Towards Autonomic Virtual Organization

MohammadReza Nami   Mohsen Kahani

Computer Engineering Laboratory
Faculty of Electrical Engineering, Mathematics, and Computer Science
Ferdowsi University of Mashhad
The Iran
E-mail: nami@iau-saveh.ac.ir kahani@ferdowsi.um.ac.ir

Abstract

The evolution of networks and the Internet, which have presented high scalable and available services have made environments more complex. The increasing complexity, cost, and heterogeny in them have motivated researchers to investigate a new idea to cope with the management of complexity in IT industry. Autonomic Computing Systems (ACSs) have been introduced. Moreover, the increasing demands for extended products and services along with advances in IT industry have metamorphosed traditional business toward creating Virtual Organizations (VOs) in order to better respond to business opportunities. Purpose of this paper is to present a survey of ACSs and VOs. It includes some characteristics, issues, and challenges in both. Applying autonomous behavior in improvement of VOs is also proposed in conclusion.

Keywords: Agent, Multi-agent System, Autonomic Computing Systems, Self-managing Systems.

1 Introduction

Advances in communication technology specially the Internet and increasing desktops power have helped distributed computing systems to become so complex. Distributed computing can be considered as an environment where use power of networked computers to work together. The need to integration has revolutionized the management of such environments. IT organizations has encountered with the grand challenges in management and maintenance of large scale heterogeneous distributed computing systems. Automation is the capability to dynamically monitor intervention. They have to manage an IT infrastructure to meet the business goals without human intervention. Autonomic Computing Systems (ACSs) have been introduced as an approach for developing large scale distributed computing systems with the aim of decreasing the cost of developing and managing the complex such systems. They manage themselves while hiding their complexity from the view of end users. Autonomics is used for acting or responding involuntary. Current technologies such as grid computing, middleware, P2P, networking, database, and agent technology have been described as a part of autonomic computing. Moreover, advances in computer networks have also affected marketing and business systems so that traditional business systems have been metamorphosed. The increasing customer needs to extended services and products with high quality, seeing business opportunities, and competitive advantage for organizations in distributed environments have emerged a new concept. Virtual Organizations have introduced as a new organizational schema including a temporary set of geographically organizations collaborating, sharing skills and resources to fulfill customer requests in a networked environment. Beside, The increasing complexity, cost, and heterogeny in such environments have motivated researchers to use autonomic behavior in VO environment. A Multi-Agent System (MAS) consists of a collection of autonomous agents that interacts to each other and can build a self-organization VO.

This paper is organized as follows. We present a survey of related works about ACS and VO in Section 2. In section 3, we state an overview of ACSs includig definition, their characteristics, Autonomic Elements (AEs) architecture as building blocks in ACSs, and some challenges such as robustness and relationships among AEs. Different VO definitions, characteristics, and some issues are discussed in section 4. Finally, we present conclusions and propose further researches.

2 Related Works

With the growth in the Internet technologies and services, IBM and other vendors such as HP, Sun, and Microsoft have recognized importance of complexity in the development and management of such systems. On March 8, 2001, Paul Horn [10] presented a link between pervasiveness and self-regulation in body 's autonomic nervous system and introduced ACSs to the National Academy of Engineering at Harvard University. With choosing the term *autonomic*, researchers attempted to make autonomic capabilities in computer systems with the aim of decreasing the cost of developing and managing them. S. White et al in [19], and R. Sterritt and D. Bustard in [17] have described some general architectures for ACSs and their necessary elements called autonomic elements. J. A. McCann and M. C. Huebscher in [15] have proposed some metrics to evaluate ACSs like cost and adaptability. Some performance factors such as security and availability have been discussed by others [4]. ACS properties have been discussed by many researchers. These properties include self-optimization [20], self-configuration [18], self-healing [9], and self-protection [17]. Grand challenges in engineering and scientific have been discussed in [12]. Different projects and products have been developed in both by the industry and the academic. M. Salehie and L. Tahvildari have outlined some of these products in [16].

