Towards a Bio-inspired Architecture for Autonomic Network-on-Chip

Mohamed Bakhouya
Universite de Technologie de Belfort Montbeliard
90010 Belfort Cedex, France
mohamed.bakhouya@utbm.fr

ABSTRACT

In the past few years, research in the domain of network-on-chip has been concentrated on application-specific approaches. These approaches are design-time parameterized approaches and, however, do not consider runtime configuration of different network-on-chip parameters, which are hard to predict in early design stage. Network-on-chip architectures must be adaptive, for example, by rerouting at runtime traffic from congested area or by dynamically changing link's bandwidth. The objective is to efficiently use the ressources by dynamically reconfiguring the network-on-chip according to the user and/or system requirements. In this paper, a decentralized system inspired by the immune system for autonomic network-on-chip is introduced. The immune system has a useful set of organizing principles, such as self-configuration, self-optimization, and self-healing principles, that can guide the design of autonomic network-on-chip.

KEYWORDS: System-on-Chip, Autonomic NoC, Adaptive approaches, Bio-inspired techniques, Immune system.

1. INTRODUCTION

Network-on-chip (NoC) has been recently proposed for System-on-Chip (SoC) applications design. The NoC infrastructure is the combination of various elements (e.g., switches, links) and protocols (e.g., routing, switching) that determine the communication architecture and modes. In open environments, with the rising need for on-demand services, a high degree of self-management and automation is required in NoC infrastructures. NoC must be enhanced by adaptive capabilities such as self-configuration, self-optimization, and self-healing. For example, NoC elements must be able to change the traffic route at runtime in order to efficiently avoid the faulty areas or hotspots. Therefore, a autonomic NoC infrastructure that provides these self-* features is required.

Self-* features have been mainly studied to develop large and adaptive distributed systems [2, 20, 21, 29, 35]. Furthermore, this is in part the aim of autonomic computing, which was an IBM initiative to deal with IT servers complexity [14] by developing systems that mange themselves. As described in [17], autonomic computing focuses on creating computer systems that manage themselves according to an administrators goals. Autonomic systems should be designed with adaptive capabilities, which are Self-configuration, Self-healing, Self-optimization, and self-Protection, to deal with an unpredictable environment such as user's behavior or system changes.

Self-* capabilities have been seen in natural and biological systems and have inspired many researches to develop autonomic systems. In other words, biological and natural systems have been exploited in a variety of computationally systems and been perceived as an efficient system model for developing autonomic systems [2, 4, 6, 20] and reconfigurable/evolvable hardware [25, 28, 32] with self-* capabilities.

In this paper, BNoC, a Bio-inspired architecture inspired by biological immune system is introduced for autonomic NOCs. Biological immune system could allow the design of autonomic NoCs with promising self-* capabilities. The objective is to implement these capabilities within the system to adapt to environmental changes and the dynamics of its computing elements. BNoC could react like an immune system against pathogens that have entered the body. It detects the infection (i.e., applications behavior or system state changes) and delivers a response to eliminate it (i.e., adapt to changes). The aim of this paper is to highlight how the principles of biological immune system can be incorporated into the design of autonomic middleware for NoCs.

The remainder of this paper is structured as follows. In section 2, we present approaches mainly proposed for autonomic NoC architectures. Section 3 presents an overview of biological immune system. In section 4, we present a bio-inspired architecture for autonomic NoCs. Section 5 presents some preliminary simulation results. Conclusions and perspectives are given in section 6.

2. RELATED WORK

Several approaches have been proposed for autonomic NoCs and can be classified into two main categories, adaptable approches and adaptive approches. Adaptable approches are generally tailored an application domain or a specific application by providing an application-specific NoC. All parameters, such as the on-chip interconnect architecture (i.e., topology), routing and switching schemes are defined at design time [23]. Adaptive approaches, however, provide techniques that allow NoCs to autonomously modify its structure and their behavior during the course of their operation (i.e., in runtime). Recently, there has been a great deal of interest in the development of adaptive approches for autonomic NoC.

