SURVEY ON CHARACTERISTICS OF AUTONOMOUS SYSTEM

Taranjeet Singh and Avadhesh Kumar

School of Computing Science & Engineering, Galgotias University
Greater Noida, Uttar Pradesh, India

ABSTRACT

Earlier software systems used to be very small in size and were developed and maintained by only few IT experts, but era changed and the requirements of user's started increasing day by day. Due to heterogeneous and dynamic behaviour of current software systems their structure is becoming complex which leads to rapid rise in their cost and complexity, now more IT professionals are required to develop, design and maintain them. If the complexity keeps on increasing in this manner then one day software systems would be unmanageable, very costly, low maintainable, and least reliable. Like biological systems software systems must be designed which will be once developed and deployed to users then these systems will be able adapt and change accordingly according to user's requirements. Can these software systems be developed? The answer to this is yes, autonomic computing based software systems can do the job, many of these systems have already been developed and future waits for software systems to be entirely autonomous. This paper tends to explain autonomic computing, its characteristics and their relation with ISO 9126 quality factors.

KEYWORDS

Autonomic computing, Self properties, ISO 9126 quality factors.

1. INTRODUCTION

These Biological systems such as human body have ability to adapt according to surrounding environmental conditions, it can recover itself, prevent itself from danger, manage its day today tasks and similarly Paul horn from IBM provided us with new approach in 2001 to develop computing systems and named the approach autonomic computing [1]. System developed using autonomic computing is known as autonomic computing system, they have ability to adapt unpredictable conditions of IT environment, manage themselves, protect themselves recover from failures and faults with minimum or no human interference. Even software systems can also be injected using properties of autonomic computing and are known as autonomous software systems or autonomic computing based software systems examples of such systems are DB2, IBM Tivoli risk manager and many more. After 2001 many IT companies seemed interested in this new computing approach these companies are HP, SUN, DELL, Microsoft etc. Although there are many characteristics or properties of autonomous systems, these characteristics are *self configuring, self healing, self-optimizing, self protecting, self aware, context aware, open and anticipatory* [2], these are known as self properties and are discussed in later section of this paper. The major goal of autonomous systems is self management which is only possible when all the components of the system work together to achieve it. Architecture of autonomic systems contains five building blocks these building blocks are *autonomic manager, knowledge source, touch points, manual manager and enterprise service bus* [3]. These building blocks work together to achieve self management. Autonomic manager manages the managed element which can be software or hardware. It uses MAPE (Monitor, Analyze, Plan and Execute) function to do

DOI:10.5121/ijcsit.2016.8210

so. The monitor function monitors the and collects details from the managed element, analyze function analyzes the collected details and check if any change is to be made in the managed element or resource then plan function develops the necessary actions that are required to be taken these actions are performed using execute function. Knowledge Source can be implemented using database, dictionary or repository. It consists of knowledge that autonomic manager uses to perform self management tasks, knowledge can be either passed to autonomic manager through policies or is retrieved from external knowledge source or is created itself by autonomic manager [3]. Changes are made in the managed element using policy which is provided to the autonomic manager. Reference model of autonomic manager is provided by Sahadeva et.al [17]. Autonomic manager communicates using touch points which consist of sensors and effectors. Sensors sense the information about the current state of the managed element and effectors checks the changes that are made to manage element by autonomic manager. Some of the functions of autonomous system can be managed by IT professionals manually using manual manager. Enterprise Service Bus is used to integrate and connect various building blocks of autonomic system's architecture [3].Zhu et.al [16] various patterns of implementing Autonomic elements while Hubescher et.al [13] also explains the architecture of Autonomic systems.

2. CHARACTERISTICS OF AUTONOMOUS SYSTEMS

Characteristics provide self management capabilities to autonomic system and these characteristics are often known as self properties. Salehie et.al [4] states that these characteristics are either major or minor, Characteristics like *self healing, self configuring, self optimizing, self protecting* are major characteristics while characteristics like *self aware, context aware, open and anticipatory* are minor characteristics of autonomous system. Both major and minor characteristics are shown in figure 1 and explained in this section.

2.1. Major Characteristics

Autonomic computing system has four major characteristics, these characteristics are *self healing, self configuring, self optimizing, self protecting* and these are known as self CHOP [4]. This section explains these properties.

