23th International Scientific Conference
Strategic Management and Decision Support Systems in Strategic Management

April 26.-27, 2018, Subotica, Republic of Serbia

Margarita Janeska
Faculty of Economics
Prilep, Macedonia

Kosta Sotiroski
Faculty of Economics
Prilep, Macedonia

Dejan Zdraveski
Faculty of Economics
Prilep, Macedonia

CONTROL LOOPS-KEY FACTOR FOR AUTONOMIC COMPUTING SYSTEMS REGULATION AND OPTIMIZATION

Abstract: Increased complexity, heterogeneity, uncertainty, and scale require new paradigms to design, control and manage systems and applications. Autonomic computing systems provide a solution to these issues. An autonomic computing system is a system that manages itself. Autonomic computing is a new attractive paradigm to tackle the problem of growing software complexity. These systems are characterized by their self-*properties: self-configuration, self-healing, self-optimization, and self-protection. Autonomic computing systems, also known as self-*systems, can regulate and maintain themselves without human intervention. Also, the target of autonomic computing is improving the manageability of IT process, it has great implications on a company's ability to transition to n demand business, where business processes can be rapidly adapted to realize on demand goals. These systems are constantly checked in terms of their optimization and are automatically adapted to changing conditions.

The goal of an autonomic computing architecture is to reduce intervention and carry out administrative functions according to predefined policies. Namely, an autonomic system is made of a connected set of autonomic elements that contain resources and deliver services to humans and other autonomic elements. Autonomic elements will manage their internal behaviors and their relationships with other autonomic elements in accordance with policies that humans or other elements have established. Autonomic computing theory integrates several fields: distributed computing, artificial intelligence, security and reliability, systems and software architecture, control theory, and systems and signal processing theory.

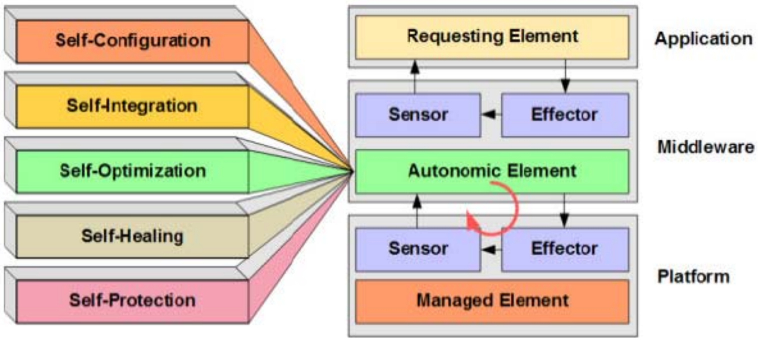
At the heart of an autonomic system is a control system, which is a combination of components that act together to maintain actual system attribute values close to desired specifications. The problem of controlling autonomic computing systems is gaining importance due to the fact that computing systems are becoming more and more dynamically reconfigurable or adaptive, to be flexible in their environment and to automate their administration. The control is realized using a control loop that involves sensors and actuators that are connected to the process i.e., the system to be controlled.

The objectives of this paper are to underline: the importance of autonomic systems and the need for their development, the correlation between the levels of autonomic systems and business objectives, and the role and functioning of the control cycle, which occupies a special place in the autonomic computing systems architecture.

Key words: autonomic computing, self-adaptive systems, control loops, autonomic computing element, regulation.

1. THE NEED FOR THE DEVELOPMENT AND IMPLEMENTATION OF AUTONOMIC COMPUTING SYSTEMS

Autonomic computing represents a fundamental shift in managing IT. Autonomic computing is term introduced by IBM. Autonomic computing systems are characterized by the inherent self-management properties of computer systems. Autonomy is accomplished through key aspects, such as self-regulation, self-customization, self-organization, self-optimization, self-configuration, self-diagnosis of errors, self-protection and autonomy. They are intelligent open systems that manage complexity, know themselves, continuously adapt to unpredictable conditions, prevent and recover from failures and provide a safe environment. Autonomic computing systems allow them to adapt to changes in accordance with business policies and goals, with "minimum human participation". An autonomic computing system manages and controls the functioning of computing systems and applications without any user input or intervention, in the same way as autonomic nervous system regulates the human body systems without conscious input from the individual.[1] The realization and implementation of autonomic computing systems will be achieved through a combination of process changes, the evolution of skills, emerging technologies, appropriate architecture and open industry standards. The human administrator can more reality concentrate on defining high-level business objectives, or policies, and are freed from dealing with the lower-level technical details necessary to achieve such objectives, as these tasks are now performed by the autonomic system.The architecture of autonomic computing contains two main entities: autonomic manager and managed resources (Janeska M, 2017) (More details about the architecture will be presented in the following text when it is explained about the meaning and functioning of the control loop). It needs to be pointed out that there is a correlation between self-*properties and autonomic architecture, which can be seen from the following figure:



Picture 1. Correlation between self-*properties and autonomic architecture
Source: Sakthi Nathiarasan A, 2014

The need for autonomic computing systems is very urgent. The basic reasons for the emergence of autonomic computer systems are:
Harder to anticipate interactions between components at design time, i.e. need to defer decisions to run time;

- Computer systems are becoming too massive, complex, to be managed even by the most skilled IT professionals;
- The workload and environment conditions tend to change very rapidly with time.
- Over and above, with autonomic computing systems it is also achieved:
- Reducing total cost of ownership;
- Improving quality of service;
- Accelerating time to value;
- Managing IT complexity.

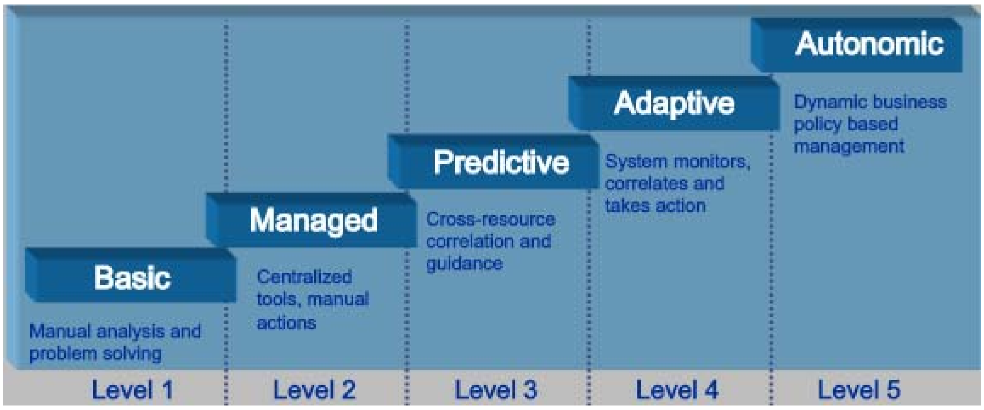
For successful handling, the objectives of the design of the system must transform the so-called "ordinary" performance/price requirements into stability and management issues. Efforts must be made to simplify and automate the management of systems. Today's systems must be developed to become more self-management, i.e. self-configurable, self-correcting, self-optimizing and to provide self-protection. In other words, there is only one answer: technology should be managed on its own. It is necessary to develop real software, real architecture, real mechanisms, so intead of technology to function in its usual way and need a person to do everything, it starts to function more like an "intelligent" computer and begin to take care of its needs. If someone attacks it, the system needs to recognize the attack and take action to deal with it. If more computing power is needed, it complements itself and there is no need for man to intervention.(I. Wladawsky-Berger , 2001)

IBM hardware and software systems have already made significant progress in introducing the named functionalities of autonomic computing systems. But the integration of these functionalities has not yet been achieved.(IBM Corporation, 2001) Efforts to achieve a cohesive system must overcome improvements in individual components. These components must be consolidate, include an integration architecture that will establish instrumentation, policies, and collaboration technologies so that asset groups can work together, for example, through network systems. System management tools will play a central role in coordinating the activities of the system components, providing a simplified mechanism for the administration system and for transforming business goals into executive policies to regulate the activities of available IT resources.