Beside, evolution of the Internet and rapid changes in customer demands for extended services and products have motivated organizations toward a new cooperation schema including geographically and legally organizations that collaborate to achieve the goal. This cooperation is supported by computer networks. Jan Hopland, Nagel and Dove, and Davidow and Malone have introduced concepts such as virtual company, virtual corporation, and Virtual Enterprise (VE) in the early 1990s, but definitions and concepts of VO/VE are still completed. Although, VO is still studied, but many researchers are working on different issues and various projects are defined in this field. R. Camacho and et al [2] have presented a reference model that integrates all information related with VO planning and launching. A VO management should allocate and coordinate resources, manage VO members cooperation, evaluate partners, and define some rules to achieve the goal. Effective VO management needs VO to respond to fast changes in the environment. I. Karvonen and et al have presented some VO management approaches in [11]. L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus [3] are researchers who have many publications in this context. Since VO is a new idea which has not been fixed yet, there are many challenges and issues in designing, implementing, and managing them. Trust management [8], performance measurement [6], competency management [1], Applying agents in implementation of virtual enterprise [5], Virtual organization Breeding Environment (VBE) management, and Definition of a comprehensive model for VBEs [1] are some challenges in this field. Different projects and case studies have been Also performed in virtual organization field. L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus [3] have outlined some of these products.

3 Autonomic Computing Systems Perspective

3.1 Definitions, Characteristics, and Architecture

The autonomic concept is inspired by the human body 's autonomic nervous system. The human body has good mechanisms for repairing physical damages. It is able to effectively monitor, control, and regulate the human body without external intervention. An autonomic system provides these facilities for a large-scale complex heterogeneous system. In fact, an ACS is a system that manages itself. ACS components are often referred to as self-CHOP characteristics namely, self-Configuration, self-Healing, self-Optimization, and self-Protection. Our definition of autonomic computing is related to its properties. The initial definition of ACS has been presented by Paul Horn in 2001. According to it, an ACS is a self-management system with the following eight elements (characteristics):

1. **Self-configuration:** An ACS must dynamically configure and reconfigure itself under varying and unpredictable conditions.

2. **Self-healing:** An ACS must detect failed components, eliminate them, or replace them with another components without disruption of the system. It must predict problems and prevent failures.

3. **Self-optimization:** This is the capability of maximizing resource allocation and utilization for satisfying user requests. Resource utilization and work load management are two significant issues in self-optimization.

4. **Self-protection:** An ACS must identify and detect attacks and cover all aspects of system security at different levels such as the platform, operating system, applications, etc. It must also predict problems based on sensor reports and attempt to avoid them.

5. **Self-awareness or Self-knowledge:** An ACS needs to know itself. It must be aware of its components, current status, and available resources. It must also know which resources can be borrowed or lended by it and which resources can be shared.

6. **Context-awareness or environment-awareness:** An ACS must be aware of the execution environment to react to environmental changes such as new policies.

7. **Openness:** An ACS must operate in a heterogeneous environment and must be portable across multiple platforms.

8. **Anticipatory:** An ACS can anticipate its optimal required resources while hiding its complexity from the end user view and attempts to satisfy user requests.

Self-configuration, self-healing, self-optimization, and self-protection are considered as major characteristics and the rest as minor characteristics.

3.1.1 Autonomic Element Architecture

Autonomic Elements (AEs) are the basic building blocks of autonomic systems and their interactions produce self-managing behavior. We can consider AEs as software agents and ACSs as multi-agent systems. Each AE has two parts: Managed Element (ME) and Autonomic Manager (AM). An ME is a component from system such as server or database, and can be a single resource or a collection of resources. Sensors retrieve information about the current state of the ME, then compare it with expectations that are held in knowledge base by the AE. The required action is executed by effectors. Therefore, sensors and effectors are linked together and create a control loop. Autonomic Managers (AMs) are the second part of an AE. An AM uses a manageability interface to monitor and control the ME. It has four parts: monitor, analyze, plan, and execute. The monitor part provides mechanisms to collect information from a ME, monitor it, and manage it. Monitored data is analyzed. It helps AM to predict future status. Plan uses policy information and what is analyzed to achieve goals. Policies can be a set of administrator ideas and are stored as knowledge to guide AM. Plan assigns tasks and resources based on the policies, adds, modifies, and deletes the policies [19]. AMs can change resource allocation to optimize performance according to policies. Finally, the execute part controls the execution of a plan and dispatches recommended actions into the ME. These four parts provide control loop functionality.

