

A Survey about Self-Healing Systems (Desktop and Web Application)

Amjad A. Hudaib¹, Hussam N. Fakhouri², Fatima Eid Al Adwan³, Sandi N. Fakhouri²

¹Department of Computer Information Systems, The University of Jordan, Amman, Jordan

²Department of Computer Science, The University of Jordan, Amman, Jordan

³Department of Counseling and Special Education, The University of Jordan, Amman, Jordan

Email: ahdaib@ju.edu.jo, h.fakhouri@ju.edu.jo, f.aladwan@ju.edu.jo, sandi_fakhouri@yahoo.com

How to cite this paper: Hudaib, A.A., Fakhouri, H.N., Al Adwan, F.E. and Fakhouri, S.N. (2017) A Survey about Self-Healing Systems (Desktop and Web Application). *Communications and Network*, 9, 71-88.

<https://doi.org/10.4236/cn.2017.91004>

Received: March 11, 2016

Accepted: January 22, 2017

Published: January 25, 2017

Copyright © 2017 by authors and
Scientific Research Publishing Inc.

This work is licensed under the Creative
Commons Attribution International
License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The complexity of computer architectures, software, web applications, and its large spread worldwide using the internet and the rapid increase in the number of users in companion with the increase of maintenance cost are all factors guided many researchers to develop software, web applications and systems that have the ability of self-healing. The aim of the self healing software feature is to fast recover the application and keep it running and available for 24/7 as optimal as possible. This survey provides an overview of self-healing software and system that is especially useful in all of those situations in which the involvement of humans is costly and hard to recover and needs to be automated with self healing. There are different aspects which will make us understand the different benefits of these self-healing systems. Finally, the approaches, techniques, mechanisms and individual characteristics of self healing are classified in different tables and then summarized.

Keywords

Computer Architecture, Self-Healing Systems, Self-Healing Principles, Software Self-Healing, Web-Self-Healing

1. Introduction

Software development engages a lot of researchers for the development of software system which is typically secure, significantly used, reliable, configured, most effective, self controlled, facilely updatable, healed and distinguished. The word self-healing is deeply connected the field of Autonomic computing.

The development of computer and software systems and applications that can manage themselves in accordance with high-level guidance from humans has been called the Autonomic Computing systems (ACS), which is inspired by the

human autonomic nervous system which regulates vital body functions without the need for conscious human involvement. *i.e.* human body is able to heal himself in case of injury using the control of the brain and nervous system with corporation of different body systems [1] [2] [3]. ACS extends the adaptive behavior of the nervous system to computing systems and software. Autonomic systems respond to changes in their environment according to goals set by the system administrator. A dedicated self-management infrastructure is then responsible for maintaining a system state that adheres to those goals. This is achieved by automating low-level decisions and tasks while allowing administrators to specify system behavior as high-level policies [4] [5].

Autonomic computing (AC) was IBM's suggest resolution to the issue related to the growing involvement of software and computing systems developing. Autonomic computing beginning was published in October 2001 and depicted a view of computing systems which administer themselves related to main level goals. The model looks to qualify the load of gathering and administering extremely comprehend systems with raising automation and objective nomination [7]. The phrase autonomic is come from autonomic nervous system (ANS) of human, which detects dynamic physical functions with no need to the necessity for aware of human partnership. Like, while someone gets into climate that has high temperature, the ANS directly contains race to cold the body and keep a stable temperature. Autonomic computing expands the fit actions of the ANS for software and computing systems [8]. Autonomic systems reply to alteration in its environment related to the objectives set through system administrator. Present with self administered infrastructure that is consequently in charge of keeping system condition which involves these objectives. This is accomplished through automating minimum level of tasks; by letting administrators to determine system attitude as top scale of policies [9].

Self healing term is also applied to web applications, self healing of web application require a 24/7 auto monitoring of the web application and a fast mechanism of recovery that can keep the online functionality and service offering to the customer available all the time [10] [11] [12]. The importance of developing fast automated self-healing web applications was generated from the effect that may be caused if the web application for a business or a company is stopped to run for few hours. For example, an online business such as a bank may lose customer trust and lose financially if it is not functioning for a few hours. Many factors may affect a web application and cause it to stop. These factors may be either internal or external. The internal factor include viruses, worms that may affect the server that hosts the web application. The external factors include attackers that attack the website and change the content of web pages for a web application for different reasons including the use of different methods such as xss, sql injection. You need to have references for this information! [14] [15] [16].

After publishing the web application on the hosting server many problems rise including deletion of a component, replacing of a component or modifying a component. The risk of having one of these three factors is very high. For exam-

ple replacing a web application checkout component by an attacker that function the same way as the main component but with minor changes would allow the attacker to steal credit card information which will cause a major problem and loss to the web application owner and customers. Most of web application owner do not perform tests to check if the component has been changed or not and that is because of the complex architecture of the web application and lack of knowledge at the owner level [17].

Self healing main aim is to have an automated system that is able to heal itself without human interaction. It has different pre defined actions and procedures that are suitable for system recovery from different expected failure states. It is able to make the system to change from the infected state to the healthy state. The self-healing property has control over a set of environmental factors that the system run into [18].

The rest of this paper is organized as follows. Section 1 provides a definition of some terminology related to the term self-healing, section two to section ten try to answer the following questions [19].

- What is self healing? History and Definition.
- Why using self healing software?
- What are the types of self healing and the mechanisms used?
- Where we can use self healing software?
- What is the future perspective of self healing?
- What are the methodology, techniques, tools developed in self healing and there classification.

It provides a deep look to the self-healing software systems history, definition, and aim of using, type, mechanisms, use and future perspective [20] [21] [22].

2. Related Terminology

2.1. Self-Adaptation

The systems that are able either to enhance their status or keep it running under different situations; it is able to give evaluation for it current status or after enhancement has been done to the system. And it is able to decide it need any enhancement [17].

2.2. Self-Optimization

It means to find the optimal solution either minimum or maximum to meet specific goal by the system itself, and that the system adjust its path according to resources provided to find the goal [26].

