

# Self-Organization of Open Complex Distributed Systems

Huaglory Tianfield

*School of Computing and Mathematical Sciences, Glasgow Caledonian University  
70 Cowcaddens Road, Glasgow G4 0BA, United Kingdom  
E-mail: h.tianfield@gcal.ac.uk*

## Abstract

Self-organization has been investigated for long in a very broad range of areas. However, only till recently has self-managing and self-organizing computing received intensive attention. The purpose of this paper is to discuss some of important basic problems of self-managing and self-organizing computing, including the essence of self-managing computing, generic architecture of computing, real challenge of self-managing and self-organizing computing, and feedback-control point of view for self-managing and self-organizing computing. After a discussion of the essence of self-managing computing, the paper puts forward a generic architecture of computing processes. Then, the paper argues that the real challenges of self-managing and self-organizing computing are from open complex distributed systems. To address these challenges of self-managing and self-organizing computing, the paper presents a new definition of self-organization, from a structural perspective, and puts forward feedback-control framework for holonic self-organization of multi-agent systems.

## 1. Introduction

For some long time, the computing and information technology has been concentrated on the pursuit of high speed, VLSI, powerful computing capacity, etc. As a consequence, today's computing and information infrastructures have reached a level of complexity that is far beyond the capacity of human supported (i.e., manual) system administration. The unmanageable complexity incurred at the end of this pursuit has forced both academia and industries to re-think the fundamental issues of computing and information technology.

In recent years the international computing community has seen a great move to a new paradigm of computing, i.e., autonomic computing [1][2]. In (distributed) computing and networking, especially since 2001, a number of initiatives of self-organization and self-management have come into play in IT industries and academics worldwide. For example, early 2001 IBM Research launched its "Autonomic

Computing" Initiative, which aimed to design and build "Self-Managing" computing systems that would work like human autonomic nervous systems regulating human bodies. Immediately following, Microsoft has launched its "Dynamic Systems Initiative (DSI) -- The Drive to Self-Managing Dynamic Systems". At the same time, other world IT players have also branded their products/solutions with self-organizing and self-managing, e.g., "Adaptive Enterprise" by HP, "Self-Healing Networking", "Self-Defending Network" and "Adaptive Services Framework" by Cisco, "Self-Managing Database" by Oracle, and so forth.

This move is featured with the change from the previous pursuit of high speed, VLSI, powerful computing capacity to the pursuit of self-managements of computing systems. Self-managing computing will undoubtedly bring forth profound impacts both upon the discipline of computing and the IT industries.

## 2. Understanding self-managing computing

A human autonomic nervous system controls the human body incredibly well so that human beings can manage ever changing and unpredictable circumstances. Moreover, the control in human body works so well that normally no humans' interference and even consciousness are required. Primarily inspired by the functioning of human autonomic nervous systems, self-managing computing is to design and build computing systems that function as a human autonomic nervous system controls the human body. Essentially the self-managing computing paradigm is to equip computing systems with self-managements.

Essentially two mechanisms are primary for self-managing computing, i.e., system adaptation and the complexity hiding from users.

From a system science point of view, human beings are complex adaptive systems. For human beings, adaptation is the marrow. So is for self-managing computing system, which aim to mimic a human autonomic nervous system controlling the human body.

A self-managing computing system is autonomous in that it completely hides the complexity with the realization of system adaptation. Complexity hiding

from users means that self-managing computing will provide users with a computing environment that allows them to concentrate on what they want without worrying about how it has to be done. Uttered in a simple way, system adaptation is to be carried out without requiring the user's consciousness or significant effort. This is to say that complexity hiding is letting the machines rather than the human user do the job of system adaptation, i.e., getting automation of the system adaptation. Thus there comes a formula:

$$\begin{aligned} \text{Self-managing Computing} &= \text{System Adaptation} + \text{Complexity Hiding} \\ &= \text{Automation of System Adaptation} \end{aligned} \quad (1)$$

System adaptation, once possessed, is an inherent capability of a system, whether the system is automated or manually operated. Once a system possesses the primary mechanisms of system adaptation and complexity hiding, it will be able to exhibit a range of attributes, such as autonomy in the control and management of the resources inside the system and the service provisions to the outside, reliability and trustworthiness, robustness to un-modeled properties of the environment, adaptability to the environment, fault tolerance and resilience, protectiveness against attacks, etc.

