

A Review of the Autonomic Computing Models and Applications

Leandro Zambrano
Méndez
Facultad Ingeniería
Informática. ISPJAE, Cuba

Alejandro Rosete Suárez
Facultad Ingeniería
Informática. ISPJAE, Cuba

Humberto Díaz Pando
Facultad Ingeniería
Informática. ISPJAE, Cuba

ABSTRACT

Today more and more business processes of companies rely on their infrastructure ICT (Information Technology and Communications). Along with this business evolution and the broader economic environment, these computing infrastructures have grown exponentially in the storage capacity and ease and accessibility from anywhere in the world using multiple devices. However, this development has not been accomplished by similar advances in the operation of these infrastructures, on the contrary way; infrastructures have become increasingly complex to manage. The complexity that has achieved the systems and communications infrastructure of any company makes their management is very complicated and resource intensive, which ultimately impacting negatively the bottom line of the company and therefore its ability to develop the expected business.

Autonomic Computing is a concept that brings together many fields of computer science, in order to create computing systems that are self-managed. In this study, a preview of the state of the art on this topic shows. To this end, offers first an introduction to the motivation and concepts of autonomic computing. This study raises some benefits and features of this proposal, as well as some properties of itself. In addition present some of the main applications of autonomic computing.

General Terms

Autonomic Computing.

Keywords

Autonomic Computing, Self Management.

1. INTRODUCTION

As with the birth of the Internet, one of the most notable projects by self-management was initiated DARPA for military use in 1997 [1].

Advances in network technology and computing plus software tools have enabled an explosive growth in network applications and information services and covers done all aspects of our lives. Such applications and services are very complex, heterogeneous and dynamics. Moreover, the underlying information infrastructure globally adds a lot of independent computing. Communication resources, data stores, are themselves equally large s, heterogeneous, dynamic and complex. The combination has resulted in the implementation, development, configuration and administration of complexities breaking paradigms, behaviors, interactions and compositions. Consequently, applications, programming environments and infrastructures have information have become rapidly in fragile aspects is difficult to handle and is unsure. This has led to the investigation of an

alternative paradigm for the system and application design, which is based on strategies used by biological systems to deal with challenges similar scale, complexity, heterogeneity and uncertainty, a view that has been referred to as autonomic computing [2].

The autonomic computing paradigm [1] has been inspired by the autonomy of the human nervous system. Its main objective is to detect computer systems and applications that can be handled themselves in accordance with high level guidance from humans. The fulfillment of the great challenges of autonomic computing requires scientific and technological advances in a wide variety of fields, as well as a new programming paradigm, and that the software and systems architectures give a support effective integration of the constituent technologies [2].

This paper presents a brief introduction to autonomic computing and current state of the art is presented. First overviews of the architecture of the nervous system are displayed and how it is used to motivate the paradigm of autonomic computing. The following summarizes the main challenges of autonomic computing and an overview is presented autonomic computing systems and existing applications.

2. CONCEPTUAL ISSUES

2.1 The human nervous system

The human nervous system is probably the most complex system known and organized. Autonomy is an example of behavior or existing in nature. He is the captain of the body, the controller monitors changes inside and outside the organism, integrates sensory stimuli and the effects of the appropriate response. In conjunction with the endocrine system, the nervous system is able to regulate and maintain it constantly as a principle of regulation [2].

2.2 The autonomic computing paradigm

An autonomic computing paradigm, following the model of the autonomic nervous system, must have a mechanism by which changes in essential variables can trigger changes in the behavior of the computer system so that the system is restored back into balance with regard to the environment. This state of stable equilibrium is a necessary condition for the survival of the organism. For a system of autonomic computing, you cannot think of survival as the system's ability to protect, recover from failures, reconfigured as required by changes in the environment, and always keep your s functions almost optimum performance. Your balance is affected by both and the internal environment (for example, memory usage / excessive CPU) as the external environment (for example, protection against external attack) [2].

An autonomic computing system requires:

- Channels sensor to detect the change in the internal and external environment.
- Channels engine to react and counter the effects of changes in the environment to maintain balance.