Adaptive approaches can be classified based on level in which the adaptivity is done, at the application-layer by, for example, re-mapping the application, at the communication-layer by, for example, dynamically switching between packet and circuit switching scheme, or/and at architecture-layer by, for example, changing links bandwidth. At the infrastructure layer, the performance and the efficiency of the NoC are highly depend on the On-chip interconnect. Switches constitue the active component that influences on the latency and the throughput. For example, designing efficient switches with minimum buffer size represents a critical issue for the success of the NoC design [3]. Links are components that transport data between switches, efficient allocation of their capacity, decreasing their length or increasing their number may increase the performance of SoC applications.

On-chip interconnect configuration, bandwidth allocation, and buffer minimization are three main issue that should be addressed when designing and carrying out autonomic NoC. For example, a NoC architecture, called ReNoC (Reconfigurable NoC), is proposed in [27] to enable the network topology to be configured by the application running on the SoC architecture. More precisely, the topology can be customized for the application that is currently running on the chip, including long links and direct links between IP cores. The NoC architecture viewed by the application as a logical topology built on top of the real physical architecture. To create this application-specific topologies, ReNoC combines packet-switching and circuit-switching in the same topology switch. This makes it possible to create application-specific topologies in a general NoC-based SoC platform. The evaluation studies showed that the power consumption was decreased by 56% when configuring an application specific topology, compared to the static 2D mesh topology.

An approach, called 2X-Links, is proposed in [11] to allow links to change at runtime its supported bandwidth on-demand. Links capacity are not design-time parameterized but can adapt at runtime depending on the current traffic. More precisely, a runtime observability infrastructure analyzes the communication architecture and self-adapts depending on the monitoring traffic on when and how a certain switch should be configured for a certain transaction. This approach was evaluated using real-time multi-media and the E3S application benchmark suits and reported results show that 2X-Links provides a higher throughput and high fault tolerance.

A technique, called Blind Packet Switching (BPS), has been proposed in [12]. In this technique, buffers of the switch ports are replaced by simple latches and every latch can store one flit. Therefore, buffer control logic has been removed. Furthermore, latches do not store the flits while packets are blocked because they work as repeaters by storing a flit during a clock cycle. The reported results showed that this technique improves the performance while reducing the area and power consumption. A flow control technique, called ViChaR, is proposed in [22] to improve buffering efficiency by dynamically adjusting the depth and number of virtual channels based on network traffic.

An approach was proposed in [1] to allow NoC to dynamically configure itself with respect to the switching modes with the changing communication requirements of the system at run time. The main objective of this approach is to provide low latency, low power, and high data throughput. This approach is decentralized because the decision to switch between packet and circuit switching scheme is made at router level. Therefore, the system continues performing its functions in the presence of node failure. Simulations were conducted and results were reported to show that this approach use resources efficiently compared with traditional approach wherein all decisions are made at design time.

An approach to switch between deterministic and adaptive routing based on the network's congestion conditions is proposed in [15]. This approach is based on Odd-Even routing algorithm [7] and combines the advantages of these two techniques by proposing a technique called DyAD, from Dynamically switching between Adaptive and Deterministic modes. A routing approach, called dynamic XY (DyXY) routing, is proposed in [19] to avoid packets to be forwarded to congested switches. By monitoring the congestion status in the proximity, routing decisions made efficiently limiting a packet to traverse the network only following one of the shortest paths between the source and the destination. Analytical models based on queuing theory and simulations are conducted to compare DyXY with static routing and odd-even routing algorithm for a 2D Mesh architecture. Reported results showed that DyXY outperforms these algorithms by achieving better performance in termes of latency.

A dynamically-allocated VC approach with congestion awareness is proposed in [18]. This approach extends deep VCs at low traffic rate in order to reduce packet latency, while it increases VC number and avoids congestion situations to improve throughput in high rate. Simulations are conducted and results showed that at different injection rates or traffic patterns this dynamic allocation approach provides 8.3% throughput increase and 19.6% latency decrease on average.

