Chapter 1

Introduction to Autonomic Concepts Applied to Future Self-Managed Networks

Nazim Agoulmine

DEFINITION AND SCOPE

Network area has seen tremendous changes during the last decade through several waves. The first wave has been the Internet, which has completely changed the way network services are provided and helped global usage networked applications worldwide. The second wave came with the cellular and wireless technologies that have allowed the provisioning of telephony and data services anyplace, anytime. Cellular technologies have also drastically changed our behavior, allowing any person to be able to make phone calls from anywhere. They have also helped poor countries to develop efficient and cost-effective telephony infrastructure quickly, which was not possible with wire technologies. The third wave has surprisingly come from the POTS (Plain Old Telephony System) last mile access. Indeed, development of DSL (Digital Subscriber Line) technologies has enabled operators to provide high-speed IP access through a telephone line without having to pay for optical fiber installation and by taking advantage of existing telephone lines. This has allowed the emergence of the so-called Triple Play Services to the home (IP data access, Telephony of IP, and TV over IP). The fourth ware is probably under way with the convergence of the services Triple Play plus Mobile (sometimes called Quadruple Play) but also All-in-One emerging services such as P2P, social networking, and presence.

In this ideal picture of technology development, there are many behind-the-scenes issues. Indeed, slowly but steadily, the users' focus has changed from the high-speed network to value-added services. Users no longer care about the technologies of networks that are deployed (as they are lost in the different terminologies) but rather are more concerned about the quality of services they can use anywhere, anytime and at an acceptable cost. Internal architectures and protocol are only of interest to experts and are not important to customers whose

Autonomic Network Management Principles. DOI: 10.1016/B978-0-12-382190-4-00001-2

sole concern is the benefit they can get from the services. For the operator, however, internal architecture, protocols and so on are very important as they drive their capabilities to respond to customers' needs. Like any other system but with hundreds of orders of magnitude of complexity, the network has to change and evolve regularly. Operators are constantly integrating new services, new components, and new technologies without interrupting the ongoing one to fulfill new customers' needs, resolve problems, increase capacity, and the like. Every year operators' networks have become larger and more complex, dealing with a numerous heterogeneous sources (hardware, software, services, etc.). Relations between operators to allow services to span their networks to fulfill the end-to-end businesses of their customers have also added to this complex picture. And the picture becomes even more complicated as operators and equipment builders evolve in a highly revenue-generative but also deregulated area where they are all pushed to reduce their cost while facing newly entering competitors. These competitors are more aggressive as they usually enter with the latest technology without having to support several years of investments in older technologies.

On one side, the network complexity is increasing the need for more expertise and for more efforts from highly skilled people to maintain the infrastructure (Figure 1.1), and on the other side, the deregulated market is pushing for more competition and lower prices. It seems that the old operators and constructors have somehow found their way in the competition, though at a higher cost than the newly entered actors. The cost is of course related to CAPEX (Capital Expenditure) in the acquisition of new equipments and also OPEX (Operational

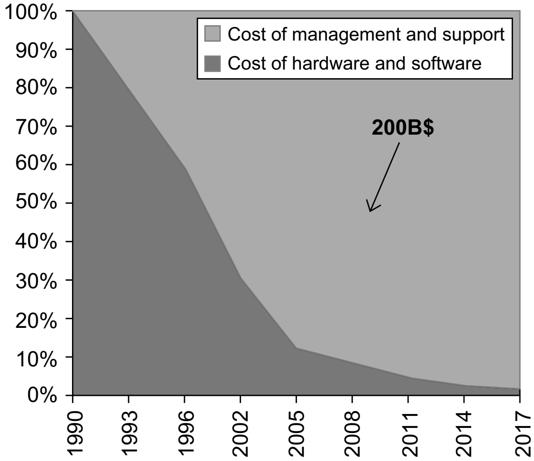

FIGURE 1.1 Evolution of the Costs of Hardware and Software and Their Management. The cost of management and support of computing and networking system has increased drastically in the past year due to the complexity of the technologies requiring ever more skilled engineers and administrators.

Cost) to control and maintain this increasingly sophisticated equipment as well as existing infrastructure and corresponding line of products.

This situation is very difficult for old operators and constructors as they could rely on huge benefits in the past; however, if a similar situation arises in the future, it will be much harder for them to survive as they will not have the necessary resources to compete.

We can easily imagine the emergence of a new network technology in the future that will be cheaper to deploy and maintain than the existing ones and could therefore supersede all existing technologies. If new actors enter the market with this technology, they will be able to provide better services at a lower cost while the remaining operators will still need to amortize their costly existing equipment and associated human resources.

Operators and constructors recognized this problem several years ago not only in the IT world but also in the networking area. Unfortunately, network and system management solutions made significant advances and are no more capable to deal with the increasing complexity; they still rely on very expensive and rare human experts to solve problems, which themselves are beyond the capacities of the experts. Many problems also arise from these experts' intervention, such as misconfigurations (wrong configuration, tuning, etc.). These misconfigurations are among the most complex problems to solve; they are very difficult both to understand and locate and therefore to fix. Operators now understand that it is vital for them to control this increased, uncontrollable operational cost (OPEX) (including the deployment cost) deploying breaking approaches.

The only response to this unsustainable situation is innovation in the way networks are managed and controlled. It is necessary to develop new networks that are able to automatically adapt their configurations to the increases and changing requirements of end users and service providers. Soon, we'll see drastic developments in the end users' services with the introduction of high-speed access networks that are either fixed with the deployment of FFTH or wireless with LTE and WiMAX technologies. Future networks need to be more flexible, capable of reorganizing in an autonomic way when new types of equipment or services are introduced, reducing the need for human intervention and consequently associated costs. Future networks should be able to improve their performances when needed to respond to unusual changes in the traffic pattern. The innovation should help to design new types of equipments, protocols, and network architectures and even services that are able to be self-managed, to reduce the operational burden on the operators by themselves making decisions in terms of configuration, optimization, and the like.

If networks and services are able to exhibit some level of autonomy that will allow them to themselves solve their problems in any context, then the operator will be able to reduce the need for intervention by human experts and therefore reduce their operational costs (OPEX). It is time that significant progress be made in how to manage and control these complex infrastructures at the early stage of their design.

Many initiatives have been launched to push toward innovations in this area. These initiatives have different names, but all converge to the emergence of a new generation of intelligent equipments, networks, and services that are able to exhibit self-properties. These initiatives are variously named—for example, Autonomic Communication (AC), Autonomic Networks (AN), Autonomic Network Management (ANM), Self-Managed Networks (SFN), Situated Networks (SN). Differences in the focus of the various approaches can explain roughly the differences in the terminology, but all of them have one thing in common: They all seek to introduce self-adaptive capabilities in the network, avoiding human interventions as much as possible.

EPIDEMIOLOGICAL DEFINITION OF AUTONOMICS

According to the Oxford English Dictionary, "autonomic" is the adjective derived from "autonomy," meaning self-governing or independent [1]. With respect to physiology, the autonomic nervous system is that part of the human body that functions independently of the will. The Cambridge physiologist John Newport Langley (1852–1925) was the first scientist to apply this term, in his publication in the *Journal of Physiology* [2] in 1898: "I propose the term 'autonomic nervous system' for the sympathetic system and allied nervous system of the cranial and sacral nerves, and for the local nervous system of the gut."