2.3 Self-Management

The essence of autonomic computing systems is self-management, purpose is the free system administrators, from the details of a system operations, maintenance, and to provide users with an operating at maximum infrastructure 1 of the 24-hour yield by 7 days of the week.

Like their biological counterparts, the autonomic systems or sAngels maintain and adjust their operation to the evolution of the component is, workloads, demands and to external hardware or software failures whether accidental conditions or malicious.

An autonomous system can continuously monitor their own use, and check the actualizations component, for example. If you think that improvements are worth it, the system will be installed, reconfiguring as needed, and run a regression test to make sure all is well. When it detects errors or modifications, the system returns to the previous version, while it's automatic algorithms for determining problems trying to isolate the source of the error [3]. Frequently cited IBM four aspects of the self, which in table 1 are summarized.

Table 1. Four aspects of self-management as they are now and would be with autonomic computing [3].

Concept	Current Computing	Autonomic Computing
Self-configuration	Corporate data centers have multiple vendors and platforms. Installation, configuration and integration of system is time-	Automated configuration of components and systems follows high-level policies. Rest of system adjusts automatically and

	consuming and more error prone.	seamlessly.
Self-optimization	Systems have hundreds of manually set, nonlinear tuning parameters and their number increases with each release.	Components and systems continually seek opportunities to improve their own performance and efficiency.
Self-healing	Problem determination in large, complex systems can take a team of programmer's weeks.	System automatically detects, diagnoses, and repairs localized software and hardware problems.
Self-protection	The detection and recovery from attacks and cascading failures is performed and resolved manually.	The system automatically defends attacks or cascading failures. Use early warning to anticipate and prevent system failures.

2.4 Architecture of an autonomous element

An autonomous element (see Figure 1) is the smallest of an application or system autonomic. This is a system of independent software modules or input / output specified interfaces and explicit context dependencies. Mechanisms for self-management, which are responsible for the execution of its functions, export limitations, management of behavior, according to the context and policies, and interaction with other elements are also incorporated. The applications systems are constructed from autonomic and dynamic elements such as dynamic compositions, opportunistic and or ephemerals [2]. These compositions are to be defined to by the policies and context, and can be traded. The key parts of a separate element as described below [2].

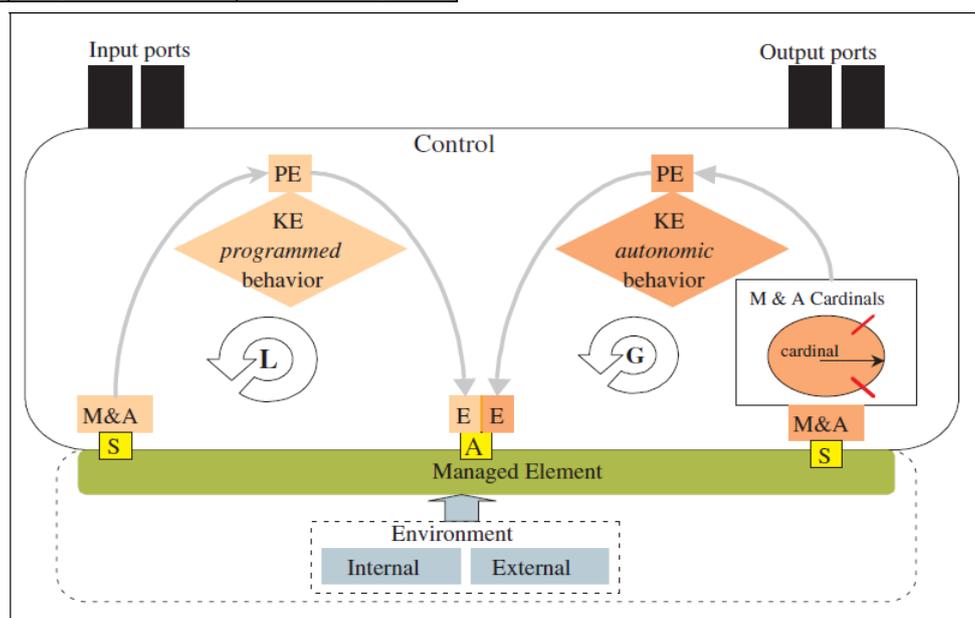


Fig 1:Autonomous element taken from [2].

