autonomic computing:

IBM's Perspective on the State of Information Technology the information technology industry loves to prove the impossible possible.

We obliterate barriers and set records with astonishing regularity. But now we face a problem springing from the very core of our success — and too few of us are focused on solving it.

More than any other I/T problem, this one — if it remains unsolved — will actually prevent us from moving to the next era of computing. Interestingly enough, it has little to do with the usual barriers that preoccupy us.

It's not about keeping pace with Moore's Law, but rather dealing with the consequences of its decades-long reign. It's not directly related to how many bits we can squeeze into a square inch, or how thinly we can etch lines in silicon. In fact, a continued obsession with the smaller/faster/cheaper triumvirate is really a distraction.

It's not a barrier of "machine intelligence," either, that threatens our progress. It has less to do with building "thinking machines" that embody the popular conception of artificial intelligence (AI) than automating the day-to-day functioning of computing systems. It may sound odd coming from the creators of Deep Blue, but we don't really need a better chess-playing supercomputer — or sentient machines and androids programmed to love and laugh — to overcome the largest obstacle standing in our way.

The obstacle is complexity. Dealing with it is the single most important challenge facing the I/T industry.

It is our next Grand Challenge.

We in the I/T industry continue to create increasingly powerful computing systems. Why? To make individuals and businesses more productive by automating key tasks and processes. *And for good reason:* in the evolution of humans and human society, automation has always been the foundation for progress. Relegate life's mundane requirements to "automatically handled," and we free our minds and resources to concentrate on previously unattainable tasks. Few of us worry about harvesting the grain to grind the flour to bake bread — we buy it at a nearby store — or about how we'll connect with a friend halfway across the globe — we simply pick up the phone. **SEE FIGURES 1 & 2**

But evolution via automation also produces complexity as an unavoidable byproduct. *Computing systems especially have proved this true.*

Follow the evolution of computers from single machines to modular systems to personal computers networked with larger machines and an unmistakable pattern emerges: incredible progress in almost every aspect of computing — microprocessor power up by a factor of 10,000, storage capacity by a factor of 45,000, communication speeds by a factor of 1,000,000 — but at a price. Along with that growth has come increasingly sophisticated architectures governed by software whose complexity now routinely demands tens of millions of lines of code. Some operating environments weigh in at over 30 million lines of code created by over 4,000 programmers!

FIGURE 1

Progress in Agriculture

Nearly two centuries of innovations in automating manual tasks in agriculture have enabled efficiencies in both the production and yield of crops. Within this time frame farming as a percentage of the labor force decreased from 90 percent to 2.6 percent and labor hours to produce 100 bushels of wheat dropped from 300 hours to just 3 hours. (Source: U.S. Department of Agriculture)

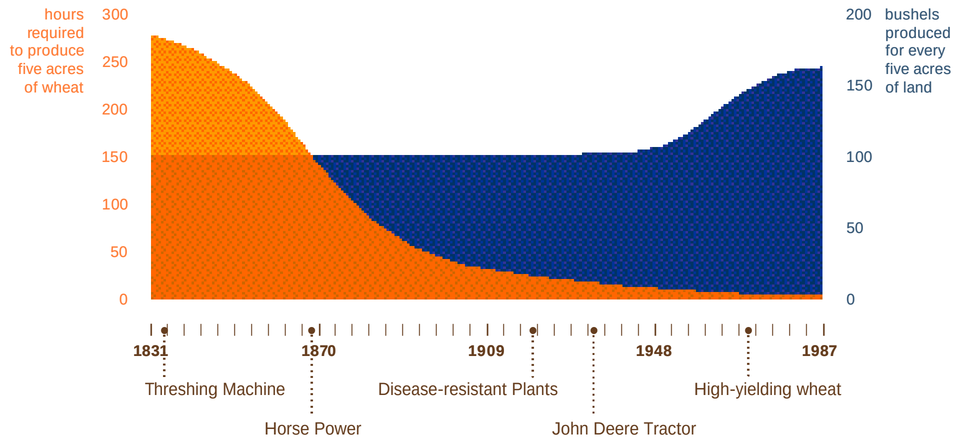
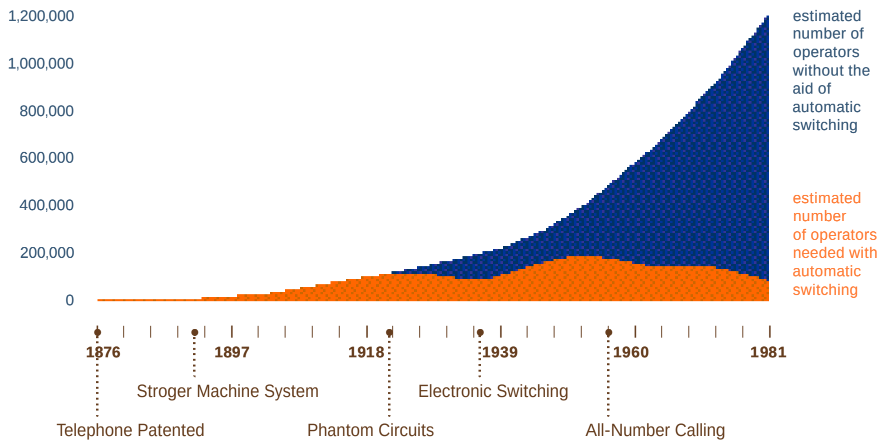


FIGURE 2

Progress in Telephony (U.S. Only)

AT&T/Bell System's implementation of an automated switching protocol in the 1920s allowed it to meet demand for telephones without outpacing the supply of human switchboard operators. (Source: AT&T/Bell Systems)



The Internet adds yet another layer of complexity by allowing us to connect — some might say entangle — this world of computers and computing systems with telecommunications networks. In the process, the systems have become increasingly difficult to manage and, ultimately, to use — ask anyone who's tried to merge two I/T systems built on different platforms, or consumers who've tried to install or troubleshoot DSL service on their own.