The autonomic nervous system (ANS) has of course an important role in the biological system as it regulates involuntary activity in the body by transmitting motor impulses to cardiac muscle, smooth muscle, and the glands. The ANS controls all the vital muscular activities of the heart and of the circulatory, digestive, respiratory, and urogenital systems. The autonomic nervous system governs our heart and body temperature, thus freeing our conscious brain to deal with higher level activities.

THE NEED FOR AUTONOMIC SYSTEMS

The tremendous complexity of computing systems during the last decades has exponentially increased the management and operation expenses of these systems. Operators and system administrators are envisioning IT systems that can self-govern and solve their configuration problems to achieve objectives in an autonomic. Although this idea has been the subject of many works in the area of artificial intelligence (AI), what is different today is that on one hand technologies have evolved in an impressive way, allowing new types of solutions, and on the other hand the operator's requirements are much more precise than the general case AI tried to solve in the past.

IBM was the first company to use the business keyword autonomic computing (AC), aiming at developing a new generation of intelligent computer systems [6]. Paul Horn of IBM introduced AC in October 2001 in a pioneer work in the new wave of autonomics. AC is used to embody self-managing. In the IT industry, the term *autonomic* refers to the ability of a component to self-manage based on internal stimuli. More precisely, “autonomic” means the act of acting and occurring involuntarily—that is, systems that are able to manage themselves based on high-level administration objectives. With AC human administrators would no longer need to deal with low-level management and could then concentrate on the higher level management process.

AUTOMATIC, AUTONOMOUS, AND AUTONOMIC SYSTEMS

There are some semantic differences between autonomic, autonomous, and automatic that can be summarized in the following definitions [see Collins, *Educational Dictionary Millennium Edition,* 2000] [21]

- **Definition of Automatic (Adv Automatically)**: An autonomic action is an action that is performed from force of habit or without any conscious thought. Many examples can of course be found in the bio-system or artificial system. The human body is able to perform a number of reflex or involuntary actions, while an automatic system is designed to perform some specific actions as a consequence of some occurring events or known problems. Automatic systems do not have any knowledge outside the predefined one and no ability to extend it. An automatic system will always exhibit the same behaviors for the same input. In automatic theory, this behavior is called transfer function. Even though the word “automatic” comes from the Greek word *automatous*, which means acting for one’s own will, it has become associated with mechanical terms that are predefined and not running of one’s free will [22].
- **Definition of Autonomous (Adv Autonomously)**: An autonomous system is a system that exhibits a large degree of self-governance. This system takes its decision without referring to any external entities and in complete independence. The autonomous entity defines its own rules and principles to achieve its own goals. An autonomous behavior is the ultimate freedom.
- **Definition of Autonomic (Adv Autonomically)**: The word “autonomic” suggests the idea of self-governance within an entity based on internal policies and principles, which can also be described as autonomics. Autonomic relating to the autonomic nervous system (ANS) is based on internal stimuli that trigger involuntary responses. In medical terms, autonomic means self-controlling or functionality independent.

The autonomic system in the IT world uses a holistic approach to the design of highly distributed and complex distributed computing environments resulting in self-managed systems. Autonomic systems are inspired by ANS, where ANS manages the important bodily functions devoid of any conscious involvement. It describes a system in which humans define goals as input to a self-managing autonomous distributed and more likely heterogeneous system.

IBM'S APPLICATION OF AUTONOMICS TO COMPUTERS

IBM has introduced an evolutionary process to evaluate the level of autonomic behavior in a computing system. The approach also defines the basic self-managing concept. This evolutionary process defines different levels of evolution of the system management process and describes how this process evolves with the adoption of autonomic concepts and technologies [5]. The five levels of evolution toward effecting a fully autonomic system are as follows:

- ***Level 1 or Basic Level.*** In this system, the operator needs to monitor and configure each element manually during its entire life cycle from installation to uninstallation.
- ***Level 2 or Managed Level.*** In this level, the system operator can take advantage of a set of system management technologies to monitor multiple systems and system elements simultaneously using management consoles with appropriate human machine interfaces.
- ***Level 3 or Proactive Level.*** Advances in analytical studies of the system allow the development of a system with predictive capacities that allow analyzing gathered information and identifying and predicting problems and therefore propose appropriate solutions to the operator of the system for deployment.
- ***Level 4 or Adaptive Level.*** In this level, the system is not only able to gather monitored information and predict situations but also to react automatically in many situations without any human intervention. This is based on a better understanding of system behavior and control. Once knowledge of what to perform in which situation is specified, the system can carry out numerous lower level decisions and actions.
- ***Level 5 Autonomic Level.*** this is the ultimate level where the interactions between the humans and the systems are only based on high-level goals. Human operators only specify business policies and objectives to govern systems, while the system interprets these high-level policies and responds accordingly. At this level, human operators will trust the system in managing themselves and will concentrate solely on higher level business.

These levels range from a totally operator-managed system to an autonomically managed system based on high-level objectives. The goal of the autonomic research domain is to achieve a fully Level 5 system by researching methods and innovative approaches leading to the development of an autonomic system.

IBM AUTONOMICS COMPUTING

IBM breaks down autonomic self-management into four categories: self-configuration, self-healing, self-optimization, and self-protection. The overall goal of these concepts is to deliver a self-managing solution that is proactive, robust, adaptable, and easy to use. In accordance with Roy Sterritt's Autonomic Computing Tree 0[7], the Autonomic Computing initiative will require four primary objectives known as CHOP [8]: Self-Configuration, Self-Healing, Self-Optimization, and Self-Protecting. Figure 1.2 lists the attributes of self-Aware, environment-aware, self-monitoring, and self-adjusting, which are necessary in achieving the CHOP objectives. The last branch on this tree lists the approaches that must be undertaken to provide an autonomic solution.

Systems should automatically adapt to dynamic changing environments [5]. This self-configuration objective of self-management enables the introduction of new entities such as new devices, roaming devices, software services, and even personnel, into a system with little or no downtime or human interaction. Configuration will be under the control of well-defined SLA (service level agreements) or high-level directives or goals. When a new entity is introduced, it will integrate itself effortlessly, making its presence and functionality public knowledge while the host system and existing entities reconfigure themselves if necessary.

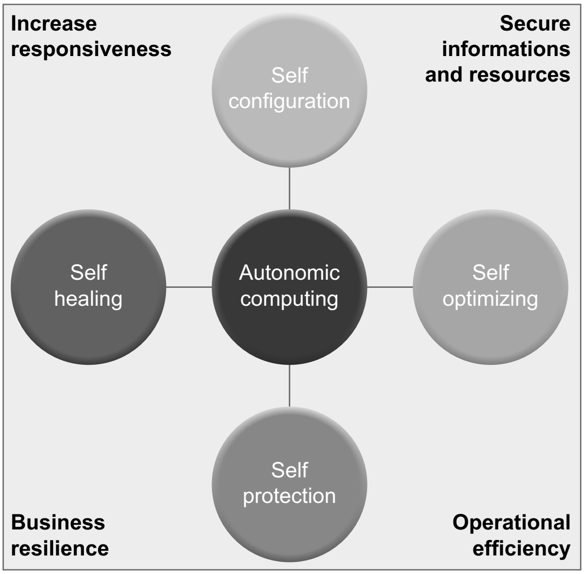

FIGURE 1.2 IBM Autonomic Computing Self-CHOP. By implementing self-CHOP Management, computing systems are themselves able to solve any operational problems related to configuration, healing, optimization, or protection reducing the burden on human administrators.

