Autonomic Computing: Further Maturation in IT Industry

Divya Bahl[1], Ritika Wason[2]

[1]MCA Scholar
Institute of Information Technology and Management
divyabahl@yahoo.com

[2]Assistant Professor
Institute of Information Technology and Management
rit_2282@yahoo.co.in

Abstract: ***IT industry in the past few decades has been defined by a passion for faster, smaller, and powerful software applications. The current emphasis is on raw processing power from smaller components with greater capacity to store, process and move data. This paper is a critical analysis of the autonomic computing paradigm that aims to eliminate the need of human intervention on executable computer systems. The paper reviews the IBM autonomic computing initiative, analyzing its applications and examines the challenges and issues autonomic computing faces while achieving its objective.***

Keywords: Autonomic Computing, Self-healing, self-CHOP [12], Holistic computing.

1. Introduction

As an implication of all the technological gifts due to more fast and cogent systems, the requirement of trained manpower for the maintenance of such complex systems has come up as a major implication for the IT industry. The past two decades have witnessed the rapid proliferation of raw computing power coupled with the exponential growth of complex computing devices. As a result new demand for skilled people to manage and maintain such intricate computer systems has also emerged.

In terms of computing, we are at a threshold right now. Millions of organizations, billions of humans, and trillions of devices all require the services of the I/T industry to run [1].However, there is a shortage of skilled I/T workers to manage, install, configure, operate, tune, and maintain all smart gadgets and devices that have become an inseparable part of increasingly convoluted systems.

The probability of this growing complexity had been indicated as early as 1965 by the Intel co-founder, Gordon Moore [2]. As part of his research, Moore observed that the number of transistors per square inch on integrated circuits had doubled every year since their invention. Moore predicted the trend to continue into the foreseeable future. The IT industry's exploitation of the technologies in accordance with Moore's law has brought us on the verge of a complexity crisis [3]. The human race has become dependent in thought and action to the machine we have created to serve us. The need of the hour is that we start making our systems smarter so that they can manage themselves without any human intervention. The same vision has led to the origin of a new computing paradigm called autonomic computing [1]. The rest of the paper is organized as follows: Section 1.1 discusses the basic idea of autonomic computing, Section 1.2 highlights the major drivers behind autonomic computing research. Section 2 analyzes the milestones in autonomic computing achieved by IBM. Section 2.1 observes IBM's autonomic initiatives. Section 3 discusses the basic components constituting an autonomic system. Section 3.1 explains the generic architecture to construct an autonomic system. Section 4 highlights the varied applications of the autonomic paradigm. Section 4.1 discusses how IBM has included autonomic capabilities in its various products. Section 4.2 introduces the positive & negative issues associated with every autonomic system. Section 5 illustrates the future of the paradigm. Section 6 concludes the discussion.

1.1 Origin of Autonomic Computing

IT corporate IBM in 2001 first identified the complexity crisis and introduced a new model of computing called-Autonomic Computing as a solution for the same[15].Automation in computing refers to execution of important computing operations without the need for human intervention.

Natural systems are by nature self-balancing, self-adjusting and self-cleansing. However, technology works on human crutches. In order to make technology analogous to nature, IBM started with its initiative of building a systematic way of computing, modeled after self-regulating biological system. Since then IBM has been working on the idea of developing self-regulating computing systems that can regulate and protect themselves just like human nervous system [2]. Figure 1 below depicts the parallelism that can be drawn between an autonomic system and human nervous system.

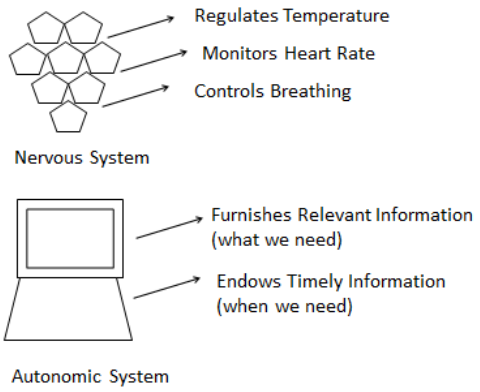

Fig 1: Parallelism between the Nervous System & an Autonomic System

In simple words, Autonomic systems are based on the self-healing property of the human nervous system. An autonomic system is expected to configure, tune and even repair itself independently, without human intervention. Hence, Autonomic Computing can be formally defined as designing and building computing systems capable of running and managing themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently their workloads [4]. In other words, Autonomic Computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity from operators and users. An autonomic system is capable of making decisions on its own, using high-level policies; it can constantly check and optimize its status and automatically adapt itself to changing environmental conditions.[14]

