SELF-HEALING SYSTEMS: APPLICATION AND METHODOLOGIES-A REVIEW

Ugwuanyi F. O. [1]✉ iD , Prof. Anigbogu S. O [2]✉ , Anigbogu K. S [3]✉

[1] Department of computer Science, Federal College of Education, Pankshin, Plateau State, Nigeria
[2, 3] Department of Computer Science, Nnamdi Azikiwe University, Awka, Anambra State, Nigeria

---

ABSTRACT

Self-healing in software applications is patterned after the human cells which regenerates after a damage has been done to it. There are always attacks on software applications that sometimes render the user helpless, since most users are not technicians. If these applications will be able to recover from attacks and get back to normal state before it was attaced without letting the user know that such attack ever happened, a self-healing mechanism has been achieved in that application. In this paper, we tried to look at what is self-healing, methodologies that some researchers have proposed in order to achieve self-healing in any given system, system faults and its remedies, self-healing life cycle, and applications of self-healing in a system.

---

1. INTRODUCTION

1.1. BACKGROUND

An approach to computer, software systems and applications that can take care of themselves without human intervention is known as *automatic computing system*. This approach takes after human nervous system. Which means that they will be able to function without human interference. According to Hudaib, Fahkouri, Al-Adiwann and Fahkouri, (2017). The automatic computing system is inspired by the human automatic nervous system. The system administrator has to set the objectives for the system. And it is expected that the system reacts towards these objectives. To achieve the objectives, a concecrated self-manaement substructure has to be maintained. To Koldehole, Mayer, Umakishore and Volze, (2013). The administrator has to determine the system's behaviour as upper stage policies. Self-healing objectives are to have a machine-controlled situation where the computer, software, or the application can mend themselves without people interfering. When a system gets infected the self-healing will be able to make it to come back to its normal state. According to Hudaib et al (2017), the self-healing attributes has authority over the external conditions that surrounds the system. Self-healing came to be because the several measures attempted to proffer solution to systems reliability has not yielded effective result. Since reliability,

performance and assurance is an issue in most systems, self-healing has come to take care of these conditions. Self-healing's major attribute is to identify the faults in any given system, and try everything possible to get rid of it.

1.2. RELATED LITERATURE

In the course of this study, several researchers were reviewed. It was observed that focuses are on the ubiquitous (existing everywhere) nature of the systems and error handling. Some of the presentations are the Gaia which is a middleware design and has the ability to take care of its wealth in a given environment. This Gaia was proposed by Chetan, Rajnathan, and Campbell (2005). Also reviewed is the Self-Adaptive Fault Tolerant Middleware (SATM) by Cai, Peng, Jiang and Zhang (2002). SATM is an error resistant middleware that exists everywhere in any given network. SATM discovers error by examining continuously the system's components such as the CPU, Memory, Operating System, Input/output etc, and then constructs the self-adaptive technique according to types of errors it detected. Also reviewed is the self-healing panel of the Middleware Adaptability for Resource Discovery Knowledge usability and self-Healing (MARKS) of Sharmin, Ahmed and Ahmed (2006). This middleware is known as *Efficient, Transparent and Secure (ETS).* They posited that it has a healing manager that takes care of errors in any given system. This is achieved by carrying out an observation of the conditions of the device components. If it finds errors in any of the components, it isolates the service and asks the device manager to provide an alternative device to run pending when the isolated device is repaired. The work of Silas, Ezra and Rajsingh, (2012) were also reviewed where they proposed Faults Tolerant Device Selection (FTSS). FTSS records every services that are allocated to all the authorized user and observe if the performance of the given services is completed successfully.