2.3. Self-Monitoring:

The software system will have the capability and functions that is required to monitor its internal functions as well as its performance. The system will also be able to generate reports that has feedback and learning or adaptive capabilities [20] [21].

2.4. Self Testing

Is the process that is happening by the device when it is turned on for malfunction, discover any change in its configuration or miss in its component, so error message is being displayed by the device to specify what is happening and what user should do [23].

2.5. Self Diagnoses

Process to identify and diagnoses condition in one self to reduce errors, that could be dangerous if unsuitable decisions are taken on the basis of a misdiagnosis [24].

2.6. Self Management

The techniques in which computer system manage their own functions without human intervention self repair is the process that software is automatically find a solution to software bugs, without human intervention, by observe unacceptable error behavior [25].

2.7. Self Control

It is the process of controlling that agent has over its own state and behavior. Depending on the complexity of the agents internal state which is evaluated as a function of structural complexity and internal state size and of its behavior complexity [26].

2.8. Self Configuration

Process in which establishment and monitoring system configurations by a specific component, that the area of execution is supporting configuration testing by reusing of the created tests with every supported hardware and software configurations [27].

3. Self Healing Definition

The term self-healing for software is inspired from the biological healing process for human and animals, where the body heals itself by repairing the affected tissue or bone, the process of healing is carried out internally from inside the body, the cells will gather in the place that has been affected (ex. Tissue insured or bold vessel cut) and the heal process retain the affected place to its original health status [28]. The process of healing is controlled by the brain and nervous system. Note that some types of damages can't be self healed to retain to its original status (*i.e.* Hand cut) but a method of compensating to stop the damage from leading the system to complete fail will be applied (ex. Stopping the blood bleeding by cell gathering at the place of hand cut will prevent dying). Some healing processes may fail from and the reason is external factors (*i.e.* Bacteria may infect the hand cut place and lead to death of body or the place of cut very big and bleeding continuo). These different processes and cases have been applied in the field of self healing for software systems [29] [30] [31].

The term self healing first appeared with the appearance of the concept autonomic computing that was invented by IBM, Autonomic computing includes designing modern systems which includes four major design objectives. Related to IBM, systems like this should be: self-healing, self-configuring, self-optimizing, self-protecting. Ever after the cast of this prevision, several significant researches have been generated concentrating fundamentally on basics and architecture for designing distributed autonomic systems. Every schema participate a general objective of inquiring much perceivable future of computing [32] [33] [34]. That is set to be completed through developed and implement systems that efficiently keep them with small or not skilled intervention. Self Healing Characteristics are shown in **Figure 1** [11].

When design a self healing software system many questions should be taken into consideration:

- Will the Self healing software system that be able to heal the software in case of failure?
- Will the Self healing software system that be able to heal every failure type?
- Will the Self healing software system that be able to heal the software if there is other affecting or attacking factors? [35] [36].

4. Automatic Self-Healing Software

Automatic Self-Healing Software concept is a kind of highly efficient, adaptive and a thoughtful system design which is further applied to security for appropriate reaction against every possible threat. As far as any software is concerned, it is a kind of never stopping process. It is a bridge between a long term reliability and Short-term unity [6].

Every self-healing software system will have the ability to return from the abnormal state to normalcy state [20]. In some cases, Automatic Self-healing software is considered to be a subordinate to every fault tolerant system [13]. Self healing contain the term self recovery that include the following methods:

1. Stabilizing.
2. Replacement.
3. Security.
4. Isolation.

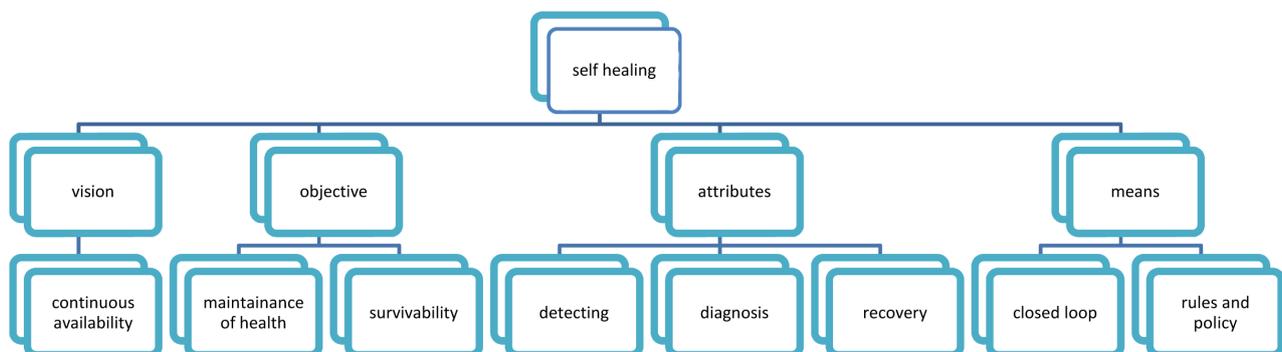


Figure 1. Self Healing Characteristics (Dionysis et al., 2011).

5. What Are the Types of Self Healing and the Mechanisms Used?

At this stage, what are the types of self healing and the mechanisms used? Before moving on with the concept, this is another question that we need to answer immediately. Self-healing is a kind of virtual mechanism based on four different stages:

First of all, Automatic Self-healing software monitors everything.

Secondly, according to this mechanism, the problem is diagnosed.

Thirdly, the relevant solution is automatically found.

Lastly, comes the most crucial stage when the solution is tested and finally implemented to restore the software back to normalcy. After the problem is fixed, the software is tested once again and starts a Self-Healing loop. Self-Healing loop is the most defining stage of self-healing process which is a cycle that loops throughout the process and performs the five most important functions. Self healing life cycle are shown in **Figure 2** (Herald Pasier-Scharam Dustdar, 2010 [36]).

6. Self Healing Life Cycle

1) Monitoring

It is a process for gathering all the vital information. For this purpose, information are brought into use for analyzing the task.

2) Analyzing

Analyzing the task helps you determine the action to be taken that is entirely done by comparing the status info to the system's requirements.

3) Diagnosis

It is a process for capturing all the vital faults, errors or changes in the system. Using comparison or different techniques.