3.2 Autonomic Computing Challenges

Since AC is a new concept in large-scale heterogeneous systems, there are different challenges and issues [13]. Some of them have been explained in the following:

3.2.1 Issues in Relationships among AEs

Service relationships among AEs have a key role in implementing self-management. This Relationships have a life cycle consists of specification, location, negotiation, provision, operation, and termination stages. Each stage has its own challenges. Expressing set of output services that an AE can perform and set of input services that it requires in a standard form and establishing syntax and semantics of standard services for AEs can be a challenge in specification. As An AE must dynamically locate input services that it needs and other elements that need its output services must dynamically locate this element with looking it up, AE reliability can be a research area in location stage. AEs also need protocols and strategies to establish rules of negotiation and to manage the flow of messages among the negotiators. One of challenges is the designer to develop and analyze negotiation algorithms and protocols, then determine which negotiation algorithm can be effective. Autonomated provision can be also a research area for next stage. After agreement, The AMs of both AEs control the operation. If the agreement is violated, different solutions can be introduced. This can be a research area. Finally, after the both AEs agree to terminate the negotiated agreement, internal resources of both are freed. The information about service relationships can be recorded in a local or in a global database.

3.2.2 Learning and Optimization Theory

A question raises this challenge: how can we transfer the management system knowledge from the human experts to ACSs? The master idea is that by observing that how several human experts solve a problem on different systems and by using traces of their activities, a robust learning procedure can be created. This procedure can automatically perform the same task on a new system. Of course, facilitating the knowledge acquisition from the human experts and producing systems that include this knowledge can be a challenge. One of reasons of the success of ACSs is their ability to manage themselves and react to changes. In sophisticated autonomic systems, individual components that interact with each other, must adapt in a dynamic environment and learn to solve problems based on their past experiences. Optimization can be also a challenge, because in such systems, adaptation changes behavior of agents to reach optimization. The optimization is examined at AE level.

3.2.3 Robustness

There are many meanings for robustness. Robustness has been served in various sciences and systems such as ecology, engineering, and social systems. We can interpret it as stability, reliability, survivability, and fault-tolerance, although it does not mean all of these. Robustness is the ability of a system to maintain its functions in an active state and persistence, when changes occur in internal structure of the system or external environment. The persons often mistake it with stability. Although both stability and robustness focus on persistance, but robustness is broader than stability. It is possible that components of a system are not themselves robust, but interconnections among them make robustness at the system level. A robust system can perform multiple functionalities for resistancing without change in the structure. With design of instructions that permit system to preserve its identity even when it is disrupted, the robustness in systems can be increased. Robustness is one of grand scientific challenges which can be also examined in programming.