Autonomic computing frees system administrators from many routine management and operational tasks just as the autonomic nervous system frees our brain from the burden of having to deal with vital but lower-level functions. This capability shall empower corporations to devote more of their IT skills towards their core businesses, instead of spending increasing amount of time dealing with the maintenance tasks of computing systems.

The autonomic paradigm shall shift the fundamental focus of technology from computing, to data. Access to data from multiple, distributed sources, in addition to traditional centralized storage devices will allow users to transparently access information when and where they need it. At the same time, this view of computing will necessitate changing the industry's focus on processing speed and storage to one of developing distributed networks that are largely self-managing, self-diagnostic, and transparent to the user.[5]

1.2 Essentials of Autonomic Computing

Technology has the shelf life of a banana [6]. This assertion is absolutely correct and relevant for the current times. Technology is transforming constantly at a fast pace and so is the associated infrastructure, middleware, and services that maintain them. The bright side to this is that the processor's speed, storage capacity and network connectivity have also improved dramatically over the year. However, the ever increasing system complexity is reaching a level beyond manageability. The present scenario, suffers from shortage of skilled administrators along with the complexity of IT Infrastructure. As a result, we are unable to fully exploit the current available IT infrastructure. This situation necessitates a radical shift in our approach to computing. IBM's Autonomic Computing paradigm promises to be the same. Figure 2 below depicts the market forces behind autonomic computing

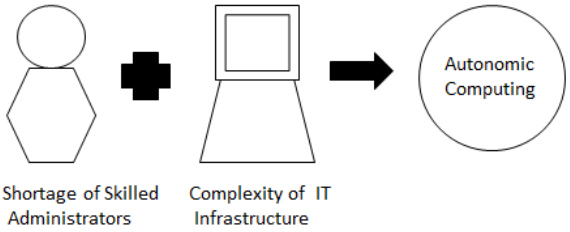

Fig 2: Need of Autonomic Computing

Autonomic Computing is the new computing model that will not only automate the management of increasingly complex systems but will also cut costs and ensure maximum system uptime.

2. BIG BLUE and AUTONOMIC COMPUTING

IBM popularly known as BigBlue is an IT conglomerate that has always focused on discovering new technologies and delivering innovative products and services. As the frontrunner of Autonomic Computing, the company is currently working on Policy Management for Autonomic Computing – A policy-based autonomic management infrastructure that simplifies the automation of IT and business processes.

2.1 Semblance of IBM in Autonomic Systems

A complete autonomic system is far from reality at the moment. However, the initiative itself of creating self-managing computing systems is nothing less of a Grand Challenge.

With autonomic computing, human intervention in most tasks associated with systems management will seem as archaic and unnecessary just as asking an operator for help for making a phone call seems today. Also, tasks like server load balancing, process allocation, monitoring power supply, automatic updating of software, pre-failure warning, memory error-correction, automated system backup and recovery, will be automated for autonomic sysems. Partially autonomous systems have already been achieved and are being successfully implemented. Some such notable examples are listed in Table 1 below.

Table 1: Industry Initiatives for Autonomic Computing [13]

Company	Autonomic Application
IBM	Autonomic Computing?
SUN MICROSYSTEMS	N1
HEWLETT PACKARD	Adaptive Enterprise
MICROSOFT	Dynamic Systems Initiative
INTEL	Proactive Computing

Key features of these industrial research projects are as follows-

- Sun – N1

  N1 is Sun's vision for the next-generation data center. It manages widely distributed computing resources such as servers, storage, software, and networks, and enables them to operate as a single entity. The core concepts of N1 are:-Virtualization (The process of modeling all the components in the network), Application and service level provisioning (The process of configuring and deploying software on systems) and Dynamic policy management (Automations of the configuration and deployment of software in the system).

