World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

Applying Autonomic Computing Concepts to Parallel Computing using Intelligent Agents

Blesson Varghese and Gerard T. McKee

International Science Index, Computer and Information Engineering Vol:3, No:7, 2009 waset.org/Publication/3751

***Abstract*—**The work reported in this paper is motivated by the fact that there is a need to apply autonomic computing concepts to parallel computing systems. Advancing on prior work based on intelligent cores [36], a swarm-array computing approach, this paper focuses on 'Intelligent agents' another swarm-array computing approach in which the task to be executed on a parallel computing core is considered as a swarm of autonomous agents. A task is carried to a computing core by carrier agents and is seamlessly transferred between cores in the event of a predicted failure, thereby achieving self-ware objectives of autonomic computing. The feasibility of the proposed swarm-array computing approach is validated on a multi-agent simulator.

***Keywords*—**Autonomic computing, Intelligent agents, Swarm-array computing.

I. Introduction

THE increase in complexity of distributed computing systems has led to concern about management and cost related issues. With the aim of building large scale systems, reducing cost of ownership and reallocating management responsibilities from administrators to the computing system itself [1] - [8], autonomic computing principles have paved necessary foundations towards self-managing systems. Self-managing systems are characterized by four objectives, namely self-configuration, self-healing, self-optimizing and self-protecting and four attributes, namely self-awareness, self-situated, self-monitoring and self-adjusting. [9] - [11].

Autonomic computing researchers have adopted six different approaches, namely emergence-based, component/service-based, control theoretic based, artificial intelligence, swarm intelligence and agent based approaches to achieve self-managing systems.

The emergence based approach for distributed systems considers complex behaviours of simple entities with simple behaviours without global knowledge [12]. Intelligent behaviour is thus repercussions of interactions and coordination between entities. One major challenge in emergence based approaches is on how to achieve global coherent behaviour [13]. Autonomic computing research on emergence based approaches is reported in [12] - [15].

Blesson Varghese is a research student with the Active Robotics Laboratory, School of Systems Engineering, University of Reading, Reading, Berkshire, United Kingdom, RG6 6AY.(e-mail: hx019035@ reading.ac.uk).

Gerard T. McKee is Senior Lecturer in Networked Robotics, School of Systems Engineering, University of Reading, Reading, Berkshire, United Kingdom, RG6 6AY. (e-mail:g.t.mckee@reading.ac.uk).

The component/service based approach for distributed systems employ service-oriented architectures. With advancements in software engineering practices, component/service based approaches are also implemented in many web based services. The autonomic element of the autonomic system is a component whose interfaces, behaviours and design patterns aim to achieve self-management. These approaches are being developed for large scale networked systems including grids. Autonomic computing research on component/service based approaches is reported in [16] - [19].

The control theoretic based approach aims to apply control theory for developing autonomic computing systems. The building blocks of control theory such as reference input, control input, control error, controller, disturbance input, measured output, noise input, target system and transducer are used to model computing systems and further used to study properties like stability, short settling times, and accurate regulation. Using a defined set of control theory methodologies, the objectives of a control system namely regulatory control, disturbance rejection and optimization can be achieved. These objectives are closely associated with the objectives of autonomic computing. Research on control theoretic based approaches applied to autonomic computing is reported in [20] - [22].

The artificial intelligence based approaches aim for automated decision making and the design of rational agents. The concept of autonomy is realized by maximizing an agent's objective based on perception and action in the agent's environment with the aid of information from sensors and in-built knowledge. Work on artificial intelligence approaches for autonomic computing is reported in [23] [24].

The swarm intelligence based approaches focuses on designing algorithms and distributed problem solving devices inspired by collective behaviour of swarm units that arise from local interactions with their environment [25] [26]. The algorithms considered are population-based stochastic methods executed on distributed processors. Autonomic computing research on swarm intelligence approaches is reported in [27] - [29].

