Adaptivity: One Phenomenon, multiple perspectives of study

Rosalia Caya and João José Neto

Abstract—**This article studies adaptivity as a general phenomenon, a subject of matter, permeable to different perspective within the research areas. From simple and general definitions grounded in Biology and Psychology to sophisticated and particular definitions within Computer technology areas, such as Computer Science, Information Technology, and Software Engineering, this work focus in retrieving the core features that characterize adaptivity, its needs, the different connected areas and the characteristic problems it is suited to solve. We present a proposal for the holistic characterization of adaptive behavior that contributes in the establishment of a framework to share scientific knowledge regarding specification, representation, modeling, design and implementation of adaptive behavior in the technological arena.**

Index Terms—**Adaptivity, self-* systems, complexity, cybernetics, autonomic computing.**

I. Introduction

THE idea of an entity capable of modifying its own behavior according to characteristics of its environment and its own particular goals it's not a novelty, as a matter of fact it has been around from ancient times. From Daedalus, the most ingenious inventor of Greek myth, credited with making the first "living statues" that appeared to be endowed with life because of its capability to make human-like movements, wept and even vocalized [1]. Automata, from Greek perspective self-operating entities made to obey particular goals[2], were also engineered by Hephaestus, the Greek god of invention and technology. Talos, the gigantic animated bronze warrior programmed to guard the island of Crete, was one of Hephaestus's creations. Many variations of the myth exists and the analysis of the story of Talos has risen several books and studies. However, most of them agree on some characteristics of Talos: autonomous entity, with an specific task given by a superior authority that needs to be aware of the circumstances in its surroundings to be able to fulfill its goal, through the acquisition of hidden knowledge or underlying truth [3]. This seem to presage today's scientific "cybernetic organism"[1].

Automatic machines were also created by Italian inventor Leonardo da Vinci. Leonardo's robot (or Leonardo's mechanical knight) was a humanoid automaton designed and possibly constructed by Leonardo da Vinci around the year 1495. The robot knight is capable of performing several human-like motions: he could stand, sit, raise its visor and independently maneuver its arms. t is partially a result of Leonardo's anatomical research in the Canon of Proportions as described in the Vitruvian Man focusing in the complexity of the movements in human body.

A new attitude towards automata is to be found in Descartes when he suggested that the bodies of animals are nothing more than complex machines. France in the 17th century was the birthplace of those ingenious mechanical toys that were to become prototypes for the engines of the Industrial Revolution following a reductionist approach. This way of thinking culminated in unprecedented economic growth and development, and machines entered in everyone's lives. However, techniques continue to evolve along with technology and computing power, so, new models can be developed that better reflect the real world and its complexity [4]. Most real-world problems deal with complexity, uncertainty and optimization of some type where information must be exploited as acquired so that performance maintains or improves apace. This characteristics compound the basic definition for adaptive behavior we will present in the next section, and permeates problems in several areas of knowledge as diverse as ecology, psychology, economy, artificial intelligence, computational mathematics, sociology, and others. This way, over the years, some fields within the technological arena have dedicated special efforts to study adaptive behavior and developed approaches to deal with it in its particular domain.

However, in the last decades the study of adaptivity is gaining attention as a multi-disciplinary concern due to the rising demand for intelligent and more realistic applications. Applications mimicking human behavior, considering continually changing conditions, critical systems or high-definition simulation of real life situations have introduced back into technology the complexity cut off by the reductionist approach at the beginning of computer's era. This opens the door for many potential applications that require real-time perception and reaction. In fact the rising of new applications empowered by technologies, such as Internet of Things, Ubiquitous Computing, Multi-agent Systems, Evolving Systems, Cyber-Physical Systems, autonomic computing and others demand the support of some sort to process adaptive behavior while aiming a particular goal.

As consequence a variety of spaces to develop research related to adaptivity have been created. Some institutions and research groups specifically address adaptive behavior, complexity and dynamic change. Well established venues on technology have incorporated issues on adaptivity within their main tracks as academic work and research projects related to the topic continue to rise, as detailed in [5]. In fact, to deal with this growth dedicated venues from different natures:

R. Caya was with the Department of Electrical and Computer Engineering, Polytechnic School, University of São Paulo, São Paulo, SP - 05508-010, Brazil, e-mail: rosalia.caya@usp.br.

Manuscript received September 29, 2017.

academic (workshops, conferences, and journals) and mainstream (magazines, blogs and trade journals) have been created to cover topics on adaptivity. The same can be observed within the industry where inclusion of assisted technology and human-centered approaches welcome variety, complexity and uncertainty in every-day technology.

However, despite the growth and advances shown by research, this scenario and promising tendency reveal one major challenge: the need for a unified approach for adaptivity. The aforementioned approaches taken by specific fields present ad-hoc solutions to deal with adaptivity within their domains. On one side, this situation allows that different areas respond with autonomy to the challenges as they appear, creating new knowledge process and understandings. On the other side, the same situation is an ideal environment for creation of domain specific terminology, methods, models and resources, that sometimes remain invisible or even conflict between each other. We believe that by facing that challenge researchers collaborate to highlight the multi-disciplinary nature of adaptive behavior and empower the growth of a unified field.

To address this challenge, in this article we propose an study on literature of four different fields to create an holistic framework matching theoretical and practical approaches to define, model and create technology considering adaptive behavior.

Paper Organization The rest of the paper is organized as follows: first, in Section II, we analyze the historical scientific interest in the study of change processing within technological fields. In Section III we look at the approaches this phenomenon has taken within different technological areas aiming to solve problems related to their particular subject matter. We took four main fields within technological arena related to adaptivity that have developed solid approaches for dealing with adaptive behavior: self-* systems, cybernetics, complex systems, and autonomic computing. In Section IV, we propose an holistic vision of adaptivity, the factors that allow such approach, the common ground between the theory in the analyzed fields, and the contribution and advantages it will bring. Later, in Section V, we took a revision of related works on unified initiatives for adaptive behavior and highlight the difference they present with this work. Finally, in Section VI, we elaborate the conclusions about this work and present the future directions to develop an holistic vision of adaptivity.

II. Adaptivity: the phenomenon

The evolution and grown of technology have developed systems more complex and heterogeneous. At the same time, the interaction with such systems, made that the demand from users for mechanisms that allow personalization, reconfiguration, flexibility and autonomy passed from preference to necessity. When consulting literature about computer science and technology one can find several studies that highlight a novel ability from the systems to adjust themselves to events in their surrounding. Systems with this ability can be called *"self-adaptive systems"* within software engineering, or *"dynamically adaptive systems"* for the dynamic change community, or as part of *"autonomous"* or *"evolving"* systems. As matter of fact, there have been several keywords, over the years and across disciplines, that aim to describe some aspects of interest of a wider phenomenon: adaptive behavior.