2. METHODOLOGIES

2.1. SELF-HEALING METHODOLOGIES

We will look at the different techniques, tools and methodologies of self-healing in this section. Hudaib et al (2007) examined different researchers with the techniques and methodologies they adapted in order to achieve a self-healing in an existing system. Some of the methodologies and techniques adopted by researchers which he examined are presented on the table below:

Table 1: Methodologies and Techniques of Self-Healing adopted by some Researchers. Source: Hudaib et al (2017)

Researcher	Methodologies and Techniques
Angelos et al (2007)	They focused on architectural changes to the software under protection as against the block level system identification.
Locasto et al (2007)	Brought the idea of application communities where collection of large numbers of independent instance of the same application and showed the practicability of the strategy using Selective Transactional Emulation (STE) as both the observing and ratification mechanics for low-level software faults.
Goutam, K.S (2007)	Presented a self-loading system that depends on the on-the-fly fault discovery and repair of web application or service agent code and data
Dashofy et al (2012)	Described a procedure that applied sub-structure elements in the contents of an overall architecture-based creativity for building self-healing systems. They observed that there is gap in the current sub-structure and the overall dream and ways to accomplish the dream.
Philip (2013)	Presented a hierarchy for identifying problem space for self-healing system. He also included fault theory, system reactions, system integrity and the settings of the plan.
Jiang et al (2002)	Described a generic modelling structure to ease the evolution of self-healing system. They also presented a structure to determine and execute self-healing on the system software.
George et al (2002)	Described a programming structure based on biological cell. They also speculated self-healing software system that will be inspired by a biological approach that will be very robust.
Faud et al (2009)	They matched a fault situation to an existing fault structure and tried to find a self-healing actions in this way.

Azim et al (2002)	Focused on Android application and described an idea that will use automatic detection and patch construction in providing a self-healing ability.
Michael (2004)	Presented an idea where setting a self-healing element for robust, concurrent and distributed software design.
Naftaly (2003)	Pointed out preconditions for self-healing or self – repair in a large system and presented a way to take care of these conditions especially when the software is not the same and are spread out.
Park et al (2006)	Presented a self-healing design which supports automatic code generation trying to use productions of design level. And also, automatic code operation idea to find out faults springing up from outside and inside of the surrounding.
Dinkek et al (2012)	Described a novel approach where self-healing in a distributed embedded system contains black-box application software
Dabrowsky et al (2002)	Presented a design structure to qualify how designs, configurations and consistency-maintenance mechanics and future recovery approach, each come together to achieve self-healing during communication error.
Amand et al (2012)	They are of the opinion that self-healing systems are very important in forestalling system software-based attacks which recovers and secures themselves from failed services. Self-healing systems are able to detect faults and repair the faults without an intervention of human being.
Gorla et al (2013)	Presented the issues that faces the design of self-healing system and focused on operational errors, and went on to propose methods to construct software system that can heal such errors without human intervention.

3. DESIGN OF SELF-HEALING SYSTEM

3.1. SELF-HEALING DESIGN REQUIREMENTS

According to Saha, (2007), a self- healing system is capable of modifying its own actions if there is a change in the surroundings. These changes can be unevenness on its resources. The needs of the end users, its movements and the system faults.

Depending on the self–healing life cycle and the features of the existing design consequences, the following are the design demands of the self-healing:

- **Cognisance:** It helps in the observation of the system's execution. These includes the reliability, the correctness, state, behaviour and the recognition of abnormalities in their executions.
- **Flexibility:** These enables adjustments of a static system
- **Dynamical:** It has the ability to change at any time.
- **Self-Sufficiency:** It is able to take care of the animosities that are observed as it functions. This will help to plan, deploy and enact the changes that can guarantee economy.
- **Observable:** This enables the careful watch as a result of self – healing systems performance environment.
- **Validity:** It provides the power for a system to efficaciously react to unexpected operational circumstances. The outer condition of the system like a malicious attack or unexpected behaviour can be an infliction.
- **Distribution:** It supports effective execution as a result of the system in front of the various deployment profiles.
- **Mobile:** It provides the power for the system to change the location of the systems components.
- **Traceable:** this relates to the systems design and the systems operational level to make a change in order to support the traceable requirement (Salah, 2007).