- Autonomic manager (AM) implements a control loop called MAPE that implements the life cycle from monitoring managed elements and autonomically modifies their state to fulfill the administrator goals:

TABLE 1.1

Monitor Function	Allow the AM to collect, aggregate, filter, and report details (e.g., metrics, topologies) from the underlying management element(s) under its responsibility.
Analyze Function	This allows the management element (ME to analyze the collected details to understand the current system state. This analyze function requires the use of complex models of the various situations in which the management elements(s) could evolve.
Plan Function	Once the situation is identified, the ME needs to define the set of actions needed to achieve the high-level goals and objectives.
Execute Function	This function allows the ME to change the behavior of the managed resource using effectors.
Managed Resource	The managed resources are the underlying controlled system components. These can be a server, a router, a cluster or business application, and so on.
Manageability Interface	The manageability interfaces are all the management services that are made available by the managed resource to manage them, such as the sensors and the effectors used by an autonomic manager.
Sensor	The sensor interface allows retrieving information about the current state of a managed resource. It also allows receiving asynchronous events (unsolicited, asynchronous messages or notifications) that can occur.
Effector	The effector interface allows the change of different aspects of the state of the managed resource to influence its behavior.

FROM AUTONOMIC COMPUTING TO AUTONOMICS NETWORKING

Similarly to autonomic computing that studies the possibilities of developing new computer that are able to manage themselves, the idea of autonomic networking research is to study new network architectures and protocols in order to develop networks that can behave with a certain level of freedom and adapt themselves efficiently to new situations (possibly unknown initially) without any direct human intervention [9]. Systems management process can be simplified by automating and distributing the decision-making processes involved in optimizing the system operation. This will save cost, enabling expensive human attention to focus more on business logic and less on low-level device configuration processes.

> *"Device, system and network* ***intelligence*** *that enable the services and resources offered to be* ***invisibly adapted*** *to changing user needs, business goals, and environmental conditions according to* ***policy*** *and* ***context****."*
>
> —John Strassner

It is very important to understand how to build such a dynamic system in which any change in an individual element (networked elements) could affect many others and vice versa.

Therefore the main complexity involved in applying the autonomic concepts to networking lies precisely in understanding how desired behaviors are learned, influenced, or changed, and how, in turn, these affect other elements, groups, and networks. This could concern any aspect of the network or support services such as configuration, performance, security, or routing.

An Autonomic Network Elements (A-NE) (Figure 1.3) performs the following tasks:

TABLE 1.2

Sensing its environment	This is similar to the sensors in the autonomic computing architecture. The A-NE should continuously monitor the managed element (s) under its control using different types of sensors that could be software or hardware local or remote. Sensors in this case should be able to intervene at different levels of the communication stack (hardware, protocol, service, application, etc.), which makes it very complex.
Perceiving and analyzing the context	When information is collected from the sensors, the A-NE needs to understand its context. Indeed, the information collected could have different meanings based on the context in which the A-NE is evolving. The network environment is intrinsically very complex. Therefore, it is a very difficult task as the A-NE will need to interpret heterogeneous monitored information using other levels of information (local or global). The information can be related to different aspects of the network or the supported services. Historical information is very important to analysis of the context and understands in which situation the A-NE is now. Based on high-level goals, the A-NE will try to maintain itself in a set of "desired states" according to the context. Otherwise, the A-NE will need to autonomously perform some actions to change its state and try to reach a desired one. The desired state could sometimes be the optimal one, but in some situations, this state could only be

(*Continued*)

TABLE 1.2 (*Continued*)

	a safe one that would ensure that the system will always deliver the service. This state could then be changed to move toward an "optimal one."
Learning	During its lifetime, the A-NE will face different contexts and situations to which it has reacted, implementing different strategies to always fulfill its assigned goal. During these trials, the A-NE will be able to evaluate the usefulness of the implemented situation and learn from the performed action to adapt itself to future known or unknown situations. Autonomic adaptation is the capability by which the A-NE will improve by learning (increasing its knowledge) the best strategies to plan in order to react to situations. This is what humans do to improve their knowledge and skill but in autonomic networking, this should be an inner capacity of network elements.
Participating in Groups	A-NE cannot improve their knowledge if they do not interact with other A-NE to improve its knowledge and skill. A-NEs need to communicate, collaborate, andexchange information and knowledge to improve their capabilities to solve problem, better their performance, and secure themselves. These group interactions are also important in collectively achieving a global goal that cannot be reached without a certain level of coordination. These communications should be achieved within purposeful (structured and unstructured, ad hoc) groups or clusters. Information should be understandable to the A-NEs, though it is exchanged by autonomic entities.
Planning	Once the context is identified and its situation is evaluated, the A-NE should define the strategy (list of actions) that should be taken to either reach a "desired state" in case it is in an "undesired state" or to reach another "desired state" that is better from a different perspective, that is, performance, security, organization, and the like. Therefore the planning process will encompass a set of strategies that allow the A-NE to continuously fine tune the underlying managed elements and adapt to new contexts while always seeking to be in "desired states." With the distributed nature of the network, the planning can be very difficult as it is not possible to enforce an action instantaneously; when a set of actions are identified, it is not possible to activate them also at the same time. As the A-NEs take their decision in an autonomic way, a consistently among the actions should be ensure in a completely distributed and decentralized way which is a real challenge. Here we seek convergence as actions can be inconsistent, and therefore convergence time becomes an important aspect.
Actuating its state	Finally, the A-Ne should have the full control of itself and the parameters that affect its local behavior. This shall happen through a set of actuators that are linked to the underlying physical and logical resources that are part of the A-NE's boundaries.

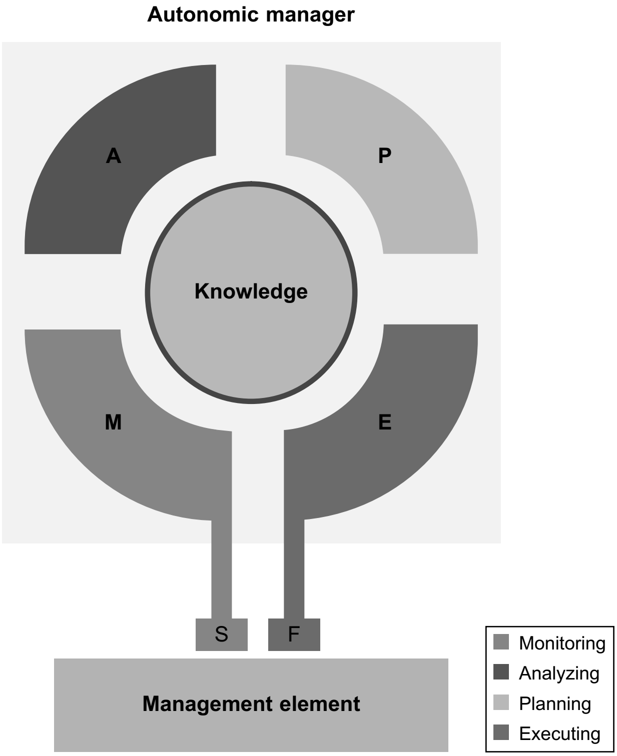

FIGURE 1.3 IBM Autonomic Element Architecture and Inner MAPE Loop. The autonomic element continuously senses the underlying management element state and executes the MAPE loop to identify the appropriate actions to enforce so that the management element is always in the desired state.

AUTONOMIC (NETWORKING) DESIGN PRINCIPLES

Autonomic networking can be built on a set of design principles that have already been proven to provide some level of autonomic behavior in many areas. Many research projects have followed different directions in achieving autonomic systems with different levels of success. Indeed, the principles of autonomic behavior already exist, not only in the natural system (nature, biology), the social environments (society, communities, etc.), but also other areas of IT in fields such as space, vehicular, robotics. These principles can be applied solely or collectively to build the foundation of autonomic networks.