2.1.1. Self Configuring

By this property autonomic system is able to add new components, configure its components, remove old or faulty components, and reconfigure them by themselves with a little or no human interference [5]. These components will be able to adapt the IT environment and would provide users with desired performance and quality.

2.1.2. Self Healing

Autonomic system can heal itself and its components through this property. It is able to detect faulty components, try to diagnose these components using some corrective mechanism. But self healing process should not harm other components of the system [5]. ROC (Recovery Oriented Computing) is the application for performing self healing process; it provides us with various mechanisms for recovery of the system from many failures [4].

2.1.3. Self Optimizing

It means autonomic system can optimize itself and its components for improving its efficiency, tune its resources and satisfy requirements of different users. Resource utilization and workload management are aspects of these characteristics [4]. *Self optimizing* is also called as self adjusting or self tuning property [7].

2.1.4. Self Protecting

As biological systems are capable of protecting themselves from dangers and other natural calamities similarly autonomic systems have self protecting property, by this property they are capable of defending themselves and their components from various malicious attacks. Autonomic systems can detect hostile behaviours; take various corrective actions for guarding themselves from any attack [5].

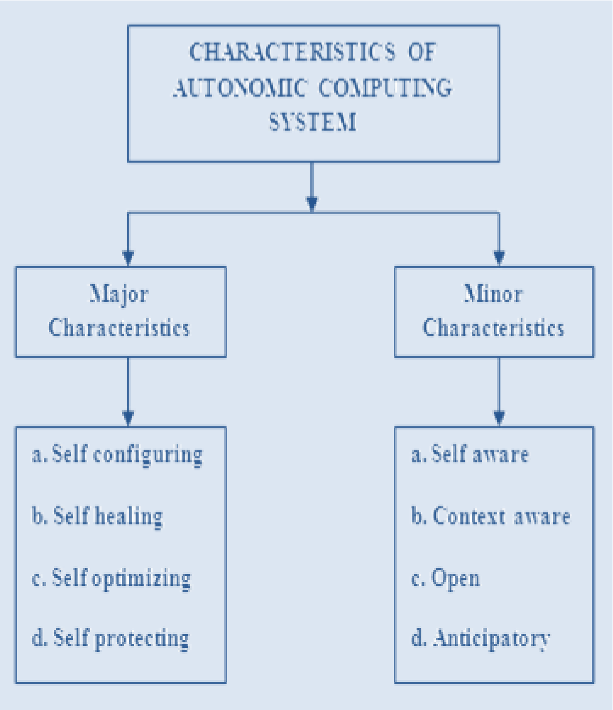


Figure 1:- Characteristics of autonomic computing [4].

2.2. Minor Characteristics

Autonomous systems have four minor characteristics these are *self knowledge* or *aware, open, context aware* or *environment aware* and *anticipatory* [4].These are explained below.

2.2.1. Self aware

The system is aware of its internal states, components and their behaviours [7].

2.2.2. Open

With this property autonomous system can perform its tasks in heterogeneous environment only not hermetic environment [1] [4].

2.2.3. Context aware

Autonomic system must be aware of the environment where it is currently being executed and it should be capable enough to react to changes in its operational environment [7].

2.2.4. Anticipatory

Autonomic system can anticipate its optimized resources while keeping its complexity hidden from the users [1].

Other characteristics were later defined by many of the researchers, Sterritt et.al[6], Li et.al[12] and Dobson et.al[14] provide us with many other characteristics of these systems these characteristics are *self adjusting, self monitoring, self adapting, self critical, self defining, self destructing, self diagnosis, self governing, self recovery, self reflecting ,self simulation* and *self regulating*. While developing any autonomic computing based software system then it is necessary to follow international standards for developing quality software system. ISO 9126 defines set of factors to measure software quality these factors are *functionality, reliability, usability, portability, efficiency* and *maintainability* [8] [9]. Next section explains ISO 9126 quality factors and how these factors are related to characteristics of autonomic computing.