4) Healing: The healing consist of planning and execution.

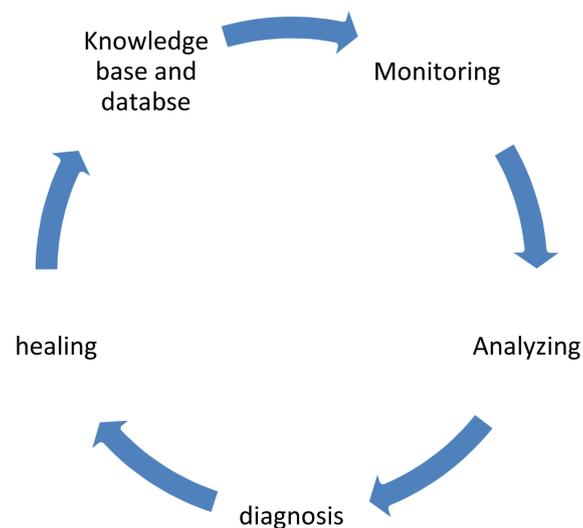


Figure 2. Self healing life cycle.

A plan is charted out to carry the task successfully like Accuracy, sound as well as properly planned action which is definitely required. Execution: At this stage, the whole plan is executed to perfection for achieving the desired result of a healthy software or system [37] [38].

5) Knowledge

This phase is important as it contains all the relevant knowledge consumed and produced by the earlier four tasks [21].

It is clearly evident that the process of developing Self-healing Software is not as easy as it appears to be [9]. The self-healing life cycle which is a vital component of self-healing approach that It is a vital design component related to the design element .At this stage, the idea of constant multi-stage processing loop is good for self-healing approaches [1] [42]. The operating environment of self-healing extensions holds a lot of features such as:

1. Unreliable systems.
2. Many types of sources related to errors.
3. Fluctuating over time.

7. Self-Healing Policies and Executed Approaches and Applications

According to the researches that is influenced by human behavior and conducted by AI Research, there are at-least three level model of self-healing. These models are based on following policies [43] [44] [45]:

1. Reaction.
2. Routine.
3. Reflection.

All of these three levels further vary in terms of deep processing (Norman DA, *et al.*, 2003). This processing is further involved between the following two factors:

- Evaluation of the nearby world.
- Cognition of the world.

Many of self-healing approaches has been emerged in various application areas with various features such as:

1. Survey index and guide.
2. Embedded systems.
3. Operating systems.
4. Architecture based.
5. Cross-layer based.
6. Multiple agent based.
7. Reflective Middleware.
8. Legacy application and Aop.
9. Website services and Qos Based [39].

8. Classification

It is challengeable task that only a failure that can help to move the deployment of appropriate recovery strategies impeccable either affects one unit or the entire

system. Despite all of this, general classification of failure can be found in self-healing concerned researches, System fitness and evaluation which is an integral part of this process.

Self-healing mechanisms, systems, tools provide a great solution for the huge and complicated system that require great effort, cost, time and knowledge to be recovered from different types of errors either big or simple. And form the errors that are hard to manage by human. The concept of self healing that were raised from autonomic computing included the different categories of autonomic computing (self-configuration, self-optimization, self-restoration and self-protection) [10] [46] [47]. When we talk about the computer world and the automation of systems then we should certainly develop a Software that has the ability to actively control itself rather than passively being controlled through a human. Maximum self properties can be achieved by the responsibility of standalone self reliable entity which controls a hierarchy of different situations that the software face in the environment.

Self healing and autonomic computing may be categorized to centralized and decentralize (**Figure 3**) from the controlling perspective. The centralize have a control manager which track, analyse, make plans and execute. Whereas the decentralized don't have a controlling manager and require human interaction, it may have rules and policies to be applied [48] [49].

Furthermore, researchers have done many researches for the implementation of self-healing techniques. Such researches are still developing. Despite this, Hewlett-Packard, an American global information firm took a giant leap in this direction and launched self healing software services to support self-healing [5]. In accordance with the reviews provided by the experts, this move is expected to be a positive and a productive push for the software engineering industry to produce self-healing software and operating systems. Moreover, this will be a great help to ease the problem of repairing faults. The company expects the operating systems to be extra smart. In simple words, the operating system should be able to react exactly like a human body. Being this smart will allow the operating systems to take appropriate actions to fight against the possible threats. [40] [52] [53].

The concept of Automatic self-healing software appears to be very interesting. Most of the IT companies get attracted towards it. Again, the reality is not what it actually appears to be! Very frankly, the reality is entirely vice-a-versa and involves a lot of snags. These snags are a kind of road block in the process of replacing the existing software technologies with the one that can heal itself automatically [49] [50].

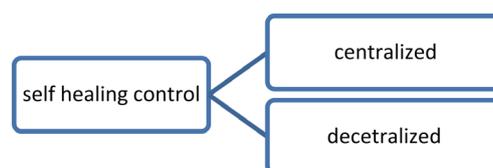


Figure 3. Self healing control type.

Table 1. Summary of self healing methodology, tool, and technique developed and the name of the researchers that have developed them.