Managed Element: This is the functional unit small application and contains the executable code (programs, data structures). It also exports its functional interfaces, and behavioral treatment, behavioral attributes and limitations. At runtime, the managed element can be affected in different ways, for example, may encounter a failure, run out of resources, be attacked externally, or may fall into a bottleneck for the impacts of poor performance [2].

Environment: The environment represents all factors that can affect the managed element. The environment and the managed element can be viewed as two subsystems that form a stable system. Any change in the environment makes the whole system and pass a stable state to an unstable state. This change is compensated below by reactive changes in the managed element cause that the system is moving backwards from unstable state to a different stable state. Is observe or that the environment consists of two parts, internal and external. The internal environment consists of changes internal to the managed element, which reflects the status of the application / system. The external environment reflects the state of the runtime environment [2].

Control: Each element has its own autonomous manager about:

- 1 -The user agrees specified requirements (performance, fault tolerance, security, etc.).
- 2 - Interrogate element and characterize their status.
- 3 - Detect the overall system state / application.
- 4 - Define state the environment.
- 5 -Using information is to control the operation of the managed element in order to effectively achieve specified behaviors.

This control process is repeated continuously throughout the lifetime of the autonomous element. As show in Figure 1, the control part knows two control loops of the local loop and the overall circuit.

The **local loop** can only handle states known environment and is based on knowledge is n d within the element. His knowledge engine contains the mapping state of the environment and behavior. For example, when the load on the local system exceeds the threshold value, the local control loop to balance work either by loading the control of local resources for their availability & managed element or by reducing the magnitude of the problem handled by this element. This will only work if local resources can handle the computational requirements. However, the local loop is blind to the overall performance of the entire application or system and therefore cannot achieve the desired global objectives. In nary scenes in which the entire system is to affected, the local loop continues repeating the local optimization may lead to the degradation in performance and the result the suboptimal or chaotic behavior.

At some point, one of the essential variables of the system can exceed its limits. In this case the global loop at work in action.

The **global loop** can handle unknown states and ambient signals may involve machine learning, artificial intelligence and / or human intervention. Use four points for monitoring and analysis of managed elements. These are performance, configuration, protection and security [2].

As shown, the paradigm of autonomic computing is closely related to the paradigm of agent-oriented programming, which

arises from five trend of computing, ubiquitous computing devices, the increasing interconnection between them, the systems on increasingly complex and intelligent time, these systems will increasingly delegating tasks, and there is a tendency to see systems using higher levels of abstraction [7]. Autonomic computing is clearly framed in these trends.

3. RESEARCH ISSUES AND CHALLENGES

The fulfillment of the great challenges of autonomic computing presents significantly critical mind and research challenges that span all levels, from the conceptual level of architecture, middleware and applications. Below the main research topics and challenges are presented [2].

Conceptual challenges: The research is conceptual issues and challenges include [2]:

- 1 -Definition of suitable models and abstractions for specifying, understanding, control and implementation autonomous behaviors.
- 2 - Adaptation of the model years and the classical theories of machine learning, optimization and control systems for multi agents dynamic.
- 3 - Providing effective trading models for autonomic elements that can be used to establish multilateral relations with each other.
- 4 - Statistical design models large networked systems that allow autonomous systems elements or detect or predict general problems of a group of sensor data from individual devices.

In [3] the authors presented in a clear definition of autonomic computing and a state of the art of this issue by citing what some authors have on autonomic computing and determining key aspects of both research and academic enterprise level. A significant contribution and the original contribution as it handles the challenges of autonomic computing at the level of engineering and scientific level ago.