In fact, the growing complexity of the I/T infrastructure threatens to undermine the very benefits information technology aims to provide. Up until now, we've relied mainly on human intervention and administration to manage this complexity. *Unfortunately, we are starting to gunk up the works.*

Consider this: at current rates of expansion, there will not be enough skilled I/T people to keep the world's computing systems running. Unfilled I/T jobs in the United States alone number in the hundreds of thousands. Even in uncertain economic times, demand for skilled I/T workers is expected to increase by over 100 percent in the next six years. Some estimates for the number of I/T workers required globally to support a billion people and millions of businesses connected via the Internet — a situation we could reach in the next decade — put it at over 200 million, or close to the population of the entire United States.

Even if we could somehow come up with enough skilled people, the complexity is growing beyond human ability to manage it. As computing evolves, the overlapping connections, dependencies, and interacting applications call for administrative decision-making and responses faster than any human can deliver. Pinpointing root causes of failures becomes more difficult, while finding ways of increasing system efficiency generates problems with more variables than any human can hope to solve.

Without new approaches, things will only get worse. Paradoxically, to solve the problem — make things simpler for administrators and users of I/T — *we need to create more complex systems.*

How will this possibly help?

By embedding the complexity in the system infrastructure itself—both hardware and software — then automating its management. For this approach we find inspiration in the massively complex systems of the human body.

Think for a moment about one such system at work in our bodies, one so seamlessly embedded we barely notice it: *the autonomic nervous system*.

It tells your heart how fast to beat, checks your blood's sugar and oxygen levels, and controls your pupils so the right amount of light reaches your eyes as you read these words. It monitors your temperature and adjusts your blood flow and skin functions to keep it at 98.6°F. It controls the digestion of your food and your reaction to stress—it can even make your hair stand on end if you're sufficiently frightened. It carries out these functions across a wide range of external conditions, always maintaining a steady internal state called homeostasis while readying your body for the task at hand. **SEE FIGURE 3**

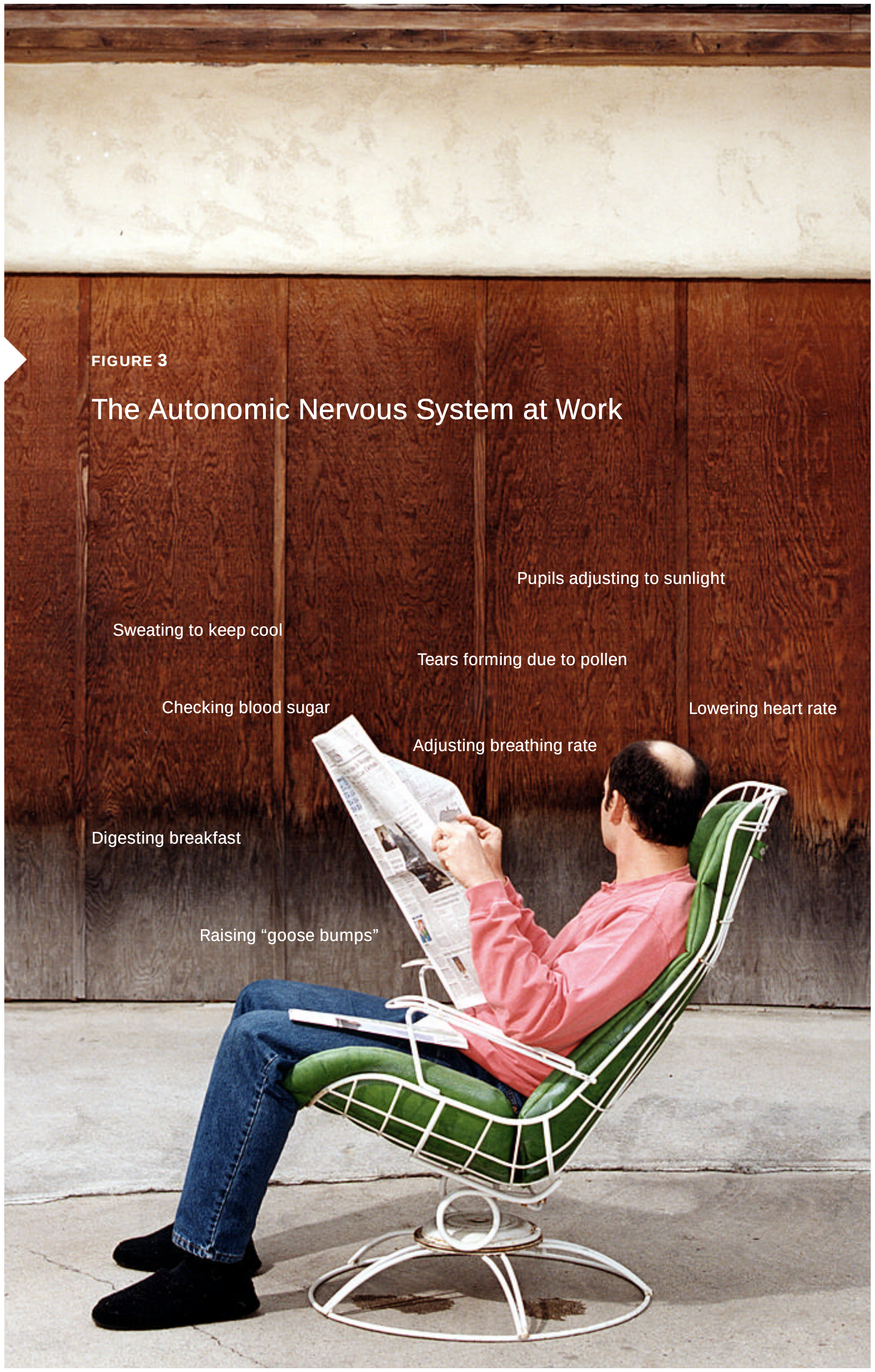
But most significantly, it does all this without any conscious recognition or effort on your part. This allows you to think about what you want to do, and not how you'll do it: you can make a mad dash for the train without having to calculate how much faster to breathe and pump your heart, or if you'll need that little dose of adrenaline to make it through the doors before they close.