When studying these areas, it is possible to identify some general design principles that can help to build autonomic systems. These principles can also be used in the area of networking, taking into account its particular specificities such as heterogeneity, scalability, and distribution.

Living Systems Inspired Design

Living systems have always been an important source of inspiration for human-designed systems. Living systems exhibit a number of properties that make them autonomic, and their understanding is valuable for the design of artificial autonomic systems [19]. Among these properties two characteristics are especially interesting for the design of autonomic systems (1) bio-inspired survivability and (2) collective behavior.

- Bio-Inspired Survivability

  The body's internal mechanisms continuously work together to maintain essential variables within physiological limits that define the viability zone. In this very complex system, adaptive behavior at different levels is directly linked with the survivability. The system always tries to remain in an equilibrium zone, and if any external or internal stimulus pushes the system outside its physiological equilibrium state the system will autonomically work toward coming back to the original equilibrium state. This system implements what is called positive and negative forces [28]. The system implements several internal mechanisms that continuously work together to maintain essential variables within physiological limits that define the viability zone

  In the biological system, several variables need to be maintained in the viability zone, that is, upper and lower bounds (e.g., sugar level, cholesterol level, body temperature, blood pressure, heart rate). However, environmental change may cause fluctuations (food, efforts, etc.). The autonomic mechanisms in the body continuously control these variables and maintain them in the viability zone (Figure 1.4). This is called homeostatic equilibrium. Three types of adaptation to environmental disturbance are available to higher organisms:

TABLE 1.3

Short-term changes	This adaptation allows the biological system to respond to a stimulus immediately: For example, an environmental temperature change moves the body temperature variable to an unacceptable value. This rapidly induces an autonomic response in the (human) organism, that is, either perspiring to dissipate heat or shivering to generate heat. Such adaptation is quickly achieved and reversed.
Somatic changes	Prolonged exposure to environmental temperature change results in the impact of the change being absorbed by the organism, that is, acclimatization. Such change is slower to achieve and is reversible once the individual is no longer in that specific environment.
Genotypic changes	A species adapts to change by shifting the range of some variables, for example, in a cold climate a species may grow thicker fur. Such genotypic change is recorded at a cellular level and becomes hereditary and irreversible in the lifetime of the individual. The adaptation here is through mutation and hence evolution.

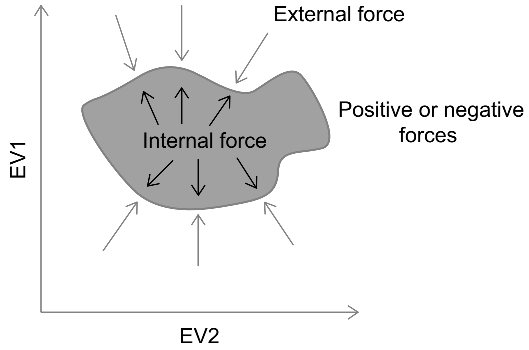

FIGURE 1.4 Biological Physiological Equilibrium. The environment in which the biological system evolves is always pushing it outside its viability zone. The biological systems have inner autonomic physiological mechanisms that create reverse forces to maintain the bio-system in viability. (This equilibrium is called homeostasis equilibrium.)

These adaptation mechanisms have a primordial role in the biological system's ability to adapt to changes in environment for survivability and evolution toward more sophisticated systems. This can and should inspire the design of future autonomic systems as these intrinsic properties of biological systems could similarly benefit artificial autonomic systems such as autonomic networks [39].

- Collective Behavior

  The term *collective behavior* was coined by Robert E. Park and employed by Herbert Blumer to refer to social processes and events that do not reflect existing social structures (laws, conventions, and institutions), but that emerge in a "spontaneous" way [31]. Among the collective behavior one needs to understand, "social movement" is particularly relevant to understand—notably, an understanding of how from an individual autonomic behavior some general behavior emerges. Social movement is a form of collective behavior that is identified by a type of group action performed by the individuals within the movement. It generally emerges from a large informal grouping of individuals and/or organizations focused on specific political or social issues. The growth of the social movement is sometimes not under any control and emerges because many factors and circumstances take place at the same time. Recently, some modern movements have utilized technology such as cellular networks with the SMS service or the Internet with Twitter to mobilize people globally. From this point of view, it is interesting to notice how local information becomes global and can influence the behavior of all the members of the community. The architecture of autonomic systems can learn a lot from such social collective behavior, which represents a real large-scale prototyping of autonomic entities (humans) interacting to fulfill different and sometimes conflicting goals. In this context, economists and social science researchers have made use of game theory to try to model the behavior of individuals able to take autonomous decisions in strategic situations. The objective is to understand how an individual's success in making choices depends on the choices of others.

The objective is to identify potential equilibrium which is beneficial to all parties. In an equilibrium state, each player of the game has adopted the most appropriate strategy and any change in the strategy will not be beneficial to any of them. Among these equilibriums, the Nash equilibrium is an attempt to capture this idea. These equilibrium concepts are motivated differently depending on the field of application, so do Autonomic Networking. This methodology is not without criticism, and debates continue over the appropriateness of particular equilibrium concepts and the appropriateness of equilibrium altogether in term of fairness. For example, what is the best configuration of a base station to share the radio resources among a number of users attempting to access the network? AN-Es could have to take their own decision in their own situation (local knowledge) and maybe a partial knowledge of what is happening at the global level.

These collective behavior studies can inspire the design and development of novel network architecture that exhibit flexible and dynamic organization. This collective behavior could rise from individual AN-Es interacting and reorganizing among themselves according to some high-level objective such as resource management, topology change, or economic need. Figure 1.5 shows how an individual AN-E seamlessly integrates an already organized network (e.g., ad hoc network, new network equipment in an existing infrastructure network, etc.). The figure also shows how two networks can integrate seamlessly together without any manual intervention.

Policy-Based Design

"Policy is a rule defining a choice in the behavior of a system" [11, 12]. Policy concept has already been widely used in the area of networking to introduce some level of automation in the control and configuration of network equipment behavior based on a set of predefined event-condition-action rules defined by the administrator. A policy is typically described as a deliberate plan of action to guide decisions and achieve rational outcome(s). The human expert defines the right policies, and therefore these policies are enforced in the network [14]. Once enforced, the network could govern itself, freeing the administrators from having to deal with all the known situations and appropriate reactions. Policy-based behavior is very useful but cannot be used by itself because situations can change and it is not always possible to know about all situations and the autonomic network needs to exhibit other properties such as adaptability to varying contexts and situations [15]. Policy-based design permits a level of separation of concerns to facilitate the design of the autonomic system. This will be made feasible by means of behavior definitions and conflict resolution [11–13]. It is thought that a number of *meta-roles* associated with the