4 Virtual Organization Overview

4.1 VO Definitions and Characteristics

Advances in Information and Communication Technology (ICT) and trends such as agility, globalization, and increasing demands for products and services with high productivity have motivated different organizations to cooperate and come together to explore business opportunities and fulfill customer tasks. There are three concepts used in this context: Virtual Organization (VO), Virtual Enterprise (VE), and Virtual organization Breeding Environment (VBE). A VE is defined as a networked, reconfigurable, and temporary collection of enterprises that cooperate and share resources, knowledges, and competencies for better responding to business opportunities. A VO is defined as a temporary coalition of reconfigurable, independent, networked, geographically dispersed organizations including high level trust and competencies that collaborate and share their resources and competencies in order to comply the customer request. L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus [3] also define VO as a "set of co-operating (legally) independent organizations, which to the outside world provide a set of services and act as if they were one organization. It is also reconfigurable. The co-operation is supported by computer networks". As mentioned, partners in a VO should collaborate in order to achieve business opportunities. Trust among them and operation according to a common agreement are essential things for collaborating. Networks or breeding environments are an appropriate context for effective creation of dynamic VOs. [1] has called this context as VBE and defined it as "an association of organizations and their related supporting institutes, adhering to a base long term cooperation agreement, and adoption of common operating principles and infrastructures, with the main goal of increasing both their chances and their preparedness towards collaboration in potential VOs". A VBE should identify and obtain new business opportunities, know the competencies and capabilities of its members, then select an appropriate set of partners for creation of new VO. Of course, it is possible to VBE administrator searches and recruits new organizations as member for new VO. As a result, A VE is a subset of VO and VOs are initiated within the VBEs. On the other hand, VBEs are a context to select the best partners for creation new VO with high agility according to the business opportunity characteristics, competencies and capabilities of the partners. The aim of a VBE is to improve preparedness of the partners [1].

Some benefits of VOs are to save time to market and fast development, decrease costs and risks in the delivery for the reason of partner competencies, improve quality factors such as performance and flexibility, exchange and share knowleges, and marketing in high scale (global networks). Since VOs have been formed from temporary organizations, they can match themselves with dynamic changes in marketing and assemble the organizations for better respond to business opportunities. According to definitions of VOs and VEs, the following characteristics can be considered for them [3]:

1. **Temporalization:** As a VO is created for a specific business opportuinity, it has a life cycle including creation, operation, evolution, and dissolution stages. During these stages, interorganizational processes and relations should be performed in a temporal limit. After customer delivery, the VO lifetime terminates.

2. **Dematerialization:** This property is derived from the term "virtualization". It means there are virtual forms in products, communities, services, and so on. The works are not done in an office work. Existing mutual confidence for members, absence of physical attributes and administrator can affect system performance and flexibility.

3. **Delocalization:** Since VOs are formed in an virtual environment without space and capacity limitation, they work in such environments and do not need a particular location.

4. **Individualization:** One of ways for capturing market is to attend to mass production along with personal needs. This property combines both in order to improve competitiveeness and better fulfill customer demands.

5. **Non-Institutionalization:** Because operations are performed in a virtual environment without physical attributes, institutionalization of interorganizational relationships in such environments can be waived.

6. **Asynchronization:** This attribute causes members to asynchronously communicate and interact with each other via ICT in the context of innovations with the release of time. Some companies globally schedule their works in three shifts between dispersed locations.

7. **Integrative Atomization:** This prorerty refers to integrate all atomized core competencies of the participants for satisfying customer.

These properties can be categorized in three groups: product and service, VO conditions and environment, and effective VO operation characteristics. It is outlined in table 2.

Properties/Type	Product/Service	Environment	Operational
Temporalization		*	
Dematerialization	*		
Delocalization		*	
Individualization	*		
Non-Institutionalization			*
Integrative Atomization			*
Asynchronization		*	*

Table 1: A Form of Types of VO Properties

4.2 Virtual Organization Issues and Challenges

Since VO is fixing as a master component of dynamic collaborative networks, there are different issues and challenges in VO creation, management, design, and implementation. We discuss some of them in the following:

4.2.1 VO Life cycle

As mentioned, a VO framework is created for responding to a business opportunity and terminated after completion. The VO life cycle is similar to VBE life cycle. It is formed from three stages: creation, operation along with evolution, and termination. VO Creation consists of identifying business opportunities, examining the partner competencies, selecting partners from within or outside the VBE (network), forming the best partnership in terms of the competencies, creating the necessary databases, registering new memebres, and VO setupping are the key tasks in VO creation. VO Operation and evolution is next stage. At this stage, the activities of the partners are integrated, competencies and common knowledge are managed, cooperations are organized, and collaboration is improved in order to the best respond to the business opportunities. Performance measurment and evolution play key roles at this stage. Finally, each VO has a limited lifetime. Therefore, when a VO fulfill the business opportunity, It is dissolved. Of course, the partner competencies and relationships between the partners are kept in the network (VBE) knowledge. The collected knowledge can be used in creating another VO in the VBE.