It's as if the autonomic nervous system says to you, *Don't think about it — no need to. I've got it all covered.*

that's precisely how we need to build computing systems—an approach we propose as *autonomic computing.*

FIGURE 3

The Autonomic Nervous System at Work



it's time

to design and build computing systems capable of *running themselves*, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them. These autonomic systems must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there.

Why the urgency? Consider some major challenges faced by organizations embracing new e-business technologies:

As a proliferating host of access devices becomes part of the corporate computing infrastructure, enterprises must transform both their I/T systems and the business processes to connect with employees, customers and suppliers. Not only must they manage desktops, workstations and PCs, but also PDAs, cell phones, pagers, etc. Annual compound growth of these devices is expected to exceed 38 percent over the next three years. Companies must also manage the very products they produce, such as network-enabled cars, washing machines, and entertainment systems, as part of this integrated "system," extending the system concept well beyond traditional corporate boundaries. This demands a reliable infrastructure that can accommodate rapid growth and hide system complexity from its users — the company's customers, employees and suppliers.

Emerging "Web Services" standards promise to make delivery of valuable services over the Internet possible. In one recent *InfoWorld* survey, close to 70 percent of respondents said they would develop Web Services strategies within the year, and roughly the same percentage felt Web Services likely to emerge as the next viable business model of the Internet. I/T services, in particular, are a likely candidate for delivery in a utility-like fashion, a trend we call e-sourcing. But such services cannot become widespread unless I/T systems become more automated and allow true economies of scale for e-sourcing providers. Customers also must gain enough confidence in this model to turn over critical business data and processes, confidence unlikely to develop if system reliability remains dependent on an inadequate supply of I/T workers.

The underlying technologies to enable greater automation of complex systems management are ripe for innovation. The emergence of XML and a host of new standards give us a glimpse of the glue we'll need to bind such self-governing systems, and advances in workload management and software agents promise possible incremental paths to autonomic computing.

But focusing on automating
the piece parts of computing systems
will not be enough.

Think again of the functioning of the body's autonomic nervous system. it is the self-governing operation of the entire system, and not just parts of it, that delivers the ultimate benefit. An increase in heart rate without a corresponding adjustment to breathing and blood pressure would be disastrous, just as a functioning brain stem would prove useless without an "always available" connection to the organs and muscles it directs.

So too with an autonomic computing system. Bringing autonomic capabilities to storage systems would certainly be an improvement, but if the computing systems that mine the data in those storage repositories become next to impossible to manage, that partial automation will not yield much overall benefit.

That's why we need a systemic approach that allows for coordination and automatic management across entire networks of computing systems—systems built on various platforms and owned (or even shared) by various entities. *Autonomic computing is thus a "holistic" vision* that will enable the whole of computing to deliver much more automation than the sum of its individually self-managed parts.

More is at stake than general system reliability or "ease of use" for I/T customers. Only when I/T complexity becomes embedded in the infrastructure, effectively eliminating its visibility from users, can the next wave of I/T-driven economic productivity occur. *In other words*, if we make it simpler for our customers to use our technology, new and even unpredictable applications of I/T will emerge. And more people will use them than ever before.

Both the Internet and the PC revolution accelerated computing's inevitable migration to the masses. But the vast majority of the earth's population has yet to touch a computer or benefit directly from its potential. For them to do so, interacting with computing systems will have to become much more natural. *How natural?*

As early as the 1960s, Captain Kirk and his crew were getting information from their computing systems by asking simple questions, and listening to straightforward answers — all without a single I/T administrator to be seen on the ship. Subtly, this has become the expectation of the masses: let me interact with computers as if I'm dealing with a person, but let the system have information processing capabilities no human could ever possess. While this goal may still sound futuristic,

why not apply that thinking to everyday business?

Why, for instance, should business owners have to spend so much time and money figuring out how to install and manage I/T systems? For some enterprise resource-planning systems, installation and customization fees can run several times the initial licensing cost of the system. All that should matter to the business owner is: *what does my business need to accomplish?*

ask i/t customers what they want, and they'll tell you: *let me concentrate on setting the business policy for, say, security, then have the system figure out the implementation details.* In much the same way programming languages have moved closer to natural language, companies should be able to frame their "instructions" for the computing system in plainspeak: watch my competitors, and if they're beating us in a particular market, adjust prices and the corresponding supply chain to support the new pricing.

To really benefit I/T customers, autonomic computing will need to deliver measurable advantage and opportunity by improving interaction with I/T systems and the quality of the information they provide, and enabling e-sourcing to be adopted as the future of I/T services delivery.

Then I/T customers will at last be able to focus entirely on the information services they want and "forget" about the systems providing them.

THE ACCOMPANYING FIGURES ILLUSTRATE THIS POTENTIAL.

FIGURE 4

A large retail chain with hundreds of outlets, a network of warehouses, delivery fleets, employee services, customer service call centers, web interfaces and more — an autonomic computing system manages all these distinct (and quasi-independent) I/T systems as one and provides integrated time-sensitive functionality, as well as *"always available"* access through web interfaces.