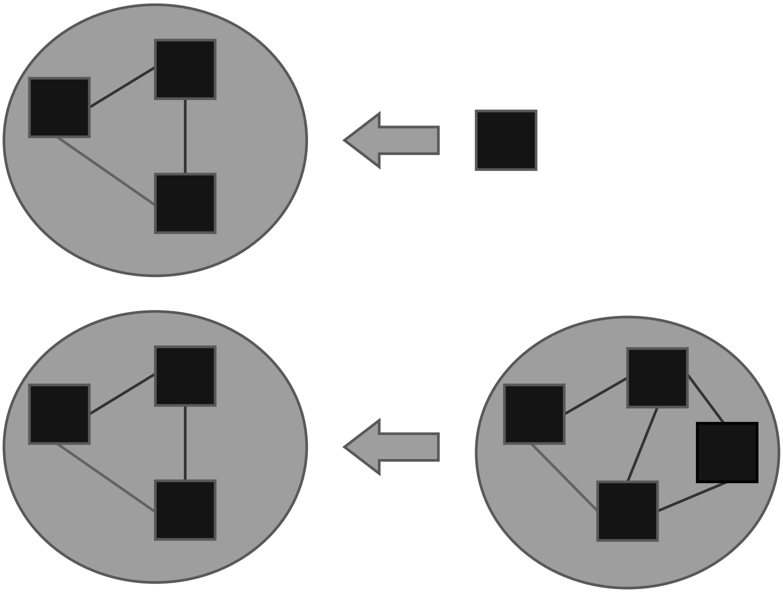

FIGURE 1.5 Autonomic composition et decomposition of Autonomic Networks from individual or group of Autonomic Element(s). Collective behavior in a society helps one to understand how individual interactions lead to more efficient collective behavior. The self-organization and communications mechanisms that are used in this context could inspire the development of autonomic networks exhibiting the same interesting properties.

role's actor, such as role director and role observer, might be needed. Role actors become active when triggered by role-specific headers found in IP datagrams. Role directors are triggered by a need to specify/activate/enforce new behavior at associated role actors [40]; role observers are triggered (as specified in their behaviors) by any of the above events pertaining to the task of monitoring or auditing the behaviors of role actors and directors.

Autonomic networking could be inspired by the policy-based design (Figure 1.6) proposing a proactive approach, in which autonomic networks will self-organize based on these policies [10]. Policies can be used at different levels of the autonomic loop: monitoring, analyzing, planning, and enforcing. Autonomic networking could take advantage of policy monitoring, policy analyzing, policy decision, policy enforcement logical points, and so on, to implement the autonomic loop. Of course, the same conflict problems could rise from using these policies at different decision levels, and this approach will also require defining efficient conflict resolution mechanisms.

Context Awareness Design

Annind Dey defines context as "any information that can be used to characterize the situation of an entity" [17]. Dr Annind K. Dey from Berkey Research Labs defines context "as any information relevant to an interaction that can be used

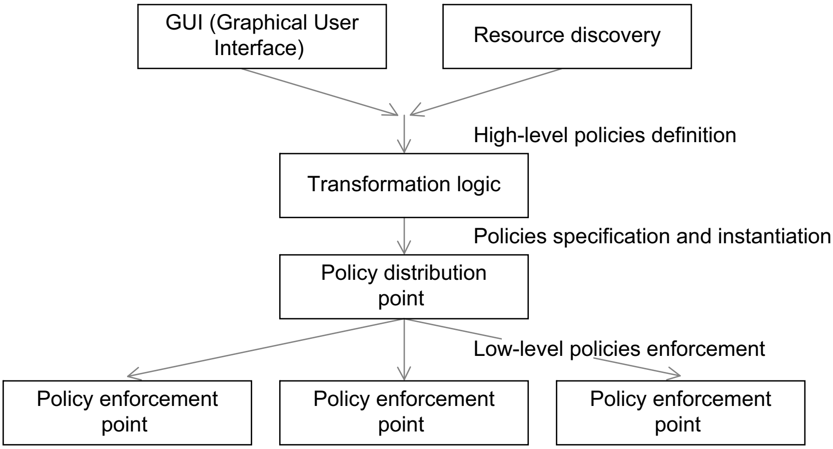
GUI (Graphical User Interface)
Resource discovery
High-level policies definition
Transformation logic
Policies specification and instantiation
Policy distribution point
Low-level policies enforcement
Policy enforcement point
Policy enforcement point
Policy enforcement point

FIGURE 1.6 Policy Based Design. High-level policies are specified by experts and introduced in the system to be enforced in the network elements. To ease the work of the administrator, high-level policy languages are used, and the system introduces several levels of transformation to transform the high-level policies into low-level policies that are distributed by the policy decision point to the different policy enforcement points that control the network elements. Policy-based design gives administrators a way to delegate to the system some level of control of the underlying network elements.

to characterize the situation of an entity: An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [18]. In general, we can define a context as a set of rules of interrelationship of features in processing any entities (including the entity solely). Computer science has introduced the idea of context awareness in order to develop applications that can sense their environment (localization, time, user identity, etc.) and react accordingly. Chen and Kotz argue that context can be categorized as having two main aspects: active context and passive context [20]. Active context is context information that influences the behaviors of an application, whereas passive context represents information that is relevant but not critical to an application. Gwizdka distinguishes between context that is internal and that which is external to the user. Internal context encompasses work context, while the external context defines the state of the environment. In autonomic networking, J. Strassner presents a more abstract and extensible definition of the context as follows: "The Context of an Entity is the collection of measured and inferred knowledge that describes the state and environment in which an Entity exists or has existed" [19]. Indeed, the idea is that A-NE could take into account the information about the circumstances under which they are able to operate and react accordingly. Context awareness is also sometimes extended to the concept of situation awareness, which aims also to make assumptions about the user's current situation. But this is somehow limiting as even a machine, service, or protocol, and not just a user, can be in a particular situation. From a practical point of view, the main problem with the context awareness design lies in deciding how to model the context in order to capture all the states as well as the semantics of the interactions. For example, Strassner proposes an approach whereby the context of an entity is modeled as a collection of data, information, and knowledge resulting from gathering measurements of and reasoning about that entity. The originality of the approach is the use of modeling patterns and roles to feature an extensible representation of context. The proposed model could then be refined or extended to deal with specific domains.

Self-similarity Design Principle

The term *autonomic computing* also refers to a vast and somewhat tangled hierarchy of natural self-governing systems, many of which consist of myriad interacting self-governing components that in turn comprise large numbers of interacting autonomous, self-governing components at the next level down as introduced by Jeffrey O. Kephart and David M. Chess from IBM [15]. This vision of autonomic computing applies also to networking area however with more complexity than in computing systems due to the distributed, large scale and heterogeneous nature of networks. Interacting autonomic element need to organize among themselves so that some system level properties autonomically emerge from that organization and make the whole system autonomic (Figure 1.7). Such a scalable organization can also be found in what is called self-similarity. This design principle suggests that the same organization will be found at different scales of the system. This allows using the same template to create higher level templates having the same properties such as ice snowflakes or glass cracks. The same principle can be used to design the autonomic system at a small scale, which is then able to grow to large scale while maintaining small-scale properties. This will simplify the design of the system and solve the

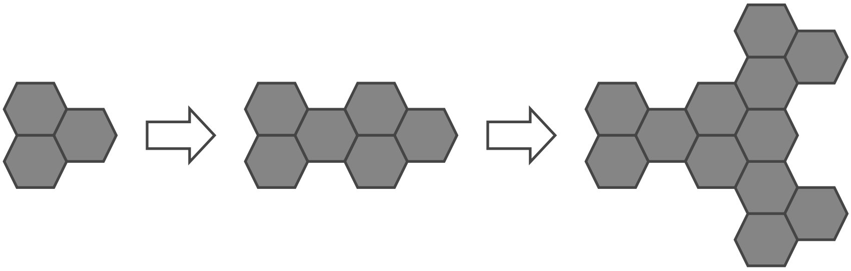

FIGURE 1.7 Self-Similarity as a Design Principle for Autonomic Networks. The natural and biological systems have shown very interesting self-similarity properties allowing them to grow in a structured way and exhibiting scalability properties. Scalability is an important problem in autonomic systems that is difficult to reach and sometimes to prove. Inspiring from self-similarity properties of exiting natural systems could help to design an autonomic system with the required properties.

problem of scalability, which is one of the main issues involving the Internet today.