The agent based approaches for distributed systems is a generic technique adopted to implement emergence, component/service, artificial intelligence or swarm intelligence based approaches. The agents act as autonomic elements or entities that perform distributed task. The domain of software engineering considers agents to facilitate autonomy and hence have a profound impact on achieving the objectives of autonomic computing. Research work based on multi-agents supporting autonomic computing are reported in [5] [30] - [35].

International Science Index, Computer and Information Engineering Vol:3, No:7, 2009 waset.org/Publication/3751

However, though all of the autonomic computing approaches above aims towards the objectives of autonomic computing, few researchers have applied autonomic computing concepts to parallel computing systems. This is surprising since most distributed computing systems are closely associated with the parallel computing paradigm. The benefits of autonomy in computing systems, namely reducing cost of ownership and reallocating management responsibilities to the system itself are also relevant to parallel computing systems.

How can a bridge be built between autonomic computing approaches and parallel computing system? The work reported in this paper is motivated towards bridging this gap by proposing swarm-array computing, a novel technique to achieve autonomy for distributed parallel computing systems.

Swarm-array computing is biologically inspired by the theory of autonomous agents in natural swarms, abstracted and implemented in swarm robotics. This technique considers the computational resource as a swarm of resources and the task to be executed as a swarm of sub-tasks. Hence, the approach considers complex interactions between swarms of sub-tasks and swarms of resources. The interactions between swarm agents bring about the notion of intelligent agents or swarm agents carrying the sub-tasks and intelligent cores or swarm of cores executing the sub-task. The interactions between different swarms give rise to the notion of landscapes. In other words, the approach can be viewed as a computational approach emerging from the interaction of multi-dimensional arrays of swarm agents.

In this paper, a swarm-array computing approach is proposed as a solution that aims to apply autonomic concepts to parallel computing systems and in effect achieve the objectives and attributes of self-managing systems. Unlike another swarm-array computing approach proposed earlier [36], in the second approach proposed in this paper, the task to be executed on parallel computing cores is considered to be a swarm of autonomous agents.

The remainder of the paper is organized as follows. Section II considers three approaches of swarm-array computing. The second approach is of focus in this paper. Section III investigates the feasibility of the proposed approach by simulations. Section IV concludes the paper and considers future work.

II. The Approach

Swarm-array computing is a swarm robotics inspired computing approach proposed as a path to achieve autonomy in parallel computing systems. The foundations of swarm-array computing are the existing computing paradigms of parallel and autonomic computing. There are three approaches based on intelligent cores, intelligent agents and swarm of intelligent cores and intelligent agents that bind the swarm-array computing constituents (the computing system, the problem/task, the swarms and the landscape) together. Each approach is briefly considered in the following subsections.

A. Intelligent Cores

In the first approach, the cores of a parallel computing system are considered to be intelligent. These intelligent cores are implemented as autonomous swarm agents and form the landscape representing the computing space. A parallel task to be executed resides within a queue and is scheduled onto the cores by a scheduler. The intelligent cores interact with each other to transfer tasks from one core to another at the event of a hardware failure.

The cores of a parallel computing system can be considered as a set of autonomous agents, interacting with each other and coordinating the execution of tasks. In this case, a processing core is similar to an organism whose function is to execute a task. The focus towards autonomy is laid on the parallel computing cores abstracted onto intelligent cores. The set of intelligent cores hence transform the parallel computing system into an intelligent swarm. The intelligent cores hence form a swarm-array. A parallel task to be executed resides within a queue and is scheduled onto different cores by the scheduler. The swarm of cores collectively executes the task.

The intelligent cores are an abstract view of the hardware cores. But then the question on what intelligence can be achieved on the set of cores needs to be addressed. Intelligence of the cores is achieved in two different ways. Firstly, by monitoring local neighbours. Independent of what the cores are executing, the cores can monitor each other. Each core can ask the question 'Are you alive' off its neighbours and gain information. Secondly, by adjusting to core failures. If a core fails, the process which was executed on the core needs to be moved to another core where resources previously accessed can be utilized. Once a process has been moved or relocated, all data dependencies need to be re-established.