In [8] is expressed addition to the basic concepts of autonomic computing categorization into two research groups: those that describe the technologies related to autonomic computing and autonomic computing those who develop as a unified project. Performs conceptual comparisons between authors which is useful in the realization of the state of the art and future challenge exposed as autonomic computing implementing a self-healing system in a virtual organization in which some parties may fail.

In [10] a rich conceptual framework based on systems theory and artificial life for autonomic computing topic arises. Defines three approaches that research has been done on the subject: Multiagent Systems, Dynamical Systems and decentralized control. In the conceptual modeling group's basic elements of the systems, as their tendency to be analyzed over time, the condition of the autonomic elements to be dynamic systems which can be defined through attractors if a long period of time is regarded. Handles concepts like entropy and controllability of decentralized feedback [10]. It also provides key concepts to be considered in the communication between agents as given by the same environment, through agents that are placed in the environment (Beacon).

In [11] the author proposes a model relevant aspect of self-protection techniques are employed to control the integrity of

software components running in insecure environments. States open problems that could be conceptualized from the perspective of autonomic computing.

Architecture Challenges: The applications and autonomic systems to be built from autonomous elements that manage their internal behavior and their relationships with autonomous elements in accordance with the policies that humans or other elements established.

In [9] the authors present and meet the challenges of autonomic computing research from two perspectives: scientific challenges and challenges for engineering. In the part of self-recovery shows the challenge of troubleshooting, *micro rebooting* process and approach that closely resembles formulated.

Application Challenges: The key challenges application-level focus on the design and development of systems and applications that are able to be managed (configured, adapted, optimize, protect and oneself) themselves. This includes programming models, frameworks and services middleware supporting the definition of autonomous elements [2].

3.1 SCEL: Language autonomic computing

The autonomic computing paradigm is proposed to cope with the magnitude, complexity and dynamism of software and infrastructure contemporary technology. The challenge for designers is to design appropriate language and primitive linguistic abstractions to deal with the large size of the systems, and their need to adapt to changes in the work environment and the evolving requirements. A set of programming abstractions that can represent behaviors, knowledge and aggregations according to the specific policies and programming to support context-awareness, self-awareness and adaptation is proposed. Based on these abstractions, SCEL (Ensemble Software Component Language) [4] a kernel language whose foundations are based on formal reasoning and behavior of autonomic systems is defined.

One possible answer to the problems posed by such complex systems is to make them able to manage themselves, continually monitoring their behavior and working environment by selecting actions to be taken for the proper handling of the current state of things. Self-management could also be harnessed to address situations in which the intervention of humans is limited or nonexistent and components that work together to achieve specific goals. This requires systems to build capacity for self-management and ensure what is now known as self-property (self-configuration, self-healing, self-optimization, self-protection) of autonomic computing [3, 5].

4. THE PANORAMA OF AUTONOMIC COMPUTING

There has been a number of research, both in academia and in the industry versus autonomic computing concepts and research of the problems described above. There are projects and products that can be classified as:

- 1 - Systems incorporating autonomic mechanisms for problem determination, tracking, analysis, management, etc, on the technological infrastructure.
- 2 - Systems investigating models and programming paradigms as development environments that support the development of

systems and autonomic applications. A sample of systems belonging to these categories summarized in tables 2 and table 3, respectively.

Table 2. Systems that incorporate autonomic properties.

System	Field of Application	Autonomous Topics Addressed
OceanStore [2]	Comprehensive, coherent, high availability and storage of persistent data.	The self-healing, self-optimization, self-configuration, self-protection. Policy based at caching, the autonomous replication, the continuous monitoring, testing and repair.
Storage Tank [2]	Multi-platform storage management universal access.	Self-optimization, self-healing. Policy storage and data management based redirection servers.
Oceano [2]	Cost effective for scalable management of computing resources for business software.	Self-optimization, self-awareness. Demands autonomic distribution, constant monitoring of components.
Smart DB2 [2]	Reduction human intervention and cost DB2.	Self-optimization index determining autonomous self-configuration, disaster recovery, continuous monitoring of health DB2 and alert DBA.
AutoAdmin [2]	Reduction TCO.	Self-administration. Usage tracking, Index Tuning and recommendation based workload.
Sabio [2]	Rank autonomic many documents.	Self-organization, self-consciousness. Documentary according to the word and utilizing the sentence.
Q-Fabric [2]	Continuous Support System of online management.	The self-organization. Ongoing management quality on-line through the "personalization" of QoS of each application.