Self-similarity will also permit achieving self-organization in the autonomic system. S. Camazine [23] defines self-organization as “a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system.” In this self-organization schema, autonomic entities have access only to local information without reference to the global pattern. Still, the system as a whole is able to achieve a global pattern. In the autonomic network context, this aims at designing autonomic network elements that are able to organize among themselves using their own internal processes and not relying on any global entities to control them. This self-organization is the only guarantee of the same properties exhibited at the large scale as those exhibited at a smaller scale. These could be any aspect of interest in networking such as performance, security, and reliability.

Adaptive Design

By definition an autonomic network should be adaptive—that is, be able to change its operations, state, and function to cope with situations that could affect the general goals for which it was built. A-NE should be designed so that their inner behavior can adapt to various situations.

According to Laddaga, self-adaptive software may be defined as “software that evaluates and changes its own behavior when the evaluation indicates that it has not [been] accomplishing what it is intended to do, or when better functionality or performance is possible” [26]. The self-adaptive approach presents an attractive concept for developing self-governing systems that partially or fully accommodate their own management and adaptation activities. Self-adaptation in the design of A-NE can be fulfilled in a local or coordinate way but should always seek to put the global network in a desired state. The adaptation capabilities of each A-NE should be able to deal with temporal and spatial changes (e.g., topology change, structural change), operational changes (e.g., malicious attack, faults), and strategic changes (e.g., objectives of the operators, SLAs). The temporal adaptation could be inspired from the adaptation mechanisms in biology as described in the bio-inspired design. The Autonomic Computing approach suggests an adaptation inspired from the automatic theory to implement a first and fast adaptation loop called the automatic loop. This approach is also applied to the area of networking, however this requires another level of adaptation called cognitive loop that aims to improve the efficiency of the first loop in fully distributed manner over time. This suggests that the A-NE will improve its inner capacity to react to situational changes by learning in the long term from the experiences while with the automatic loop; it is capable of responding to immediate situations enforcing the best actions with the available knowledge. As its knowledge increases in time, A-NE’s skill to respond efficiently to various known and unknown situations will also increase.

Knowledge-Based Design

The knowledge plane for the entire chapter is a high-level model of what the network is supposed to do. Some researchers argue that there can be no autonomic network without building an efficient and complete knowledge plane able to capture all the properties of underlying network, protocols, and supported services. However, the heterogeneity, complexity of interacting, and manipulation underlying technology needs to disappear from the user's perspective. The so-called Future Internet is based on the knowledge plane concept, which is a kind of meta-control plane for future intelligent management of the Internet [16]. The knowledge plane was originally proposed as a research objective designed to build "a fundamentally different sort of network that can assemble itself given high level instructions, organize itself to fulfill new requirements change, automatically discover new autonomic elements and integrate them seamlessly when necessary and automatically tries to fix detected problems" [33]. The used term plane comes from the fact that network functionalities are organized into layers (Figure 1.8), each layer is responsible for the management of its own data or information. The knowledge plan come above these layers to aggregate and give semantic meaning to all underlying information as well to any new knowledge that helps the network to fulfill its objectives. Hence, the knowledge plane was envisioned as "a new construct that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network" [37]. This approach advocated the use of cognitive and artificial intelligence (AI) techniques to achieve the above goals.

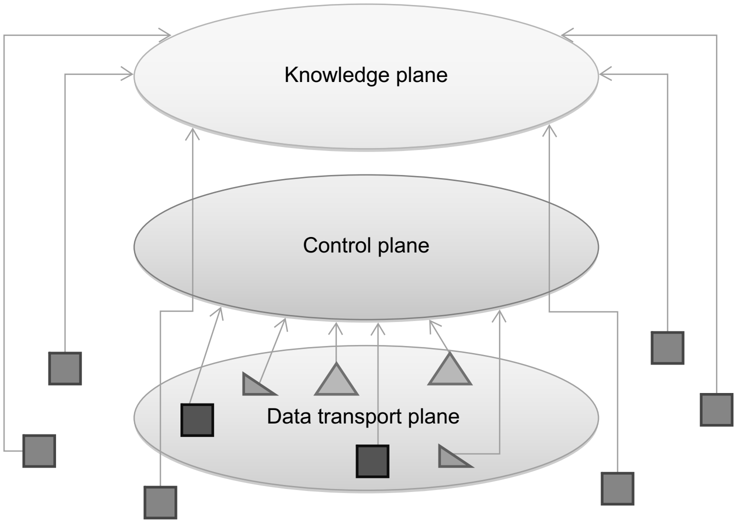

FIGURE 1.8 Knowledge Plane: The knowledge plane is a high-level model of what the network is supposed to do. It constitutes a lingua franca build on information models and ontologies, which can serve as a lexicon to translate between different heterogeneous control interfaces. It constitutes a meta-control plane for future autonomic networks.

The knowledge plane is based on three principal mechanisms: a new architectural construct that is separate from the existing data and control planes; the ability to harmonize different needs; and a cognitive framework (giving it the ability to make decisions in the presence of partial or conflicting information).

Although the knowledge plane approach appears to be very promising, many issues need to be addressed to build autonomic networks. Indeed, existing contributions on knowledge plane do not define how to express business goals and how to link these goals to underlying services and resources. While existing data models are excellent at representing facts, they do not have any inherent mechanisms for representing semantics, which are required to reason about those facts [29]. To build such as knowledge plane, it is necessary to specify how to work with heterogeneous technologies and devices. Several ontology-based approaches have been proposed to solve this problem of integrating heterogeneous domains and to dynamically discover the services and capabilities of autonomic entities. Ontology provides a vocabulary of classes and relations to describe a domain aiming at representing knowledge and sharing it. First, the Web Service Definition Language (WSDL), a standardized service description language of the World Wide Web Consortium, has been used to describe the functional aspects of services by defining the semantics of their input and output parameters, but it remains limited as the internal semantic of the service is not described. For that, OWL-based Web Service Ontology (OWL-S) is being positioned as a good candidate to enhance the descriptions of these services using ontology-based semantics. In general, the Web Ontology Language (OWL) is considered as a candidate for knowledge representation in autonomic ystems and more particularly the contextual information as well as the interactions between the entities and how a change in the state of an entity will affect the state of the other entities [25]. From this point of view, OWL-S marks interesting progress in the area as it provides an unambiguous, computer-interpretable semantic description of a service by providing rich definitions of the semantic of the IOPE (Inputs, outputs, preconditions, and effects), in addition to the description of the resources used by the service [24]. In a service-oriented architecture (SOA), which could constitute the foundation for the interaction between A-NEs, the semantic specification of the IOPEs of the A-NE services could help to specify the knowledge A-NEs require to interact, compose, and cooperate to fulfill their global goal. This should also be associated with some level of reasoning.

While these approaches sound promising, there is still a long way to go before the objective of this knowledge plane specification and instrumentations can be achieved.

FROM AUTONOMIC NETWORKING TO AUTONOMIC NETWORK MANAGEMENT

Autonomic network management is one facet of autonomic networking that focuses on developing new solutions to allow the networks to self-manage.