- HP – The Adaptive Enterprise

HP's approach to autonomic computing is embraced in its Adaptive Enterprise strategy. It Provide tools, services, and products that deliver IT service levels that match the flow of real-time business activities or changes whenever needed. HP's adaptive technologies fall into three main areas:-Dynamic resource optimization, Automated and intelligent management and continuous and secure operations.

- Microsoft – Dynamic Systems Initiative

This initiative is targeted at helping IT teams to capture and use knowledge so as to design more self-.managing dynamic systems and automate ongoing operations, resulting in reduced costs and more time to proactively focus on what is most important to the organization. A key technology component of the DSI is the System Definition Model (SDM) which provides a common language, or meta-model, that can be used to create models that capture the organizational knowledge relevant to entire distributed systems.

- Intel – Proactive Computing

  This initiative introduces a new era of computing which focuses on human-supervised operation, where the user stays out of the loop as much as possible until required to provide guidance in critical decisions as depicted in Fig.3

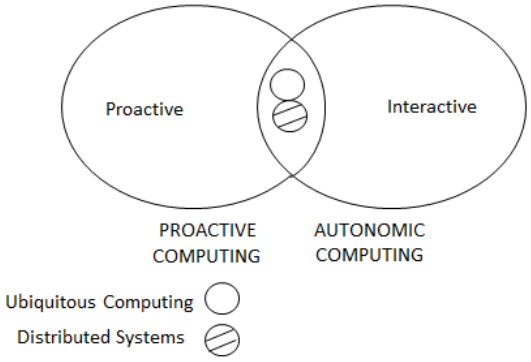

Fig 3: Relationship of computing paradigms

Various academic initiatives are also being pursued for autonomic computing. Some notable ones are Rutgers University (active middleware), CMU (self-healing systems), Columbia University (retrofitting legacy systems), and Imperial College (autonomic management of ubiquitous eHealth systems and autonomic web).

3. Core of autonomic computing

According to Big Blue, there are 8 key characteristics of any autonomic system that form the foundation of this software technology. These are as follows:-

- An autonomic system maintains comprehensive and specific knowledge about all its components;
- it has the ability to self-configure itself to suit varying and possibly unpredictable conditions;
- it continually monitors itself for optimal functioning;
- it is self-healing and is able to find alternate ways to function when it encounters problems;
- it is able to detect threats and protect itself from them;
- it is able to adapt to environmental conditions;
- it is based on open standards rather than proprietary technologies;

- And it anticipates demand while remaining transparent to the user.

Further these eight defining characteristics of autonomic systems can be further broadly classified as 4 fundamental properties by which the autonomic paradigm is identified, popularly known as Self-CHOP properties [12]. The same are depicted in Fig 4 below-

Table 2: Major Characteristics of Autonomic Computing [7]

Concept	*Current Trend*	*Autonomic Computing*
Self-configuration	Corporate data centers use multiple vendors and platforms. Installing, configuring, and integrating systems is time consuming and error prone.	Automated configuration of components and systems. Rest of system adjusts automatically.
Self-healing	Problem determination in large, complex systems can take weeks or even more.	System automatically detects, diagnoses, and repairs localized software and hardware problems.
Self-optimization	Systems have hundreds of manually set, nonlinear tuning parameters, whose number increases with each release.	Components and systems seek opportunities to improve their own performance and efficiency.
Self-protection	Detection of and recovery from attacks and cascading failures is manual.	System automatically defends against malicious attacks or cascading failures.

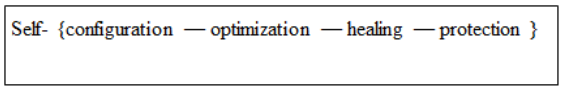

Fig 4: Fundamentals of Autonomic Computing

i) Self-configuration: vehemently adjust to changing environment

Self-configuration refers to the ability of the system to automatically adapt to changes in its physical topology as well as its software environment. An autonomic System should be able to adjust dynamically to variances in its environment like changes occurring from dramatic alterations in the system characteristics, deployment of new components or the removal of existing components. Desired dynamic adaptation can be achieved by using policies provided by the IT professional. Self-configuration improves system reliability by reducing human configuration errors and minimizing downtime to increase system resource availability. This ensures continuous strength and productivity of the IT infrastructure, resulting in business growth and flexibility. [7]