Basically a self-managing computing system involves double automation [3]. This can be exemplified with an e-business system. While the basic business computing system has already automated the request/service processes from/to the business processes, what the self-managing computing system is equipping with on top of the basic business computing system is actually the automation of the system management. As depicted in Figure 1, based on the awareness and assessment of the operations of the business computing system and the requirements from the business process, the self-managing computing system enforces the adaptation actions upon the manageable patterns of the business computing system.

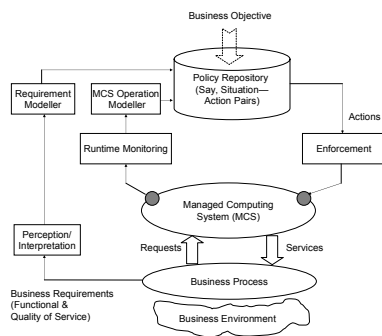


Figure 1 Generic architecture of self-managing computing systems

Therefore, one of the major challenges for self-managing computing system will be how to design explicitly the system adaptation and the complexity hiding in addition to conventionally the design of the basic computing systems.

### 3. Feedback-control frameworks of self-managing and self-organizing computing

Computing is the process where information is processed to fulfill a task by means of some hardware and software infrastructures. Essentially computing is the realization of a problem solving process. A generic architecture of computing processes can be put forward as depicted in Figure 2. As can be seen in Figure 2, computing is not simply information processing, but is a process of purpose-oriented information processing. Problem solving is normally not a one-stop case but an iterating process involving a complicated convolution of intermittent results and refinement of original goals and constraints. Some simple mechanisms for problem solving include recursion, means-ends analysis, etc. Run-time management of hardware and software infrastructures is the kernel of self-managing and self-organizing Computing.

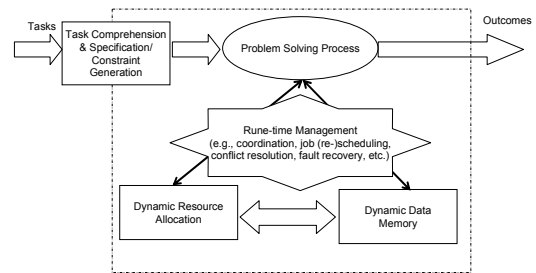


Figure 2 Generic architecture of computing processes

The primary category of self-managing computing system involves establishing control loops on top of the basic computing systems to bring forth passive adaptation to the systems. Specifically, this category involves equipping various types of computing systems (e.g., data storage systems, web service systems, networking systems, security/firewall systems, Grid resource management systems, etc.) explicitly with control loops, as sketched in Table 1 in conjunction with the generic architecture of self-managing computing system [3] (as depicted in Figure 1).

Table 1 Some elaborations of the adaptation scenarios for various types of computing systems in terms of the generic architecture of self-managing computing system

Managed computing system	Basic infrastructure	Instance of situation	Instance of action
Data storage systems	Data storage devices	Unsatisfied/varied storing patterns	Enabling, disabling, or outsourcing of data storage devices
Web service systems	Web servers	Unsatisfied/varied web service requests	Enabling, disabling, or outsourcing of web servers
Networking systems	Network servers, LAN/WAN infrastructures	Unsatisfied/varied networking patterns	Enabling, disabling, or outsourcing of network segments
Security/firewall systems	Web and/or network servers,	Detected attacks, disruptions	Reinforcement of firewall policies
Grid resource management systems	Cross-organizational distributed computing resources	Unsatisfied/varied computing service requests and computing resources availability patterns	Re-allocated and/or re-scheduled computing resources

It can be seen that feedback-control frameworks have widely existed in common computing processes. Common examples include artificial intelligence planning, Autonomous Agents, and cognitive computing, etc.

#### 4. Open complex distributed systems ---- real challenges for self-managing and self-organizing computing

Run-time management of hardware and software infrastructures is not entirely new. It has actually already existed in some fixed environments of common computing, e.g., the operating systems (OS's) of single machines or a fixed LAN (local area network). Table 2 illustrates the different types of management problems of hardware and software infrastructures.

Table 2 Different types of management problems of hardware and software infrastructures

		Service	Service
		Fixed	Re-Configurable
Infrastructure	Fixed	OS of desk-/lap-top PC's	e.g., web servers
Infrastructure	Re-Configurable	NOS, IoS, middleware of networks, e.g., LAN,	Open complex distributed (computing) systems, e.g., ad hoc wireless networks, wireless sensor networks, Grid computing, etc.