To shift a process from one core to another, there is a requirement of storing data associated and state of the executing process, referred to as checkpointing [37]. This can be achieved by a process monitoring each core or by swarm carrier agents that can store the state of an executing process. The checkpointing method suggested is decentralized and distributed across the computing system. Hence, though a core failure may occur, a process can seamlessly be transferred onto another core. In effect, awareness and optimizing features of the self-ware properties are achieved.

B. Intelligent Agents

In the second approach of swarm-array computing, only the task to be executed is considered to be intelligent. The intelligent swarm agents are autonomous and form the landscape. A parallel task to be executed resides in a queue, which is mapped onto carrier swarm agents by the scheduler. The carrier swarm displace through the cores to find an appropriate area to cluster and execute the task. The intelligent agents interact with each other to achieve mobility and successful execution of a task. Figure 1 shows the approach diagrammatically.

The task to be executed on the parallel computing cores can be considered as a swarm of autonomous agents. To achieve this, a single task needs to be decomposed and the sub tasks need to be mapped onto swarm agents. The agent and the sub-problems are independent of each other; in other words, the swarm agents are only carriers of the sub-tasks or are a wrapper around the sub-tasks.

International Science Index, Computer and Information Engineering Vol:3, No:7, 2009 waset.org/Publication/3751

The swarm displaces itself across the parallel computing cores or the environment. The goal would be to find an area accessible to resources required for executing the sub tasks within the environment. In this case, a swarm agent is similar to an organism whose function is to execute on a core. The focus towards autonomy is laid on the executing task abstracted onto intelligent agents. The intelligent agents hence form a swarm-array.

The intelligent agents described above are an abstract view of the sub-tasks to be executed on the hardware cores. Intelligence of the carrier agents is demonstrated in two ways. Firstly, the capabilities of the carrier swarm agents to identify and move to the right location to execute a task. In this case, the agents need to be aware of their environments and which cores can execute the task. Secondly, the prediction of some type of core failures can be inferred by consistent monitoring of power consumption and heat dissipation of the cores. If the core on which a sub-task being executed is predicted to fail, then the carrier agents shift from one core to another gracefully without causing an interruption to execution, hence making the system more fault-tolerant and reliable. An agent can shift from one core to another by being aware of which cores in the nearest vicinity of the currently executing core are available.

C. Swarm of Intelligent Cores and Intelligent Agents

In the third approach of swarm-array computing, both the intelligent cores and intelligent agents are considered to form the landscape. Hence, the approach is called a combinative approach. A parallel task to be executed resides in a queue, which is mapped onto swarm agents by a scheduler. The swarm agents can shift through the landscape utilizing their own intelligence, or the swarm of cores could transfer tasks from core to core in the landscape. The landscape is affected by the mobility of intelligent agents on the cores and intelligent cores collectively executing a task by accommodating the intelligent agent.

A combination of the intelligent cores and intelligent swarm agents leads to intelligent swarms. The intelligent cores and intelligent agents form a multi-dimensional swarm-array. The arena in which the swarms interact is termed as a landscape.

The landscape is a representation of the arena of cores and agents that are interacting with each other in the parallel computing system. At any given instance, the landscape can define the current state of the computing system. Computing cores that have failed and are predicted to fail are holes in the environment and obstacles to be avoided by the swarms.

A landscape can be modelled from three different perspectives which is the basis for the swarm-array computing approaches discussed in the next section. Firstly, a landscape comprising dynamic cores (are autonomous) and static agents (are not autonomous) can be considered. In this case, the landscape is affected by the intelligent cores. Secondly, a landscape comprising of static cores and dynamic agents can be considered. In this case, the landscape is affected by the mobility of the intelligent agents. Thirdly, a landscape comprising of dynamic cores and dynamic agents can be considered. In this case, the landscape is affected by the intelligent cores and mobility of the carrier agents.