ii) Self-healing: ascertain, analyze and act to prevent disruptions

Self-healing is concerned with the ability of the systems to automatically recover from faults. An autonomic system should be proficient in detecting its malfunctions using some form of monitoring and thereby initiating as required, policy-based corrective actions like effecting changes in other
environment components or involving a product altering its own state, without disarraying the IT locale. The goal is to repair only that component that has failed without bringing down the
entire system so that resource availability is maintained even if the QoS is somewhat degraded. This warrants that the IT system becomes more resilient as day-to-day operations are less likely to fail [7].

iii) Self-optimization: attune resources and harmonize workloads to escalate the use of information technology resources.

Self-Optimization means an act of making a system as fully perfect, functional, or effective as possible. Accordingly, an autonomic system should be capable of tuning resources automatically .The tuning actions could mean reallocating resources, such as in response to dynamically changing workloads-to improve overall utilization, and ensure that particular business transactions can be completed in a timely
fashion. Self-optimization assures a high standard of service for both the system's end users and a business's customers [7].

iv) Self-protection: predict, discover, recognize and guard against threats

Self-protection is concerned with protecting the system from malicious attacks. An autonomic system should have an innate capacity to anticipate, detect, identify and protect against threats from external sources as well as attacks that are initiated from within the system. It should be able to detect hostile behaviors resulting from unauthorized access and use, virus infection and proliferation, and denial-of-service attacks, as they occur and take corrective actions like providing backup and recovery capabilities that are as secure as the original resource management systems. This allows businesses to consistently enforce security and privacy policies [7].

The above four aspects of autonomic computing as they are at present and how will these likely transform as contributing technologies mature, is tabulated in table 2 above. It shall also be noted that the road to this ideal of autonomic computing is expected to pass via a progression of 5 system self-management levels as shown in table 3 below[10].

Table 3: Levels of Autonomic Computing [3]

BASIC LEVEL 1	MANAGED LEVEL 2	PREDICTIVE LEVEL 3	ADAPTIVE LEVEL 4	AUTON-OMIC LEVEL 5
● Multiple sources of system generated data ● Require extensive, highly skilled IT staff	•Consolidation of data through management tools • IT staff analyses and takes actions	● System monitors, correlates, and recommends actions ● IT staff approves and initiates actions	● System monitors, correlates and takes action ● IT staff manages performance against SLAs	● Integrated components managed by business rules ● IT staff focuses on enabling business needs
	• Greater system awareness • Improved productivity	● Reduced dependency on deep skills ● Faster and better decision making	● IT agility and resilience with minimal human interaction	● Business agility and resiliency

3.1 Autonomic Systems Architectonics

The design of technical systems usually focuses on the intended functionality of the system and often obeys the "design follows function" principle[16].Consequently, a system is organized into components that implement the application specific functions. Such a system is then embedded onto some runtime environment that deals with execution failures and captures exceptions. Often such a design cannot satisfy the requirements of autonomic systems and therefore a generic architecture that introduces system components not at the level of application-specific functionalities, but at the level of functionalities derived from the key features of autonomic systems is required.

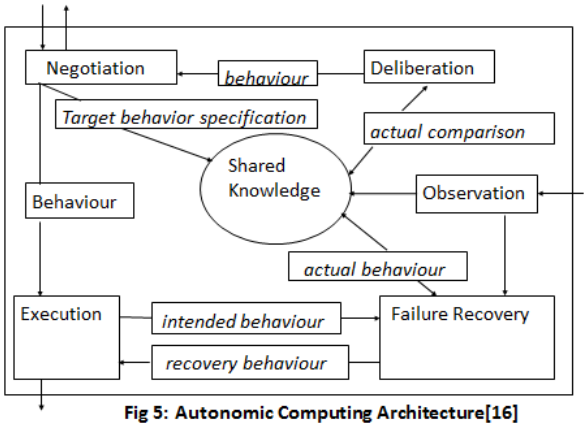

Fig 5: Autonomic Computing Architecture[16]

Each autonomic system runs in some environment or context. The interaction between the system and its environment occurs through three system components: negotiation, execution, and observation.