Researcher	Self healing Methodology, tool, and technique developed
Angelos D., <i>et al.</i> , 2007	Attempted to characterize self-healing software systems by surveying some of the existing work in the field. focusing on systems that effect structural changes to the software under protection, as opposed to block level system reconfiguration.
Locasto <i>et al.</i> , 2007	introduced and explored the concept of Application Communities: collections of large numbers of independent instances of the same application, and demonstrated the feasibility of the scheme using Selective Transactional EMulation (STEM) as both the monitoring and remediation mechanism for low-level software faults, and provided some preliminary experimental results using the Apache web server as the protected application.
Goutam K. S., 2007	described various issues on designing a self-healing software application system that relies on the on-the-fly error detection and repair of web application or service agent code and data, illustrated critical points of the emergent research topic of Self-Healing Software System
Dashofy, <i>et al.</i> , 2002	Presented tools and methods that implemented infrastructure elements in the context of an overall architecture-based vision for building self-healing systems, and concluded with a gap analysis of current infrastructure vs. the overall vision, and plans for fulfilling that vision.
Philip Koopman, 2003	Proposed a taxonomy for describing the problem space for self-healing systems including fault models, system responses, system completeness, and design context.
Jiang, <i>et al.</i>	presented a generic modeling framework to facilitate the development of self-healing software system and introduced a modeling framework to specify and implement self-healing focusing on the software aspects of a system
George <i>et al.</i> , 2002	Presented a cell-based programming model inspired from biology and speculate on biologically inspired strategies for producing robust, scalable and self-healing software systems.
Fuad, <i>et al.</i> , 2009	presented a new technique of finding self-healing actions by matching a fault scenario to already established fault models
Azim, <i>et al.</i> ,	presented an approach that uses automatic error detection and patch construction towards providing a certain degree of self-healing capabilities to Android apps.
Michael E. Shin	described an approach to designing self-healing component for robust, concurrent and distributed software architecture
Montani <i>et al.</i> ,	described a CBR approach for providing large-scale, distributed software systems with self-healing capabilities, and demonstrated the practical applicability of methodology by means of some experimental results on a real world application.
Park <i>et al.</i>	proposed a self-healing mechanism that monitors, diagnoses and heals its own internal problems using self-awareness as contextual information.
Naftaly H. Minsky	identified one of the necessary conditions for self-healing, or self-repair, in complex systems, and proposed means for satisfying this condition in heterogeneous distributed software
Diaconescu, <i>et al.</i> ,	Proposed the use of component redundancy for enabling self-optimisation, self-healing and dynamic adaptation capabilities in component-based software systems.
Keromytis	provided first attempt to characterize self-healing software systems by surveying some of the existing work in the field and focused on systems that effect structural changes to the software under protection, as opposed to block level system reconfiguration.
PARK, <i>et al.</i> ,	Proposed a self-healing software architecture supported automatic code generation approach using productions of design level. and proposed an automated code generation approach to determine problems arising in external and internal system environment.
Brumley <i>et al.</i>	Presented a self-healing architecture for software systems where programs self-monitor and detect exploits, self-diagnose the root cause of the vulnerability, self-harden against future attacks, and self-recover from attacks.

Continued

Dinkel <i>et al.</i> ,	Presented a novel approach for self-healing in distributed embedded systems containing black-box application software. Abdullah <i>et al.</i> presented a self-healing architecture for software system based on one of the biological processes that have the ability to heal by itself-the wound-healing process, which is divided into two layers, functional layer and healing layer.
Michael E. Harald <i>et al.</i> ,	Focused on the self healing branch of the research and gave an overview of the current existing approaches.
Dabrowski <i>et al.</i> , 2002	Used architectural models to characterize how architecture, topology, consistency-maintenance mechanism, and failure-recovery strategy each contribute to self-healing during communication failure.
Elkorobarrutia <i>et al.</i> ,	described an approach of inserting a self-healing mechanism in components that are specified according to a state chart and whose implementations also offer the possibility to act on them in terms of state; <i>i.e.</i> forcing the component to some state and rolling back one transition.
Anand <i>et al.</i> ,	self-healing systems prove increasingly important in countering system software based attacks, which recover and secure to the data from interrupted services. Self-healing systems offer an active form of decision support, without human intervention that can detect the fault and recover from the fault.
Gorla <i>et al.</i> ,	Discusses the challenges underlying the construction of self-healing systems with particular focus on functional failures, and presents a set of techniques to build software systems that can automatically heal such failures.
ánchez <i>et al.</i> ,	Provided SDN with fault management capabilities by using autonomic principles like self-healing mechanisms.
Alessandra Gorla,	Argued that a disciplined design approach can enable a wide and effective range of self-healing mechanisms, thus overcoming many limitations of the current approaches.
Ardagna <i>et al.</i> ,	Proposed a classification of Web service faults, discussing three levels of faults, namely infrastructural & middleware, Web service, and Web application, and their mutual dependencies.
Azim <i>et al.</i>	Presented an approach that uses automatic error detection and patch construction towards providing a certain degree of self-healing capabilities to Android apps.
Zhang <i>et al.</i> ,	Proposed a fully dynamic solution to locating execution omission errors using dynamic slices. We introduce the notion of implicit dependences which are dependences that are normally invisible to dynamic slicing due to the omission of execution of some statements.
Fuad <i>et al.</i>	Proposed a new technique of matching unknown fault scenarios to already established fault models. By capturing runtime parameters and execution pathways, stable execution models are established and later are used to match with an unstable execution scenario.
Ehlers <i>et al.</i> ,	Proposed an approach for localizing performance anomalies in software systems employing self-adaptive monitoring.
Kramer <i>et al.</i>	Focused on architectural approaches to self-management, not because the language-level or network-level approaches are uninteresting or less promising, but because we believe that the architectural level seems to provide the required level of abstraction and generality to deal with the challenges posed.
Katti <i>et al.</i>	Presented and compares two novel failure detection and consensus algorithms and proposed algorithms are based on Gossip protocols and are inherently fault-tolerant and scalable.
Huang <i>et al.</i> ,	Investigated the effects of using an unsupervised log data abstraction method to aid the supervised learning processes of problem determination.
Ada Diaconescu	Proposed a framework that uses component redundancy for enabling self-adaptation, self-optimisation and self-healing capabilities in component-based enterprise software systems.
Janssen <i>et al.</i> ,	Zoltar toolset was discussed, which adopts a technique to localize software faults based on statistical information retrieved from an instrumented version of the program under analysis.
Carzaniga <i>et al.</i>	Develop the idea of automatic workarounds. They proposed a general architecture for the deployment of automatic workarounds and examine its essential requirements.
Thorat <i>et al.</i>	Proposed a self-healing SDN framework which can optimize the recovery by applying autonomic principles. Included a rapid recovery (RR) mechanism to perform an immediate link recovery at the switch level without overburdening the controller.
Stehle <i>et al.</i>	Presented a computational geometry technique and a supporting tool to tackle the problem of timely fault detection during the execution of a software application.

Table 2. Classification of self healing features according to the features and research developed to investigate the feature.