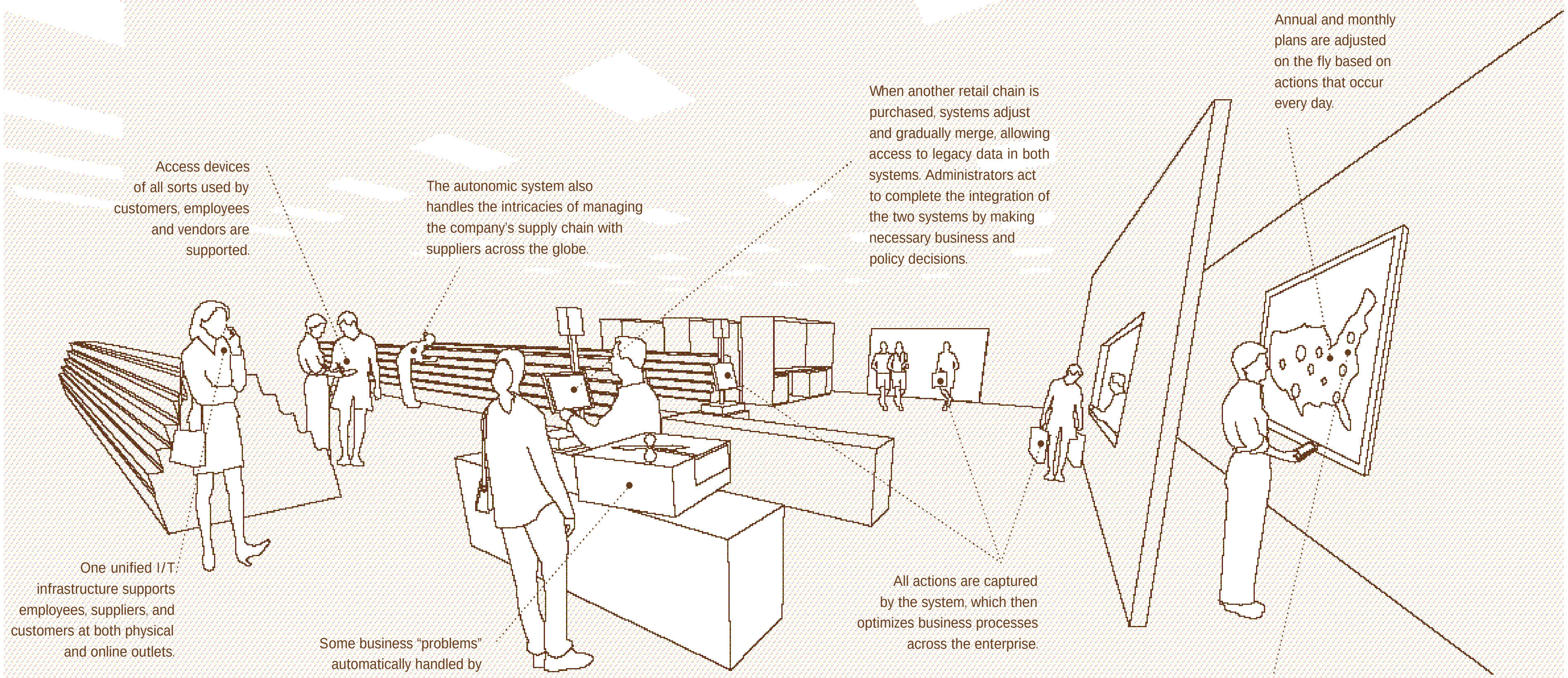

Access devices of all sorts used by customers, employees and vendors are supported.
The autonomic system also handles the intricacies of managing the company's supply chain with suppliers across the globe.
When another retail chain is purchased, systems adjust and gradually merge, allowing access to legacy data in both systems. Administrators act to complete the integration of the two systems by making necessary business and policy decisions.
Annual and monthly plans are adjusted on the fly based on actions that occur every day.
One unified I/T infrastructure supports employees, suppliers, and customers at both physical and online outlets.
Some business "problems" automatically handled by following management rules and guidelines: rerouting merchandise to low-inventory areas; adjusting orders and pricing based on the latest financial and climate models (e.g., heatwave = more electric fans needed).
All actions are captured by the system, which then optimizes business processes across the enterprise.
The system provides real-time visualization of the business as an organic whole: sales, costs, inventories, most efficient markets, etc., mapped to its thousands of retail stores. This enables rapid decision-making by management.

FIGURE 5

A medical emergency in a foreign country — dealing with such an urgent situation requires instant access and integration across multiple disparate systems. Autonomic systems management based on global medical standards enables distinct systems to act in a unified manner and exchange and integrate data. The result: a quick and accurate diagnosis as well as the best course for immediate life-saving treatment.

Even though the patient is initially unconscious, autonomic computing allows immediate access to overseas records, including complete family histories, x-rays and MRIs, which rule out some common diagnoses and indicate something more serious that requires surgery.

Doctors evaluate all current prescriptions and possible interactions as well as allergies, which alters plans for the type of anesthesia to be used.

Systems link to insurance records, obtain all necessary approvals and submit all forms.