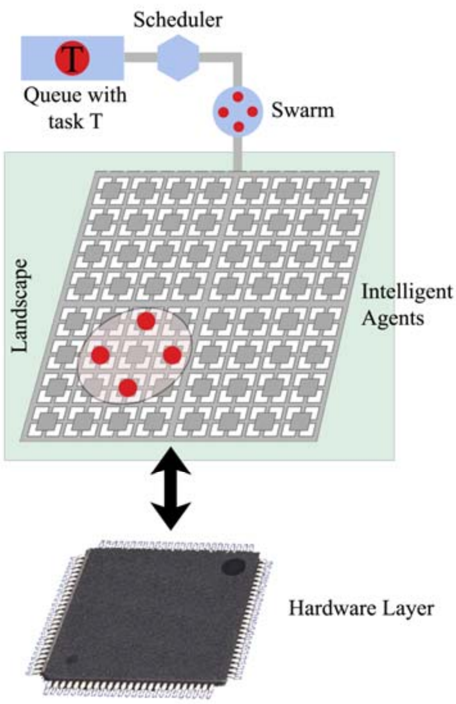

Fig. 1. Second Approach in swarm-array computing

In this paper, the focus is on the second approach based on intelligent agents. The feasibility and experimental studies based on the first approach is reported in [36]. The third approach will be reported elsewhere.

III. Experimental Studies

Simulation studies were pursued to validate and visualize the second proposed approach in swarm-array Computing. Since the approach proposed in this paper considers executing tasks as agents, a multi-agent simulator is employed. This section is organized into describing the experimental platform, experimental environment, experimental modelling and results.

A. Experimental Platform & Environment

The computing systems available for parallel computing are multi-core processors, clusters, grids, field programmable gate arrays (FPGA), general purpose graphics processing units (GPGPU), application-specific integrated circuit (ASIC) and vector processors. With the objective of exploring swarm-array computing, FPGAs are selected as an experimental platform for the proposed approaches.

FPGAs are a technology under investigation in which the cores of the computing system are not geographically distributed. The cores in close proximity can be configured to achieve a regular grid or a two dimensional lattice structure. Another reason of choice to look into FPGAs is its flexibility for implementing reconfigurable computing.

The feasibility of the proposed swarm-array computing approach was validated on the SeSAm (Shell for Simulated Agent Systems) simulator. The SeSAm simulator environment supports the modelling of complex agent-based models and their visualization [38] [39].

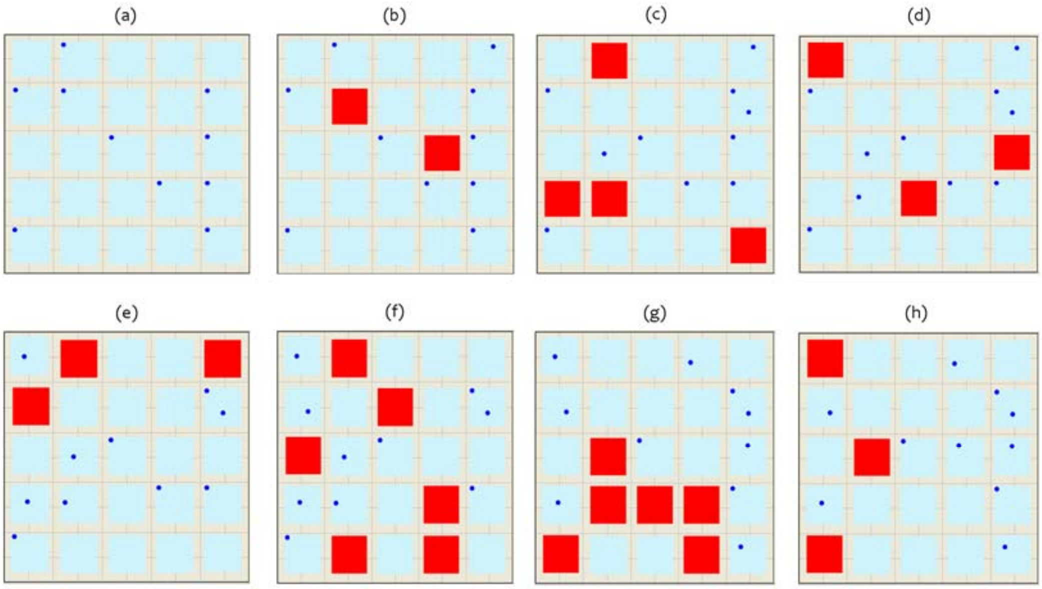

Fig. 2. Sequence of eight simulation screenshots (a) – (h) of a simulation run from initialization on the SeSAm multi-agent simulator. Figure shows how the carrier agents carrying sub-tasks are seamlessly transferred to a new core when executing cores fail.