The negotiation component has a two-way interaction with the environment that allows the system to receive requirements from the environment, negotiate the fulfillment of the requested requirements, making itself known to other systems, or communicate its own requirements to other autonomic systems it is aware of. The main purpose of this component is to receive and actively construct a target behavior specification based on its interaction with the environment. This target behavior Specification is added to the shared knowledge of the system components.

The architecture in figure 3 above, highly abstracts the knowledge contents and format, and the sharing mechanisms between the various autonomic system components. We only assert that the knowledge base contains a representation of the actual system behavior, the system itself and the environment as perceived by the system. When a new target behavior is added to the shared knowledge, which differs from the actual behavior, a deliberation process is triggered that will produce a new behavior. The deliberation process sends the new behavior to the negotiation component that decides whether this behavior should be executed. The decision can for example be based on whether new requirements have been received that make the behavior already obsolete.

The execution component has a one-way output interaction with the environment to execute any behavior that was determined by the deliberation component and forwarded by the negotiation component. The execution component concentrates solely on executing the behavior in a specific environment, e.g., on expanding high-level action descriptions in sequences of lower-level system commands.

The observation component has a one-way input interaction to receive status information from the environment. The component observes the effect of what the execution component is executing without knowing what was actually executed. It adds its observations to the shared knowledge and produces a representation of its observations for analysis by the failure recovery process.

Limiting the interaction between the AC system and its environment helps to address the key factors of self-protection and hidden complexity. A system with a controlled interaction is less vulnerable to attacks and hides its internal complexity by exposing only clearly defined interfaces to its environment. The types of interaction introduced above (one-way, two-way) emphasize the predominant, not necessarily the only flow of information.

Two components that do not interact directly with the environment: deliberation and failure recovery. The deliberation component computes new behaviors for the autonomic system and encapsulates the "normal" application-specific functional components. It is responsible for fulfilling the key factors of self adaptivity and self-optimization. Two major fields of Artificial Intelligence (AI) play a dominant role in the development of deliberation components: machine learning and AI planning.

The failure recovery component adds self-healing and self protection capability to the AC system. Interestingly, it does not interact directly with the environment, but interacts with the execution and observation components only. The reason for this design principle lies again in the need to reduce the complexity of the system and enhance its robustness at the same time. The failure recovery receives information about the intended behavior of the system from the execution, i.e., the execution component tells it, for example, what action or command it intends to execute next. This information is used by the failure recovery to build an internal expectation of what will happen next in the system environment. The observation component tells the failure recovery what it actually observed happening in the environment. As execution and observation are completely decoupled in this architecture, they cannot inadvertently influence each other.

Further, the failure recovery analyzes the deviations between the intended and the independently observed changes occurring in the environment. For minimal deviations (that need to be precisely defined when implementing this architecture), it computes simple recovery behaviors that it sends to the execution component for immediate recovery. If greater deviations occur, it updates the shared knowledge with a new actual behavior. This will trigger an actual/target comparison and a new deliberation process that may lead to the replacement of the behavior in the execution component.

Any particular autonomic system will be based on a sophisticated implementation of the generic architecture explained above. In particular, the sharing of knowledge or information between the various components will usually distinguish between globally shared knowledge between all components and locally shared knowledge between only selected components. Furthermore, more than one instance of each component or complex components that are autonomic systems themselves can be included. In particular, the deliberation component will probably involve a hierarchical decomposition into application specific functional components, which is already common in realistic application systems. Self configuring autonomic systems can be expected to involve several deliberation components—specialized in computing system behaviors or computing new system configurations. The illustrated architecture is meant for a general design principle that will always require refinements and even modifications when instantiated for a particular IT application.

4. Applications of autonomic computing

Autonomic systems have found a wide variety of applications today. Perhaps the first application of autonomous computing were 'software agents' that made waves around 1999[11]. A prime example is Computer Associates' Neugents. Neugents, which are included in CA's Unicenter systems management software and its Jasmine object-oriented database, can look at up to 1,200 variables and make sense of it all.

'Spiders' or software agents from search engines are another popular example. Also called 'Bots', these agents hunt the Web looking for new websites and then return to the search engine and update its database with the new URLs.