METHOD	Research
Monitoring at runtime	(. W. Cheng, 2004) "architecture-based self-adaptation with reusable infrastructure" B. Schmerl, P. Steenkiste, N. Hu,
	G. Kaiser, 2004 "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" P. Gross, G. Kc, J. Parekh,
	M.G. Merideth "Proactive containment of malice in survivable distributed system" P. Narasimhan, "Proactive containment of malice in survivable distributed system, International Conference on Security and Management"
	J. Aldrich, "Architecturecentric programming for adaptive systems, Proceedings of the First Workshop on Self-Healing Systems" V. Sazawal, C. Chambers, D. Nokin,
	S. Michiels, "Self-adapting concurrency: the DMonA architecture, Proceedings of the First Workshop on Self-Healing Systems" L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten, Grishikashvili,
	I. Georgiadis: Self-organizing software architectures for distributed systems " J. Magee, J. Kramer,
	M.W. Knop," Windows performance monitoring and data reduction using watchtower, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" J. M. Schopf, P.A. Dinda,
	Y. Hong, "Closed loop design for software rejuvenation, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" D. Chen, L. Li, K. Trivedi,
	S. Michiels, "Self-adapting concurrency: the DMonA architecture, Proceedings of the First Workshop on Self-Healing Systems" L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten,
	M. G. Merideth," Proactive containment of malice in survivable distributed system" S. George, D. Evans, S. Marchette, Locasto <i>et al.</i> , E. M. Dashofy, A. V. D. Hoek, R.N. Taylor,
self-management	M. G. Merideth, "Proactive containment of malice in survivable distributed system" P. Narasimhan, "Proactive containment of malice in survivable distributed system, International Conference on Security and Management" E. M. Dashofy, A. V. D. Hoek, R.N. Taylor,
	G. Kaiser, "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" P. Gross, G. Kc, J. Parekh.
	Keromytis "2009Automatic Self-healing and Present and Future" S.W. Cheng, "architecture-based self-adaptation with reusable infrastructure" B. Schmerl, P. Steenkiste, N. Hu, N. Combs, J. Vagle,
	G. Kaiser, "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" P. Gross, G. Kc, J. Parekh. Ada Diaconescu
decision and control	S. Michiels, "Self-adapting concurrency: the DMonA architecture, Proceedings of the First Workshop on Self-Healing Systems" L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten
	S.W. Cheng, "architecture-based self-adaptation with reusable infrastructure" B. Schmerl, P. Steenkiste, N. Hu, Grishikashvili, Huang <i>et al.</i> ,

Continued

feedback control loop, cycle	<p>G.E. Kaiser, "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems"</p> <p>Aldrich <i>et al.</i>, "Architecturecentric programming for adaptive systems, Proceedings of the First Workshop on Self-Healing Systems"</p> <p>C. Dabrowski, K.L. Mills" Understanding self-healing in service discovery systems, Proceedings of the First Workshop on Self-Healing Systems"</p> <p>O. Raz, P. Koopman, M. Shaw,</p> <p>Y. Hong, "Closed loop design for software rejuvenation, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" D. Chen, L. Li, K. Trivedi,</p> <p>B. Schmerl,</p> <p>S.W. Cheng "architecture-based self-adaptation with reusable infrastructure"</p> <p>B. Schmerl, P. Steenkiste, N. Hu, Grishikashvili,</p>
adaptation and reconfiguration	<p>M.W. Knop, "Windows performance monitoring and data reduction using watchtower, Workshop on Self-Healing, Adaptive and self-MAN aged Systems" J. M. Schopf, P.A. Dinda,</p> <p>S. Michiels, "Self-adapting concurrency: the DMonA architecture, Proceedings of the First Workshop on Self-Healing Systems" L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten, Grishikashvili,</p> <p>E.M. Dashofy, A.V.D. Hoek, R.N. Taylor,</p> <p>Jiang, <i>et al.</i></p>
Network based system	<p>C. Dabrowski, K.L. Mills, "Understanding self-healing in service discovery systems, Proceedings of the First Workshop on Self-Healing Systems"</p> <p>PARK, <i>et al.</i></p>
notification	<p>C. Dabrowski, K.L. Mills," Understanding self-healing in service discovery systems, Proceedings of the First Workshop on Self-Healing Systems"</p> <p>M. G. Merideth, "Proactive containment of malice in survivable distributed system"</p> <p>P. Narasimhan, "Proactive containment of malice in survivable distributed system, International Conference on Security and Management"</p> <p>N. H. Minsky,</p> <p>Michael E. Shin</p> <p>Naftaly H. Minsky</p>
Distributed systems	<p>I. Georgiadis, "Self-organizing software architectures for distributed systems"</p> <p>J. Magee, J. Kramer</p> <p>M.G. Merideth, "Proactive containment of malice in survivable distributed system"</p> <p>P. Narasimhan, "Proactive containment of malice in survivable distributed system, International Conference on Security and Management"</p> <p>Dinkel <i>et al.</i></p> <p>N. H. Minsky</p>
Healing on same system	<p>S. Michiels, "Self-adapting concurrency: the DMonA architecture, Proceedings of the First Workshop on Self-Healing Systems" L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten</p> <p>C. Dabrowski, K.L. Mills "Understanding self-healing in service discovery systems, Proceedings of the First Workshop on Self-Healing Systems"</p> <p>S.W. Cheng, "architecture-based self-adaptation with reusable infrastructure" B. Schmerl, P. Steenkiste, N. Hu,</p> <p>I. Georgiadis, "Self-organizing software architectures for distributed systems" J. Magee, J. Kramer</p> <p>G.E. Kaiser, "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems"</p> <p>O. Raz, P. Koopman, M. Shaw,</p>