A. The intuitive definition

In the most basic approach, we define adaptivity in the following intuitive terms:

Adaptivity is the ability of an *entity*, at any moment, to *decide* the modification, by *executing* a set of proper actions, of its own features, structure and/or behavior, or even its environment, when facing new coming events *perceived* in its surroundings while pursuing a particular *goal* to *suit more efficiently* the new *context* of its functioning.

The term *entity* can take a wide range of meanings for example: individual, system, structure, agent, being, and so on, encompassing different magnitudes from a single molecule to an interacting group of organisms. The concrete meaning is largely determined by the field of study. The *decision* to effectively change any of its components comes uniquely from its own analysis about the benefits gained in doing so. The direct *execution* of the proper set of actions matching the situation can have indirect repercussions in other available actions that define the behavior of the entity. The analysis mechanism is performed due to stimulus *perceived*, or sensed, in the current situation. The high-level *goal* the entity aims generally is set by an external high-level authority: a leader, a manager, a need or even evolution. The motivation in performing adaptations is a better suiting of the entity and its goal into the new situation, this have a variety consequences, such as: incorporation/drop of new features to take advantage of opportunities, avoidance of threats, or preparation for facing danger. By being adaptive the entity tries to respond to changes in the *context* in which it performs at the moment. The context is define by its internal features (behavior and structure) the characteristics of the external environment (resources, events, objects within) and the channels of interaction between them.

III. Adaptivity: the multiple perspectives

Adaptive behavior is an ability that has been studied within technological and computational arena for a long time from different perspectives. Most of this perspectives correspond with a particular domain, and were developed taking into account the focus of the field, the terminology and the concepts proper of that field. However, as we will see, when analyzing adaptivity they present profound similarities. We will present four main fields in technology that study adaptive behavior: Self-* systems, Cybernetics, Autonomic Computing and Complex Systems. For each of them we will give a concise definition of the field, it main focus, the general characterization of its architecture and it relationship with adaptivity.

A. Self-* Systems

A self-* system is a computer system that maintains at least one aspect of its operation automatically to relieve some of the burden that complex systems put over human administrators. The aspects it automatize are known as self-* properties, they are a set of features that characterize the behavior of some complex system [6]. The *self-* prefix highlight the autonomous nature of this property, meaning that the system has the power to decide, perform and control over this feature on its own.

These properties, even when particularly applied to describe technological entities, were inspired and observed first in natural organisms [7] whom use them efficiently to achieve particular goals or overcome difficulties. Thereby, the self-* properties are related to adaptive behavior by describing the requirements and consequences of applying adaptivity within the systems. Figure 1 shows a hierarchical categorization of some of the most referenced self-* properties over the years. From these self-* properties two of them differentiate in nature from the others: self-management and self-adaptivity. Self-management is a vision, it can be thought as the parent or superclass of all self-properties. It describes a system that has at least one self-* property [8], [9]. Self-adaptivity or self-adaptiveness is the ability of an entity to evaluate its own behavior and change it when the evaluation indicates that its not accomplishing what the software is intended to do [10]. So self-adaptivity it can be understood as ability to process dynamic change in an autonomous and intelligent way, adjusting some characteristics in its behavior according to the environment in which the system performs. Therefore, we can say self-adaptivity is subsumed by other, more narrow self-* properties [8].

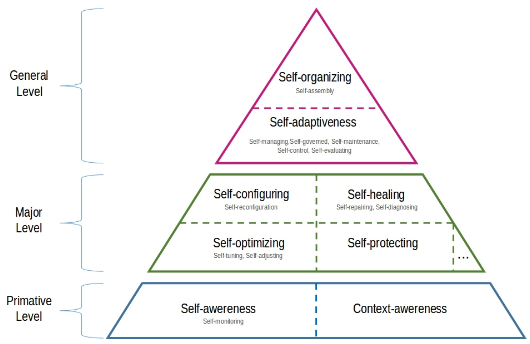


Fig. 1. Hierarchy of Self-* properties. Adapted from [11], [12], [9], [13]

Properties at the primitive level of this hierarchy, self-awareness and context awareness, are necessary condition for any system to process some type of change. Properties at the major level embedded more complex functions, hence, they are provided with more advanced mechanism of evaluation and control.

The four properties highlighted are the more referenced of the major level in the literature of different fields related to adaptive behavior. They are explicitly named as fundamental for the Autonomic Computing of IBM [14]. However, since the 2001 launch of Autonomic Computing by IBM, the set of self-* properties has grown substantially[15] and other self-* properties have been defined in the last years for computer systems[16]. In [6] and [9] the authors present a detailed, however not closed or definitive, list of self-* properties together with example of some types of systems implementing them.

Properties at the general level are those of higher complexity present in a system that processes changes and they are usually performed and observed by more sophisticated entities that control functionalities in different parts of the system. It is important to point out that in literature the term self-adaptivity has several terms closely related, particularly, self-managing and self-governing are used interchangeable. The attention, study and analysis in self-* properties, particularly self-adaptive, correspond to the interest of several areas of knowledge in dealing with problems with complexity and difficult to manage with traditional approaches [17], [18].

Characterization of Self- Systems:* As Self-* Systems permeates a wide set of properties, and their particular behaviors, the concrete architecture of a self-* system may vary from one to another depending of which ones are implemented. Moreover, a general approach for self-* systems can only point out the basic elements, conditions and mechanism, looking more as a guideline than an architectural model. Today self-* systems draws upon several well developed technologies that may be used as building blocks[19]. The idea of programs that reason about their own behavior and are able to manipulate their own semantic representation at runtime is not new, in fact it's very well founded and developed. Reflection and exception handling mechanisms are techniques that can be found in the majority of Programming Languages and support the basic traits of adaptivity. Reflection has been around long enough for efficient implementation methodologies to be developed. Reflection provides the tools for writing such programs but it doesn't provide any guidance in how it should be done. To help addressing this challenge there are many other contributing technologies , such as: model-based computing, theorem provers, models for reasoning about uncertainty, and agent based systems to name a few. [19].

From design perspective, as pointed out in [20] three metaphors have been useful to early researchers on self-adaptive software: coding an application as a dynamic planning system, or coding an application as a control system, or coding a self-aware system. In each case, self-* systems features are mapped into the paradigm's structure (planning system, control theory, or self-aware) aiming that some valuable insights and techniques can be borrowed to self-adaptive systems.

The work in [21], [22] sum up some of the effort performed by Software Engineering community to analyze models and patterns that can be applied to develop self-adaptive systems.