2. CORRELATION BETWEEN THE LEVELS OF AUTONOMIC COMPUTING SYSTEMS AND BUSINESS OBJECTIVES

In order to implement autonomic computing systems, the business must monitor their evolution and provide improvements to existing systems that will provide meaningful self-managing values to customers without requiring them to completely replace their current IT environment. New open standards must be developed to define new mechanisms for inter-heterogeneous systems.
Self-management systems can be designed to support different maturity levels of self properties:

- basic;
- managed;
- predictive;
- adaptive and
- autonomic.
- These are shown in the following figure:



Picture 2. Levels of autonomy
Source: IBM Corporation, 2001

This figure, in fact, represents the correlation between autonomy level and business goals. As shown in the figure, the *basic level* is the starting point where some IT systems are today. In the basic level process is informal and tools are local. Each element of the system is managed independently by IT professionals, who install, monitor and eventually replace them. Information technology sees the organization as an expense, with variable labor costs being preferred in terms of investment. At the *management* level, systems management technologies can be used to gather information from different systems, reducing the time the administrator needs to collect and synthesize information, given that systems become more complex for work. At management level process is formal, i.e. documented and used consolidated resources for solving problems with the management system. At this level, IT organizations are measured in terms of the availability of their management resources, the time they need to solve problems in their problem management system and the time it takes to complete formal work requirements. To improve these measurements, IT organizations document their processes and continuously improve them through feedback cycles and adopting best practices. IT organizations get efficiency through consolidation of management tools from the set of strategic platforms, and also through the hierarchical organization for managing problems. At the *predictive* level, new technologies are introduced that provide a connection between several elements of the system, whereby the system itself can begin to recognize models, anticipate optimum configuration, and provide advice on the course of the activity to be undertaken by the administrator. At this level process is proactive and are utilized more sophisticated tools: role-based consoles with analysis and recommendations, product configuration advisors, real-time view of current & future IT performance, automation of some repetitive tasks, common knowledge base of inventory and dependency management.(IBM White Paper, 2002) At predictive level, IT organizations are measured in terms of the availability and efficiency of their

business systems and their return on investment. In order to improve these measurements, IT organizations measure, manage, and analyze the performance of a transaction. Prediction tools are used to project future IT performance and many tools provide recommendations for their improvement. At the *adaptive* level, not only does the system monitor, correlate and develop action plans, the system also takes actions according to established policies. This level allows the staff to manage performance against service level objectives. Users can communicate with the system to track business processes or change targets. At this level the process is characterized by many resource and transaction and policy management tools that drive dynamic change based on resource specific policies. At the adaptive level, IT resources are automatically predicted and tuned in order to optimize transaction efficiency. Business policy, business priorities, and service contracts streamline the behavior of the autonomic infrastructure. IT organizations are measured in terms of the number of end-to-end responses to the business system (transaction performance), the degree of efficiency, and their ability to adapt to changing work tasks. At an *autonomic level*, the functioning of the system is regulated by business policies. At this level there is a complete automation of all IT resources and all IT services and are utilized costing/financial analysis tools, business and IT modeling tools automation of some e-business management roles etc.
At an autonomic level, IT organizations are measured in terms of their ability to make the business successful. To improve business metrics, IT tools understand financial parameters related to e-business activities and support IT activities. Advanced modeling techniques are used to optimize e-business performance and quickly develop new optimized e-business solutions.
The following table gives an overview of the autonomic maturity index

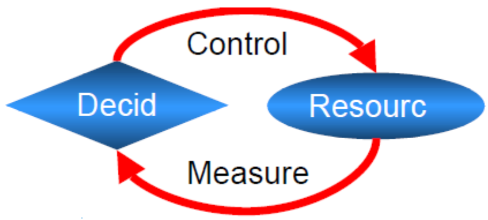
	Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Characteristics	Rely on system reports, product documentation, and manual actions to configure, optimize, heal and protect individual IT components	Management software in place to provide consolidation, facilitation and automation of IT tasks	Individual IT components and systems able to monitor, correlate and analyze the environment and recommend actions	IT components, individually and collectively, able to monitor, correlate, analyze and take action with minimal human intervention	Integrated IT components are collectively and dynamically managed by business rules and policies
Skills	Requires *extensive, highly skilled* IT staff	IT staff *analyzes and takes actions*	IT staff *approves and initiates actions*	IT staff *manages performance* against SLAs	IT staff *focuses* on enabling business needs
Benefits	Basic requirements addressed	Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better decision making	Balanced human/system interaction IT agility and resiliency	Business policy drives IT management Business agility and resiliency
	Manual				Autonomic