Continued

	S. George, D. Evans, S. Marchette
	M.G. Merideth, "Proactive containment of malice in survivable distributed system"
	P. Narasimhan, "Proactive containment of malice in survivable distributed system, International Conference on Security and Management"
	B. Schmerl,
	J. Aldrich, "Architecturecentric programming for adaptive systems, Proceedings of the First Workshop on Self-Healing Systems"
	V. Sazawal, C. Chambers, D. Nokin,
Healing on different system	R. Sharman, H.R. Rao, S. Upadhyaya, P. Khot, S. Manocha, S. Ganguly,
	M.W. Knop, "Windows performance monitoring and data reduction using watchtower, Workshop on Self-Healing, Adaptive and self-MAN aged Systems"
	J.M. Schopf, P.A. Dinda
Prediction	Fuad, <i>et al.</i> , "Self-Healing by Means of Runtime Execution Profiling"
	Philip Koopman "Elements of the Self-Healing System Problem Space"
	C. Dabrowski, K.L. Mills "Understanding self-healing in service discovery systems, Proceedings of the First Workshop on Self-Healing Systems"
Self evaluate	G. Kaiser, "An approach to autonomizing legacy systems, Workshop on Self-Healing, Adaptive and self-MAN aged Systems"
	P. Gross, G. Kc, J. Parekh.
Self healing detection	Brumley <i>et al.</i> "Sting: An End-to-End Self-Healing System for Defending against Internet Worms"

Table 3. Classification of self healing features and the name of researchers who have investigated this feature deeply in there researches.

METHOD	Researcher
	S.W. Cheng, D. Garlan, B. Schmerl, P. Steenkiste, N. Hu,
	G. Kaiser, P. Gross, G. Kc, J. Parekh, G. Valetto,
	M.G. Merideth, P. Narasimhan,
	J. Aldrich, V. Sazawal, C. Chambers, D. Nokin,
	S. Michiels, L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten, Grishikashvili,
	I. Georgiadis, J. Magee, J. Kramer,
	M.W. Knop, J.M. Schopf, P.A. Dinda,
	Y. Hong, D. Chen, L. Li, K. Trivedi,
Monitoring at runtime	S. Michiels, L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten,
	Montani <i>et al.</i>
	M.G. Merideth,
	S. George, D. Evans, S. Marchette,
	Locasto <i>et al.</i> ,
	E.M. Dashofy, A.V.D. Hoek, R.N. Taylor,
	M.G. Merideth, P. Narasimhan,
	E.M. Dashofy, A.V.D. Hoek, R.N. Taylor,
	G. Kaiser, P. Gross, G. Kc, J. Parekh, G. Valetto,
	Angelos D.
self-management	Keromytis
	S.W. Cheng, D. Garlan, B. Schmerl, P. Steenkiste, N. Hu,
	N. Combs, J. Vagle,

Continued

	G. Kaiser, P. Gross, G. Kc, J. Parekh, G. Valetto, Ada Diaconescu
decision and control	S. Michiels, L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten S. W. Cheng, D. Garlan, B. Schmerl, P. Steenkiste, N. Hu, Grishikashvili, Huang <i>et al.</i> , G. Valetto, G.E. Kaiser, Aldrich <i>et al.</i> , C. Dabrowski, K.L. Mills O. Raz, P. Koopman, M. Shaw, Y. Hong, D. Chen, L. Li, K. Trivedi, D. Garlan, B. Schmerl, S.W. Cheng, D. Garlan, B. Schmerl, P. Steenkiste, N. Hu, Grishikashvili, M.W. Knop, J.M. Schopf, P.A. Dinda,
feedback control loop, cycle	S. Michiels, L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten, Grishikashvili, E.M. Dashofy, A. V. D. Hoek, R.N. Taylor, Jiang, <i>et al.</i> C. Dabrowski, K.L. Mills, PARK, <i>et al.</i>
adaptation and reconfiguration	C. Dabrowski, K.L. Mills, M.G. Merideth, P. Narasimhan, N.H. Minsky, Michael E. Shin Naftaly H. Minsky
Network based system	I. Georgiadis, J. Magee, J. Kramer M.G. Merideth, P. Narasimhan, Dinkel <i>et al.</i> N. H. Minsky
notification	S. Michiels, L. Desmet, N. Janssens, T. Mahieu, P. Verbaeten C. Dabrowski, K.L. Mills S.W. Cheng, D. Garlan, B. Schmerl, P. Steenkiste, N. Hu, I. Georgiadis, J. Magee, J. Kramer G. Valetto, G.E. Kaiser, O. Raz, P. Koopman, M. Shaw, S. George, D. Evans, S. Marchette M.G. Merideth, P. Narasimhan, D. Garlan, B. Schmerl, J. Aldrich, V. Sazawal, C. Chambers, D. Nokin,
Distributed systems	R. Sharman, H.R. Rao, S. Upadhyaya, P. Khot, S. Manocha, S. Ganguly, M.W. Knop, J.M. Schopf, P. A. Dinda Fuad, <i>et al.</i> , Katti <i>et al.</i> , Philip Koopman C. Dabrowski, K.L. Mills
Healing on same system	G. Kaiser, P. Gross, G. Kc, J. Parekh, G. Valetto, Goutam Kumar Saha. Azim <i>et al.</i> Brumley <i>et al.</i> Stehle <i>et al.</i> Diaconescu, <i>et al.</i> Thorat <i>et al.</i>
Healing on different system	
Prediction	
Self evaluate	
Self healing detection	

Primarily, the development of the Automatic Self-healing software will focus on the following areas:

1. Self-optimization.
2. Self-protection.
3. Self-Configuration [5].

The possibility to develop self-healing software that can be work effectively. At the same time, it works to perfection. a lot of research is required in this direction. This is also the part of complexity engineering [7] [41] [51].

9. Researcher Investigated Self Healing Methodology, Tool, and Technique Developed

Table 1 presented a summary of self healing Methodology, tool, and technique developed and the name of the researchers that have developed them. While **Table 2** shows a classification of self healing features according to the features and research developed to investigate the feature. And **Table 3** shows other authors who investigated the self healing features and it shows a classification of self healing features and the name of researchers who have investigated this feature deeply in there researches [41].

10. Summary

There are different features that characterize the self healing mechanisms; different researches explored these researches. Looking at the tables provided above, it is clear that the Self-healing software can be a reality as many researchers have investigated this field deeply as it took the intention of many top researches and researchers. Although some precious standards are required to ensure that application's participative approach can be very useful in this direction.