B. Cybernetics

Cybernetics is, in general terms, defined as the science that studies the abstract principles of organization in complex systems, focusing in how systems use information, models, and control actions to steer towards and maintain their goals, while overcoming difficulties [23]. By being inherently interdisciplinary, cybernetic reasoning can be applied to understand systems of any kind and it has influenced many fields included computer science, robotics, management, sociology, political science, economics, psychology and philosophy.

The field of cybernetics was created after WWII by a group of intellectuals interested in studying ”circular causal and feedback mechanisms in biological and social systems” to develop a general theory of organizational and control relations in a system [23].The first use of the term in English was at 1948 by mathematician Norbert Wiener in its seminar book *”Cybernetics: Control and communication in the animal and the machine”* [24]. There, Wiener states that Cybernetics is the study of **communication and control** in animals and machines, communication being the receiving and digesting of information, and control the use of this information in a direct action. Wiener’s approach is in fact mechanistic, correspondent to the kind of machines used back on the days, and its known as Cybernetics of the 1st order or first-order Cybernetics.
According to Beer in [25]: ”cybernetics studies the flow of information round a system, and the way in which this information is used by the system as a means of controlling itself: it does this for animate and inanimate systems indifferently”. Later, organization theorists regard Cybernetics as a science of information processing, decision-making, learning, adaptation, and organization, whether this occurs in individuals, groups, organizations, nations, or machines [26].

After the Control Engineering and Computer Science disciplines become fully independent, some remaining cyberneticians felt the need to differentiate from the mechanistic approach by emphasizing autonomy, self-organization, cognition, and the role of the observer in constructing the model of a system. In this approach the system being study is interpreted as an agent in its own right, interacting with another agent, the observer. Hence, the observer too is a cybernetic system, trying to construct a model of another cybernetic system. To understand this process, we need a ”cybernetics of cybernetics”, i.e. a ”meta” or ”second-order” cybernetics. Therefore, 2nd. Order cybernetics, tries to understand adaptive autonomy and, further, shifting adaptability.

Nowadays, the second order perspective is firmly ingrained in the foundations of cybernetics overall. It is undeniable that many of the core ideas of cybernetics were assimilated by other disciplines, or inspired the development of new contemporary fields, and hence, continue influencing scientific developments. More generally, the philosophy of cybernetics is starting to permeate popular culture.

1) Cybernetics Model: According to [27] the operation of cybernetic systems can be characterized by a cycle with five stages, as shown in Figure2:

1) goal activation, this stage is about discovering goals, intentions and expectations that can be achieved, in the current state of the system. The triggering of this stage is the perception of an event (any kind of stimuli, disturbance, perturbation) from the environment through the system’s sensory mechanism. Selecting the features of a system to pay attention to is inherit in the perceptual process;
2) action selection, this stage enacts the decision making mechanism of the system. Based on the goal it pursues this stage select the suiting plan or strategical actions, depending of the repertoire available, for achieving it;
3) action, this stage is responsible for the execution of the strategy, the implementation of the necessary context and the detailed directions for the actual behavior of the system in the environment;
4) outcome interpretation, at this stage the system retrieves the data and facts from the environment to interpret the consequences of the behavior performed before. This information is passed to the system as feedback using memory updating and storing. Some of the criteria used to elaborate the feedback are: effects on the environment, collateral effects in the system, and nature of the outcomes (long term vs short term, abstract vs concrete, approach vs avoidance);
5) goal comparison, this stage is about examination, deliberation and revision of the information from the feedback to reveal the achieving of the selected goal. If the goal has not being achieved yet, then some actions may be done to approach its completion. This actions can be towards to take advantage of an opportunity, or for avoiding a threat. Finally, some actions promoting changes in the system may be requested. This can be sum up as analyzing the mutual influence between the system and the environment in which it performs.

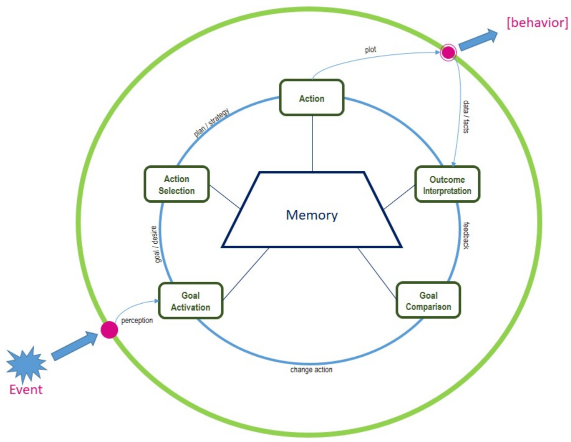


Fig. 2. Cycle in Cybernetic systems

These elements can be divided into two basic categories:

- First,there is a collection of mechanisms that evolved to carry out the different processes associated with each stage of the cycle. The mechanisms called into play by encounter with uncertainty are of two fundamental types, reflecting the unique status of the unknown as the only class of stimuli that is simultaneously innately threatening and innately promising: Stability and plasticity. The first of these needs is to maintain the stability of ongoing goal-directed functioning. The second is the need to engage in exploration that integrates novel or anomalous information with existing knowledge. Stability reflects variation in the control mechanisms that prevent the cybernetic system from being disrupted by emotional impulses.Stability reflects the capacity of the cybernetic system to resist disruption. The term ”plasticity” is probably most often encountered in neurobiology, but here we understand plasticity as the general tendency toward exploration, with exploration defined as the creation of new goals, interpretations, and strategies [27]. All exploration involves learning).Plasticity reflects the degree to which the cybernetic system is prone to generating new goals, interpretations, and strategies, not only when required by stressors that have caused instability and disintegration, but also voluntarily, in response to the incentive reward value of the unknown;

- Second, stored in memory is a collection of goals, actions, and knowledge about the world (strategies, standards, behavioral repertoire, and patterns that exist in the world). Most of these are learned through experience rather than innately preprogrammed. These learned, updateable memory contents of the cybernetic system are deployed by the mechanisms described in the first category. Goals, interpretations, and strategies represent the information used by the cybernetic system to function in any situation, and they always reflect the manner in which the individual has adapted to that situation, even if they are one-off, never repeated. This means that not all adaptations are characteristic. To be considered ”characteristic,” the adaptation must have enough stability to be a useful descriptor of the person for some reasonable length of time.

2) Cybernetics and Adaptive behavior: Following the approach given by Wiener in [24] cybernetics can be also understood as the study of goal-directed self-adaptive systems [27], [28], [29]. By being a general theory of the regulation of systems it can be use as a framework to study management and organizations, encompassing adaption, self-organization and reflexivity. Cybernetics, as mentioned before, had a crucial influence on the birth and developing of various modern science [23]. Particularly, complexity, self-organization, self-reproduction, autonomy, networks, connectionism and adaptation were concepts first explored by cyberneticians during 1940’s and 1950’s [23].