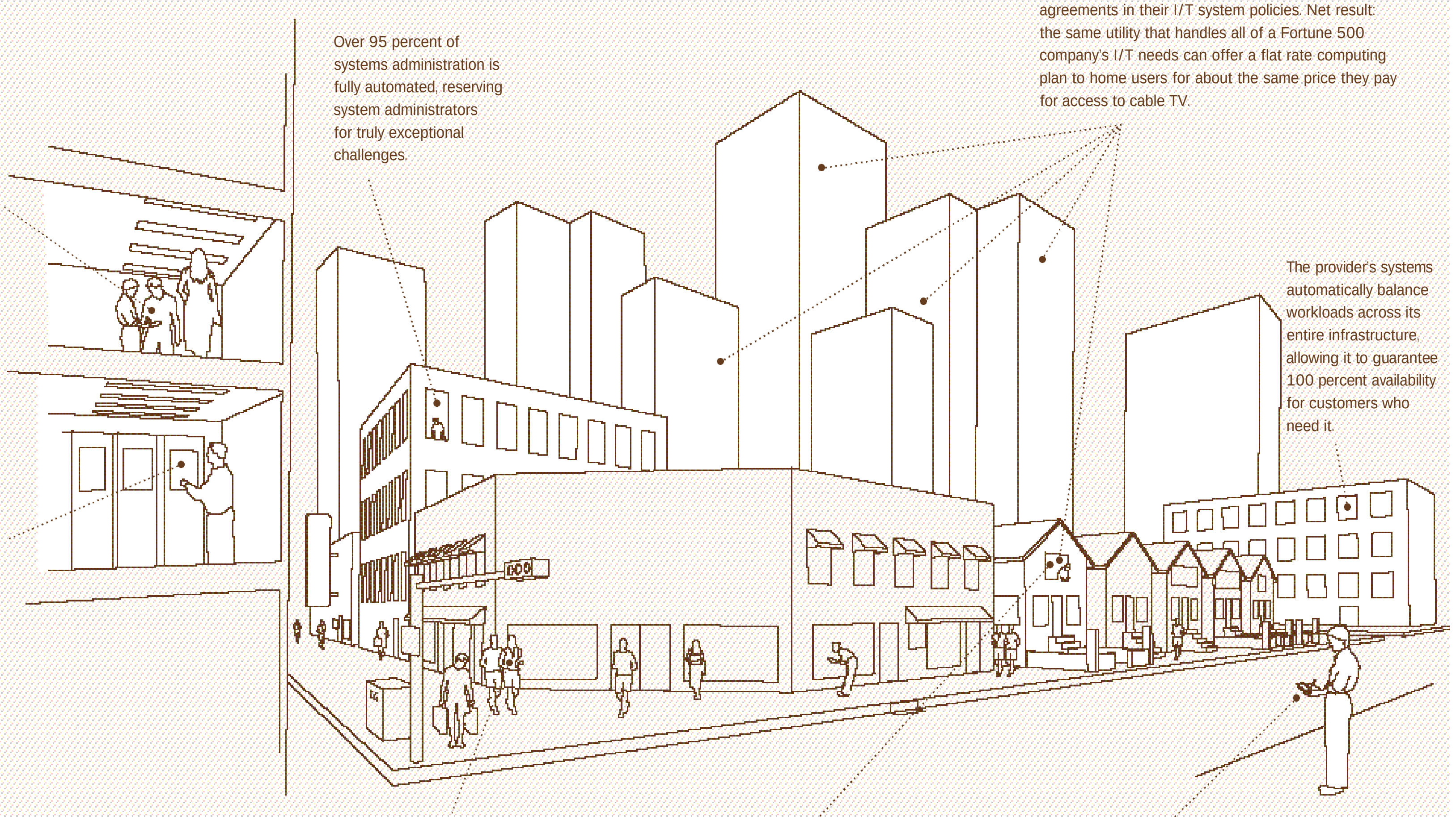
Patient's DNA record and known genomic data compared when evaluating potential treatments. Super-computing simulations run to ascertain best possible drug therapy, including a patient's unique systemic reaction to novel drugs.

Records accessed and appended only through trusted sources and with full protection of confidentiality based on standard medical processes embedded in software agents.

Reliability of the medical network — one which comprises many different systems across the globe belonging to different doctors and hospitals — uncompromised by local outages, storms or network traffic spikes.

FIGURE 6

An e-sourcing provider that services thousands of customers ranging from Fortune 500 companies to individuals — autonomic computing allows it to offer many ranges of service while profitably aggregating customers' varying demands across its global I/T infrastructure.


Customers can enter into contracts with more than one e-sourcing provider to minimize costs and provide an added layer of redundancy. For example, a service level agreement with a local e-sourcing provider might specify higher charges during peak hours from 8am to 5pm, at which time a customer's autonomic systems might switch to a provider on the other side of the globe.
Over 95 percent of systems administration is fully automated, reserving system administrators for truly exceptional challenges.
Varying "quality of service" agreements allow the utility to accommodate all types of customers, from large corporations to individuals, and embody those service agreements in their I/T system policies. Net result: the same utility that handles all of a Fortune 500 company's I/T needs can offer a flat rate computing plan to home users for about the same price they pay for access to cable TV.
The provider's systems automatically balance workloads across its entire infrastructure, allowing it to guarantee 100 percent availability for customers who need it.
Distributed redundancies protect critical customer data, allowing the utility to guarantee the highest level of integrity for those customers willing to pay for it.
The provider can accommodate any access devices its customers prefer by automatically managing them as part of its system.
Customers can purchase special "pay-as-you-use" services, such as access to supercomputing.
The provider offers plans where the customer receives the latest access devices at no charge — most will never again own a "computer," yet they will rely on computing services more than ever before.

This level of functionality won't come easily, and may sound like the description of a panacea rather than an approach to the next era of computing. In fact, it's only the beginning of what we can imagine this next era will deliver. Unfortunately, that next era will never come if we continue managing systems and I/T complexity as we do today.

so, if autonomic computing "systems" are the inevitable answer, what specifically are they?

First, let's define a "system." A system can be thought of as a collection of computing resources tied together to perform a specific set of functions. Following this logic, an individual server can constitute a system, as can a microprocessor containing varying integrated elements on a single chip — the so-called "system-on-a-chip" approach. These lower-level systems combine to form larger systems: multiprocessors to form servers, servers with storage devices and client or access devices to form networked systems, and so on. The principle of autonomic computing must govern all such systems—*i.e.*, at some level, they must be able to manage their own processes — although most of the defining elements of autonomic computing refer specifically to larger, higher-level systems.

Our bodies have a somewhat similar hierarchy of self-governance: from single cells to organs and organ systems (one such system being the autonomic nervous system). Each level maintains a measure of independence while contributing to a higher level of organization, culminating in the organism — us. We remain thankfully unaware, for the most part, of the daily management of it all, because these systems take care of themselves and only "escalate" to a higher level function when they need help.

So, too, with an autonomic computing system. In the end, its individual layers and components must contribute to a system that itself functions well without our regular interference to provide a simplified user experience. *Such a high-level system could be described as possessing at least* eight key elements *or characteristics.*

1

To be autonomic, a computing system needs to "know itself"—and comprise components that also possess a system identity.

Since a "system" can exist at many levels, an autonomic system will need detailed knowledge of its components, current status, ultimate capacity, and all connections with other systems to govern itself. It will need to know the extent of its "owned" resources, those it can borrow or lend, and those that can be shared or should be isolated.