Note: NOS – networking operating system, IoS – internetworking operating systems

Re-configurability of infrastructures entails openness of systems, and re-configurability of services entails complex, dynamically changing environments. From Table 2, entries corresponding “fixed” infrastructure/service have their management problems more or less addressed by existing operating systems. What remains unresolved and challenging for self-managing and self-organizing computing is the entry of re-configurable infrastructures and re-configurable services, i.e., Open complex distributed systems [4],

where self-management and self-organization are urgently required.

#### 4.1 Self-organization of multi-agent systems

Self-organization is a concept that stems from the nature and the human society [5]. It has been observed that self-organization is the basic element of living organisms and open social systems and is a phenomenon widely existing in complex adaptive systems, including living systems and socio-economic/technical, and human organizational/social systems, and so forth. As such, self-organization is studied intensively in biology, sociology, management science and organizational theory.

While self-organization has been extensively studied in terms of complex adaptive systems, multi-agent systems, swarm intelligence, many mechanisms and basic principles of self-organization are still remaining unclear. This has stimulated intense interests and it is even more challenging to devise and build self-organization mechanisms to solve complex problems, which is broadly about how to engineer artificial self-organizing systems.

Conventional studies on self-organization are more a behavioral perspective, e.g., what characteristics (e.g., emergence, persistence, etc.) self-organization has, how self-organization behaves, how self-organization should be observed. Correspondingly, conventional studies on self-organization are more about the mechanisms and principles in existing complex adaptive systems, e.g., swarm intelligence.

Differently, our study on self-organization essentially follows a structural perspective, which is directly about how a self-organizing system can be constructed and how a self-organizing process works.

Specifically, we define self-organization of a system as the run-time adjustment of the system itself to any organizational attribute of the system. Essentially, self-organization of a complex system is the process in which feedback is taken advantage of via the interactions between the system components. Here feedback and control play a pivotal role. In fact, self-

organization of complex systems can be formulated as run-time feedback control loops between the system components. Such a formulation requires analysis of the re-configurability and manageability of system organization (system components and interactions) and analysis of the run-time decision-making on re-configuration of the system.

If a system is very general, its organizational attributes will be numerous, and accordingly the ways of adjusting organizational attributes of the system will be various, too. Since self-organization is closely related to complex adaptive systems, a complex adaptive system can, without loss of generality, be viewed as a multi-agent system, with the different components of the system as semi-autonomous agents which interact with one another within the system. In the context of multi-agent systems, the ways of adjusting organizational attributes of the system can be largely restricted. Furthermore, if we further restrict the process of self-organization as a recursion of self-similar entities, i.e., holons, the ways of adjusting organizational attributes of the system can be even more dramatically reduced. This is so-called holonic self-organization. Since a finite number of recursion of holonic processes will be able to approximate a wide variety of complex system properties, holonic self-organization is no loss of generality.

Our definition of self-organization fits very naturally to the engineering of artificial self-organizing systems.

#### 4.2 Feedback-control architectures of holonic self-organization of multi-agent systems

In human society, the process for the design of human organization can be understood through its feedback-control framework, as illustrated in Figure 3.

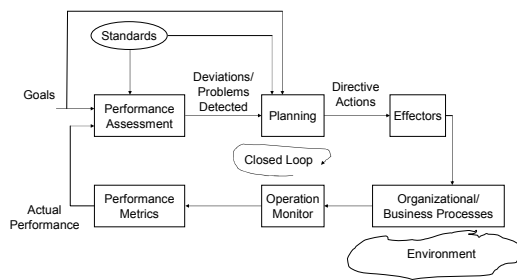


Figure 3 Feedback-control framework of the design process of human organization

If the feedback-control framework of the design process of human organizations can be translated into an automated variant for artificial organizational systems, that will straightly entail a scheme of self-organization.

Organizational concepts have been adapted for the design of multi-agent systems in the so-called organization based design methodology. For example, Zambonelli et al. suggest to analyze firstly the requirements an organization should meet, then to define which organizational form best fits the requirements. Finally, the designer should constrain the self-interested agents by creating a formal structure (i.e., create a control regime by designing a role model, specify organizational roles, assign agents to these roles) [6]. The multi-agent systems will be accepted as self-organizing if the process of the organization based design methodology can fully be automated.