Traditional network management solutions called Simple Network Management Protocol and Common Management Information Service/Protocol have shown limitations with regard to the increased scale and complexity of existing networks as the intelligence of solving the problems was always outside the network and usually humancentric. Policy-based management solutions have certainly provided a certain level of simplification by automating some aspects of management, but not enough to cope with the ever increasing complexity. The objective of autonomic network management is to investigate how to design new management solutions that will be able to cope with the increasingly complex, heterogeneous scalability of today's and future networks. The solution should benefit from the autonomic concepts to reach the required flexibility and adaptability to deal with any unforeseen situation. The idea behind the autonomic network management solutions is to develop management systems that are capable of self-governing and reducing the duties of the human operators who are not able to deal with increasingly complex situations. The systems should exhibit some level of intelligence so that their capability can improve over time, assuming more and more tasks that are initially allocated to skilled administrators. Humans will only need to interact with the system using some high-level goal-oriented language and not any low-level commands as is true today. This autonomic management of the networks and services will not only improve the end users' quality of service, as problems and quality degradation will be solved much quickly, but it will also reduce operational expenditure for network operators.

As presented earlier, the autonomic network as well as autonomic network management could make use of different techniques to exhibit the required properties. In autonomic network management, human management goals should be dynamically mapped to enforceable policies across the A-NE across the network. A-NEs should exhibit autonomic behavior in term of adaptation to changing context, improving at each stage their capacity to find better solutions. Some research entities think that these adaptations should be constrained by some human-specified goals and constraints, while others think that the emerging behaviors and collective behaviors will freely reach optimum equilibrium without any human intervention.

From an operator's point of view, the full freedom of the network is difficult to accept today, so "self-management" would be appropriate only if it were to be overseen or governed in a manner understandable to a human controller. Experience with the Internet in the past has shown that several visions could coexist.

Autonomic network management presents many challenges that need to be addressed. Among these challenges, the smooth migration from existing management to fully autonomic network management will require an accurate mapping between underlying data models and high-level semantic models in order to efficiently control the underlying heterogeneous network equipments and communication protocols (Figure 1.9). On the reverse side, a high-level governance directive should also be correctly mapped down to low-level adaptation and

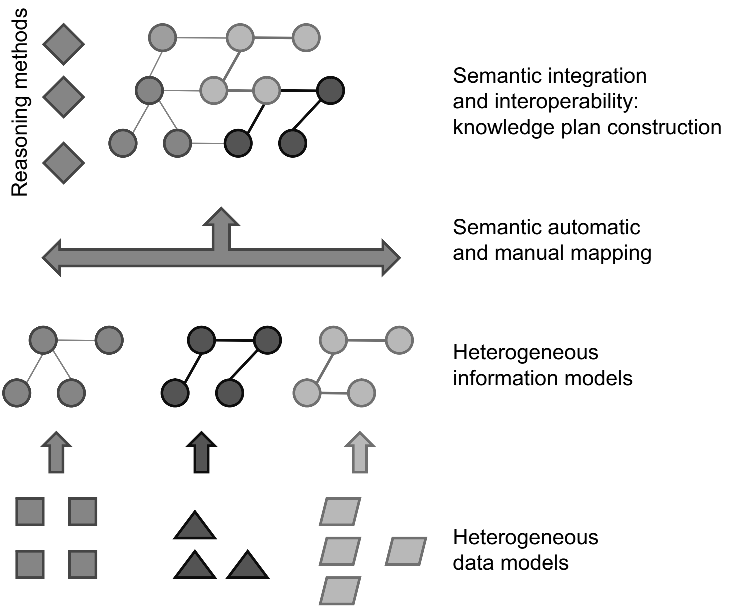
Reasoning methods
Semantic integration and interoperability: knowledge plan construction
Semantic automatic and manual mapping
Heterogeneous information models
Heterogeneous data models

FIGURE 1.9 Integration of Heterogeneous Underlying Management Information Sources. Autonomic network management cannot be achieved without a unified knowledge plan gathering all the knowledge about the underlying possibility of the complex and heterogeneous network elements to control. A high-level semantic model (sometime called knowledge plan) can be built based on a transformation process of models starting from the underlying data models up to the knowledge plan. Autonomic Network Management can then be achieved by defining appropriate Self-* mechanisms that implement the MAPE loop using this knowledge plan.

control policies to be enforced in the individual heterogeneous elements. This mapping can be even more complicated when one takes into account the context of a specific service chain or flow within a more richly connected network of managed components [30].

At the higher level of its architecture, the A-NE should maintain a knowledge base that should help to describe its situation and to reason about it to determine the right action to perform. Therefore, the A-NE should maintain different types of knowledge. Which knowledge to maintain, how to represent it, and how to reason on it are among the challenges existing and future research initiatives will address. The knowledge can be structured in different levels of abstraction [34]. The following table presents a decomposition of the knowledge in three layers: domain knowledge, control knowledge, and problem determination knowledge (Table 1.4). Domain knowledge provides a view or conceptualization of the managed objects, their properties, the relations among them, and the like; control knowledge represents the ways to manage and control the autonomic elements of the domain; and problem determination knowledge contains all the knowledge necessary to analyze and infer about situations to find appropriate solutions and describes the problems related to the domain and their corresponding applied solutions.

TABLE 1.4

Domain Knowledge	● Captures knowledge about all the elements and concepts of the target domain. ● Contains the knowledge about all the existing managed elements, how to install, configure, and control them ● Relations between the different concepts.
Control Knowledge	● Aims to determine whether or not changes need to be made in the managed elements by means of policies for example. ● Provides a uniform and neutral way to define the control policies to govern the decision-making process of the autonomic network.
Problem Determination Knowledge	● Represents the information captured or inferred. ● Allows defining context and reason on it. ● Specifies how to infer new knowledge from existing knowledge to detect symptoms and take decision. ● Allows the system to learn about situations and improve its capabilities. ● Defines a uniform approach to represent problems and associated potential solutions.

The specification of a knowledge plane for autonomic network management as well as a general architecture is not an easy task. Many initiatives have been launched to address this issue during the last decade, but there has been no agreement so far either on a standard specification of this knowledge plan or on a common architecture. It is not sufficient to find the right mapping techniques, but it is also very important to agree on the structure of the common representation of knowledge. As stated in [38], current approaches for building an Network-Knowledge-Base System (NKBS) are mainly limited to representing the network structure knowledge with some efforts to build simple models for control knowledge.

CONCLUSION

This chapter has presented a general overview of the autonomic concept as applied to networking and to network management. It highlights the different challenges faced by academia and industry. Despite the numerous efforts made in this area, as yet there is no agreement on either the architecture of autonomic networks or autonomic network management, nor is there any consensus on the knowledge plane. Standardization on this critical issue has not really started despite several initiatives with organizations such as IEEE and ETSI. This can be explained primarily by the fact that the autonomic concept has led to the introduction of fundamental changes in the way networks will be built and controlled; therefore the community has not yet reached agreement on the level of autonomic and determinism in the behavior of autonomic element. It is envisioned that some solutions will come from the industry such as those initiated by IBM in autonomic computing and if the experience is positive perhaps it will be deployed on a small scale before being deployed globally. In any case, standardization does not affect all aspects of autonomic networking or network management but rather probably only the architecture and structure of the knowledge plan. The internal mechanisms of autonomic network elements will not have to be standardized and should be part of the competition process between the concerned actors. Nevertheless, the standardization of the communication protocols to enable autonomic networks element to exchange knowledge is mandatory, allowing the autonomic network element to provide autonomic open interfaces to facilitate the interaction between autonomic network elements from different sources.