Such system definition might seem simple, and when a "computer system" meant one room-filling machine, or even hundreds of smaller machines networked within the walls of one company, it was. But link those hundreds of computers to millions more over the Internet, make them interdependent, and allow a global audience to link back to those hundreds of computers via a proliferating selection of access devices — cell phones, TVs, intelligent appliances — and we have blurred the once-clear concept of a "system." Start allowing all those devices to share processing cycles, storage and other resources, add to that the possibility of utility-like leasing of computing services, and we arrive at a situation that would seem to defy any definition of a single "system."

But it's precisely this awareness at an overall system-wide level that autonomic computing requires. A system can't monitor what it doesn't know exists, or control specific points if its domain of control remains undefined.

To build this ability into computing systems, clearly defined policies embodied in adaptable software agents will have to govern a system's definition of itself and its interaction with I/T systems around it. These systems will also need the capacity to merge automatically with other systems to form new ones, even if only temporarily, and break apart if required into discrete systems.

2 *An* autonomic computing system must configure and reconfigure itself under varying and unpredictable conditions.

System configuration or "setup" must occur automatically, as must dynamic adjustments to that configuration to best handle changing environments.

Given possible permutations in complex systems, configuration can be difficult and time-consuming — some servers alone present hundreds of configuration alternatives. Human system administrators will never be able to perform dynamic reconfiguration as there are too many variables to monitor and adjust — multiply hundreds of alternatives by thousands of servers and other system devices — in too short a period of time — often minutes, if not seconds.

To enable this automatic configuration ability, a system may need to create multiple images of critical software, such as an operating system (a kind of software cloning), and reallocate its resources (such as memory, storage, communications bandwidth, and processing) as needed. If it is a globally distributed system, it will need to leverage its multiple images and backup copies to recover from failures in localized parts of its network. Adaptive algorithms running on such systems could learn the best configurations to achieve mandated performance levels.

3 *An* autonomic computing system never settles for the status quo — it always looks for ways to optimize its workings.

It will monitor its constituent parts and fine-tune workflow to achieve predetermined system goals, much as a conductor listens to an orchestra and adjusts its dynamic and expressive characteristics to achieve a particular musical interpretation.

This consistent effort to optimize itself is the only way a computing system will be able to meet the complex and often conflicting I/T demands of a business, its customers, suppliers and employees. And since the priorities that drive those demands change constantly, only constant self-optimization will satisfy them.

Self-optimization will also be a key to enabling the ubiquitous availability of e-sourcing, or a delivery of computing services in a utility-like manner. e-sourcing promises predictable costs and simplified access to computing for I/T customers. For providers of those computing services, though, delivering promised quality of service to its customers will require not only prioritizing work and system resources, but also considering supplemental external resources (such as subcontracted storage or extra processing cycles) similar to the way power utilities buy and sell excess power in today's electricity markets.

But to be able to optimize itself, a system will need advanced feedback control mechanisms to monitor its metrics and take appropriate action. Although feedback control is an old technique, we'll need new approaches to apply it to computing. We'll need to answer questions such as how often a system takes control actions, how much delay it can accept between an action and its effect, and how all this affects overall system stability.

Innovations in applying control theory to computing must occur in tandem with new approaches to overall systems architecture, yielding systems designed with control objectives in mind. Algorithms seeking to make control decisions must have access to internal metrics. And like the tuning knobs on a radio, control points must affect the source of those internal metrics. Most important, all the components of an autonomic system, no matter how diverse, must be controllable in a unified manner.

4

An autonomic computing system must perform something akin to healing — it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction.

It must be able to discover problems or potential problems, then find an alternate way of using resources or reconfiguring the system to keep functioning smoothly. Instead of "growing" replenishment parts, as our cells do, healing in a computing system means calling into action redundant or underutilized elements to act as replacement parts. (In a sense, this is akin to what our brain does when parts of it are damaged.)

Of course, certain types of "healing" have been a part of computing for some time. Error checking and correction, an over 50-year-old technology, enables transmission of data over the Internet to remain remarkably reliable, and redundant storage systems like RAID allow data to be recovered even when parts of the storage system fail.

But the growing complexity of today's I/T environment makes it more and more difficult to locate the actual cause of a breakdown, even in relatively simple environments. We see this even with personal computers—how many times is the "solution" to a problem "shut down, reboot and see if it helps"?

In more complex systems, identifying the causes of failures calls for root-cause analysis (an attempt to systematically examine what did what to whom and to home in on the origin of the problem). But since restoring service to the customer and minimizing interruptions is the primary concern, an action-oriented approach (determining what immediate actions need to be taken given current information available) will need to take precedence in an autonomic solution.

Initially, "healing" responses taken by an autonomic system will follow rules generated by human experts. But as we embed more intelligence in computing systems, they will begin to discover new rules on their own that help them use system redundancy or additional resources to recover and achieve the primary objective: meeting the goals specified by the user.

5

A virtual world is no less dangerous than the physical one, so an autonomic computing system must be an expert in self-protection.

It must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.

Before the Internet, computers operated as islands. It was fairly easy then to protect computer systems from attacks that became known as "viruses." As the floppy disks used to share programs and files needed to be physically mailed or brought to other users, it took weeks or months for a virus to spread.

The connectivity of the networked world changed all that. Attacks can now come from anywhere. And viruses spread quickly—in seconds—and widely, since they're designed to be sent automatically to other users. The potential damage to a company's data, image and bottom line is enormous.

More than simply responding to component failure, or running periodic checks for symptoms, an autonomic system will need to remain on alert, anticipate threats, and take necessary action. Such responses need to address two types of attacks: viruses and system intrusions by hackers.

By mimicking the human immune system, a “digital immune system”—an approach that exists today—can detect suspicious code, automatically send it to a central analysis center, and distribute a cure to the computer system. The whole process takes place without the user being aware such protection is in process.

To deal with malicious attacks by hackers, intrusion systems must automatically detect and alert system administrators to the attacks. Currently, computer security experts must then examine the problem, analyze it and repair the system. As the scale of computer networks and systems keeps expanding and the likelihood of hacker attacks increases, we will need to automate the process even further. There won’t be enough experts to handle each incident.