References

- [1] Avizienis, A., Gilley, G., Mathur, F., Rennels, D., Rohr, J. and Rubin, D. (1971) The STAR (Self-Testing and Repairing) Computer: An Investigation of the Theory and Practice of Fault-Tolerant Computer Design. *IEEE Transactions on Computers*, **20**, 1312-1321. <https://doi.org/10.1109/T-C.1971.223133>
- [2] Koldehofe, B., Mayer, R., Umakishore, Rothermel, K. and Völz, M. (2013) Roll-back-Recovery without Checkpoints in Distributed Event Processing Systems. *DEBS'13*, Arlington, 29 June-3 July 2013.
- [3] Krena, B., Letko, Z., Tzoref, R., Ur, S. and Vojnar, T. (2007) Healing Data Races On-the-Fly. *PADTAD'07*, London.
- [4] Dabrowski, C. and Mills, K.L. (2002) Understanding Self-Healing in Service Discovery Systems. *Proceedings of the First Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 15-20. <https://doi.org/10.1145/582128.582132>
- [5] Saran, C. (2003) Could Self-Healing Software Be the IT Director's Way of Cutting Support Costs? <http://www.computerweekly.com/feature/Could-self-healing-software-be-the-IT-directors-way-of-cutting-support-costs>
- [6] Perez, C.-R., Stelios, S., Laadan, O., Viennot, N., Keromytis, A. and Nieh, J. (2009) Automatic Self-Healing and Present and Future. 23 June 2009.

- [7] Weyns, D., Haesevoets, R., Van Eylen, B., Helleboogh, A., Holvoet, T. and Joosen, W. (2008) Endogenous versus Exogenous Self-Management. *SEAMS'08*, Leipzig, 12-13 May 2008.
- [8] Ardagna, D., Cappiello, C., Fugini, M.G., Mussi, E., Pernici, B. and Plebani, P. (2006) Faults and Recovery Actions for Self-Healing Web Services.
- [9] Li, D., Tran, A.H. and Halfond, W.G.J. (2014) Making Web Applications More Energy Efficient for OLED Smartphones. *ICSE'14*, 31 May-7 June 2014, Hyderabad.
- [10] Lee, D., Yoo, J., Kang, H., Kim, K. and Kang, K. (2006) Distributed IPv6 Addressing Technique for Mobile Ad-hoc Networks. *SAC'06*, 23-27 April 2006, Dijon.
- [11] Athanasopoulos, D., Zarras, A.V., Vassiliadis, P. and Issarny, V. (2011) Mining Service Abstractions: NIER Track. *2011 33rd International Conference on Software Engineering*, Honolulu, 21-28 May 2011, 944-947.
- [12] Frei, R., McWilliam, R., Derrick, B., Purvis, A., Tiwari, A. and Serugendo, G.D.M. (2013) Self-Healing and Self-Repairing Technologies. *The International Journal of Advanced Manufacturing Technology*, **69**, 1033-1061.
<https://doi.org/10.1007/s00170-013-5070-2>
- [13] Ghosh, D., Sharman, R., Rao, H.R. and Upadhyaya, S. (2007) Self-Healing Systems—Survey and Synthesis. *Decision Support Systems*, **42**, 2164-2185.
<https://doi.org/10.1016/j.dss.2006.06.011>
- [14] Brahme, D. and Abraham, J.A. (1984) Functional Testing of Microprocessors. *IEEE Transactions on Computers*, **C-33**, 475-485.
<https://doi.org/10.1109/TC.1984.1676471>
- [15] Janssen, T., Arjan, R.A., van Gemund, J.C. and Zoltar, A. (2009) Spectrum-Based Fault Localization Tool. *SINTER'09*, Amsterdam, 25 August 2009.
- [16] Dashofy, E.M., Hoek, A.V.D. and Taylor, R.N. (2002) Towards Architecture-Based Self-Healing Systems. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 21-26. <https://doi.org/10.1145/582128.582133>
- [17] Frei, R. (2010) Self-Organisation in Evolvable Assembly Systems. PhD Thesis. Department of Electrical Engineering, Faculty of Science and Technology, Universidade Nova de Lisboa, Portugal.
- [18] Hudaib, A.A. and Fakhouri, H.N. (2016) An Automated Approach for Software Fault Detection and Recovery. *Communications and Network*, **8**.
- [19] Blair, G.S., Coulson, G., Blair, L., Limon, H.D., Grace, P., Moreira, R. and Parlavantzias, N. (2002) Reflection, Self-Awareness and Self-Healing in Open ORB. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 9-14. <https://doi.org/10.1145/582128.582131>
- [20] Pasier, H. and Dustdar, S. (2011) A Survey on Self-Healing Systems: Approaches and Systems. *Computing*, **91**, 43-73.
- [21] Diaconescu, A. (2003) A Framework for Using Component Redundancy for Self-Adapting and Self-Optimising Component-Based Enterprise Systems. *OOPSLA'03*, Anaheim, 26-30 October 2003.
- [22] Thatte, S.M. and Abraham, J.A. (1978) A Methodology for Functional Level Testing of Microprocessors. *Highlights from Twenty-Five Years, Twenty-Fifth International Symposium on Fault-Tolerant Computing*, 27-30 June 1995, 326.
- [23] Georgiadis, J. and Kramer, M.J. (2002) Self-Organizing Software Architectures for Distributed Systems. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 33-38. <https://doi.org/10.1145/582128.582135>
- [24] Zhou, J. and Wunderlich, H.-J. (2006) Software-Based Self-Test of Processors under