A hierarchical agent-monitored network-on-chips was proposed in [24] to provide diagnostic services to the system against failures or errors. Based on the performance provided, the agents will decide on an optimal improvement mechanism, either replacing the poor functioning processing element, or adding more processing elements, or speeding up the circuits. In this approach, the agents are imitating the adaptive behaviors observed in biological systems. They autonomously adjusted the system performance at their own level. Higher level agents in their turn are supervising over lower level agents. The objective of their work is towards developing an autonomic NoC with self-* capabilities into component level and strengthen self-design and fault-tolerance aspects.

3. AN OVERVIEW OF IMMUNE SYSTEM

In this section, we describe the structure of the immune system from a biological perspective and introduce the innate and adaptive immune systems. The immune system defends the body against harmful diseases and infections [4, 26]. Once pathogens enter the body, they will be handled by two subsystems, the innate and the adaptive immune system, which work together to achieve this task. The innate and adaptive immune system are produced primarily by leukocyte cells. Among the several different types of leukocytes, there are phagocyte and lymphocyte cells. The phagocyte cells are the first line of defense for innate immune system. They engulfs the pathogen and present it to adaptive immune system. More precisely, the innate immune system hosts defence in the early stages of infection through nonspecific recognition of a pathogen and inhibits the adaptive immune response by activating T-Cells.

The adaptive immune system primarily consists of lymphocytes that circulate through the body in the blood and lymph system. There are two categories of lymphocytes, the T-cells and B-cells. The role of T-cells is to potentiate the immune response by the secretion of specialized factors that activate other cells to fight off infection. The major function of B-cell is the production of antibodies in response to foreign antigen. The main characteristics of adaptive immunity are specific recognition of pathogen leading to the generation of pathogen specific response.

After producing the immune response, a feedback is introduced to the tissue to induce healing and to the immune system itself to modify the structure and future behavior of both its adaptive and innate arms [8]. Therefore, the immune response is formulated a posteriori by the immune system in response to the cumulative experience of the immune system in dealing with the body (the self) and with the world (the foreign).

Another mechanism proposed by Jerne in [16], called immune Network theory, in which B-cells are not isolated but form an idiotypic network for antigen recognition. These cells both stimulate and suppress each other in certain ways that lead to the stabilization of the network [9]. Two B-cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share. In this model, T-cells help is never a limiting factor for B-cell proliferation or production of antibodies.

Another model was proposed in [34], called second generation immune system in which the activation of B-cells is explicitly dependent on co-operation with activated T-cells. The production of the antibodies by the B-cells mediates idiotypic interactions between them, and control their induction. In fact, circulating antibodies are the only inhibitory influence on T-cells activation and growth. A bounded dynamics of the T-cell activity can be achieved if and only if their receptors are integrated into the idiotypic network.

The biological immune system has several features and organizing principles that can be exploited to develop autonomic systems [13]. The immune system can be seen as a massively parallel architecture with a diverse set of cells and organs, which are distributed throughout the body but can communicate using chemical signals. There is no central control, the multitude of independent cells work together resulting in the emergent behavior of the immune system. The immune system is multilayered and each layer operates in concert with all the other components to provide defense-in-depth. The immune system evolves to adapte and improve the overall system performance.

Immunological principles and functionalities from computational viewpoint have been applied in many domains [9, 13, 36, 37]. For example, in [33, 36], an interesting model inspired by the natural immune system to monitor resources and services in distributed networks was proposed. The author has suggested that an intelligent network with self-organizing and emergence capabilities can react like the natural immune system against pathogens that have entered the body. The mapping is the following, pathogens (i.e., virus) correspond to user requests and immune responses correspond to request resolutions. More precisely, each user request is considered as an attack launched against the global network. The intelligent middleware detects the infection (i.e., user request) and delivers a response to eliminate it (i.e., satisfy the user request).