All cybernetic systems receive feedback, through some kind of sensory mechanism, indicating the degree to which they are moving toward their goals and use this information to adapt and adjust their behavior, to pursue their goals. As explained in [26], Adaptivity is related to improvement, better suiting the conditions of an environment or enhance the performance of an entity. Ashby’s theory of adaptation explains the success of process improvements methods based on the distinction between working IN a process and working ON a process. Work IN a process refers to the work done to make the process function. Work ON a process is the ”discussion” about how to improve the process. Ashby show that any system having two nested feedback loops, one inside the other, would be able to display adaptive behavior. The inner loop operates frequently and makes small adjustments, the outer feedback loop operates infrequently and initiates the learning of a new pattern of behavior. Adaptation encompasses learning.

C. *Complex Systems*

In both Engineering and Science the term *complexity*, does not have a sharp definition and the demarcation with the notion of complicated systems is a challenge [30], [31], [32]. However, this does not forestall a rigorous approach to the subject matter.

A system is considered complex if it have many components that collaborate to create a functioning whole. The function of such system is governed by the dynamical interactions of the components (within the system and with the environment) and cannot be fully understood by the description and analysis of its parts in an individual manner, as with the reductionist approach [33]. Complex systems are characterized, and distinguished from *complicated* systems, by two factors: (i) it exhibits unexpected behavior, often referred as emergence of properties, as consequence of non-linear interactions between its parts, framed by the hierarchical structure that build up the system, and (2) the uncertainty in predicting the behavior of the system, named unpredictability, due to continuous change in function and structure [30], [33]. The description of such a system can be done in different levels and from different perspectives, which means that **complexity** is subjective and it describes the stance that is being taken towards a system [31], hence the definition becomes unclear and arbitrary.

Traditionally, literature on complexity has tended to come from scientific domains taking a systems perspective, such as social and natural sciences, biology, sociology, philosophy which have frequently question emergent behavior, adaptivity and evolution. However, other areas of knowledge have turn to complexity as a way to get answer at questions that would otherwise remain inaccessible and to offer a key to new kinds of understanding [30]. Therefore, Complexity has become a highly interdisciplinary topic today, building bridges between several fields. This reflects the fact that most real-world systems are complex, and so increasingly are our technologies. As pointed out in [34] complexity is considered an inherent feature of the matter, being nature its ultimate source[35], and so is technology.

According to [30] the field of complexity studies has split into two subfields: the study of *Complex Physical systems*(CPS) and the study of *Complex Adaptive Systems*(CAS). The former studies has a set of tools and questions centering on elements with fixed properties and has let to a better understanding of physical phenomena. The latter deals with elements that are not fixed, usually called agents, that learn or adapt in response to interactions with other agents, continually exchanging information.

1) Characterization of Complex Adaptive Systems: Complex system, due to its study from different perspectives, present a challenge when trying to organize the features involving its functioning and structure. In [36] Holland proposes a general way of charactering Complex Adaptive Systems by seven basics: four properties(*p*): aggregation, nonlinearity, flows and diversity, and three mechanisms(*m*): tagging mechanism, internal models and high-level reorganization patterns. These seven principals intent to be a framework to organize the different features present in complex adaptive systems.

They are not the only basic features that could be selected from all the complex systems, although, most of the other can be derived from appropriate combinations of them. Actually, literature present different analysis of complex systems characteristics according to models or basic theory. Another approach to analyze is presented in [34] where the author elaborates table summing up the basic Properties of Complex Systems and its comparison with other systems.

In the following is a list, even though not exhaustive, we collected the main characteristics identified in Complex Systems from seminar literature in the field [30], [37], [36], [32], [33]:

- Self-Organization: as the ability to structure and restructure themselves, to learn, diversify and increase their complexity;
- Decentralization of decision making;
- Nonlinear relationships or interactions between its components: a small change in the system can lead to disproportionate effects;
- Learning capacity: a computational mechanism by which a cognitive system could iteratively build up a detailed and hierarchical model of its environment;
- Default hierarchical structure to organize and manage the behavioral laws at different levels. A hierarchical structure is the natural consequence of the aggregation property mentioned in [36];
- Feedback mechanism: reinforcing learning actions are self-enhancing and lead to better performances or avoid threads in future behavior;
- Path dependency and historical information: the state of a complex system at any point in time depends on the sequence of events and decisions that preceded that point. This way learning and action mechanisms would allow the system to adapt without losing what it had learned in the past;
- Emergent behavior;
- Balancing two forces or operations: one reinforcing growth by taking advantage and keeping already known successful configurations (exploitation), and the other aiming at discovering new combinations of traits that can retrieve better building blocks yet unknown (exploration);
- Mindset and models: all complex systems creates and use internal models to prosper: This models can be of different nature: tacit or explicit, learned in a single lifespan or through evolution. Models represent both: the own adaptive agents and the environment in which it perform. It is common the models interact with stored knowledge as a way to bring efficiency to the functioning of the system;
- Adaptive interactions: the agents of complex system can perform adaptive actions to process change through behavioral rules continually adjusted through evolution and learning;
- Perpetual novelty, it is unlikely for it to reach an optimal or equilibrium.

2) Adaptation within this perspective: Some of the aforementioned characteristics even when not mentioning adaptivity directly depend upon this faculty to develop a technique for processing continuous change. Feedback, learning, self-organization, aggregation and balancing the dichotomy of forces are some of the characteristics at systems' level that laid upon adaptivity to achieve its major goals.

The direct mention of adaptivity is particularly granted to elements of the system, the agents. Agents in CAS have three levels of action:

- Performance: designates an agents' behavioral repertoire at a point in a time. It is often modeled as a set of rules and signals, and described via signal-processing approach. An agent is sensible to its environment via detectors and effectors.

Adaptation happens by changing the signal-processing rules, which corresponds to changes in the structure of the associated network. To implement such a change the CAS needs that agents be able to:

- Credit assignment: an agent requires a means of assigning a quantity (strength, grade), that rates the usefulness of different rules in helping the agent to attain important resources;
- Rule discovery: a mechanism to discover new rules by combination and/or reorganization of atoms from successful rules. The new rule is biased by the agent's previous experience.

In [36], Holland states that adaptation is the *sine qua non* of CAS. He defines CAS as systems composed by interacting agents described in terms of rules. These agents adapt by changing their rules as experience accumulates. Moreover, in [37] the authors state that lifelong research work developed by Holland, recognized that adaption is central to fields that concern populations of agents that must continually obtain information from uncertain, changing environments and be able to use it to improve its performance and chance of survival, this extend the impact of studying adaptive behavior beyond CAS.

This approach of complex system and adaptivity grant the system with the freedom of working with several techniques to implement the required mechanisms.