REFERENCES

[1] Steve R. White, James E. Hanson, Ian Whalley, David M. Chess, and Jeffrey O. Kephart, "An Architectural Approach to Autonomic Computing," Proceedings of the International Conference on Autonomic Computing (ICAC'04), November 2004.

[2] J. Simpson and E. Weiner (eds.), *Oxford English Dictionary*. Oxford, Oxford University Press, 1989.

[3] Paul Horn; "Autonomic Computing: IBM's Perspective on the State of Information Technology," Technical Report, IBM Corporation, October 15, 2001 http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[4] A.G. Ganek, "The dawning of the computing era," *IBM Systems Journal*, Vol. 42, 2003.

[5] IBM White Paper, An architectural blueprint for autonomic computing. IBM, 2003.

[6] R. Sterritt, "Autonomic computing—A means of achieving dependability," presented at 10th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), 2003.

[7] R. Sterritt, "A concise introduction to autonomic computing," *Journal of Advanced Engineering Informatics, Engineering Applications of Artificial Intelligence*, Vol. 19, pp. 181–187, 2005.

[8] IBM developer works, Can you CHOP up autonomic computing? June 2005. http://www.128.ibm.com/developperworks/autonomic/library/ac-edge4

[9] B. Melcher, "Towards an autonomic framework: Self-configuring network services and developing autonomic applications," *Inter® Technology Journal*, Vol. 8, pp. 279–290, 2004.

[10] "Autonomic communication," Research Agenda for a New Communication Paradigm, Fraunhofer Fokus, Autonomic Communication, White Paper, November 2004.

[11] E. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *IEEE Transaction on Software Engineering—Special Issue on Inconsistency Management*, Vol. 25, No. 6, November 1999.

[12] J. Moffeett and M. Sloman, "Policy hierarchies for distributed systems management," *IEEE Journal on Selected Areas in Communications,* Vol. 11, No. 9, December 1993.

[13] I. Aib, N. Agoulmine, M. Sergio Fonseca, and G. Pujolle, "Analysis of policy management models and specification languages," IFIP International Conference on Network Control and Engineering for QoS, Security and Mobility, Kluwer, pp. 26–50, October 2003, Muscat, Oman.

[14] Dinesh C. Verma and D.C. Verma, *Policy-Based Networking: Architecture and Algorithms*, New Riders Publishing, Thousand Oaks, CA, 2000.

[15] J.O. Kephart and D.M. Chess, "The vision of autonomic computing," *IEEE Computer Magazine*, Vol. 36, 2003, pp. 41–45. http://www.research.ibm.com/autonomic/research/papers/AC_Vision_Computer_Jan_2003.pdf

[16] D. Clark, C. Partridge, Chs. Ramming, and J. Wroclawski, "A knowledge plane for the Internet," ACM Sigcomm 2003, Karlsruhe, Germany, August 2003.

[17] A. Dey, "Providing architectural support for building context aware applications," Ph.D. Thesis, 2000.

[18] A.K. Dey, G.D. Abowd, and A. Wood, "CyberDesk: A framework for providing self–integrating context–aware services. Knowledge based systems," Vol. 11, No. 1, pp. 3–13, September 1998.

[19] J. Strassner et al., "Modelling context for autonomic networking," IEEE 2008.

[20] G. Chen and D. Kotz, "Solar: A pervasive-computing infrastructure for context aware mobile applications," Dartmouth Computer Science Technical Report TR2002-421, 2002.

[21] J. Gwizdka, "What's in the context?" Position paper for workshop on The What, Who, Where, When, Why, and How of Context-Awareness. CHI'2000, 2000.

[22] Martin Feeney, "Autonomic management of ubiquitous computing environments," Master's Thesis, Waterford Institute of Technology, 2006.

[23] S. Camazine, J.-L. Deneubourg, Nigel R.F., J. Sneyd, G. Téraulaz, and E. Bonabeau, Self-Organisation in biological systems. Princeton Studies in Complexity. Princeton University Press, 2001.

[24] J. Keeney, K. Carey, D. Lewis, D. O'Sullivan, and V. Wade, Ontology-based Semantics for Composable of Autonomic Elements. Workshop on AI in Autonomic Communications at 19th International Joint Conference on Artificial Intelligence, JCAI'05, Edinburgh, Scotland, July 30–August 5, 2005.

[25] Yechiam Yemini, Member, IEEE, Alexander V. Konstantinou, and Danilo Florissi, "NESTOR: An architecture for network self-management and organization," *IEEE Journal on Selected Areas in Communications,* Vol. 18, No. 5, May 2000.

[26] R. Laddaga, Active Software, in 1st International Workshop on Self Adaptive Software (lWSAS2000). Oxford, Springer-Verlag 2000.

[27] G. Di Marzo Serugendo, "Autonomous systems with emergent behaviour," chapter in *Handbook of Research on Nature Inspired Computing for Economy and Management,* Jean-Philippe Rennard (ed.), Idea Group, Hershey, PA, pp. 429–443, September 2006.

[28] G. Di Marzo Serugendo, M.-P. Gleizes, and A. Karageorgos, "Self-organisation and emergence in MAS: an overview," *Informatica* 30(1): 45–54, Slovene Society Informatika, Ljubljana, Slovenia, 2006.

[29] L. Stojanovic et al., "The role of ontologies in autonomic computing systems," *IBM Systems Journal*, Vol. 43, No. 3, 2004.

[30] Towards autonomic management of communications networks Jennings, Van Der Meer, Balasubramaniam, Botvich, Foghlu, Donnelly, and Strassner, *Communication Magazine,* 2007.

[31] H. Blumer, Collective behavior chapter in "*Review of Sociology—Analysis of a Decade,*" J.B. Gittler (ed.), John Wiley & Son, 1954.

[32] J. Strassner, N. Agoulmine, and E. Lehtihet, "FOCALE: A novel autonomic computing architecture," LAACS, 2006.

[33] D. Clark, C. Partridge, J. Ramming, and J. Wroclawski, "A knowledge plane for the Internet," Proceedings ACM SIGCOMM, 2003.

[34] D. Clark, "The design philosophy of the DARPA Internet protocols," Proceedings ACM SIGCOMM, 1988.

[35] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in systems design," Second International Conference on Distributed Systems, 1981.

[36] D. Clark et al., "NewArch: Future generation Internet architecture," Final technical report.

[37] J. Strassner, M.Ó. Foghlú, W. Donnelly, and N. Agoulmine, "Beyond the knowledge plane: An inference plane to support the next generation internet," 2007 1st International Global Information Infrastructure Symposium, GIIS 2007—"Closing the Digital Divide," pp. 112–119, Marrakech, Morroco, 2007.

[38] N. Samaan and A. Karmouch, "Towards autonomic network management: An analysis of current and future research directions." This paper appears in: *Communications Surveys & Tutorials*, IEEE, Vol. 11, Issue 3, pp. 2–35, ISSN: 1553-877X, 2009.

[39] S. Balasubramaniam, S. Botvich, N. Agoulmine, and W. Donnelly, "A multi-layered approach towards achieving survivability in autonomic network," Proceeding—2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, ICT-MICC 2007, pp. 360–365, Penang, Malaysia, 2007.

[40] C. Stergiou and G. Arys, "A policy based framework for software agents," Proceedings of the 16th international conference on Developments in applied artificial intelligence table of contents, Laughborough, UK, pp. 426–436, 2003.