4.2.2 VO Planning and Launching

VO planning activities include recieving and analyzing business opportunities, selecting proper partners, determining high level Work Breakdown Structure (WBS), and setting up VO. R. Camacho and et al [2] present a reference model for VO planning and launching. This model integrates the elements involved in VO creation in VO creation, modelling, and knowledge management dimensions. After VO planning activities mentioned in above, VO modelling is created in four views: Resource, organization, functional, and Knowledge. Resource view represents all resources used in the VO operation. Organization view represents responsibilities and authorities of the elements involved in the VO. Functional view represents the behavior of the elements involved in VO life cycle. Knowledge view represents the structure of knowledge among the elements involved in the VO and relationships between these elements (VO partners). Knowledge sharing and using among the VO partners needs to efficient management. This knowlege includes VO structure, VO members profiles, procedures of VO member responsibilities, and reports developed into VO life cycle. The authors have used Unified Modelling Language (UML) to creation of the VO models for each modelling view.

4.2.3 VO Management

As a VO is composed of different members located at dispersed sites, different issues can affect the VO. Therefore, the VO management must be examined in different aspects. We can categorize them in human issues and technical issues. For example, different experiences and cultures paly the key role in collaboration. Beside, communication between the partners, trust among them, VO planning, and security are important challenges from technical point of view. We briefly explain some of them in the following:

1. **Trust Management:** Trust among VO members (partners) is one of important issues in collaboration and VO creation, and affects the result of VO operations. It means that one organization as a VO member expects to other behave reliably in performing their tasks for achieving desirable goal of the VO. VO modeling is based on Trust. Modeling is an appropriate tools to enhance planning and implementation. The basis of trust management are Trust Parameters (TPs) such as customer response time as an operational TP and damages by natural disasters as a Economic TP. Jochen Haller [8] has presented some trust requirments such as identifying and selecting TPs, aggregating them, considering trust values for reputation, and managing trust in collaboration. Finally an architecture for reputation system has been proposed.

2. **Competency Management:** Competencies are a set of capabilities and skills that play the key role in partner selection for creating a new VO. If the partners with high level competencoes are selected, new VO will be able to achieve its goal efficiently. The important issues associated to competencies include applying robustness in competency management, introducing approaches for dynamically competencies collection and updating, and Designing an autonomic approach for collecting and upgrading competencies can be introduced as a challenge in this context [1].

3. **Security Management:** Security management should guarantee security in three aspects: legal, organizational, and technical. It is possible that legal operation is violated or new threats alter the organizational structure of VO. Technical errors can also change configuration. In this case, the VO system should have a reconfigure plan for reorganization. Security manager should continuously monitor the VO members operation and authorization, detect violation of rules and threats, then generate proper alarms for eliminating it. J. Magiera and A. Pawlak [14] have proposed a multi-layer security system for VOs. This system consists of four layers for prevention of attacks, detection of the attacks alnog with generating alarms, protection of resources, and recovery layer. The important issues in security management of VOs including members at dispersed locations are member authorization to control access to resources, security in communication between members, and data integrity.

5 Conclusions and Future works

Use of ICT in business has changed views about traditional business. With VO, temporary coalition of geographically dispersed organizations collaborate, share their resources and competencies in order to better respond to business opportunities. Beside, The evolution of networks and the Internet has increased costs and errors of managing IT infrastructures. The skilled persons who manage these systems are expensive and can't manage them in configuration, optimization, healing, protection, and maintenance. IBM proposed Autonomic Computing Systems as a solution. ACSs manage themelves. AEs are the building blocks of ACSs and their interactions produce self-managing behavior. We can consider AEs in ACS similar to organizations in a VO. policies saved in knowledge in an ACS can be also considered contract and knowledge among the elements involved in the VO. As Co-operating organizations and trust among them build major body of a VO, AEs and their behaviors make an ACS. Agent is also defined as an entity with its own state, memory, and mechanisms to sense and interact with the environment. One of agent characteristics in Multi-Agent Systems (MASs) is autonomous that can be effective in building reconfigurable VOs. Some projects have been defined about applying agent-based framework in VO [7]. Adding autonomous behavior can enhance VO performance in healing and configuration, because they dynamically need reconfiguration and self-healing in order to better respond to their goals. Negotiation is one of master challenge in VOs. Management of the flow of messages among the partners can be improved with designing an effective algorithm.