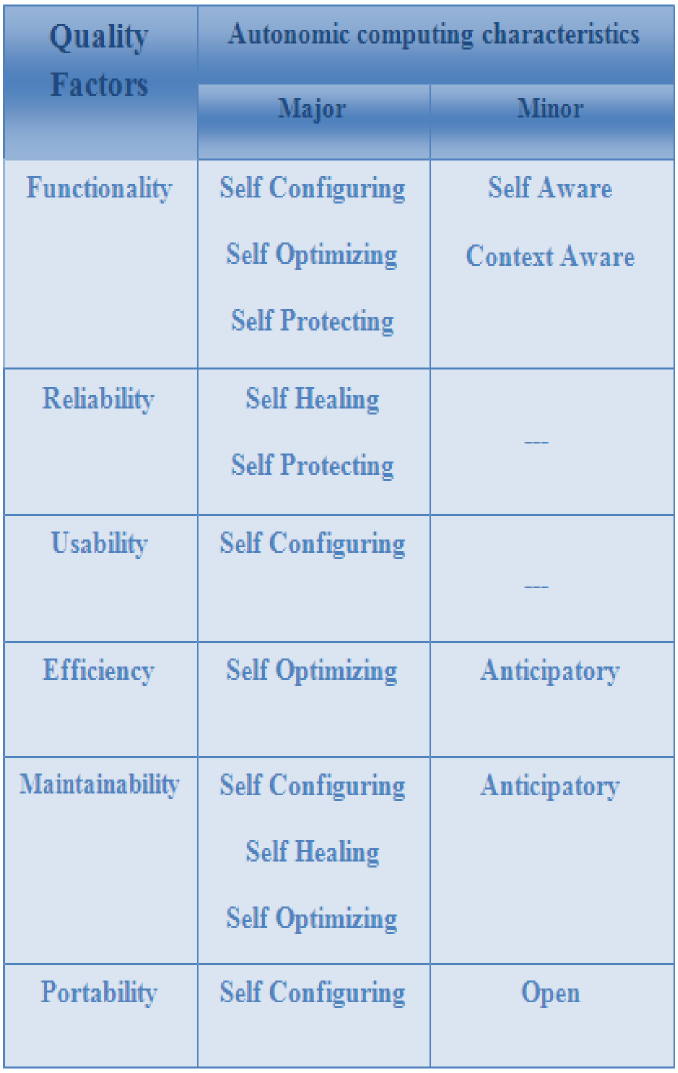
3. RELATIONSHIP BETWEEN QUALITY FACTORS AND CHARACTERISTICS OF AUTONOMIC SYSTEM

ISO 9126 defines six characteristics for measuring the quality of software system. These characteristics are *functionality, reliability, usability, efficiency, maintainability* and *portability* [8]. These software quality factors are closely related to both major and minor characteristics of autonomous system. Nami et.al [10] and Salehie et.al [4] shows relationship between characteristics and quality factors. This section will explain quality factors and their relationship with characteristics of autonomous system.

Functionality means the degree at which software system satisfies the needs of its users. Suitability, accuracy, interoperability and security are sub characteristics of this quality factor [8] [9]. *Reliability* is the amount of time system is able to provide its services to the users or time for which system is available to the users [8].*Usability* means how easily the user is able to communicate with the system and perform tasks on the system. Understandability, learnability, operability, attractiveness are sub factors of Usability [8] [9]. *Efficiency* has two sub factors time behavior and resource utilization [9]. It means degree to which software system can make use of resources and fulfil its objectives [8].*Maintainability* is the ease with which changes can be made in the software system to remove faults, improve overall system perform or for adapting the environment. Its sub factors are analyzability, scalability, changeability and testability [9].*Portability* is the ease with which software system can be transferred from one environment to another. Its sub factors are adaptability, instability, replaceability, co existence [9]. Figure 2 shows the relationship between ISO 9126 quality factors and characteristics of autonomic computing. When *self configuration* is performed then it will have impact on quality factors like *maintainability, functionality, usability and portability* [4] [10].

When *self configuration* is done then there would be one or more changes made, after the changes our system must satisfy the needs of the users, user must be able to communicate with the system easily and it is also possible that our software system can be ported to another environment too during *self configuration. Self Healing* has a relationship with *maintainability and reliability* [4] [7], because when software system is *self Healing* then it must perform any changes to its components or its state and must be available to the users for providing its services. *Self optimizing* means to optimize its tasks and tuning of the resources for improving efficiency so for *self optimization* regular changes are required in the system for improving overall *efficiency*, which depicts its relationship with *functionality, efficiency* and *maintainability* [4] [10]. *Self optimizing* is strongly related to *efficiency* [7].