D. Autonomic Computing

Autonomic Computing is a term coined by IBM's vice-president, Paul Horn at 2001, while presenting the idea in a keynote speech to the National Academy of Engineers at Harvard University[38]. It describes systems that can manage themselves, self-managing systems, given high-level goals from human administrators[39], [40], [14]. The term resembles the ability observed in the *Autonomic Nervous System* to govern and regulates a whole set of body functions, such as heart rate, body temperature, blood circulation and other "involuntary" actions without the need for conscious human involvement [39], [38], [41].The IBM initiative was formalized in [39], [14], [40] as a Grand Challenge within the information society, and particularly in the field of Information and Communication Technology (ICT) due to the complexity produced as byproduct of evolution in human society via automation. In particular, computer systems and its applications, have experience an explosive growth in the last decades and now nearly pervade every aspect of our life [41]. However, they have reached the point in which the benefits that IT aims to provide are threaten by its own complex infrastructure, making them difficult to manage and use [39].To cope with this challenge, autonomic computing systems need to be capable of running themselves, adjusting to varying circumstances and preparing their resources to handle the workloads that we put upon them [39].

To build such high-level systems the authors in [39], [14], [40] describe the 8 key characteristics shown in Fig. 3 and detailed as follows.

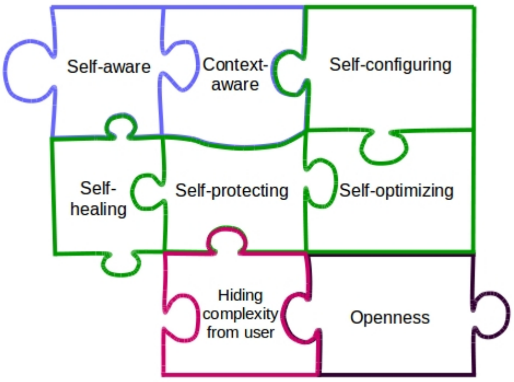


Fig. 3. Key characteristic of Autonomic Computing Systems. Adapted from [39], [14], [40].

In detail:

- Self-awareness and context awareness are inherent characteristics of Autonomic Computing systems regardless of the specific task each one performs. Any Autonomic Computer system must be able to observe their own structure an behavior to adapt or modify it, and be aware of its environmental and operational context [42].
- Self-configuring, self-healing, self-optimizing and self-protecting are the core capabilities that support self-management in Autonomic Computing. The works on [39], [40], [38] offer detailed descriptions and analysis of this capabilities.
- Hidden the complexity from user which implies autonomy in the process of decision taking to achieve the high-level policy given by the user.
- Openness, meaning that an autonomic computing system must be designed to operate in an heterogeneous environment, interacting with other technological elements (autonomic or not)[41]. Additionally, autonomic computing systems must be portable across multiple platforms [38].

The essence of Autonomic Computing initiative is to build self-managed, the intent of which is to free system administrators from low-level technical task, such as configuration, installation, updating and other system operation and maintenance tasks [40], to focus in high-level activities. It's worth notice that the automatic systems only performs the tasks that IT professionals choose to delegate to technology through policies.

1) Autonomic Computing Model: In [14] IBM describes a proposal for the necessary architectural building blocks, organization and behavior to build self-managing autonomic capability. The proposed architecture has two levels: systems' level and autonomic manager's level. At system's level the architecture organizes the elements in a hierarchy, as shown in Figure 4, governed by the manual managers, or IT specialists and assisted by knowledge sources.

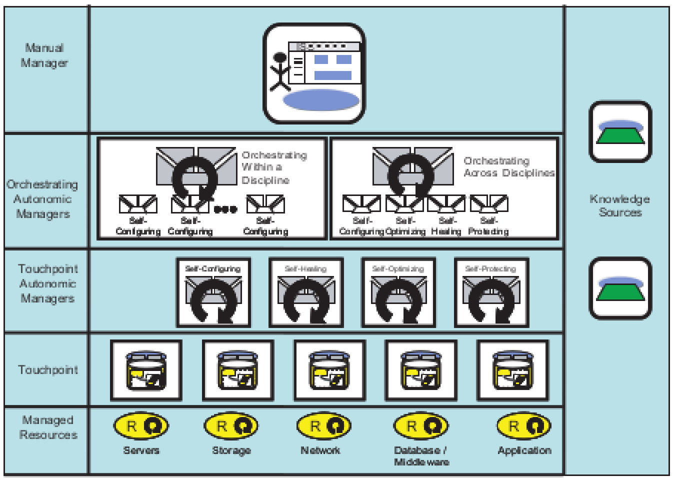


Fig. 4. Autonomic Computing reference architecture. Extracted from [14].

Layer 1, contains the managed resources (hardware/software, local/distributed), which may have embedded self-managing attributes. The resources, its types and scopes, define a set of decision-making context that are used to classify the purpose and the role of a control loop within the autonomic computing architecture. Layer 2, add a standard interface for accessing and controlled the resource called *touchpoint*. Layer 3 contains the touchpoint autonomic manager that interact directly with the touchpoints of managed resources to embody different tasks that support self-managing capability. Layer 4 contains autonomic managers that orchestrate other managers, they are the responsible for delivering the system-wide autonomic capability by implementing control loops that have the broadest view of the infrastructure.

At autonomic manager's level, the components must implement the architecture known as MAPE-K(Monitoring-Analyzing-Planing-Executing-Knowledge) architecture. Figure 5 shows the MAPEK architecture and its general elements. According to IBM, an autonomic manager must have the following features to exhibit self-managing properties [14]:

- An automated method to collect the details it needs from a managed resources, via touchpoint sensor interface and correlate them into symptoms that can be analyzed. This function is named *Monitoring*;
- An automated mechanism to observe and analyze situations to determine if some change needs to be made. To do so, in many cases, these mechanisms model complex behavior to employ prediction techniques. These mechanisms allow the autonomic system to learn about the

environment and predict future behaviors. This function is known as *Analyzing*;

- A mechanism that constructs the actions to enact a desired alteration in the managed resource, known as change plan, and logically passes that set of actions to the execute function. This mechanism can take many forms, goes from a single command to a complex work-flow. This function is known as *Planning*;
- A mechanism to schedule, control and carry out the actions on the plan change over one or more managed resources using the touchpoint effector interface of a managed resources. Finally, this function is known as *Executing*.

These mechanisms communicate and collaborate with one another and exchange appropriate knowledge and data. The information within the knowledge database consists of particular types of data (policies, symptoms, metrics, and logs) that can be access and/or modified by the mechanisms in the autonomic manager.