The environment has provisions for modelling agents, the world and simulation runs. Agents are characterized by a reasoning engine and a set of state variables. The reasoning engine defines the behaviour of the agent, and is implemented in the form of an activity diagram, similar to a UML-based activity diagram. The state variables of the agent specify the state of an agent. Rules that define activities and conditions can be visually modelled without the knowledge of a programming language. The building block of such rules is primitives that are pre-defined. Complex constructs such as functions and data-types can be user-defined.

The world provides knowledge about the surroundings the agent is thriving. A world is also characterized by variables and behaviours. The modelling of the world defines the external influences that can affect the agent Hence, variables associated with a world class can be used as parameters that define global behaviours. This in turn leads to the control over agent generation, distribution and destruction.

Simulation runs are defined by simulation elements that contribute to the agent-based model being constructed. The simulation elements include situations, analysis lists, simulations and experiments. Situations are configurations of the world with pre-positioned agents to start a simulation run. Analysis lists define means to study agents and their behaviour with respect to time. Simulations are combinations of a situation, a set of analysis items and a simulation run; or in other words a complete definition of a single simulation run. Experiments are used when a combination of single simulation runs are required to be defined.

B. Experimental Modelling& Results

Figure 2 is a series of screenshots of a random simulation run developed on SeSAm for eight consecutive time steps from initialization. The figure shows the executing cores as rectangular blocks in pale blue colour. When a core is predicted to fail, i.e., temperature increases beyond a threshold, the core is displayed in red. The subtasks wrapped by the carrier agents are shown as blue filled circles that occupy a random position on a core. As discussed above, when a core is predicted to fail, the subtask executing on the core predicted to fail gets seamlessly transferred to a core capable of processing at that instant.

The simulation studies are in accordance with the expectation and hence are a preliminary confirmation of the feasibility of the proposed approach in swarm-array computing. Though some assumptions and minor approximations are made, the approach is an opening for applying autonomic concepts to parallel computing platforms.

The simulation studies are in accordance with the expectation and hence are a preliminary confirmation of the feasibility of the proposed approach in swarm-array computing. Though some assumptions and minor approximations are made, the approach is an opening for applying autonomic concepts to parallel computing platforms.

IV. Conclusions

In this paper, a swarm-array computing approach based on intelligent agents that act as carriers of tasks has been explored. Foundational concepts that define swarm-array computing and associated elements are also introduced. The feasibility of the proposed approach is validated on a multi-agent simulator. Though only preliminary results are produced in this paper, the approach gives ground for expectation that autonomic computing concepts can be applied to parallel computing systems and hence open a new avenue of research in the scientific community.

Future work will aim to study the third proposed approach or the combinative approach in swarm-array computing. Efforts will be made towards implementing the approaches in real time and exploring in depth the fundamental concepts associated with the constituents of swarm-array computing.

International Science Index, Computer and Information Engineering Vol:3, No:7, 2009 waset.org/Publication/3751

International Science Index, Computer and Information Engineering Vol:3, No:7, 2009 waset.org/Publication/3751

REFERENCES

[1] P. Lin, A. MacArthur and J. Leaney, “Defining Autonomic Computing: A Software Engineering Perspective” in the Proceedings of the Australian Software Engineering Conference, 2005, pp. 88 – 97.

[2] R. Sterritt and M. Hinchey, “Autonomic Computing – Panacea or Poppycock?” in the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2005, pp. 535 – 539.