4. BNoC ARCHITECTURE

By analogy to the biological immune system (BIS), BNoC can be seen as a logical system hierarchy of layers, system layer, communication layer, and PE layer. BNoC can be viewed as an immune system in which the OCI corresponds to the lymph and blood network, immune cells correspond to software components or agents, body cells correspond to NoC components (i.e., PEs). As in BIS, BNoC operations imply a suite of functions performed by a federation of interacting components or agents. Structural features of the application to be maintained are represented as antigens to be cleared, while its behavioral features are represented as signals. The immune response against the antigen corresponds to the techniques selected to adapt to the behavioral/structure of the application. The mapping between these functions and organizing principles can be seen in Tab. 1.

Immune System	**BNoC**
Tissue	NoC
Body cells	PEs
Lymph/blood streams	OCI
Immune cells	Components/agents
Antigens	Tasks/system status
Immune response	Adapt to changes

Table 1. The Mapping between Immune System and BNoC

At each level, monitoring and supervising mechanisms are required to inhibit the corresponding self-* feature. Using the micronetwork stack model, the self-* capabilities can be defined as follows (see Fig. 1). This allows higher level components to allocate and supervise tasks of lower level agents. These lower level agents monitor the local PEs and adapt to the task's behavior. They also report the changes made at this level to the higher level agents.

Alike immune system, three type of agents are considered, P-agents, T-agents, and B-agents. At application layer, innate immune components must be *self-configuring*. Alike phagocyte cells that engulf the pathogen into antigens and present them to adaptive immune system (T-Cells), the P-agents analyze the application requirements, decompose it into tasks, and map them by presenting them to corresponding T-agents. Furthermore, the P-agents, should dynamically adapt to changes in the environment such dramatic changes in the system characteristics or user behavior by deploying new tasks or the removal of existing ones. This dynamic adaptation helps ensuring continuous operation of the application under unknown circumstances.

BNOC	Self-* Capabilities	Immune system
Application layer - Mapping	Self-configuration - Adapt to changes	Innate layer - Engulf pathogens - Activate T-Cells
Communication layer - Routing - Switching - Flow control	Self-healing - Act to the disruption	Adaptive layer (T-Cells) - Learn Ag structure - Activate B-Cells
Architecture layer - Buffer - Bandwidth - Link	Self-optimizing - Tune resources	Adaptive layer (B-Cells) - Learn Ag behaviour - Produce antibodies

Figure 1. The Micronetwork Stack Functions with Self-* Capabilities

At the communication layer, similar to T-cells, located in Thymus, to coordinate and regulate B-cells activation, T-agents are regional monitor to allocate and schedule the work of B-agents by selecting required routing/controlfolw technique and configuring the network within the region. T-agents have the *self-healing* capability, which allows them to discover, diagnose and react to disruptions such as deadlock. This can occur, for example, when a group of flits are blocked and cannot make progress because the flits are waiting for one another to release buffers or channels. In these situations, T-agents monitor and transmit metrics of congestion throughout to the other T-agents to be aware of network hotspots, and therefore avoid them by selecting another route. For example, based on the changing communication requirements of the system, T-agents can dynamically switching between packet and circuit switching scheme.

At the architecture layer, similar to B-cells that are local monitors and change their behavior to adapt to the antigen behavior, B-agents implement various techniques for resources tuning such as buffer allocation and bandwidth adaptation. In other words, B-agents have the *Self-optimizing* capability that allow them to automatically monitor and tune resources with the objective to increase the performance while lowering area and energy consumption. For example, based on the predicted traffic, the number of the active virtual channels can be adjusted to reduce in the buffer power consumption.

5. SIMULATION RESULTS

In this section, we present the simulation results using a specified target application. We analyze particularly the delay bound experienced by data flows sent by the PEs and the buffer size because it constitutes a dominant part of the NoC area as pointed out in [3, 5, 10]. In the simulation, the application is represented as communicating parallel processes already mapped into PEs. Each process is linked with a traffic generator that injects flits according to the CBR methods [30, 31].