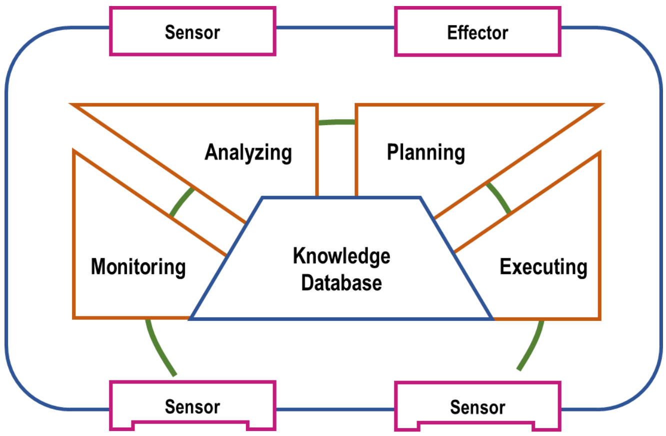


Fig. 5. Internal architecture for autonomic managers. Extracted from [14].

2) Adaptivity within Autonomic Systems: Adaptivity is the ability that supports the architecture of Autonomic Computing at both, system and component level. At a general level, the focus of Autonomic Computing approach is the design and implementation of self-management systems, which, according to [8], inherently laid upon adaptivity as main function. At the system's level the ability for rearranging components, within and between levels, exercises adaptivity to create, update or drop relations at run-time and as information arrives. Moreover, the self-* properties mentioned in the key characteristics for Autonomic Computing again laid onto adaptivity to process change. Finally, at component level, mechanism and elements organize too the basic elements to implement adaptive behavior. So adaptivity is present at Autonomic Computing as the backbone function to process dynamic change.

IV. Holistic vision

In general terms an adaptive program can be describe as a computer program that is able to evaluate its own behavior and make some changes in its own configuration (structure and functionality) at runtime when, due to new conditions perceived in its environment, its evaluation indicates :(i) that the program is not accomplishing its reason to exist, or (ii) that better functionality or performance is possible.

The variability of such a description allows formulating problems in different areas that involves optimization made difficult by substantial complexity and uncertainty where information must be exploited as acquired so that performance maintains or improves apace. Problems with this characteristics are pervasive and occur at critical points in different fields, as we have seen in the previous section, and even when the formulations to solve those problems can have a variety of guises, they give rise to the same fundamental questions.

A unified theoretical framework provides opportunities for identifying common interactions, methods and difficulties faced when studying adaptivity.

In the following subsections we will present the common ground of the four technological approaches to deal with adaptive behavior. We will draw the necessary conditions for the existence of adaptive behavior, the main properties observed in adaptive systems, and the characteristic of the problems that can found benefits in the use of adaptive behavior.

A. Necessary Conditions

- Awareness[43], [33], [44] : both self-awareness and context awareness. The former allows the program to inspect its own current configuration and recognize new needs and evaluation. The latter allows the program to perceive particular events in its application domain;
- Models[7] : system or internal model and environmental model. They can be either implicit and learned over evolutionary time, i.e. evolutionary computing, or explicit and learned (or given) over a single lifespan. The latter allows the program to reason about the impact of new-coming events in the environment in its own performance. Both models also allow the implementation of mechanisms of predictions that assist the program in decision taking during its functioning;
- Hierarchical structure[37]: hierarchy allows the application and managing of behavioral rules according to the level of abstraction of the entities and their capabilities;
- Characteristic response[37]: The normal or standard behavior that the program should perform under no changing conditions. This is the basic behavior over which the modifications, the adaptations, will be developed;
- Monitoring and selection Mechanism [44]: the mechanisms responsible for perceiving and filtering events in the environment and selecting plan and strategies to respond to them;
- Autonomy[43], [33]: the capacity to perform reasoning over its own behavior and control and acting to modify it without depending on human intervention;
- Learning Method [33], [44]: also called Learning Cycle. This is the part where the adaptivity *per se* is performed;
- Memory[37], [7]: stored knowledge.

Particularly, the learning method need to implement some basic elements:

- Feedback loop [44], [36]: the actions taken to modified the behavior and its consequences are incorporated in the stored knowledge, to generate better predictions and refine the desired behavior;
- Structural modifiers or operators [37], [7]: the elements responsible for implementing the direct modifications and its propagation. They implement the adaptive plan or strategy;
- Mechanism for comparison and assessment [44], [37], [7] : are the mechanism that allows the program to evaluate how convenient are different alternatives to modify its behavior.

B. Properties observed in Adaptive Systems

- Open-endedness [33], [37]: they are in dynamic stability, never achieve equilibrium due to balancing two opposite forces (exploitation vs exploration, or balancing loop vs reinforcement loop, or homeostasis vs homeorhesis).
- Continuity [33], [36], [7], [37], [44]: they are always perceiving changes;
- Traceability [30], [7];
- Self-* properties: usually adaptivity is used to manage the changes to achieve one or more self-* properties;
- Emergence and Propagation of features[44], [37];
- Exhibition of high-order patterns[37]: the mechanism and patterns used to implement adaptivity are not simple strategies and use sophisticated tools;
- Operation at multiple spacial and temporal scales.

C. Problem Characteristics

By identifying the main characteristics of a problem it is possible for a researcher to focus on searching resources (methods, tools, mechanisms and so on) that are aligned with the purpose of dealing with such characteristics. In this context, adaptive behavior has been credit as a good alternative in helping answering problems with the following main characteristics:

- Complexity [43], [32], [37], [7];
- Uncertainty [43], [32], [37], [7];
- Nonlinear interactions [32], [44];
- Changing Environments [37];
- An implicit optimization principle: the goal is to do the best by improving performance, and/or enhance/maintain the chance for survival with available resources [44], [37], [7].

V. Related Work

Related work naturally draws from several fields due to the multidisciplinary nature of adaptive behavior. We address related work from an unified vision of adaptivity. The most efforts to create an holistic approach comes from Complex Systems field and lately from Self-Adaptive Software Engineering. In complex System, Holland efforts for building a general theory of adaptation in complex systems are reflected in [36], [30], [37]. Moreover, the Santa Fe Institute[45] is one of the top institutions to research complex systems with a broader approach. The authors in [31] present another effort in develop a unified theoretical and practical framework for Complex Systems. Other researchers have taken the formal languages approach as a base to generate an unified model for adaptive software, such as [46], [47], [48], and other have tried to extend the concepts,models and architectures within their fields to cover different approaches, such as [41]. In [49] with can observe a unified an hybrid approach between cybernetics and formal methods.

From the field of Software Engineering, some efforts have been made for the better understanding of adaptivity. In literature we can find some of the authors from different backgrounds presenting a vision of integration with fields such as control theory [50], multi-agent systems [51], active software and so on [19]. In [21], [21] the authors explore architectures and models of Software Engineering and Testing for adaptive systems. With the same purpose some researchers have develop design patters for adaptivity [52].