Picture 3. Autonomic maturity index
Source: B.Jacob and all, 2004

3. CONTROL LOOP AUTONOMIC COMPUTING

An autonomic system is made of a connected set of autonomic elements that contain resources and deliver services to humans and other autonomic elements. Autonomic elements will manage their internal behaviors and their relationships with other autonomic elements in accordance with policies that humans or other elements have established.(J. Kephart and D.M. Chess, 2003)

At the heart of an autonomic system is a control system, which is a combination of components that act together to maintain actual system attribute values close to desired specifications, as shown in follow figure:



Picture 4. Control loop
Source: Hausi A. Müller and all, 2006

There are open and closed control loop. Open-loop control systems are those in which the output has no effect on the input. Closed-loop control systems are those in which the output has an effect on the input in such a way as to maintain a desired output value. An autonomic system embodies one or more closed control loops. The biggest challenge in an autonomic architecture is to build closed control loops, the most important concept of self-management. Closed control loops are still called feedback control loops. The feedback loop is widely used to implement reactive self-adaptation: it allows the reaction to changes in the external or internal status of the system. In general, a feedback loop starts gathering the data about the status of the system; then these data are analyzed to obtain a high-level view of the system; starting from the highlevel view, a decision phase chooses the actions that should be executed; the loop ends with the actuation of the chosen actions.

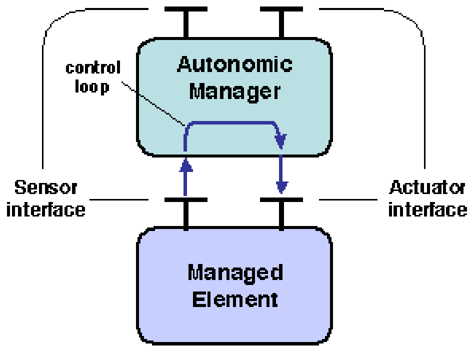
However, compared to the open loop control, closed loop structure has several characteristic features:

- *Stability.* It is a key property of a dynamic system, ensuring that the trajectory of the the system is maintained close to the desired behavior. Especially well-designed controllers are capable of improving the stability of marginally stable systems, and even stabilizing naturally unstable systems. However, the stability of the control system must be assessed;
- *Adaptability.* As control actions are continuously computed from the tracking error signals, the controller is able not only to regulate the plant output around a constant reference, but also to track moving objectives over a large range of values and dynamics, without need for on-line re-tuning;
- *Performance shaping.* As the controller amplifies and adjusts the tracking error before feeding the actuators, it is able to shape the performance of the controlled plant. For example, the closed-loop system can be tuned for a response faster than the open-loop behavior, and disturbances can be rejected in specified frequency bands;
- *Robustness*. The control actions are repeated at a rate which is fast compared with the system dynamics, hence the increments of tracking errors due to imperfect modeling are small. Feedback is able to provide precise and effective control for systems made of uncertain components.

To explain the control loop of autonomic computing systems, we will briefly explain their architecture. The autonomic computing system is organized in layers and sections. These parts are related to the so-called enterprise service bus that allow components to collaborate using standard mechanisms such as web services.

Enterprise service bus integrate various blocks, which include:

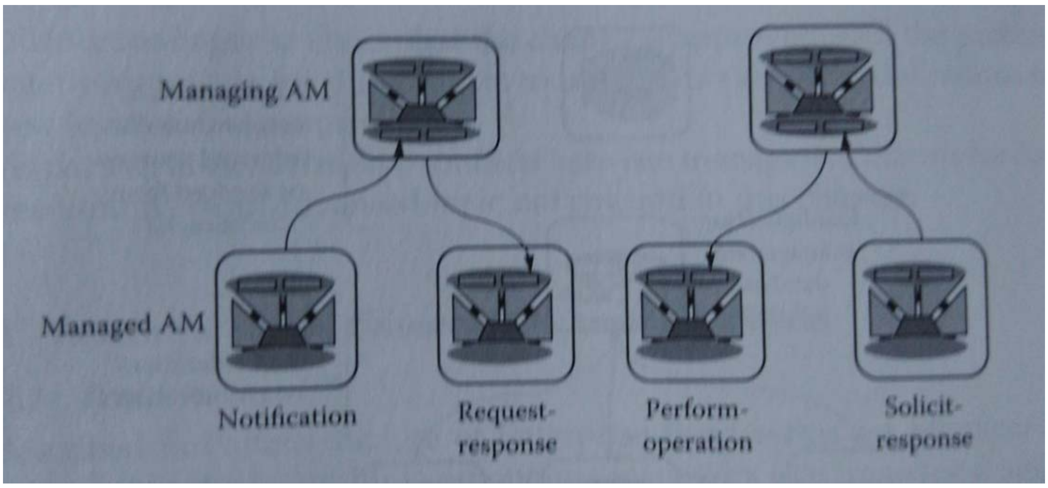
- managed resources. These managed resources can be any type of resource (hardware or software) and can have embedded self-handling properties. Intelligent control can be embedded in the configuration of the management resource itself. These embedded control nodes are one way to provide self-managing autonomic capability. The details of these embedded controls may or may not be externally visible. Control can be deeply incorporated in resources, so as not to be visible through the control interface;
- touchpoint. It is a block of the autonomic computer system that implements, or implements, the behavior of the sensor and the effector of the managed resources. It also provides a standard management interface. Access and control of management resources is managed by the management interfaces. The management interfaces use mechanisms such as files, events, commands, application programming interface (API), and configuration files. These mechanisms provide a variety of ways to collect information about the change in behavior of managed resources;
- touchpoints autonomic manager. Autonomic managers implement intelligent control that automates the combination of tasks in the IT processes. TouchPoint autonomic managers are those that work directly with management resources through their management interfaces. These self-managing managers can perform various tasks of self-management.
- *orchestrating autonomic managers*. One TouchPoint autonomic manager acts in isolation from the aspect of autonomic behavior only for the resources he is leading. However, the self-managing autonomic capabilities of Touch Point autonomic managers should be coordinated in order to allow the systemic behavior of the autonomic computer system. The organization of autonomic managers should provide this coordination function;
- knowledge sources. It is a block of a registry, a dictionary, a database, or another warehouse that provides access to knowledge in accordance with the interfaces prescribed by the architecture. In the autonomic system, knowledge consists of certain kinds of syntax and semantics data specific to this architecture, such as symptoms, policies, change of requirements, and change of plans. This knowledge can be stored in the source of knowledge so that it can be shared between autonomic managers. The knowledge found in the source of knowledge can be used to expand the skills of the autonomic manager's knowledge.;
- *a management unit of IT professionals (manual managers)*. It provides a common system for managing the interface of IT professionals using the integrated solutions console. Self-managing autonomic systems can use a common technology console to provide an interface between autonomic managers and IT infrastructure components.
- Autonomic systems are defined as a collection of autonomic elements that communicate with each other. An autonomic element consists of a single autonomic manager that controls one or many managed elements.



Picture 5. Autonomic element
Source: IBM Server Group, 2001

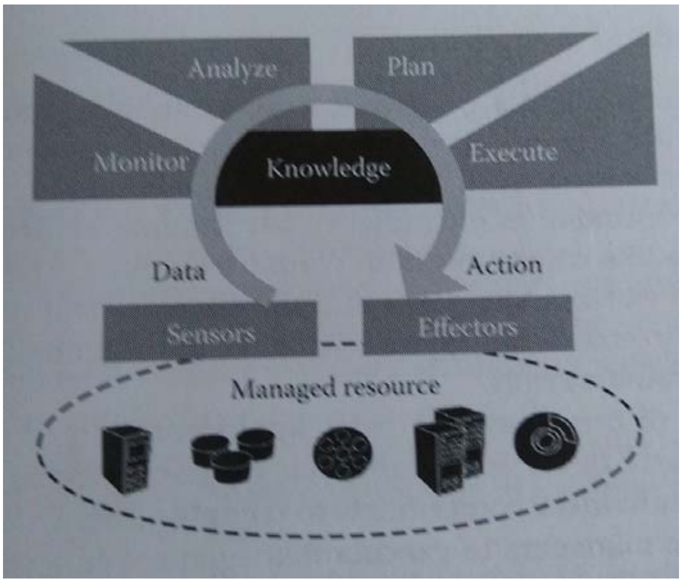
The central component in autonomic computing systems are autonomic managers. It automates certain management functions and externalizes these functions according to the behavior defined by management standards. Autonomic managers are self-managing and manage their own behavior by using policies. In addition, the autonomic manager makes use of knowledge sources to access management data, such as policies and symptom definitions, that are used to carry out its management functions. The first three blocks represent the hierarchy of autonomic managers.