Self protecting has strong relationship with *reliability* and can also be related to *functionality* [7], as system can protect itself from various attacks and failures to be available to users for satisfying their needs. For fulfilling user's requirements and satisfying them the autonomous system is self aware and context or environment aware. So characteristics *self aware* and *context aware* have *relationship with functionality* [10]. *Open* means that system should run in heterogeneous environment only, for running in heterogeneous environment autonomic software should able to run on multiple platforms so *Open* can have impact on *Portability* [4]. Autonomic system is *anticipatory* as it can anticipate its optimal resources while keeping its complexity hidden from the users [1]. *Anticipatory* has a relationship with quality factors *efficiency and maintainability* [4]. Next section explains the challenges and issues of autonomic computing.



Quality Factors	Autonomic computing characteristics	
	Major	Minor
Functionality	Self Configuring Self Optimizing Self Protecting	Self Aware Context Aware
Reliability	Self Healing Self Protecting	---
Usability	Self Configuring	---
Efficiency	Self Optimizing	Anticipatory
Maintainability	Self Configuring Self Healing Self Optimizing	Anticipatory
Portability	Self Configuring	Open

Figure 2:- Relationship between quality factors and characteristics of autonomous system [4]

3. Challenges of Autonomic Computing Systems

After 2001 many companies and researchers from all around the world started to work on autonomic computing. Many projects and applications based on autonomic computing have been developed, but here are some issues that are needed to be sorted out for making our system autonomic these are stated below.

- *Self Optimization* is a big challenge because the autonomous system need to optimize its components and tune its resources regularly, their performance should not degrade while adapting different environments [10].
- As autonomous system is *open* it can run and perform its tasks in heterogeneous environment only but in heterogeneous environment there are different platforms, so it will be a challenge for our system to run efficiently on all these environments and adapt them.
- All the autonomic elements should have a strong relationship with each other for performing self management tasks. As the IT environment conditions can change dynamically it is challenge to adapt these unpredictable conditions and maintain a strong relationship with each other [10].
- Autonomic computing approach came into existence for managing the growing complexity in IT environment [11] , As many of these systems have been developed and still infinite number of systems are still to be developed so it would be interesting challenge to see at what level these systems can manage or managing the complexity of IT environment.
- Autonomic computing is a new field many people don't even know about it. While designing these systems it became a challenge to know the problem or to identify the problem [5].
- Nami and Sharifi [10] stated that robustness is another major challenge of these systems. They component should be robust enough to work in active state. It should be capable enough to work for various functionalities for resistance without changing its structure.
- Another challenge is to measure both minor and major characteristics of the autonomic system. A framework must be developed for bridging gaps between characteristics and quality factors [4].
- How to incorporate self properties into a non autonomous or semi autonomous system is another challenge various case studies and experiments are required for doing it so [4].
- For designing autonomous system there are many tools but it should be known that which tools will be suited for what purpose [5]. Identifying these tools is itself a challenge.
- While implementing autonomic systems self managing capabilities like adapting, healing, protecting, optimizing, configuring must be incorporated. As autonomic computing is new area of research and lack of proper standards make it more difficult to implement and develop these systems [5].
- This approach does not provide us with any clear idea of what will be autonomic computing systems requiring for their development [18].
- It is very challenging to develop these systems due to challenges in all of their development stages, these stages are implementation, design, testing and verification & validation [18].
- Dealing with uncertainty is one of the major challenges of these systems; Uncertainty can be change in environment, policies or goals of the system [19].
- Using Autonomic computing for current and future space missions of NASA (National Aeronautics and Space Administration) will also be challenging [15]. ANTS (Autonomous NanoTechnology Swarm mission) are major example of such space missions [20].

4. CONCLUSION

Autonomic computing is new and promising area for developing applications for IT environments. The major issue which IT is facing is complexity, with self managing capabilities autonomic system is capable of handling all kind of complexities and adapting any changes in their environment. Many researchers have shown their interest in this new approach of computing. Various applications and projects are already developed after 2001. In this paper various challenges are discussed that this approach is facing today future work can be resolving any of these challenges. Researchers from all around the globe are working in this area but still lot of work has to be done.

