to (B) a series of network-resident, server-hosted active models that virtualize the many interactions and groups in which the device, and its user/customer, is participating. Each device reports its image of the world about it to this virtual, composite model. There this information is matched to, and combined with, all the reports from all the other devices about it. Each device can really only know itself, but the network can collect state and data from all the plurality of interconnected devices: sometimes directly, sometimes trading for it through collaborative interfaces to other service providers. We call this network resident model the" **Operational Consciousness"** of the device. It can be used to apply broad management context to policy and behavior of the network and of the device - so that together, they reach efficient interactions. The fish can keep schooling and the birds keep flocking because the sonar image and the photo reside in the hosted virtual model. Someone will eventually provide the ultimate management service. We feel it goes beyond looking at components and frameworks. We see the opportunity to completely reinvent the Service Provider for the 21st century.

If you have news you'd like to share with Pipeline, contact us at editor@pipelinepub.com.