Organization of multi-agent systems has a wide range of attributes:

- Constituent entities
- Roles played by the entities, so as the whole organization to fulfill a purpose (functionalities)
- Relationships, static and run-time, between roles
  - Organizational structure: hierarchical, flatted, nested, etc.
  - Bindings via agreement / commitment, joint constraints
- Interaction formats:
  - Control: centralized, decentralized, etc.
  - Channel of feedback (report)

Therefore, self-organization of a multi-agent system can be defined as run-time adjustment of the system itself to any organizational attribute of the multi-agent system.

The use of a priori pre-defined organizational forms of multi-agent system is important for classifying methods for self-organization of multi-agent system. In general, the methods for self-organization of multi-agent systems are not beyond two types. Firstly, there are no multi-agent systems organizational forms that are known a priori. Secondly, there are a finite number of a priori pre-defined organizational forms of multi-agent systems. For the latter, there is actually a repository of policies which are pairs of System-performance Patterns and the corresponding proper organizational forms as follows, where a measurement of system performance embodies the changes in the environment.

System-performance Pattern 1	→	Organizational form 1
System-performance Pattern 2	→	Organizational form 2
.....		
System-performance Pattern N	→	Organizational form N

The feedback-control formworks for both types of self-organization of multi-agent systems can be put forward as depicted in Figures 4 and 5, respectively.

The first method for self-organization of multi-agent systems is too generic and hard to realities. The second method for self-organization of multi-agent systems works well within limited domains over predictable changes.

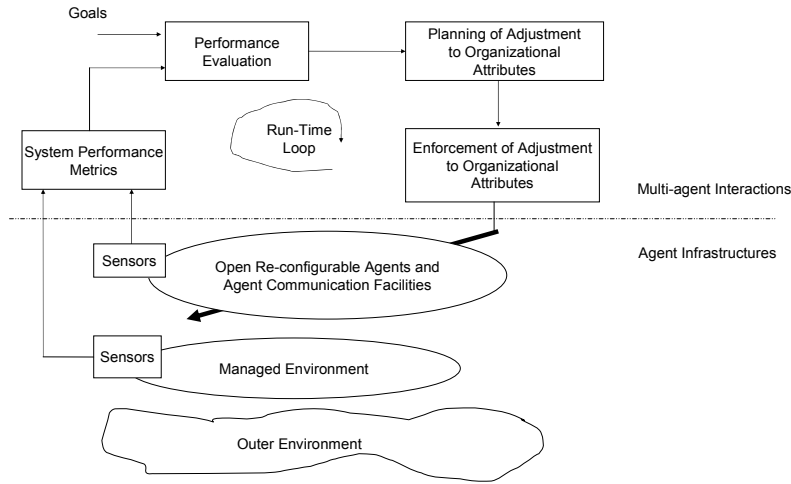


Figure 4 Feedback-control framework of generic self-organization of multi-agent systems: with no a priori pre-defined organizational forms

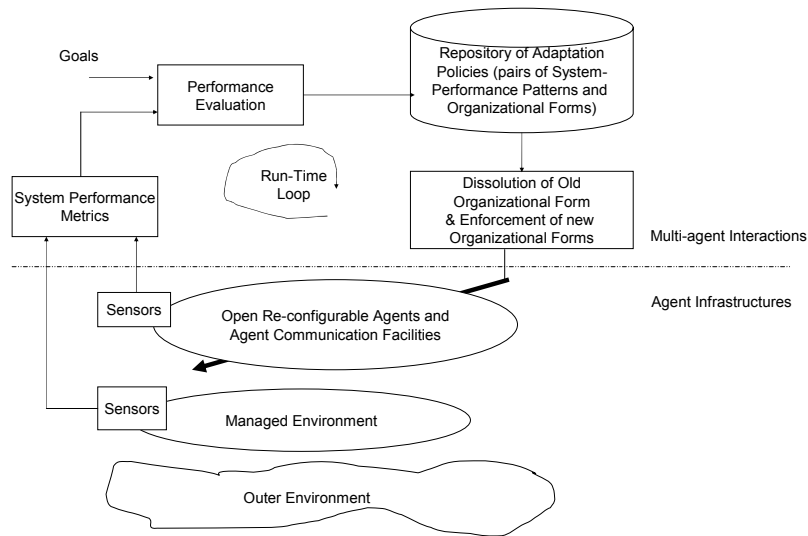


Figure 5 Feedback-control framework of discretized self-organization of multi-agent systems: With a finite number of a priori pre-defined organizational forms

The word “holon” is a compound from the Greek holos, which means the whole, with the suffix -on, which, as in proton or neutron, suggests a particle or part. It was coined by Koestler [7] to describe a basic unit of an organization in social and biological systems. Holons model recursive and self-similar structures in biological and sociological entities.