This organization allows autonomic managers to focus on specific disciplines such as performance, availability, security, and others within a single domain of interest. This hierarchical autonomic manager pattern is accomplished by using the external interfaces described earlier; an autonomic manager can manage or be managed by other autonomic man agers in a manner similar to the way in which autonomic managers manage resources by using the sensor and effector of the external interfaces of the autonomic manager. This interactions are illustrated in follow figure:



Picture 6. Manager-of-manager interaction use the same sensor and effector interface
Source: M. Parashar, 2007

An autonomic manager contains an intelligent control loop that implements four functions: monitor, analyze, plan, and execute.The major approach to self-management is to use one or multiple feedback control loops. The most common implementation of the feedback loop is the MAPE-K loop (Monitor, Analyze, Plan, Execute, over a Knowledge base). (Jeffrey O Kephart and David M Chess, 2003) Namely, the traditional MAPE loop is enriched with a knowledge base that is exploited to makes the decisions in the planning phase. MOPE-K loop is presented in follow figure:



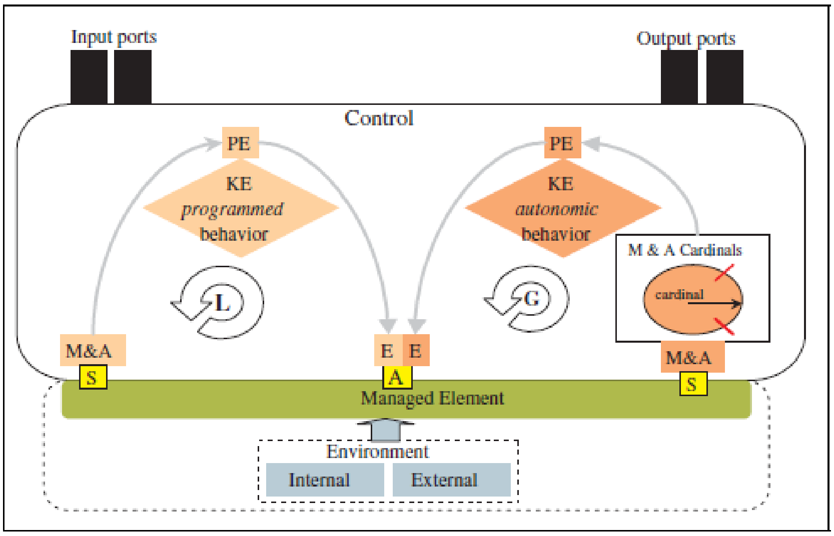
Picture 7. Autonomic control loop
Source: Adapted from IBM, 2008

As shown in the figure, the architecture of this control loop is divided into four parts that share knowledge. The monitoring function provides mechanisms that collect, aggregate, filter out information and report on the details of the managed resource. The analysis function provides mechanisms that correlate and model complex situations (for example, time series prediction and application of other models). These mechanisms enable the autonomic manager to become familiar with the IT environment and to anticipate future situations. The planning function provides mechanisms for building the activities needed to achieve the goals and objectives. The planning mechanism uses an information policy for the realization of its work. The execution function provides mechanisms that control the execution of the plan according to the dynamic update. A knowledge source provides information about the managed resources and data required to manage them, such as business and IT policies. The sensor consists of one or both of the following options:

- a set of properties that provide information on the current state of the management resources and is accessed through standard operations;
- a set of management events (unwanted, asynchronous messages or notifications) that occur when management resources are susceptible to changes for which notification is received.
- The effector consists of one or both of the following options:
- a set of operations that allow management resources to change the situation in some way;
- a collection of a set of operations carried out by autonomic managers that allow management resources to make demands from their manager.