4. IMPLEMENTATION OF SEL-HEALING SYSTEM

4.1. SELF-HEALING SOFTWARE ERRORS AND REMEDIES

In a complex computing environment where you have web applications and web services being used often, errors in the services of the software are expected. A study by Li et al (2014) shows that 72% of the most used web sites has errors than are easily seen by the users. Saha (2007) observed that these errors are as a result of java exception, deadlock threads, aging, source code buy etc.

A table below will try to identify some of these errors and how to fix them as observed by(Li et al, 2014) in a complex 2EE

Table 2: Errors and Remedies of Self-healing System

Errors	Remedies
Glitch in source code	Restart the system and inform the administrator
Java exception	Saha (2007) suggested that the best thing to do is to micro-boot EJB
Deadlock threads	Saha (2007) suggested that the remedy is to micro-boot the EJB, and then kill hung querry.
Buffer controversy	When there is buffer contention one is advised to partition the memory again across the several buffers.
Senescent	When aging occurs, one is expected to restart in order to reclaim the leaked assets.
Read/write controversy	When there is controversy in the read/write, it is expected that the table balance is partitioned again.

5. DISCUSSION AND CONCLUSION

5.1. SELF-HEALING CYCLE

According to Hudaib (2017), self-healing is a sort of virtual mechanics passed on various levels. The levels are listed below. The software that intends to heal-itself (Self-healing) will have to first of all observe everything around itself. Then based on what was found, the problem is analyzed in order to find the real remedy. Then comes the level where the solution found is tried-out to know its workability, and then lastly enforced to restore the system to its original state. When the error is remedied, the system is tried out again. This can be described as self-healing loop. According to Pasier and Dustdar, (2010) self-healing is a process which cycles its loops and adheres to five most important roles which are; to observe (monitor), to examine (analyse), to explore (diagnose), to cure (heal) and Domain (knowledge base).
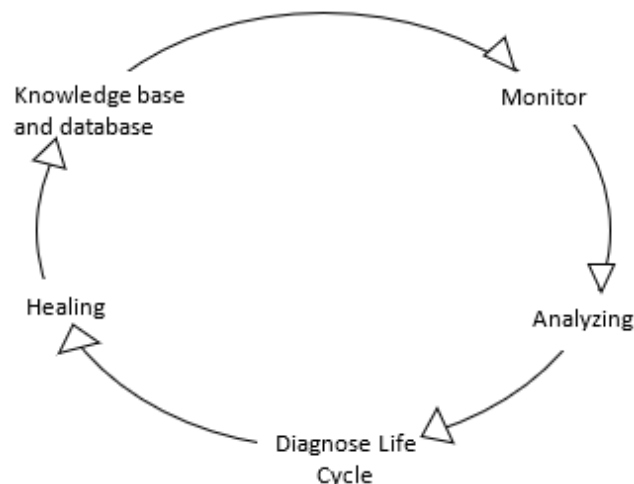
Figure 1: Self-healing life cycle
Source: Pasier and Dustdar (2010)

Let us try to explain the five important roles mentioned above by a self-healing system.

- **Observing (Monitoring):** This is a procedure in which relevant information are gathered.
- **Examination (Analysing):** Takes proper examination of the situation enables you to find out exactly the actions to take. According to Prasier and Dustder (2010) This is entirely done by comparing the status into the systems requirements.
- **Exploration (Diagnosis):** This is a procedure for getting all necessary information and errors and or misbehaviour in the system. This can be done using comparison or various methods.
- **Curative (Healing):** This consist the adequate preparation and carrying out of a pre designed arrangements as planned. These predesigned arrangements may include accuracy, sound and aforethought acts that are definitely required. Performance at this level is where the whole arrangements is carried out in order to achieve the required result of a system.
- **Domain (Knowledge Base):** The domain consists of all the important knowledge that the system has produced in order accomplish the task its intended to. This is very important. To Li and Halford (2014), the development of self-healing is not as easy as it looks.