Windows XP also incorporates autonomic features. When an application crashes, the operating system uses its autonomic recovery utility to recover the application from crash. This operating system also offers to report program errors to the Microsoft Support team. Further, Windows XP looks out for updates and automatically downloads these when available. Recent versions of Microsoft Office also include a repair feature. So if key program file (such as Winword.exe) gets corrupted or accidentally deleted, the software can reinstall it.

Plug-and-play is another element of autonomous computing. It allows automatic detection of new devices plugged into a computer system. The operating system will then fire up its hardware wizard, which guides you through the process of installing the appropriate drivers for the new device.

A notable IBM initiative for autonomic systems is project eLiza. The objective of this project is to develop hardware, software, and networks (total solutions) that will be able to allocate computing resources as required, to safeguard data, and ensure business continuity in case of a disaster. Details of this project are outlined in Table 4 below-

Table 4: A modernize on IBM's project eLiza[8]

With eLiza, IBM first started incorporating autonomic capabilities at the infrastructure level. That was Phase 1.Now in moving to Phase 2, which is building autonomic capabilities on the enterprise level. Till date the project has achieved the following milestones-

- eLiza technology has been incorporated in the IBM pSeries, xSeries and iSeries servers. It manages heterogeneous server integration and maintains cross-server security.
- eLiza e-business-management services match IT resource availability with business requirements to make sure business-performance levels are met.
- IBM's Intelligent Resource Director (IRD), a self-managing operating system for the eServer z900, allows the server to dynamically reallocate processing power to a given application as workload demands increase.
- Chipkill technology, which is used in IBM mainframes to virtually eliminate memory failures, is now available on IBM eServer p620 and p660.
- Software rejuvenation enables IBM's Windows servers to reboot automatically when they sense an impending problem that could crash the server.
- eLiza technology is also incorporated in IBM's Tivoli systems management software. It enables Tivoli to use the information it gathers to recognize its optimal configuration and fine-tune parameters accordingly.
- Currently IBM is incorporating an autonomic security feature called Enterprise Identity Mapping (EIM), which will allow a single log-on across an infrastructure. It is deploying a feature called Enterprise Workload Manager, or eWLM, which makes a heterogeneous infrastructure self-optimizing which will be the industry's first heterogeneous workload manager.

From table 4 above it is clear that IBM has incorporated different elements of eLiza in its different applications as well as servers. An initiative similar to eLiza is Project Oceano. It will enable a group of Linux servers to share jobs, and reassign jobs when new servers are added or removed from the cluster.

Compaq is also pursuing autonomic computing. It is offering a suite of tools collectively called Proliant Essentials. The tool with autonomic characteristics is Compaq Insight Manager. This software delivers pre-failure alerts for Compaq ProLiant servers, thereby proactively detecting potential server failures before they result in unplanned system downtime. Ancillary tool in the suite is ActiveUpdate, an advanced Web-based application that provides proactive notification and automatic download of software updates for all Compaq systems that range from handhelds to servers.

4.1 IBM Autonomic Applications

IBM coined the term autonomic computing in 2001 and since then has been beating the automation drum across its software and hardware product lines. The premise of a self-healing, self-protecting, self-optimizing and self-managing data center caused some industry watchers to scoff, but Big Blue stuck to its plans and now has more than 400 autonomic features shipping in 36 distinct products.

With virtualization, service-oriented architecture (SOA) and other innovative technologies increasing complexity across corporate data centers, automation is required more than ever and this is why the firm is introducing autonomic features across the entire portfolio -- from chips to business systems management software. It is actively working on a series of products in its four key software group brands--DB2, WebSphere, Tivoli and Lotus-that reflect varying levels in the progression toward autonomic computing.