Through this control loop, autonomic managers monitor resource details, analyze those details, plan adjustments, and execute the planned adjustments – using both information from humane, i.e. administrators as well as rules and policies both defined (by humans) and learned by the system. Each autonomic element has its own manager that (1) accepts user specified requirements (performance, fault tolerance, security, etc.), (2) interrogates the element and characterizes its state, (3) senses the state of the overall system/application, (4) determines state of the environment, and (5) uses this information to control the operation of the managed element in order to effectively achieve the specified behaviors. This control process repeats continuously throughout the lifetime of the autonomic element. As shown in follow figure, the control part consists of two control loops - the local loop and the global loop.

The local loop can only handle known environment states and is based on knowledge that is embedded in the element. The local loop is blind to the overall behavior of the entire application or system and thus can not achieve the desired global objectives. While, The global loop can handle unknown environment states and may involve machine learning, artificial intelligence and/or human intervention. It uses four cardinals for the monitoring and analysis of the managed elements. These are performance, configuration, protection and security.



Picture 8. The key parts of an autonomic element
Source: B.Jacob and all, 2004

4. CONCLUSION

Computer systems are becoming more and more adaptive as they have to respond and adapt of their surroundings in order to improve performance. On the other hand, complex systems are too large to continue to administer manually and must be automated in order to avoid mistakes or to make quicker decisions. The goal of autonomic computing is to add sufficient intelligence to systems to allow those systems to adapt to changes, to dynamically protect themselves, to automatically apply patches and fixes, and to alert a human when things go really awry. With such automation, it is possible to manage various aspects such as computer and communication resources, service quality, tolerance of errors, and so on. Automation isn't about removing control, it's about increasing efficiency and removing some of the human error introduced into operating system and software maintenance. In an autonomic environment, components work together, communicating with each other and with high-level management tools. They can manage or control themselves and each other. Components can manage themselves to some extent, but from an overall system standpoint, some decisions need to be made by higher-level components. Therefore, there is a need for control of autonomic systems that will contribute to their proper behavior. This is achieved by the so-called control loop. The control loop of the autonomic manager includes monitor, analyze, plan and execute functions.The control loop includes sensors and actuators that are related to the process or system to control it. In other words, only the sensors and effectors have direct access to the managed resource. Different control loops are used for different self-behaviours (self-healing, self-protecting, self- configuration, self-optimizing). Control loop provides stability, short settling times, and accurate regulation. Control loops provide the generic mechanism for self-adaptation. A control loop allows to adjust operations according to differences between the actual output and the desired output. Therefore, the control loop should be a challenge for further research in order to increase the performance of autonomic computing systems.

REFERENCES:

Janeska M., and all. (2017), Autonomic Computing Systems as Perspective for Integration of Business Processes, International Balkan and Near Eastern Social Sciences Congress Series-Russe/BULGARIA, April 8-9

Janeska M., and all.(2017), Performance evaluation of autonomic computing systems, SM 2017, Subotica

Sakthi Nathiarasan A. and all. (2014), Autonomic Computing: Overview, International Journal of Advancement in Engineering Technology, Management&Applied Science, Vol.1, Issue 3, August

Wladawsky-Berger, (2001) "Advancing E-business into the Future: The Grid," Kennedy Consulting Summit 2001, New York

IBM Server Group, (2001) eLiza: Building an Intelligent Infrastructure for E-business—Technology for a Self-Managing Server Environment, G520-9592-00, IBM Corporation

Autonomic Computing Concepts, IBM White Paper, October 2002, 10.02.2018 from http://www-3.ibm.com/autonomic/pdfs/AC_Concepts.pdf.

Bart Jacob, Richard Lanyon-Hogg, Devaprasad K Nadgir, (2004) A Practical Guide to the IBM Autonomic Computing Toolkit, IBM

J. Kephart and D.M. Chess.(2003), The vision of autonomic computing, in the IEEE Computer Journal, Vol. 36: 41-50

Hausi A. Müller and all. (2006), Autonomic Computing, Carnegie Mellon University,

D. Kaminsky, An Introduction to Policy for Autonomic Computing, IBM Corporation, 13.02.2018 from http://www-128.ibm.com/ developerworks/autonomic/library/ac-policy.html

M. Parashar, S. Hariri, (2007) Autonomic Computing, Concepts, Infrastructure and Applications, CRC Press

Olof Hagsand, (1995) Computer and Communication Support for Interactive Distributed Applications, 1995

Jeffrey O Kephart and David M Chess. (2003) The vision of autonomic computing. Computer, 36(1):41–50,