There are also some features that are necessary for an operation of self-healing extension. According to Li and Halford (2014), these features are as follows:

- Not reliable system
- Several types faults in the source
- The system being unpredictable

5.2. SELF-HEALING APPROACHES AND APPLICATIONS

Researchers like (Katti, Di-Fatta, Naughton and Engelmann, 2015; Ehler, Van-Hoorn, Walker and Hasselbring, 2011; Sanchez, Benyahia and Crespi, 2014) has posited three models of self-healing. These models are categorized because of these policies; which are:

- Responses (Reactions)
- Procedure (Routine)
- Contemplation (Reflection)

When you take these further, you will begin to see their variations in terms of processing between some factors. For Li et al (204), these factors are

- Assessment of the things around us, and
- Knowledge of the things around us.

There is an emergence of so many self-healing overtures in some application space that have different characteristics, Riz, Koopma and Shaw (2012) outlined these characteristics and are shown below:

- Study of the index and the direction
- Enclose the system firmly in a surrounding mass
- Operating system
- Based on the design
- Based on multiple level
- Based on multiple agents
- Contemplative middleware
- etc

5.3. AUTOMATIC SELF-HEALING SOFTWARE

Self-healing in software is a situation where software are able to recover from attacks and returns to its normal condition before the attack. It is designed such that it will withstand every possible attack. According to Periz Laadan, Viennot, Keromytis and Nich (2009), it has to be efficient, adaptative and thoughtful system architecture that are

applied to security for concise response to attacks. These supposed to be a continuous process. For Hudiab et al (2017), it is a mediator for long term dependability and short-term wholeness. For a software to be able to heal itself, it must have the ability to return to its normal state after every attack. There are other terms associated with self-healing. Gbosh, Sharmon, Rao and Upadhyay, (2007) listed these terms as follows:

- Stabilizing
- Replacement
- Security
- Isolation

5.4. RELATED TERMINOLOGIES

- **Self-Adaptation:** This is a process of a system adjusting to something, especially its environment. Frei (2010) stated that for a system to be self-adapted it has to give an examination of its present condition or after an improvement has been done to it, to decide whether there still need for further improvement.
- **Self-Improvement:** This is the process of the system to make itself as favourable as it can. According to Trumler, Bagci, Petcold and Ungere, (2005), it is to choose the optional result that will meet an exact target by the system itself, and the system will adjust to its path following the available resources.
- **Self-Observation:** Self-observation is a situation where the software will be capable to examine its internal functions and also its operations. The software will be required to produce a report that will tell about its actions in the past.
- **Self- Examination:** Self-examination is carried out by the system when it is put on and it starts malfunctioning. If this happens, it will be able to discover any alteration in its constellation, or if there is missing components and therefore warning is given by the device stating what has gone wrong.
- **Self-Administration:** Trenler et al (2005) posited that self-administration is when the system can take care of its operation automatically. Whenever the system finds solution to any bug without human interference, it is said to be self-management.
- **Self-Authority:** This is the ability for the system to have control over its operations. Schneider, Barker and Dobson (2003) stated that these depend on the complex nature of the system's inner condition which is examined as a result of the complexity of the morphology, and also the complex nature of its conduct.
- **Self-Conformation:** This is a process that establishes and monitors system configuration of an exact parts. For Schneider et al (2003), the space of operation supports constellation examination by denial of the created observation with all encouragement from hardware and software configuration.

6. CONCLUSION

Various factor characterizes the self-healing mechanism. Researchers has examined these various factors. Most of them has come to conclusion that it's possible to create systems with self-healing capabilities.

Self-healing system when its fully developed will help users of computing devices focus on tasks intended to achieve with such device than on the technicality of the devices they are using. With the development of the mobile systems, the usage has increased tremendously and these systems are bound to fail without given prior notice. If it fails and is able to repair itself without the user noticing this will be a huge achievement.

SOURCES OF FUNDING

CONFLICT OF INTEREST

The author have declared that no competing interests exist.