REFERENCES

[1] P. Horn, "Autonomic Computing: IBM Perspective on the state of information technology", IBM 2001.

[2] M. Parashar and S. Hariri, "Autonomic computing: An Overview", Springer Verlag, pp. 247-259, 2005.

[3] IBM white paper, "An architecture blueprint for autonomic computing", page 1-32, 2003.

[4] M. Salehie and L. Tahvildari, "Autonomic computing: emerging trends and open problems", ACM 2005.

[5] K. Ahuja and H. Dangey, "Autonomic Computing: An emerging perspective and issues", International conference on issues and challenges in Intelligent computing techniques (ICICT), IEEE, page no 471-475, 2014.

[6] R. Sterritt, M. Parashar, H. Tianfield and R. Unland, "A concise introduction to autonomic computing", Advance engineering informatics science, page no 181-187, 2005, ELSEVIER.

[7] M. Salehie and L. Tahvildari, "Self Adaptive software: Landscape and Research Challenges", ACM Transactions on Autonomous and Adaptive systems, Vol. V, March, 2009.

[8] P. Narman, P. Johnson and L. Nordstrom, "Enterprise Architecture: A Framework Supporting System Quality Analysis", page-130 -141,11th IEEE International Enterprise Distributed Object Computing Conference, 2007, IEEE.

[9] H. Jung, S. Kim, and Chung, C. "Measuring Software Product Quality: A Survey of ISO/IEC 9126", page 89-92, 2004, IEEE software, IEEE Computer Society, IEEE.

[10] M. R. Nami and M. Sharifi, "Autonomic computing: A new approach", proceeding of first Asia international conference on modelling and simulation (AMS'07), IEEE, 2007.

[11] J. Kephart and D. Chess, "The vision of autonomic computing", IEEE computer society, 41-50, 2003, IEEE.

[12] X. Li, H. Kang, P. Harrington and J. Thomas, "Autonomic and trusted computing paradigms," Springer-Verlag Berlin Heidelberg, page 143-152, 2006, Springer.

[13] M. C. Huebscher and J. A. Mccann, "A survey of Autonomic computing - Degrees, Models and Applications," ACM Computing Surveys, pp.1-31, volume 40, no 3, August 2008, ACM.

[14] S. Dobson, K. Sterritt, P. Nixon and M. Hinchey, "Fulfilling the vision of Autonomic Computing", IEEE Computer Society, page 35- 41, 2010, IEEE.

[15] R. Sterritt, "Apoptotic computing: Programmed death by default for computer-based systems," IEEE Computer society, page 37-43, 2011, IEEE.

[16] Q. Zhu, L. Lin, H. M. Kienle, and H. A. Muller, "Characterizing Maintainability Concerns in Autonomic Element Design", page no 197-206, 2008, ICSM,IEEE.

[17] K. Sahadeva, S. K. Yadav, and A. Sharma, "Minimizing the cost of software by injecting autonomic computing elements", page no 685 -690,2012, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, IEEE.

[18] P. Lin, MacArthur, A., and J. Leaney, "Defining Autonomic Computing: A Software Engineering Perspective", Proceedings of the 2005 Australian Software Engineering Conference IEEE computer society, IEEE.

[19] M. A. Yahya, and A. Dahanayake, "Autonomic computing: A Framework to identify Autonomy Requirements", Advance engineering informatics science; page 235 – 241, 2013, ELSEVIER.

[20] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and A. C. Rouff, "Autonomous and Autonomic Systems for Future Space Exploration Missions", IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, volume - 36,no -3,May 2006,IEEE.

AUTHORS

[1]Taranjeet Singh completed his B.Tech in Computer Science and Engineering From Guru Gobind Singh Indraprastha University Delhi, India in 2013.Now he is pursuing M.Tech from Galgotias University Greater Noida, India. His areas of research are Software Engineering and Soft Computing.

[2]Dr. Avadhesh Kumar completed his PhD in Computer Science from Thapar University Punjab, India. He is currently working as Professor and Dean Planning at Galgotias University Greater Noida, India. His areas of research are Software Eng, Aspect-Oriented Software System, Component-Based Software Development and Soft Computing.