However, the lack for a general examination that helps researchers in mapping elements from one approach to the other is the gap we are trying to fill with this work.

VI. Conclusion and Future Directions

By the end of this work we have come to the following conclusions:

- In the same sense that was previously mention in [32] the building of a common ground framework, where elements intent to be in its basic form, allows the appropriation of concepts and results from other complex systems for the purposes of explaining this concepts within adaptive technologies.
- Understanding the participants involved in adaptive behavior allows to identify core components, even when named under different terminologies, that perform the same task. By doing so it is possible to establish parallels between theories and techniques, resulting in opportunities to share knowledge [35].
- This holistic approach based on aggregation gives freedom to the researcher to integrate techniques, mechanisms, algorithms and other technological resources to different frameworks, representation or architectures according to the particular kind of adaptive behavior it needs to perform.
- Cross-comparisons provide the advantage of putting in evidence some characteristics that in one field are subtle and hard to extract while in other are salient and easy to examine [36].

Future work in the effort for building an holistic understanding of adaptivity are:

- Standard terminology should be build up because the current domain specific terminology does not help the holistic approach by being ambiguous or sometimes overlapping.
- Exploration and incorporation of new approaches for adaptivity in other fields to empower and support the holistic approach. In this sense, this work pretend to be only a small but significant contribution to such common

approach from the perspective of computing, we enthusiastically encourage researchers from other areas dealing with adaptive behavior to take an holistic approach and complement this effort.

- Elaboration of scientific analysis from areas such as Bibliometrics, Semantic Analysis and Semiotics can bring highly valuable information about how adaptivity is understood and the meaning it carries, helping to refine the holistic model. This studies are beyond the scope of the present work, but we encourage researchers in these areas to contribute with such an specialized knowledge to a common approach.
- Study of models and approaches for different parts of software development besides design and implementation, such as: testing, benchmarks,and others.
- Experimentation with hybrid approaches by exercising component model approach and aggregation properties.

REFERENCES

[1] A. Mayor. (2012, March) The world's first robot: Talos. The World's First Robot: Talos, Wonders and Marvels. [Online]. Available: http://www.wondersandmarvels.com/2012/03/the-worlds-first-robot-talos.html

[2] K. Anthis. (1990, Apr.) Talos. Automatons: Robots of Legend. [Online]. Available: https://sites.google.com/site/automatonrobotsoflegend/mischeif/talos

[3] W. Hyman, *The Automaton in English Renaissance Literature*, ser. Literary and scientific cultures of early modernity. Ashgate, 2011. [Online]. Available: https://books.google.com.br/books?id=TAIkn8WtjnIC

[4] M. Mitchell, *Complexity: A Guided Tour*. Oxford University Press, 2011.

[5] D. Enke, "Part i: Adaptive systems," *Procedia Computer Science*, vol. 20, pp. 11 – 13, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913010314

[6] S. Poslad, *Autonomous Systems and Artificial Life*. John Wiley and Sons, Ltd, 2009, pp. 317–341. [Online]. Available: http://dx.doi.org/10.1002/9780470779446.ch10

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[8] E. U. Warriach, T. Ozcelebi, and J. J. Lukkien, "Self-* properties in smart environments: Requirements and performance metrics," in *In Workshop Proceedings of the 10th International Conference on Intelligent Environments*. IOS Press, 2014, pp. 198–205.

[9] A. Berns and S. Ghosh, "Dissecting self-* properties," in *2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Sept 2009, pp. 10–19.

[10] R. Laddaga, "Self adaptive software baa 98-12 due 1/27/98," DARAP Broad Agency Announcement, Discussion paper, Tech. Rep., 1998.

[11] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 2, pp. 14:1–14:42, May 2009. [Online]. Available: http://doi.acm.org/10.1145/1516533.1516538

[12] J. S. Bradbury, "Organizing definitions and formalisms for dynamic software architectures," *Techinical Report*, vol. 477, 2004.

[13] M. Rahman, R. Ranjan, R. Buyya, and B. Benatallah, "A taxonomy and survey on autonomic management of applications in grid computing environments," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 16, pp. 1990–2019, 2011. [Online]. Available: http://dx.doi.org/10.1002/cpe.1734

[14] "An architectural blueprint for autonomic computing," IBM, Tech. Rep., Jun. 2005.

[15] R. Sterritt, M. Parashar, H. Tianfield, and R. Unland, "A concise introduction to autonomic computing," *Adv. Eng. Inform.*, vol. 19, no. 3, pp. 181–187, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.aei.2005.05.012

[16] D. B. Abeywickrama and E. Ovaska, "A survey of autonomic computing methods in digital service ecosystems," *Serv. Oriented Comput. Appl.*, vol. 11, no. 1, pp. 1–31, Mar. 2017. [Online]. Available: https://doi.org/10.1007/s11761-016-0203-8

[17] R. Frei, R. McWilliam, B. Derrick, A. Purvis, A. Tiwari, and G. Di Marzo Serugendo, "Self-healing and self-repairing technologies," *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 5, pp. 1033–1061, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00170-013-5070-2

[18] F. D. Macías-Escrivá, R. Haber, R. del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7267 – 7279, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413005125

[19] P. Robertson, R. Laddaga, and H. Shrobe, "Introduction: the first international workshop on self-adaptive software," in *Self-Adaptive Software*. Springer, 2000, pp. 1–10.

[20] M. M. Kokar, K. Baclawski, and Y. A. Eracar, "Control theory-based foundations of self-controlling software," *IEEE Intelligent Systems*, vol. 14, no. 3, pp. 37–45, May 1999. [Online]. Available: http://dx.doi.org/10.1109/5254.769883

[21] B. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds., *Software Engineering for Self-Adaptive Systems*, ser. Lecture Notes in Computer Science. New York: Springer Berlin Heidelberg, 2009.

[22] R. de Lemos, H. Giese, H. Müller, and M. Shaw, Eds., *Software Engineering for Self-Adaptive Systems II. International Seminar, Dadstuhl, Germany, October 24-29, 2010. Revised and Invited Papers*, ser. Lecture Notes in Computer Science. New York: Springer Berlin Heidelberg, 2013.

[23] F. Heylighen and C. Joslyn, "Cybernetics and second order cybernetics," *Encyclopedia of physical science & technology*, vol. 4, pp. 155–170, 2001.

[24] N. Wiener, *Cybernetics: Control and communication in the animal and the machine*. Wiley New York, 1948.

[25] S. Beer, *Decision and Control: The Meaning of Operational Research and Management Cybernetics*. John Wiley, 1966.

[26] S. A. Umpleby, "Cybernetics," in *International Encyclopedia of Organization Studies. Volume 1*. Sage, 2007, pp. 350–353.