ACKNOWLEDGMENT

REFERENCES

[1]   Altanasopoulos, D, Zarras A.V., Vassilliadis P., and Issammy V, (2011): Mining service Abstracts. NIER track 2011 33rd international conference software engineering, Honolulu, 21-28 may 2011, 944-947

[2]   Cai, H., Peng, C., Jiang, L., & Zhang, Y. (2012, April). A Novel Self-Adaptive Fault-Tolerant Mechanism and Its Application for   a Dynamic Pervasive Computing Environment. In Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IE

[3]   Carzaniga, A, Gorla, A. and Pezze, M (2008) Self-healing by means of automatic work around SEAMS 08, Leipzy, 12-13 may.

[4]   Chetan, S., Ranganathan, A., & Campbell, R. (2005). Towards fault tolerance pervasive computing. Technology and Society Magazine, IEEE, 24(1), 38-44.

[5]   Ehlers, J. Van-Hoorn, A., Waller, J and Hasselbring, W. (2011). Self-adaptative software system monitoring for performance anomaly localization. ICACII, Karlzuhe, 14-18 June.

[6]   Frei. R (2010) self-organization in Evaluation assembly system PhD Thesis Department of Electrical Engineering Univasidade nova de Lisbia, Portugal.

[7]   Ghosh, D, Sharman R., Rao, H.B., and Upadhayaya, S. (2007). Self healing systems survey and synthesis, Decision support systems, 42,21,64-2185

[8]   Hudaib, A.A, Fakhouri, H.N, Al-Adwan, F.F and Fahkouri, S.N (2017) A survey about self-healing systems (Desktop and web Application) Communication and networks, 9, 71-88

[9]   Hudaib, A.A and Fahkouri; H.N (2016). An Automated approach for software fault detection and receiving communication and network 8.

[10]   Katti A., Di- Fatta, G., Naughton, T. and Engelmann, C. (2015). Scalable and fault tolerant failure dectection and consensus. Euro MPI 15, Bordeaux, 21-23 September

[11]   Koldehofe B.; Mayer R., Umakishhove, R.K and Volz, M. 2013) Roll-back- Recovery without checkpoint in distributed event processing systems. DEBS 13, arlingtom29 July-3rd July

[12]   Lee.D, Yoo .J, Kang.H, Kim .K, and Karg.K, (2006). Distributed IP v 14 addressing Technique for mobilizing Ad-hoz Networks SACOL 23-77April,

[13]   Li. D, Tran, A.H., and Halford, in G.J (2014) making web application more Energy Efficient for OLED Smartphone. ISCE 14 31 May 7 June 2014 Hyderebad.

[14]   Pasier, H and Dustdar, S (2011). A Survey on self-healing systems. Approaches and systems Computing, 91, 43-73

[15]   Perez, C. R, Stelios, S Laadan, O, Viennot N, Keromytis A, and Nieh J. (2009) Automatic self-healing: present and future.

[16]   Saha.G.K (2007) software- Implemented self-heading systems ACM Ubiquity. Vol. 8 No 12 self-healing system Techniques and application Dijon

[17]   Schueder, C, Barker, A and Dobson, S (2013) A survey of self-healing systems Frameworks. Software: Practice and Experience, 45, 13, 75-1398

[18]   Schuchez, J. Ben-Yahia, I.G and Crespi, N (2014) POSTER: Self-healing mechanism for software-defined networks.

[19]   Sharmin, M., Ahmed, S., & Ahamed, S. I. (2006, April). MARKS (middleware adaptability for resource discovery, knowledge usability and self-healing) for mobile devices of pervasive computing environments. In Information Technology: New Generations, 2006. ITNG 2006. Third International Conference on (pp. 306-313). IEEE.

[20]   Silas, S., Ezra, K., & Rajsingh, E. B. (2012). A novel fault tolerant service selection framework for pervasive computing. Human-centric Computing and Information Sciences, 2(1), 1- 14.

[21]   Trumler, W., Bagei, F., Petzold, J., and Ungerer, T. 92005). AMUN-Autonomic middleware for Ubiquities' Environment applied to the snar Doorplate Project informatics, 19, 243-252