CHAPTER SIX: CONCLUSION AND FUTURE WORKS

This chapter is divided into two sections; the first is the conclusion of our works. The second section present a recommendation for future works.

6.1 Conclusions

Networked systems are created by composing a complex set of hardware and software components that are subject to continuous upgrade, replacement, and scaling. The networked systems' behavior is highly dynamic, in response to unpredictable events such as anomalies (hardware failures, residual software defects, intrusions, DoS, performance failures, power loss, bandwidth limitations etc.) They must be anticipated and viewed as "routine" properties of this environment. Networked system resilience can be determined by its ability to recover rapidly from anomalies.

In respond to these malicious we implement the capability of Autonomic computing which is self-regenerative system. Autonomic computing systems are those systems that automatically manage themselves by carrying out tasks that have been traditionally performed by computer specialists. Its ultimate aim is to develop computer systems capable of self-management, to overcome the rapidly growing complexity of computing systems management, and to reduce the barrier that complexity poses to further growth. In other words, autonomic computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes whilst hiding intrinsic complexity to operators and users. An autonomic system makes decisions on its own, using high-level policies; it will constantly check and optimize its status and automatically adapt itself to changing conditions.

Self regenerative system is to develop technology for building computing systems that provide critical functionality at all times, in spite of damage caused by unintentional errors or attacks. All current systems suffer eventual failure due to the accumulated effects of errors or attacks. Regeneration means reproducing or reconstructing of a lost or injured body part. In computer science, To achieve the SRS program goals, the program will address several key technology areas, such as detection, configuration, updating, ect. We extract our model from biological process which is cell internal operation.

We implement our model using four agents, An agent can characterize an individual with capabilities to perceive and react to events in the environment, and, thus perform certain activities, the activities that we describe to our agent are the procedure of self regenerative goal.

Our model consist of four agents, the first is the D agent is for detection of and activation of the RC agent, then if the RC agent has activated by D agent then the RC agent will perform replication of component then the configuration of the component up to the normal mode, the prevention agent is dedicated to prevent the replication as well as suspected malicious initialization,

Theoretical formulation has been done to form and describe the process of the agents by finite state automata, and it indicate that every agent has it own state and action to perform and will end to a finale state

Case study

We prototype our mode using (JADE) java agent development framework. Afterward map the model to the distributed software, in result of this we get a system that is self-regenerative and all the procedure of the survivability are hiding from the user.

We implement two case study, the first is peer to peer system, and the second is cluster system, the result in the two case study are presented and show significant availability of the component .

6.2 Recommendation for Future Works

In order to totally fulfill the self-regenerative function, there are several challenges and requirement need to be pondered:

- The implementation of the model is done by java agent development framework, farther investigation can be conducted to look for a suitable agent development tools. Agents are the core of this model, and the agent has to be deployed in every node, the dependency of agent is an issue here and need to be solved in future works.
- This work has not covered the situation where the network is ad hoc. In ad hoc nodes are joining the domain and leaving . we can see new node joing and does not have the agent deployed , therefore we need to specify in future an algorithm to monitor and deploy agent in new nodes .
- Further investigation need to be conducted in order to combines self-healing of fault protection to our model, in self-healing there is a monitor for a fault in the component itself and not in the attacks. Therefore if we combine these two approaches we can have a more reliable distributed component.

REFERENCES

Aaron, B. B., and Charlie, R. (2005). Measuring the effectiveness of self-Healing autonomic systems. *Proceedings of the Second International Conference on Autonomic Computing (ICAC'05) (pp. 328-329). IEEE*.

ACL, Agent Communication Language, *http://www.fipa.org/specs/fipa00061/index.html*.

Bush, S.F.et a, "Genetically Induced Communication Network Fault Tolerance," Complexity Journal, 2003.vol. 9, no.2.

Camazing, S, Thakoor, et al,"self-organisation in biological system", 2003.

Carlson. B. M., "Principles of regenerative biology", 2007.

Clemens Szyperski, Dominik Gruntz, Stephan Murer, (2002)"Component Software: Beyond Object-Oriented Programming, Component Software Series", *Addison Wesley*.

Cohen, F. (1988). On the Implications of Computer Viruses and Methods of Defense. Computer & Security .

David, B. Robert Pennington .Cluster Computing: Applications. *Georgia Tech College of Computing* .