Table 3. Support systems application development and autonomic systems.

System	Approach	Autonomic Question
KX (Kinesthetics eXtreme) [2]	Retrofitting automated.	Enabling properties autonomic in legacy systems.
Anthill [2]	P2P system based ant colonies.	Adaptive behavior and complex systems P2P.
Astrolabe [2]	Distributed management of information.	Self-configuration, monitoring and control adaptation.
Gryphon [2]	Publish / subscribe middleware.	Great communication.
Smart Grid [2]	Beginning Autonomic applied to solve problems the network.	Autonomic computing network.
Autonomy [2]	Model and infrastructure for enable applications autonomous.	Autonomic Applications.
AutoMate [2]	Environment execution on applications autonomous.	Autonomic Applications.

5. CONCLUSIONS

One can argue that the paradigm of autonomic computing is inspired by biological systems, using as guidance, for example, the nervous system of the humans. It is assumed that the autonomic computing paradigm could become a good approach to solve problems current technological infrastructure incorporating self attributes.

Autonomic computing suggests the possibility of implementing strategies and algorithms that introduce attributes of autonomy to manage the complexities and uncertainties with minimal human intervention. At present there are many research ones being carried out on this topic, and approaches that could be systematized under this approach are observed as proposed by [11], but this remains an open and meaningful challenge. Finally one of the current challenges of autonomic computing is how to measure and evaluate an autonomic system is carrying out its tasks properly, or how to measure its ability to meet a certain level agreement service.

6. REFERENCES

- [1] MARKUS C. HUEBSCHER, JULIE A. MCCANN. Imperial College London. A survey of Autonomic Computing degrees, models and applications. ACM Computing Surveys, Vol. 40, No. 3, Article 7, Publication Date August 2008.
- [2] M. Parashar, and S. Hariri. "Autonomic Computing: An Overview", UPP 2004, Mont Saint-Michel, France, Editors: J.-P. Banâtre et al. LNCS, Springer Verlag, Vol. 3566, pp. 247 – 259, 2005.
- [3] Jeffrey O. Kephart, David M. Chess. IBM Thomas J. Watson Research Center. The Vision of Autonomic Computing. IEEE Computer Society, 2003, pp. 41-50.
- [4] Rocco De Nicola, Michele Loreti, Rosario Pugliese, and Francesco Tiezzi. SCEL: a Language for Autonomic Computing. IMT, Institute for Advanced Studies Lucca, Italy. Università degli Studi di Firenze, Italy. August 7, 2013.
- [5] IBM: An architectural blueprint for autonomic computing. Technical report (June 2005) Third edition.
- [6] Arles Ernesto Rodríguez Portela. Modelo autónomo con capacidades de auto-recuperación: Bibliografía Anotada. UNIVERSIDAD NACIONAL DE COLOMBIA.
- [7] M. Wooldridge: An introduction to multiagent systems, John Wiley and Sons, 2002.
- [8] Nami, M.R. and Bertels, K., A Survey of Autonomic Computing Systems. Third International Conference on Autonomic and Autonomous Systems. ICAS07, 2007, p. 26.
- [9] Kephart, J.O., Research challenges of autonomic computing. Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005, pp. 15-22.
- [10] De Wolf, T. and Holvoet, T., Towards autonomic computing: agent-based modelling, dynamical systems analysis, and decentralised control. Proceedings. IEEE International Conference on Industrial Informatics, 2003. INDIN 2003, pp. 470-479.
- [11] Nuñez Musa, Yulier. Modelo no determinista para la auto-verificación de integridad de Componentes de software. Tesis Doctoral. Departamento de Tecnología Informática y Computación Escuela Politécnica Superior, Enero 2013.