The behavior changes of the application is modeled by increasing or decreasing injection rate. The *self-healing* mechanism is developed to allow for some sources/sinks, represented by B-agents, to switch from packet-switching to circuit switching in order to adapt to the application behavior. By monitoring and controlling dataflow on a feedback basis of B-agents which monitor switches queues, if a buffer is about to overflow (greater that a fixed threshold), then T-agents inform the B-agent source and B-agent sink of this traffic to establish, for example, a circuit to avoid the hotspot, similar to affinity link or relationship in immune system.

To illustrate the benefit of the establishment of affinity links, i.e., using packet and circuit switching techniques in same configuration, we consider the WK on-chip interconnect as a case study (see Fig. 2). The cores $(c_6, c_8, c_{11}, c_{15})$ are randomly selected to be sources and the cores $(c_1, c_5, c_{10}, c_{11}, c_{12})$ considered as sinks are selected according to the following communication locality principle in which 25% of the traffic takes place between neighboring cores and 75% of the traffic is uniformly distributed among the rest.

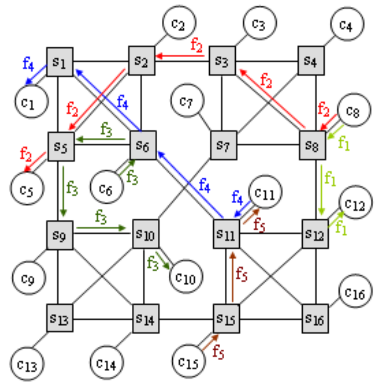

Figure 2. WK On-chip Interconnect: c_i, s_i, and f_i are Cores, Switches, and Data Flows

Fig. 3 illustrates the average buffer size required for zero flits drop under different injection rates. As the injection rate increases the buffer size increases because the network becomes more congested with heavy traffic and therefore more space is required to store flits. With only packet switching (PS), a higher buffer size is required compared with merging both packet and circuit switching (CS/PS). When adding a circuit between s_6 and s_{10} the traffic is routed through this circuit and the buffer size of the switch s_5 and s_{10} is decreased.

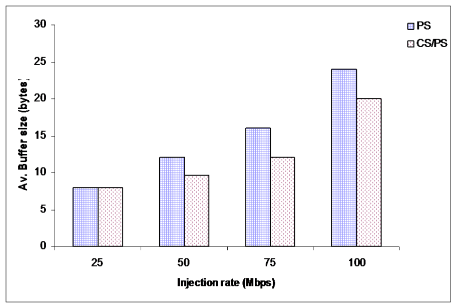

Figure 3. Average Buffer Size before (PS) and after Adaptation (CS/PS)

Fig. 4 shows the average end-to-end before and after adding a circuit. In this figure, the average delay increases linearly as the injection rate increases. When increasing the injection rate, the network becomes more congested with heavy traffic. Buffers become full causing more flits to wait and so the average latency increases. In this figure, PS shows a high latency compared to the PS/CS because adding a circuit between s_5 and s_9 decreases the number of hops for the traffic flow f_3 and so the average delay decreases.

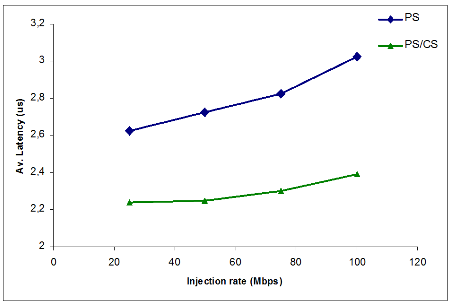

Figure 4. Average Delay before (PS) and after Adaptation (CS/PS)

6. CONCLUSIONS AND PERSPECTIVES

This paper gave an overview of state-of-the-art regarding the autonomic NoC approaches. Approaches proposed in the literature are classified as adaptable or adaptive. In adaptable approaches, all parameters/protocols are optimized/selected at design-time targeting a specific application. In adaptive approaches, the adaptation is undertaken while the system is in operation. In another words, the adaptation process is running continuously to evolve the system and make adaptive to changes in the environment. A BNOC middleware inspired by immune system was also introduced to support the adaptive capabilities, which are self-configuration, self-optimization, and self-healing. These adaptive capabilities can provide the core scheme to develop autonomic NoC at the three levels of the micronetwork stack model. The objective of this paper is to show that how to incorporate innate and adaptive immunity into the design of autonomic NoC. Some preliminary results are presented to show the benefits of using adaptive approaches, but more investigations are required and constitue our ongoing work.