Holarchies can be modeled by using whole-part relationships. It is managed in a distributed manner by

system elements or holons. Individual holons define their activities based on their local knowledge, and decide their behavior by negotiation. Accordingly, a holarchy is a hierarchy of holons, which function (a) as autonomous wholes in supra-ordination to their parts, (b) as dependent parts in sub-ordination to controls on higher levels, (c) in co-ordination with their local environment (HMS Consortium <http://hms.ifw.uni->

hannover.de/). A generic holarchy is illustrated in Figure 6.

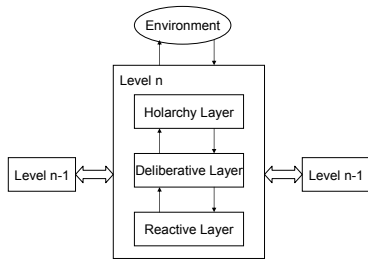


Figure 6 A Generic Holarchy

Holonic multi-agent system is an effective way for establishing self-organization of multi-agent systems. By means of holons, the design problem of

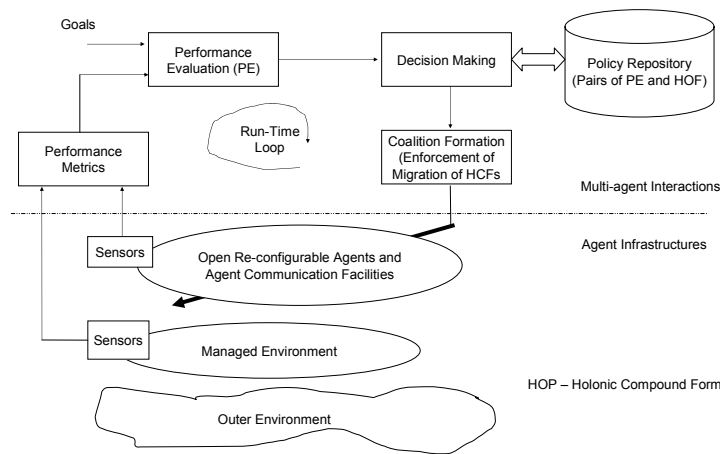


Figure 7 Generic architecture of holonic self-organization of multi-agent systems

Holonic method for self-organization of multi-agent systems is one instantiation of the second method (see Figure 5), but since specific holons can be formed infinitely according to the finite number of a priori specified holonic compound forms, holonic method actually lies between the two methods in that it is more pursuable, by the recursion of holons, and can work over unpredictable circumstances.

A feedback-control framework can be put forward for holonic self-organization of multi-agent systems, as depicted in Figure 8. The proposed architecture involves runtime dynamically self-structuring and self-optimizing. The self-structuring is basically realized through runtime holonic organization.

In conventional holonic multi-agent systems, only holonification operations are described and analyzed, but hierarchical control is not well concerned. We can envision in the self-organizing architecture to be proposed, there are runtime recognition of the situations (changes in the environment and the system

organization forms at run time, which is an infinite problem, is transformed as the decision and migration over the holonic compound forms, which are specified a priori and are finite in number.

The decision and migration over holonic compound forms at run time involves:

- How to decide/optimize --> run-time search over the policy repository
- what is based on to decide -->
  - attributes of organization forms and
  - their run-time performances, which involving performance measurements

A generic architecture can be put forward for the holonic self-organization of multi-agent systems, as depicted in Figure 7.

goals), which are to be translated to holonic roles, runtime holonification of organizational patterns, which are assumed by holonic agents, and what's important is that these two hierarchies are linked at runtime (fed back) to form hierarchical closed loops of self-organizing, by which hierarchical control of multi-agent systems is realized.

## 5. Summary

The essence of self-managing computing is the runtime automation of system adaptations for computing systems. Majority of the current works in autonomic computing is establishing control loops on top of basic computing processes to bring forth passive adaptation for the computing systems. The real challenge for self-managing and self-organizing computing is with open complex distributed computing systems.