[3] R. Sterritt and D. Bustard, “Autonomic Computing – a Means of Achieving Dependability?” in the Proceedings of the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2003, pp. 247 – 251.

[4] M. R. Nami and M. Sharifi, “Autonomic Computing a New Approach” in the First Asia International Conference on Modelling and Simulation, 2007, pp. 352 – 357.

[5] M. Jarrett and R. Seviora, “Constructing an Autonomic Computing Infrastructure using Cougaar” in the Proceedings of the 3rd IEEE International Workshop on Engineering of Autonomic and Autonomous Systems, 2006, pp. 119 – 128.

[6] S. Lightstone, “Foundations of Autonomic Computing Development” in the Proceedings of the 4th IEEE Workshop on Engineering of Autonomic and Autonomous Systemts, 2007.

[7] W. Gentsch, K. Iano, D. J.-Watt, M. A. Minhas and M. Yousif, “Self-Adaptable Autonomic Computing Systems: An Industry View” in the Proceedings of the 16th IEEE International Workshop on Database and Expert Systems Applications, 2005.

[8] G. Cybenko, V. H. Berk, I. D. G.-DeSouza and C. Behre, “Practical Autonomic Computing” in the Proceedings of the 30th IEEE Annual International Computer Software and Applications Conference, 2006.

[9] M. G. Hinchey and R. Sterritt, “99% (Biological) Inspiration” in the Proceedings of the 4th IEEE International Workshop on Engineering of Autonomic and Autonomous Systems, 2007, pp. 187 – 195.

[10] T. Marshall and Y. S. Dai, “Reliability Improvement and Models in Autonomic Computing” in the Proceedings of the 11th International Conference on Parallel and Distributed Systems, 2005, pp. 468 – 472.

[11] T. M. King, D. Babich, J. Alava, P. J. Clarke and R. Stevens, “Towards Self-Testing in Autonomic Computing Systems” in the Proceedings of the 8th International Symposium on Autonomous Decentralized Systems, 2007, pp. 51 – 58.

[12] R. J. Anthony, “Emergence: a Paradigm for Robust and Scalable distributed applications” in the Proceedings of the International Conference on Autonomic Computing, 2004, pp. 132 – 139.

[13] T. De Wolf and T. Holvoet, "Emergence as a general architecture for distributed autonomic computing," K. U. Leuven, Department of Computer Science, Report CW 384, January, 2004.

[14] F. Saffre, J. Halloy, M. Shackleton and J. L. Deneubourg, “Self-Organized Service Orchestration Through Collective Differentiation” in the IEEE Transactions on Systems, Man and Cybernetics, Part B, Dec 2006, pp. 1237 – 1246.

[15] P. Champrasert, C. Lee and J. Suzuki, “SymbioticSphere: Towards an Autonomic Grid Network System” in the Proceedings of the IEEE International Conference on Cluster Computing, 2005, pp. 1 – 2.

[16] A. Zeid and S. Gurguis, “Towards Autonomic Web Services” in the Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005.

[17] J. Almeida, V. Almeida, D. Ardagna, C. Francalanci and M. Trubian, “” in the Proceedings of the IEEE International Conference on Autonomic Computing, 2006, pp. 84 – 92.

[18] S. R. White, J. E. Hanson, I. Whalley, D. M. Chess and J. O. Kephart, “An Architectural Approach to Autonomic Computing” in the Proceedings of the IEEE International Conference on Autonomic Computing, 2004.

[19] M. Parashar, .Z. Li, H. Liu, V. Matossian and C. Schmidt, “Enabling Autonomic Grid Applications: Requirements, Models and Infrastructure” in the Lecture Notes in Computer Science, Self-Star Properties in Complex Information Systems, Springer Verlag. Vol. 3460, 2005, pp. 273-290.

[20] Y. Diao, J. L. Hellerstein, S. Parekh, R. Griffith, G. Kaiser and D. Phung, “Self-Managing Systems: A Control Theory Foundation” in the Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2005, pp. 441 - 448.