6 *An* autonomic computing system knows its environment and the context surrounding its activity, and acts accordingly.

This is almost self-optimization turned outward: an autonomic system will find and generate rules for how best to interact with neighboring systems. It will tap available resources, even negotiate the use by other systems of its underutilized elements, changing both itself and its environment in the process—in a word, adapting. This context-sensitivity includes improving service based on knowledge about the context of a transaction.

Such an ability will enable autonomic systems to maintain reliability under a wide range of anticipated circumstances and combinations of circumstances (one day perhaps covering even unpredictable events). But more significantly, it will enable them to provide useful

information instead of confusing data. For instance, delivering all the data necessary to display a sophisticated web page would be obvious overkill if the user was connected to the network via a small-screen cell phone and wanted only the address of the nearest bank. Or a business system might report changes in the cost of goods immediately to a salesperson in the middle of writing a customer proposal, where normally weekly updates would have sufficed.

Autonomic systems will need to be able to describe themselves and their available resources to other systems, and they will also need to be able to automatically discover other devices in the environment. Current efforts to share supercomputer resources via a “grid” that connects them will undoubtedly contribute technologies needed for this environment-aware ability. Advances will also be needed to make systems aware of a user’s actions, along with algorithms that allow a system to determine the best response in a given context.

7 *An* autonomic computing system cannot exist in a hermetic environment.

While independent in its ability to manage itself, an autonomic computing system must function in a heterogeneous world and implement open standards—in other words, an autonomic computing system cannot, by definition, be a proprietary solution.

In nature, all sorts of organisms must coexist and depend upon one another for survival (and such biodiversity actually helps stabilize the ecosystem). In today’s rapidly evolving computing environment, an analogous coexistence and interdependence is unavoidable. Businesses connect to suppliers, customers and partners. People connect to their banks, travel agents and favorite stores—regardless of the hardware they have, or the applications they are using. As technology improves, we can only expect new inventions and new devices —and an attendant proliferation of options and interdependency.

Current collaborations in computer science to create additional open standards have allowed new types of sharing: innovations such as Linux, an open operating system; Apache, an open web server; UDDI, a standard way for businesses to describe themselves, discover other businesses and integrate with them; and from the Globus project, a set of protocols to allow computer resources to be shared in a distributed (or "grid-like") manner. These community efforts have accelerated the move toward open standards, which allow for the development of tools, libraries, device drivers, middleware, applications, etc., for these platforms.

Advances in autonomic computing systems will need a foundation of such open standards. Standard ways of system identification, communication and negotiation — perhaps even new classes of system-neutral intermediaries or "agents" specifically assigned the role of cyber-diplomats to regulate conflicting resource demands—need to be invented and agreed on.

8

Perhaps most critical for the user, an autonomic computing system will anticipate the optimized resources needed while keeping its complexity hidden.

This is the ultimate goal of autonomic computing: the marshaling of I/T resources to shrink the gap between the business or personal goals of our customers, and the I/T implementation necessary to achieve those goals—without involving the user in that implementation.

Today our customers must adapt to a computing system by learning how to use it, how to interact with it, and how to collect, compare and interpret the various types of information it returns before deciding what to do. Even custom-made solutions rarely interact seamlessly with all a company's other I/T systems, let alone all its data and documents. While some aspects of computing have improved for general users — graphical interfaces, for instance, are far easier for most people to use than command prompts and their corresponding dictionaries of commands—tapping the full potential of entire I/T systems is still too difficult.

But does this mean autonomic computing systems must begin to possess human intelligence so as to anticipate, perhaps even dictate, a user's I/T needs? No. Think again of the analogy of our bodies, and in particular one aspect of the autonomic nervous system responsible for what's commonly known as the "fight or flight response."

When faced with a potentially dangerous or urgent situation, our autonomic nervous system anticipates the potential danger before we become aware of it. It then "optimizes" our bodies for a selection of appropriate responses — specifically, the autonomic nervous system triggers our adrenal glands to flood the body with adrenaline, a hormone that supercharges the ability of our muscles to contract, increases our heart rate and breathing, and generally constricts blood vessels to increase our blood pressure (while dilating those that feed key areas such as the skeletal muscles). The net result: our body is superbly prepped for action, but our conscious mind remains unaware of anything but the key pieces of information required to decide whether to stay and act (the "fight" response) or run for the hills.

An autonomic system will allow for that kind of anticipation and support. It will deliver essential information with a system optimized and ready to implement the decisions users make and not needlessly entangle them in coaxing results from the system.

Realistically, such systems will be very difficult to build and will require significant exploration of new technologies and innovations. That's why we view this as a Grand Challenge for the entire I/T industry. *We'll need to make progress along two tracks:*

making individual system components autonomic, and achieving autonomic behavior at the level of global enterprise i/t systems.

That second track may prove to be extremely challenging. Unless each component in a system can share information with every other part and contribute to some overall system awareness and regulation, the goal of autonomic computing will not really be reached. So one huge technical challenge entails figuring how to create this "global" system awareness and management. *Or to put it another way,* how do we optimize the entire stack of computing layers as a whole? It's not something we currently know how to do.

We know there are also many interim challenges: how to create the proper "adaptive algorithms"—sets of rules that can take previous system experience and use that information to improve the rules. Or how to balance what these algorithms "remember" with what they ignore. We humans tend to be very good at the latter—we call it "forgetting"—and at times it can be a good thing: we can retain only significant information and not be distracted by extraneous data.

Still another problem to solve: how to design an architecture for autonomic systems that provides consistent interfaces and points of control while allowing for a heterogeneous environment. We could go on, as the list of problems is actually quite long, but it is not so daunting as to render autonomic computing another dream of science fiction.