[27] C. G. DeYoung, "Cybernetic big five theory," *Journal of Research in Personality*, vol. 56, pp. 33 – 58, 2015, integrative Theories of Personality. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0092656614000713

[28] J. B. Peterson and J. L. Flanders, "Complexity management theory: Motivation for ideological rigidity and social conflict," *Cortex*, vol. 38, no. 3, pp. 429 – 458, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010945208706804

[29] C. S. Carver and M. F. Scheier, *On the self-regulation of behavior*. Cambridge University Press, 1998.

[30] J. Holland, *Complexity: A Very Short Introduction*, ser. Complexity: A Very Short Introduction. Oxford University Press, 2014. [Online]. Available: https://books.google.com.br/books?id=lmygAwAAQBAJ

[31] C.-C. Chen and N. Crilly, "From modularity to emergence: A primer in the design and science of complex systems," University of Cambridge,Department of Engineering, Cambridge, UK, Tech. Rep., 2016.

[32] M. Mitchell and M. Newman, "Complex systems theory and evolution," *Encyclopedia of Evolution*, pp. 1–5, 2002.

[33] P. Cilliers, *Complexity and Postmodernism: Understanding Complex Systems*. Taylor & Francis, 2002. [Online]. Available: https://books.google.com.br/books?id=OeCEAgAAQBAJ

[34] F. Grabowski and D. Strzalka, "Simple, complicated and complex systems: the brief introduction," in *2008 Conference on Human System Interactions*, May 2008, pp. 570–573.

[35] "How can we share solutions to complex systems problems across domains and application areas?" Mar 2017. [Online]. Available: https://phys.org/news/2017-03-solutions-complex-problems-domains-application.html

[36] J. H. Holland, *Hidden Order: How Adaptation Builds Complexity*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.

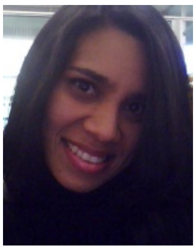
[37] S. Forrest and M. Mitchell, "Adaptive computation: The multidisciplinary legacy of john h. holland," *Commun. ACM*, vol. 59, no. 8, pp. 58–63, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2964342

[38] M. R. Nami and M. Sharifi, *A Survey of Autonomic Computing Systems*. Boston, MA: Springer US, 2007, pp. 101–110. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-44641-7_11

[39] P. Horn, "Autonomic computing: IBM's Perspective on the State of Information Technology," 2001.

[40] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1109/MC.2003.1160055

[41] P. K. Singh, A. Sharma, A. Kumar, and A. Saxena, “Autonomic computing: A revolutionary paradigm for implementing self-managing systems,” in *2011 International Conference on Recent Trends in Information Systems*, Dec 2011, pp. 7–12.
[42] T. M. King, A. E. Ramirez, R. Cruz, and P. J. Clarke, “An integrated self-testing framework for autonomic computing systems,” *JCP*, vol. 2, pp. 37–49, 2007.
[43] N. Khakpour, S. Jalili, C. Talcott, M. Sirjani, and M. Mousavi, “Formal modeling of evolving self-adaptive systems,” *Science of Computer Programming*, vol. 78, no. 1, pp. 3 – 26, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642311001742
[44] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations with CD-ROM*, 1st ed. Berlin, Germany: John Wiley & Sons, Inc., 1999.
[45] S. F. Institute. (2017) Santa fe institute website. [Online]. Available: https://www.santafe.edu/
[46] R. Bruni, A. Corradini, F. Gadducci, A. Lluch Lafuente, and A. Vandin, *A Conceptual Framework for Adaptation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 240–254. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28872-2_17
[47] E. Serral, J. D. Smedt, M. Snoeck, and J. Vanthienen, “Context-adaptive petri nets: Supporting adaptation for the execution context,” *Expert Systems with Applications*, vol. 42, no. 23, pp. 9307 – 9317, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415005400
[48] A. Bhattacharyya, A. Mokhov, and K. Pierce, “An empirical comparison of formalisms for modelling and analysis of dynamic reconfiguration of dependable systems,” *Formal Aspects of Computing*, vol. 29, no. 2, pp. 251–307, March 2017. [Online]. Available: http://dx.doi.org/10.1007/s00165-016-0405-z
[49] L. Chen, L. Huang, C. Li, and X. Wu, “Self-adaptive architecture evolution with model checking: A software cybernetics approach,” *Journal of Systems and Software*, vol. 124, pp. 228 – 246, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121216000820
[50] Y. Diao, J. L. Hellerstein, S. Parekh, R. Griffith, G. Kaiser, and D. Phung, “Self-managing systems: a control theory foundation,” in *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS’05)*, April 2005, pp. 441–448.
[51] C. Bailey, D. W. Chadwick, and R. de Lemos, “Self-adaptive federated authorization infrastructures,” *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 935 – 952, 2014, special Issue on Dependable and Secure Computing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000014000154
[52] Y. Abuseta and K. Swesi, “Design patterns for self adaptive systems engineering,” *CoRR*, vol. abs/1508.01330, 2015. [Online]. Available: http://arxiv.org/abs/1508.01330



Rosalia Edith Caya Carhuanina Was born in Lima, Peru. She received the degree of BsC (Bachelor in Science and Engineering) with major in Computer Engineer, and the professional degree of Computer and Systems Engineer from the Pontifical Catholic University of Peru (PUCP, Pontificia Universidad Católica del Perú) at Lima in 2009. Between the years 2007 and 2011, she participated of the activities at the Artificial Intelligence Research Group at the Computer Engineering Department of the same university. She holds a MsC degree in Electric Engineering, with major in the field of Computer Engineering from the University of São Paulo (USP) at São Paulo, Brazil. She is currently a PhD student developing her research in the Laboratory of Adaptive Languages and Techniques (LTA). Her main areas of research are: Adaptive Technology, Autonomous Systems, Cybernetics, Artificial Intelligence, Natural Language Processing, Human-Machine Interaction.

João José Neto graduated in Electrical Engineering (1971), Master in Electrical Engineering (1975), PhD in Electrical Engineering (1980) and Professor (1993) at the Polytechnic School of the University of São Paulo. Currently, he is an associate professor at the Polytechnic School of the University of São Paulo and coordinates the LTA - Laboratory of Languages and Adaptive Technology of PCS - Department of Computer Engineering and Digital Systems of EPUSP. He has experience in the area of Computer Science, with emphasis on the Fundamentals of Computer Engineering. He works mainly on the following topics: adaptive devices, adaptive technology, adaptive automata, and in their applications to Computer Engineering, particularly in adaptive decision making systems, analysis and processing of natural languages, construction of compilers, robotics, computer-aided teaching, modeling of intelligent systems, automatic learning processes and inferences based on adaptive technology