In this paper, we presented an overview of autonomic computing systems and virtual organizations. As future researches, the following topics can be proposed in autonomic VO domain:

1. With respect to considering each organization within a VO as an AE, designing an effective negotiation algorithm for collaborators in a VO.

2. Designing trust learning in a VO.

3. Applying robustness in VO management. For example, when a resource or member is failed.

4. Design and analysis of an autonomic manager in multi-layer P2P form, so that autonomic behavior and VO knowledge management are stored in separated layers.

References

[1] H. Afsarmanesh and L. M. Camarinha-Matos. A framework for management of virtual organization breeding environment. In *PRO-VE'05*, page September, 2005.

[2] R. Camacho and et al. An integrative approach for VO planning and launching. In *PRO-VE'05*, page September, 2005.

[3] L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus. Virtual Organizations: systems and practices. In *Springer Science*, 2005.

[4] D. M. Chess, C. Palmer, and S. R. White. Security in an autonomic computing environment. In *IBM System Journal*, volume 42, pages 107–118, January 2003.

[5] C. Cubillos, F. Guidi-Polanco, and C. Demartini. Towards a virtual enterprise for passenger transportation using agents. In *PRO-VE'05*, September 2005.

[6] F. Graser and et al. Towards performance measurement in virtual organizations. In *PRO-VE'05*, September 2005.

[7] F. Guidi-Polanco, C. Cubillos, and G. Menga. The global automation platform: An agent-based framework for virtual organizations. In *PRO-VE'05*, September 2005.

[8] J. Haller. A stochastic approach for trust management. In *22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA*, April 2006.

[9] S. Hariri and M. Parashar. Autonomic Computing: An overview. In *Springer-Verlag Berlin Heidelberg*, pages 247–259, July 2005.

[10] P. Horn. Autonomic Computing: IBM 's perspective on the State of Information Technology. In *available at http://www.research.ibm.com/autonomic/manifesto/*, October 2001.

[11] I. Karvonen, I. Salkari, and M. Ollus. Characterizing virtual organization and their management. In *PRO-VE'05*, September 2005.

[12] J. O. Kephart. Research challenges of autonomic computing. In *Proceedings of the 27th International Conference on Software Engineering*, pages 15–22, May 2005.

[13] J. O. Kephart and D. M. Chess. The vision of autonomic computing. In *IEEE Computer*, volume 36, pages 41–50, January 2003.

[14] Jaroslaw Magiera and Adam Pawlak. Security frameworks for Virtual Organizations. In *Virtual Organizations, Springer*, pages 133–148, 2005.

[15] J. A. McCann and M. C. Huebscher. Evaluation issues in autonomic computing. In *Proceedings of Grid and Cooperative Computing workshop(GCC)*, volume 15, pages 597–608, October 2004.

[16] M. Salehie and L. Tahvildari. Autonomic Computing: emerging trends and open problems. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–7, July 2005.

[17] R. Sterritt and D. Bustard. Towards an autonomic computing environment. In *14th International Workshop on Database and Expert Systems Applications (DEXA '03)*, pages 694–698, September 2003.

[18] G. Tesauro and et al. A Multi-agent systems approach to autonomic computing. In *IBM Press*, pages 464–471, March 2004.

[19] S. White and et al. An architectural approach to autonomic computing. In *Proceedings International Conference on Autonomic Computing (ICAC'04), NewYork, USA*, pages 2–9, May 2004.

[20] T. De Wolf and T. Holvoet. Evaluation and comparison of decentralised autonomic computing systems. In *Department of Computer Science, K.U.Leuven, Report CW 437, Leuven, Belgium*, March 2006.