A sampling of software in IBM's DB2 portfolio that's moving toward autonomic computing includes the following[9]:-

- DB2 UDB Version 8 includes self-managing and self-tuning features to help companies simplify and automate many of the tasks associated with maintaining databases. It's also the industry's first multimedia, Web-ready relational database management system delivering leading capabilities in manageability, reliability, performance and scalability.
- IBM's Data Management Tools are being developed along autonomic lines, with the Self-Managing and Resource Tuning (SMART) technologies, which are designed to reduce the cost and complexity of DB2 UDB tuning and managing. For instance, IBM DB2 Recovery Expert for z/OS and for Multiplatforms (Microsoft Windows, HP-UX, Sun Solaris, AIX and Linux) provides targeted, flexible and automated recovery of database assets, even as systems remain online. Its built-in SMART features provide intelligent analysis of altered, incorrect or missing database assets, including tables, indexes or data. It automates the process of rebuilding those assets to a correct point-in-time, often without taking the database or e-business operations offline.DB2 Recovery Expert's SMART interface offers database administrators (DBAs) improved efficiency in recovering a database to a specific point-in-time by automatically selecting the least time- and resource-consuming mechanism--either standard object recovery or a backout of the database changes--for a given recovery.IBM DB2 Performance Expert, with its starter set of SMART

capabilities, provides recommendations for system tuning to gain optimum throughput. New functionality in IBM's DB2 Automation Tool for z/OS, V1.3 advances the goal of enterprise autonomic computing by providing support for DB2 RECOVER and REPAIR and COPYTOCOPY utilities; predefined profiles that DBAs can use to back up or recover the catalog and/or directory; improved handling of migrated spaces; enhanced reporting and messages when building jobs and improved maintenance of the DB2 Automation Tool data repository.

- In the WebSphere family, IBM introduced autonomic computing features in WebSphere Application Server (WAS) Version 5.0. This version enables e-business applications to be developed, integrated and deployed within a reliable, secure and self-managing environment. New autonomic features enable WAS to automatically monitor, analyze and fix performance problems, allowing customers to adapt to a more fluid business environment.

In the Tivoli family, several new products possess autonomic computing features [17]:-

- IBM Tivoli Service Level Advisor is the first product of its kind to leverage a data warehouse for predictive analysis and reporting of service level agreements (SLAs) by simplifying and automating their management and linking IT service delivery to business objectives. Trend-analysis capabilities enable proactive management of existing IT resources, so that service levels can be maintained and SLA violations avoided.
- Tivoli Business Systems Manager simplifies mission-critical e-business systems management by managing real-time problems in the context of enterprise business priorities. It integrates with existing management solutions, such as monitoring and correlation, to help users view outages as they relate to the business.
- Tivoli Configuration Manager delivers an integrated solution for deploying software, as well as tracking hardware and software configurations across an enterprise.
- Tivoli Analyzer for Lotus Domino contains a Proactive Analysis Component allowing administrators to verify the availability and optimal performance of Lotus Domino servers. It provides intelligent server health monitoring and expert recommendations to correct problems.
- The Tivoli server health management and planning toolset is the first product to take advantage of new Lotus Domino statistics and activity measurements, and one of the only products that runs seamlessly inside the Domino Administrator. These tools analyze new statistics and activity measurements introduced in Lotus Domino 6 to predict sizing requirements and capacity bottlenecks.

IBM's autonomic computing software products are already stepping up to the plate to meet these requirements as they proactively manage IT systems to help ensure that e-business processes run uninterrupted, while their inherent complexity is masked from end users. These technologies are also necessary to enable grid and on-demand computing.

4.2Pros and Cons of Autonomic Computing

Autonomic computing has numerous benefits to offer. Immediate benefits include reduced dependence on human intervention to maintain complex systems accompanied by a substantial decrease in costs., Simplified user experience through a more responsive and real-time system, full use of idle processing power- including home PC's through networked system, seamless access to multiple file types as Open standards will allow users to pull data from all potential sources by re-formatting on the fly, deeper and more accurate returns due to natural language queries, optimized usage across both hardware and software because of scaled power, storage and costs, stability alongwith high availability and high security system. As fewer system or network errors will occur due to self-healing.

Long-term benefits will allow individuals, organizations and businesses to collaborate on complex problem solving.

However, Autonomic Computing faces numerous technology- and marketplace-related obstacles to widespread adoption. One of the autonomic computing key challenges is to develop approaches for different types of systems that run on various hardware and software platforms which frequently include heterogeneous components.

Another hurdle is to determine which existing techniques can yield approaches that will consistently and automatically work without user intervention. Further, there also exists a risk that technologies that monitor, manage, and heal systems on their own could also harm systems— for example, by introducing bugs, deleting important system settings, or removing executables.