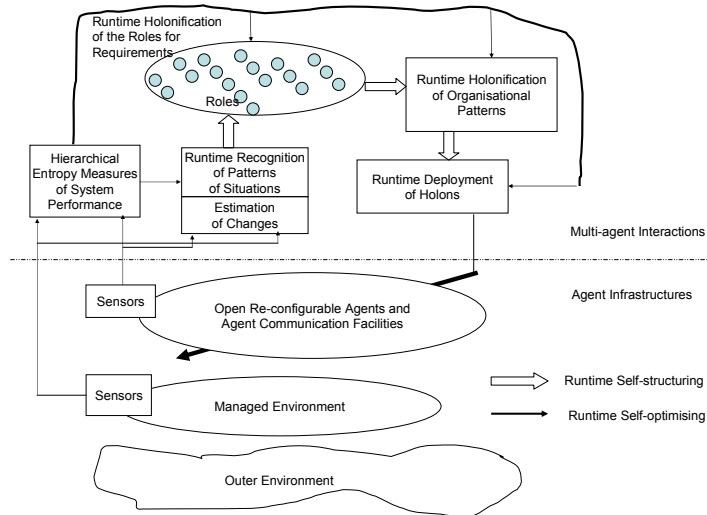


Figure 8 Feedback-control framework for holonic self-organization of multi-agent systems

While self-managing and self-organizing computing is a quite new revolutionary move in the community of computing and so far a holistic solution to self-managing and self-organizing computing has not yet appeared, some important technologies, e.g., particularly in the fields of autonomous robotics, cognitive computing, feedback control, adaptive control, intelligent systems, hierarchical control, etc., are already available for being integrated into the framework of self-managing and self-organizing computing.

The specific contributions of this paper are as follows.

- A new definition has been presented, from a structural perspective, for self-organization;
- A feedback-control framework has been put forward, for the first time, for self-organization of multi-agent systems for the methods both with no and with a priori organizational forms;
- A feedback-control framework has been put forward, for the first time, for holonic self-organization of multi-agent systems;

Apparently these works are still preliminary and need further consolidation.

Self-managing and self-organizing computing calls on interdisciplinary collaboration between feedback control, adaptive control, and computer science, etc., and especially a unified approach to self-managing and self-organizing computing including cognitive computing. However, it may be too simplistic and misleading to advocate self-managing and self-organizing computing system as feedback control of computing processes. In fact a number of intrinsic differences exist between computing systems and

conventional feedback control systems. For instance, two major differences are as follows.

(a) Conventional feedback control systems are more concerned with physical and/or engineering systems and the system dynamics normally follows physical laws, but computing processes are artificial systems, or processes of organizing hardware and software resources to solve problems, which are mostly not governed by physical laws but by human and/or social prescriptions and definitions.

(b) Conventional feedback control systems are more concerned with continuous variables, but computing systems normally evolve in terms of discrete events/states. As a consequence, models in conventional feedback control systems are rarely applicable directly to self-managing and self-organizing computing, and self-managing and self-organizing computing should take more advantage of the frameworks and methodologies of feedback control and adaptive control rather than the specific models and algorithmic procedures of feedback control systems. This of course requires tremendous efforts to properly understand and model the computing processes themselves, especially from a feedback-control point of view.

## Reference

- [1] Kephart, J. O. and D M Chess, The vision of autonomic computing. *IEEE Computer*, 2003, pp. 41-50.
- [2] Ganek, A. G. and T. A. Corbi, The dawning of the autonomic computing era. *IBM Systems Journal*, Vol. 42, No. 1, 2003, pp. 5-18.
- [3] Tianfield, H and R Unland, Towards autonomic computing systems. *Engineering Applications of*

- Artificial Intelligence, Vol. 17, No. 7, October 2004, pp. 689-699.
- [4] Unland R and H Tianfield, Shaping the future: Towards the next generation of powerful, pervasive, and service-oriented distributed systems. Multiagent and Grid Systems – An International Journal, Vol. 1, No. 3, December 2005, pp. 225-229
- [5] Tianfield H and R Unland, Towards self-organization in multi-agent systems and Grid computing. Multiagent and Grid Systems – An International Journal, Vol. 1, No. 2, October 2005, pp. 89-95.
- [6] Zambonelli, F., N. R. Jennings and M. Wooldridge, Developing Multiagent Systems: The Gaia Methodology, ACM Transactions on Software Engineering Methodology, Vol. 12, No. 3, July 2003, pp. 317-370.
- [7] Koestler, A., The Ghost in the Machine, Hutchinson & Co, London, 1967.