- Power Constraints. *Proceedings of Design, Automation and Test in Europe*, Vol. 1, Munich, 6-10 March 2006, 1-6.
- [25] Carzaniga, A., Gorla, A. and Pezzè, M. (2008) Self-Healing by Means of Automatic Workarounds. *SEAMS'08*, Leipzig, 12-13 May 2008.
- [26] Trumler, W., Bagci, F., Petzold, J. and Ungerer, T. (2005) AMUN—Autonomic Middleware for Ubiquitous Environments Applied to the Smart Doorplate Project. *Advanced Engineering Informatics*, **19**, 243-252.
<https://doi.org/10.1016/j.aei.2005.04.004>
- [27] Schneider, C., Barker, A. and Dobson, S. (2013) A Survey of Self-Healing Systems Frameworks. *Software: Practice and Experience*, **45**, 1375-1398.
<https://doi.org/10.1002/spe.2250>
- [28] George, S., Evans, D. and Marchette, S. (2003) A Biological Programming Model for Self-Healing. *1st ACM Workshop on Survivable and Self-Regenerative Systems*, Fairfax, 31 October 2003, 72-81. <https://doi.org/10.1145/1036921.1036929>
- [29] Valetto, G. and Kaiser, G.E. (2002) Case Study in Software Adaptation. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 73-78. <https://doi.org/10.1145/582128.582142>
- [30] Aldrich, J., Sazawal, V., Chambers, C. and Nokin, D. (2002) Architecture Centric Programming for Adaptive Systems. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 93-95.
<https://doi.org/10.1145/582128.582146>
- [31] Merideth, M.G. and Narasimhan, P. (2003) Proactive Containment of Malice in Survivable Distributed System. *International Conference on Security and Management*, Las Vegas, June 2003, 3-9.
- [32] Knop, M.W., Schopf, J.M. and Dinda, P.A. (2002) Windows Performance Monitoring and Data Reduction Using Watch Tower. *Workshop on Self-Healing, Adaptive and Self-MANaged Systems (SHAMAN)*, New York City, 23 June 2002.
- [33] Michiels, S., Desmet, L., Janssens, N., Mahieu, T. and Verbaeten, P. (2002) Self-Adapting Concurrency: The DMonA Architecture. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 43-48.
<https://doi.org/10.1145/582128.582137>
- [34] Fuad, M.M., Deb, D. and Baek, J. (2012) Self-Healing by Means of Runtime Execution Profiling. *Proceedings of 14th International Conference on Computer and Information Technology (ICCIT 2011)*, Dhaka, 22-24 December 2011, 202-207.
- [35] Shin, M.E. (2005) Self-Healing Component in Robust Software Architecture for Concurrent and Distributed Systems. *Science of Computer Programming*, **57**, 27-44. <https://doi.org/10.1016/j.scico.2004.10.003>
- [36] Koopman, P. (2003) Elements of the Self-Healing System Problem Space. *Proceedings of Workshop on Architecting Dependable Systems WAD'03*, Portland, 1-6.
- [37] Jiang, M., Zhang, J., Raymer, D. and Strassner, J. (2007) A Modeling Framework for Self-Healing Software Systems.
- [38] Combs, N. and Vagle, J. (2002) Adaptive Mirroring of System of Systems Architectures. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 96-98. <https://doi.org/10.1145/582128.582147>
- [39] Raz, O., Koopman, P. and Shaw, M. (2002) Semantic Anomaly Detection in Online Data Sources. *Proceedings of the 24th International Conference on Software Engineering (ICSE'02)*, Orlando, 25 May 2002, 302-312.
<https://doi.org/10.1145/581376.581378>
- [40] Raz, O., Koopman, P. and Shaw, M. (2002) Enabling Automatic Adaptation in Sys-

- tems with Under-Specific Elements. *Proceedings of the 1st Workshop on Self-Healing Systems*, Charleston, 18-19 November 2002, 55-60.
<https://doi.org/10.1145/582128.582139>
- [41] Hong, Y., Chen, D., Li, L. and Trivedi, K. (2002) Closed Loop Design for Software Rejuvenation. *Workshop on Self-Healing, Adaptive and Self-Managed Systems (SHAMAN)*, New York, 23 June 2002.
- [42] Al Zaghoul, F., Rababah, O. and Fakhouri, H. (2014) Website Search Engine Optimization: Geographical and Cultural Point of View, Computer Modelling and Simulation (UKSim). 2014 *UKSim-AMSS 16th International Conference*.
- [43] Katti, A., Di Fatta, G., Naughton, T. and Engelmann, C. (2015) Scalable and Fault Tolerant Failure Detection and Consensus. *EuroMPT15*, Bordeaux, 21-23 September 2015.
- [44] Ehlers, J., van Hoorn, A., Waller, J. and Hasselbring, W. (2011) Self-Adaptive Software System Monitoring for Performance Anomaly Localization. *ICAC11*, Karlsruhe, 14-18 June 2011.
- [45] Sánchez, J., Ben Yahia, I.G. and Crespi, N. (2014) POSTER: Self-Healing Mechanisms for Software-Defined Networks.
- [46] Locasto, M.E. (2007) Self-Healing: Science, Engineering, and Fiction. *NSPW'07*, North Conway, 18-21 September 2007.
- [47] Anand, M., Chouhan, K., Ravi, S. and Ahmed, S.M. (2011) Context Switching Semaphore with Data Security Issues using Self-Healing Approach. *International Journal of Advanced Computer Science and Applications*, **2**.
<https://doi.org/10.14569/IJACSA.2011.020608>
- [48] Gorla, A., Pezzè, M. and Wuttke, J. (2010) Achieving Cost-Effective Software Reliability through Self-Healing. *Computing and Informatics*, **2**, 1001-1022.
- [49] Dustdar, P.S. (2011) A Survey on Self-Healing Systems: Approaches and Systems. **91**, 43-73.
- [50] Abdullah, A., Candrawati, R. and Bhakti, M.A.C. (2009) Multi-Tiered Bio-Inspired Self-Healing Architectural Paradigm for Software Systems. *Jurnal Teknologi Maklumat & Multimedia*, **5**, 1-24.
- [51] Park, J., Youn, H. and Lee, E. (2009) An Automatic Code Generation for Self-Healing. *Journal of Information Science and Engineering*, **25**, 1753-1781.
- [52] Brumley, D., Newsome, J. and Sting, D.S. An End-to-End Self-Healing System for Defending against Internet Worms.
- [53] George, S., Evans, D. and Davidson, L. (2002) A Biologically Inspired Programming Model for Self-Healing Systems. *WOSS'02*, Charleston, 18-19 November 2002.

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact cn@scirp.org