REFERENCES

[1] B. Ahmad and A. T. Erdogan and S. Khawam, "Architecture of a dynamically reconfigurable NoC for adaptive reconfigurable MPSoC," Proceedings of First NASA/ESA Conference on Adaptive Hardware and Systems (AHS06), pp. 405-411, 2006.

[2] O. Babaoglu and G. Canright and A. Deutsch and G. A. D. Caro and F. Ducatelle and L. M. Gambardella and N. Ganguly and M. Jelasity and R. Montemanni and A. Montresor and T. Urnes, "Design patterns from biology for distributed computing," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 1, No. 1, pp. 26-66, 2006.

[3] M. Bakhouya, "Evaluating the energy consumption and the silicon area of on-chip interconnect architectures, *Journal of Systems Architecture*, Vol. 55, No. 7-9, pp. 387-395, 2009.

[4] M. Bakhouya and J. Gaber, "Self-organizing approach for emergent multi-agent structures," Workshop on Complexity through Development and Self-Organizing Representations (CODESOAR'06) at GECCO'06 Genetic and Evolutionary Computation Conference, 2006.

[5] A. Balkan and G. Qu and U. Vishkin, "A mesh-of-trees interconnection network for single-chip parallel processing," ASAP Proc., 2006.

[6] M. Bakhouya and J. Gaber, "Approaches for Ubiquitous Computing," Wireless Ad hoc and Sensor networks Edited by: H. Labiod,ISTE Publishing Knowledge/John Wiley and Sons Inc, pp. 111-142, 2008.

[7] G.-M. Chiu. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 7, pp. 729-738, 2000.

[8] I. R. Cohen, "Real and artificial immune systems: computing the state of the body," *Nature Reviews, Immunology*, Vol. 7, pp. 569-574, 2007.

[9] D. Dasgupta and Z. Ji and Gonzalez, "Artificial immune systems (AIS) research in the last five years," Proceedings of Congress on Evolutionary Computation (CEC), pp. 528-535, 2003.

[10] M. Bakhouya and S. Suboh and J. Gaber and T. El-Ghazawi, "Analytical modeling and evaluation of on-chip interconnects using network calculus," NoCS Proc., pp. 7479, 2009.

[11] M. A. A. Faruque and T. Ebi and J. Henkel, "Configurable links for runtime adaptive on-chip communication," DATE09: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 543-548, 2009.

[12] C. Gomez and M. E. Gomez and P. Lopez and J. Duato, "Reducing packet dropping in a bufferless NoC," Euro-Par 2008 Parallel Processing, Vol. 5168, pp. 899-909, 2008.

[13] P. K. Harmer and P. D. Williams and G. H. Gunsch and G. B. Lamont, "An artificial immune system architecture for computer security applications," *IEEE Trans. on Evolutionary Computation*, Vol. 6, No. 3, pp. 252-280, 2002.

[14] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," IBM Corporation, [website], Available www-1.ibm.com/industries/government/doc/content/bin/auto.pdf, 2001.

[15] J. Hu and R. Marculescu, "DyAD: smart routing for networks on-chip," AC04: Proceedings of the 41st Conference on Design Automation, pp. 260-263, 2004.

[16] N. K. Jerne, "Towards a network theory of the immune system," *Ann. Immunol. (Inst.Pasteur) 125C*, pp. 373-389, 1974.

[17] J. O. Kephart and D. M. Chess, ”The vision of autonomic computing,” *Computer*, Vol. 36, No. 1, pp. 4150, 2003.

[18] M. Lai and Z. Wang and L. Gao and H. Lu and and K. Dai, ”A dynamically-allocated virtual channel architecture with congestion awareness for on-chip routers,” DAC 2008, pp. 630-633, 2008.