5. Future predictions for autonomic computing

The whole world is moving towards autonomic systems and computers are following the trend. The automobile industry for instance. Engineers at Daimler-Chrysler AG (manufacturer of the highly reputed Mercedes-Benz cars), have been working on autonomic systems to ensure driver/passenger safety, for many years. The fruits of their efforts are ABS (Anti-lock braking system) and other safety innovations. ABS prevents the wheel from locking when the car goes into a skid. This ensures the car can still be steered and thus prevents accidents. The ABS comprises electronic sensors and solenoid valves in the wheel hubs.

Autonomic Computing in computers is not new anymore. Notable examples of this technology are ECC (Error-Correcting Code) memory, SMART (Self-Monitoring, Analysis and Reporting Technology) for hard disks, and fault-tolerant servers. Academia and industry alike are

working towards making such technologies more autonomous. In future, there will be minimal human intervention, and computing sub-systems will be able to proactively detect and rectify potential faults before any failure occurs. A fully autonomous computing system does not exist today, but such systems could make the concept of 24 x 7 x 365, or 99.999 percent uptime possible.

6. Conclusion

Autonomic computing is an expedition wherein progress will be made in a series of evolutionary steps. There is an urgent need of an increased research focus on availability, maintainability, scalability, cost, and performance of autonomic computing. To achieve this grand challenge of autonomic computing will involve researchers in a diverse array of fields, including systems management, distributed computing, networking, operations research, software development, storage, artificial intelligence, and control theory, as well as others. Autonomic computing is thus the ensuing era of computing and represents an exciting new research direction which must be explored to deliver the next big thing in the IT industry.

References

[1] www.research.ibm.com/autonomic/overview/

[2] Schaller B., "The origin, nature, and implications of 'Moore's Law,' the benchmark of progress in semiconductor electronics,"1996,http://mason.gmu.edu/~rschalle/moorelaw.html

[3] G. Ganek,T. A. Corbi,"The dawning of the autonomic computing era", IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003

[4] Jeffrey O. Kephart, David M. Chess,"The Vision of Autonomic Computing.", 2003.,*IEEE Computer* 36(1):41-50

[5] http://www.freshpatents.com/System-and-methods-for-it-resource-event-situation-classification-and-semantics-dt20070104ptan20070005535.php

[6] http://coolquotescollection.com/6655/most-technology-has-the-shelf-life-of-a-banana

[7] Jeffrey O. Kephart, David M. Chess,"The Vision of Autonomic Computing.", 2003,*IEEE Computer* 36(1): 41-50

[8] http://www.expresscomputeronline.com/20020819/focus1.shtml

[9] http://www.ibmsystemsmag.com/aix/trends/whtasnew/Self-Managing-Systems/?page=2

[10] http://www.ibmsystemsmag.com/CMSTemplates/IBMSystemsMag/Print.aspx?path=/aix/trends/whatsnew/Self-Managing-Systems

[11] http://hareenlaks.blogspot.in/2011/03/initiatives-of-autonomic-computing.html

[12] http://www.ibm.com/developerworks/library/ac-edge4/index.html

[13] http://www.ai.uic.edu/aiSeminarPresentations/AISeminar111805Michelle.pdf

[14] http://www.authorstream.com/Presentation/suryaprakashpand-1541088-aunomic-computing/

[15] http://seminarprojects.com/Thread-autonomic-computing-a-seminar-report?pid=30322

[16] http://www.zurich.ibm.com/pdf/ebizz/idd-ac.pdf

[17] http://www.ibmsystemsmag.com/CMSTemplates/IBMSystemsMag/Print.aspx?path=/mainframe/trends/whatsnew/Self-Managing-Systems

Author profile

Divya Bahl is currently pursuing her master's degree program in computer applications (MCA) from Institute of Information Technology and Management affiliated to Guru Gobind Singh Indraprastha University,INDIA.

Ritika Wason is a B.Sc, B.Ed, M.C.A and M.Phil. She is currently pursuing her Ph.D from Sharda University. Working as Assistant Professor with Institute of Information Technology and Management she has authored a number of books on Software Testing. She also has a number of National and International publications to her credit.