[21] S. Abdelwahed, N. Kandasamy, and S. Neema, "Online Control for Self-Management in Computing Systems," in the Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium, Toronto, Canada, 2004.

[22] Q. Zhu, L. Lin, H. M. Kienle and H. A. Muller, “Characterizing Maintainability concerns in Autonomic Element Design” in the Proceedings of the IEEE International Conference on Software Maintenance, 2008, pp. 197 – 206.

[23] J. O. Kephart and W. E. Walsh, “An Artificial Intelligence Perspective on Autonomic Computing Policies” in the Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks, 2004, pp. 3 - 12.

[24] A. Peddemors, I. Niemegeers, H. Eertink and J. de Heer, “A System Perspective on Cognition for Autonomic Computing and Communication” in the Proceedings of the 16th International Workshop on Database and Expert Systems Application, 2005, pp. 181 – 185.

[25] M. G. Hinchey, R. Sterritt and C. Rouff, "Swarms and Swarm Intelligence " in IEEE Computer, Vol. 40, No. 4, IEEE Computer Society, April 2007, pp. 111-113.

[26] J. Kennedy, R. C. Eberhart and Y. Shi, “Swarm intelligence”, Morgan Kaufmann Publishers, 2001.

[27] J. Wang, B. J. d’Auriol, Y.-K. Lee and S. Lee, “A Swarm Intelligence inspired Autonomic Routing Scenario in Ubiquitous Sensor Networks” in the Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, 2007, pp. 745 – 750.

[28] M. Hinchey, Y.- S. Dai, C. A. Rouff, J. L. Rash and M. Qi, “Modeling for NASA Autonomous Nano-Technology Swarm Missions and Model-Driven Autonomic Computing” in the Proceedings of the 21st International Conference on Advanced Information Networking and Applications, 2007, pp. 250 – 257.

[29] L. M. F.-Carrasco, H. T.-Marin and M. V.-Rendon, “On the Path Towards Autonomic Computing: Combining Swarm Intelligence and Excitable Media Models” in the Proceedings of the 7th Mexican International Conference on Artificial Intelligence, 2008, pp. 192 – 198.

[30] T. De Wolf and T. Holovet, “Towards Autonomic Computing: Agent-Based Modelling, Dynamical Systems Analysis, and Decentralised Control” in the Proceedings of the IEEE International Conference on Industrial Informatics, 2003, pp. 470 – 479.

[31] D. Bonino, A. Bosca and F. Corno, “An Agent based Autonomic Semantic Platform” in the Proceedings of the International Conference on Autonomic Computing, 2004, pp. 189 – 196.

[32] H. Tianfield, “Multi-agent Autonomic Architecture and its Application in e-Medicine” in the Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology, 2003, pp. 601 – 604.

[33] G. Pour, “Prospects for Expanding Telehealth: Multi-Agent Autonomic Architectures” in the Proceedings of the International Conference on Computational Intelligence for Modelling and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce 2006.

[34] H. Guo, J. Gao, P. Zhu and F. Zhang, “A Self-Organized Model of Agent-Enabling Autonomic Computing for Grid Environment” in the Proceedings of the 6th World Congress on Intelligent Control and Automation, 2006, pp. 2623 – 2627.

[35] J. Hu, J. Gao, B.–S. Liao and J.-J. Chen, “Multi-Agent System based Autonomic Computing Environment” in the Proceedings of the International Conference on Machine Learning and Cybernetics, 2004, pp. 105 – 110.

[36] B. Varghese and G. T. McKee, “Towards Self-ware via Swarm-Array Computing” accepted for publication in the International Conference on Computational Intelligence and Cognitive Informatics, Paris, France, 2009.

[37] J. Bacon, “Concurrent Systems Operating Systems, Database and Distributed Systems: An Integrated Approach”, Addison-Wesley, Second Edition, 1997.

[38] F. Klugl, R. Herrler and M. Fehler, “SeSAm: Implementation of Agent-Based Simulation Using Visual Programming” in the Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, Japan, 2006, pp. 1439 – 1440.

[39] SeSAm website: http://www.simsesam.de