[19] Q. Z. M. Li and W. Jone, ”DyXY-a proximity congestion-aware deadlock-free dynamic routing method for network on chip,” In proc DAC’06, pp. 849-859, 2006.

[20] N. Minar and M. Gray and O. Roup and R. Krikorian and P. Maes, ”Hive: Distributed agents for networking things,” Proceedings of First ISASA/MA, 1999.

[21] M. Moore and T. Suda, ”A decentralized and self-organizing discovery mechanism,” Proc. of the First Annual Symposium on Autonomous Intelligent Networks and Systems, May 2002.

[22] C. A. Nicopoulos and D. Park and J. Kim and N. Vijaykrishnan and M. S. Yousif and C. R. Das, ”Vichar: A dynamic virtual channel regulator for network-on-chip routers,” 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO’06), pp. 333-346, 2006.

[23] U. Y. Ogras and R. Marculescu and H. G. Lee and N. Chang, ”Communication architecture optimization: Making the shortest path shorter in regular networks-on-chip,” DATE Proc., 2006.

[24] P. Rantala ane J. Isoaho and H. Tenhunen ”Agent-based reconfigurability for fault-tolerance in network-on-chip,” Proceedings of International Conference on Engineering of Reconfigurable Systems and Algorithms, 2007.

[25] L. Sekanina, ”Nanostructures and bio-inspired computer enegineering,” In Proceedings of Nano02, (ISBN 80-7329-027-8), pp. 233-236, 2002.

[26] A. Somayaji and S. Hofmeyr and S. Forrest, ”Principles of a computer immune system,” Proceedings of the Second New Security Paradigms Workshop, pp. 75-82, 1997.

[27] M. B. Stensgaard and J. Sparso, ”Renoc: a network-on-chip architecture with reconfigurable topology,” Second ACM/IEEE International Symposium on Networks-on-Chip, 2008.

[28] C. Teuscher, ”On the state of the art of poetic machines,” available at http://citeseer.ist.psu.edu/teuscher01state.htm.

[29] M. Bakhouya and J. Gaber, ”Adaptive approach for the regulation of a mobile agent population in a distributed network,” 5th International Symposium on Parallel and Distributed Computing, pp. 360-366, 2006.

[30] S. Suboh and M. Bakhouya and S. Lopez-Buedo and T. El-Ghazawi, ”Simulation-based approach for evaluating network-on-chip interconnect architectures,” SPL 2008 Proceedings. pp. 75-80, 2008.

[31] S. Suboh and M. Bakhouya and J. Gaber and T. El-Ghazawi ”An interconnection architecture for network-on-chip systems,” *Telecom. Sys. Journal*, Vol. 37, No. 1-3, pp. 137-144, 2008.

[32] A. Upegui and Y. Thoma and E. Sanchez and A. Perez-Uribe and J. Moreno and J. Madrenas, ”The perplexus bio-inspired reconfigurable circuit,” Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 600-605, 2007.

[33] J. Gaber, ”New paradigms for ubiquitous and pervasive applications,”Research Report RR-09, Universite de Technologies de Belfort-Montbeliard (UTBM), France, 2000.

[34] F. J. Varela and A. Coutinho, ”Second generation immune networks,” *Immunology Today*, Vol. 12, No. 5, pp. 159-166, 1991.

[35] J. Gaber and M. Bakhouya, ”An affinity-driven clustering approach for service discovery and composition for pervasive computing,” International Conference on Pervasive Services ICPS’06, pp. 277-280, 2006.

[36] J. Gaber, ”Spontaneous emergence model for pervasive environments,” Proceeding of Globecom Workshops, IEEE, WDC, Nov. 2007.

[37] Y. Watanabe and A. Ishiguro and Y. Uchkawa, ”Decentralized behavior arbitration mechanism for autonomous mobile robot using immune system,” Books Artificial Immune Systems and Their Applications, ISBN 3-540-64390-7, 1999.