In fact, we're beginning to make progress in key areas.

first,

many established fields of scientific study will contribute to autonomic computing. What we've learned in artificial intelligence, control theory, complex adaptive systems and catastrophe theory, as well as some of the early work done in cybernetics, will give us a variety of approaches to explore. Current research projects at laboratories and universities include self-evolving systems that can monitor themselves and adjust to some changes, "cellular" chips capable of recovering from failures to keep long-term applications running, heterogeneous work-load management that can balance and adjust workloads of many applications over various servers, and traditional control theory applied to the realm of computer science, to name just a few.

Also,

some aspects of autonomic computing are not entirely new to the I/T industry. For instance, the protocols and standards used for forwarding packets of information across the Internet (the most well-known being TCP/IP) allow for some relatively simple functions, such as routing, to occur with little human direction. And since mainframe computers have historically been entrusted with important, "mission-critical" work for businesses and governments, they have had increasing levels of self-regulation and self-diagnosis built in. Such machines now boast "availability rates" — the percentage of time they are functioning properly — in the 99.999 percent range.

but this innovation needs to be taken to an entirely new level.

That's why at IBM we've reorganized our Research division around achieving this ambitious goal. And to accelerate the flow of this current innovation into today's hardware and software, we have undertaken Project eLiza: a major commitment of our server R&D budget earmarked for autonomic computing innovations.

this is bigger than any *single* i/t company.

It's a vision that requires the involvement of the top minds in the technology community. That's why we're forming an advisory board of leading academic and industry thinkers to help define our autonomic research agenda. We're also consulting with a large group of customers and I/T partners as part of our eLiza project to define a strategy for bringing autonomic innovations to products.

We call on our academic colleagues to drive exploratory work in autonomic computing. We propose that the research community recognize it as an important field of academic endeavor. We also call on our partners at government labs to collaborate with us on crucial projects in this area. We plan to fund a regular stream of academic grants to support research in this area, and we call on others in the I/T industry to do the same.

Finally, we call on the entire I/T industry to refocus its priorities on this essential goal. We must cooperate in developing the necessary standards and open interfaces to make this vision a reality. That's why we're working with Globus and the Grid Computing community to establish standards that will support an autonomic computing environment. *The days of pushing proprietary agendas and strategies are over. Our customers deserve better.*

We in the I/T industry have lingered too long in an era of over-specialization in which integration was just another specialty. We've made tremendous progress in almost every aspect of computing, but not enough in the one that now counts most: how to deal with the complexity generated by all that "smaller/faster/cheaper" focus.

In this heady rush we've risked losing sight of the people who buy I/T and who have come to depend on us for increased productivity and improvement in many aspects of their daily lives. We've made it unnecessarily difficult for them to tap the potential we've promised them. *It's time for a change.*

Autonomic computing represents both this change and the inevitable evolution of I/T automation. This next era of computing will enable progress and abilities we can barely envision today. But the best measure of our success will be when our customers think about the functioning of computing systems about as often as they think about the beating of their hearts.

Paul Horn

Paul Horn, *Senior Vice President*
ibm research

GLOSSARY

Adaptive Algorithm	An algorithm that can “learn” and change its behavior by comparing the results of its actions with the goals that it is designed to achieve.
Algorithm	A procedure, which can be written as a set of steps, for producing a specific output from a given input.
Artificial Intelligence (AI)	The capacity of a computer or system to perform tasks commonly associated with the higher intellectual processes characteristic of humans. AI can be seen as an attempt to model aspects of human thought on computers. Although certain aspects of AI will undoubtedly make contributions to autonomic computing, autonomic computing does not have as its primary objective the emulation of human thought.
Autonomic	1.Of, relating to, or controlled by the autonomic nervous system. 2.Acting or occurring involuntarily; automatic: an autonomic reflex.
Autonomic Nervous System	That part of the nervous system that governs involuntary body functions like respiration and heart rate.
Catastrophe Theory	A special branch of dynamical systems theory that studies and classifies phenomena characterized by sudden shifts in behavior arising from small changes in circumstances.
Control Theory	The mathematical analysis of the systems and mechanisms for achieving a desired state under changing internal and external conditions.
Cybernetics	A term derived from the Greek word for “steersman” that was introduced in 1947 to describe the science of control and communication in animals and machines.
Feedback Control	A process by which output or behavior of a machine or system is used to change its operation in order to constantly reduce the difference between the output and a target value. A simple example is a thermostat that cycles a furnace or air conditioner on and off to maintain a fixed temperature.
Globus	A collaborative academic project centered at Argonne National Laboratory focused on enabling the application of grid concepts to computing.
Grand Challenge	A problem that by virtue of its degree of difficulty and the importance of its solution, both from a technical and societal point of view, becomes a focus of interest to a specific scientific community.
Grid computing	A type of distributed computing in which a wide-ranging network connects multiple computers whose resources can then be shared by all end-users; includes what is often called “peer-to-peer” computing.

Homeostasis
A physiological constancy or equilibrium maintained by self-regulating mechanisms.

Policy-based Management
A method of managing system behavior or resources by setting “policies” (often in the form of “if-then” rules) that the system interprets.

Project eLiza
An initiative launched by IBM in April 2001 to integrate autonomic capabilities into its products and services, including servers, storage, middleware, and various kinds of services offerings. It is a core element of IBM’s autonomic computing effort.

Quality of Service (QoS)
A term used in a Service Level Agreement (SLA) denoting a guaranteed level of performance (e.g., response times less than 1 second).

Redundant Arrays of Independent Disks (RAID)
A way of storing the same data in different places on multiple hard disks. Storing data on multiple disks can improve performance by balancing input and output operations. Since using multiple disks increases the mean time between failure, storing data redundantly also increases fault-tolerance.

Service Level Agreement (SLA)
A contract in which a service provider agrees to deliver a minimum level of service.

Web Services
A way of providing computational capabilities using standard Internet protocols and architectural elements. For example, a database web service would use web browser interactions to retrieve and update data located remotely. Web services use UDDI to make their presence known.

Printed in the U.S.A.
October 2001

Design: VSA Partners, Inc.

For more information about autonomic computing, please visit our website at
http://www.ibm.com/research/autonomic