Falko Dressler , "Bio-inspired Promoters and Inhibitors for Self-Organized Network Security Facilities" Dept. of Computer Science 7 ,University of Erlangen-Nuremberg.

Falko Dressler, "Efficient and Scalable Communication in Autonomous Networking using Bio-inspired Mechanisms – An Overview". Dept. of Computer Science 7, University of Erlangen-Nuremberg Martenstr. 3, 91058 Erlangen, Germany, October 11, 2004.

FitzGerald, N. (2003). A Virus by Any Other Name - Virus Naming Updated. Virus Bulletin (1).

Ganek and T. Corbi, “The Dawning of the autonomic computing era,” IBMSystems Journal, 42(1):5-18, 2003.

George, C., Vincent, H. B., Ian, D. G.-D., and Chad, B. (2006). Practical autonomic computing. *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06) (pp. 3-14). IEEE Computer Society*.

HORN .(2001) .Autonomic Computing: IBM's Perspective on the State of Information Technology. *IBM Corporation*.

HYACINTH .(1996). Software Agents: An Overview, *Cambridge University Press* .

JADE, “Java Agent Development Framework, http://jade.tilab.com

JENNINGS. (2001) .An agent based approach for building complex software systems. *Communication of the ACM, Vol. 44, No. 4, pp. 35-41.*

JEZIC, M KUSEK, S DESIC, A CARIC AND D HULJENIC. (2003). Multi-agent remote maintenance shell for remote software operations. *Lecture Notes in Artificial Intelligence, Vol. 2774, No. 2,, pp. 675-682.*

John , Yoav Shoham , and Vanessa Teague (2007). Distributed algorithmic mechanism design and networking security.

KEPHART and D.M. CHESS (2003). The Vision of Autonomic Computing. *IEEE Computer, 36(1), pages 41-50.*

KUSEK, G JEZIC, I LJUBI, K MLINARIC, I LOVREK, S DESIC, G LABOR, A CARIC AND D HULJENIC. (2003) .Mobile agent based software operation and maintenance. *7th International Conference on Telecommunications (ConTEL), Zagreb, Croatia, Vol. 2, pp. 601-608.*

LOVREK, G JEZIC, M KUSEK, I LJUBI, A CARIC, D HULJENIC AND LABOR, S.D. AD O.(2003) .Improving software maintenance by using agent based remote maintenance shell. *International Conference on Software Maintenance (ICSM), Amsterdam, The Netherlands, IEEE Computer Society Press, Vol. PR01905, pp. 440-449.*

MCCANN, MARKUS HUEBSCHER,(2004) .Evaluation issues in Autonomic Computing. *Department Of Computing, Imperial College London, International Workshop on Agents and Autonomic Computing and Grid Enabled Virtual Organizations*.

PALUL .A Model-based system supporting Automatic self-regeneration of critical software.

PAVO´NA. (2007) .Agent-based modeling and simulation for the analysis of social patterns. *Universidad Complutense Madrid, Facultad de Informa´ tica* .

SAIDI HASSAN (2003);" Self-regenerative Software Component" Conference on Computer and Communications Security.

SCHMO¨ LZER , EGON TEINIKER 1, CHRISTIAN KREINER (2008).Model-typed component interfaces" ,*Institute for Technical Informatics, Graz University of Technology, Inffeldgasse 16, A-8010 Graz, Austria Salomon Automation GmbH, A-8114 Friesach bei Graz, Austria Received 16 February*

2007; received in revised form 20 January 2008; accepted 28 January 2008 Available online 6 February 2008.

SELVIN . (2003). A Biological Programming Model for Self-Healing", *Department of Computer Science University of Virginia Charlottesville*.

SZYPERSKI et al. (2002)"Component Software beyond Object-Oriented Programming," *Addison-Wesley, Great Britain, Second Edition*.

THOMAS R. PELTIER(2005). Information security fundamentals".

VOLYNKIN .(2007). Prevention of Information Attacks by run-time Detection of Self-Replication in Computer Codes. *journal of computer society*.

WAN et al (2006) . a comprehensive bio-inspired Model for network security. *Proceedings of the First International Conference on Semantics, Knowledge, and Grid ,IEEE.*

WEI et al,(2008) .Component behavior relativity analysis. *SIGSOFT Software Engineering Notes March.*

ZACH, P., and SAM, S .(2005)"A machine to support autonomic computing", *IEEE Region 5 and IEEE Denver Section Technical Professional and Student Development Workshop (pp. 25-31).*