

# Self-Healing Mechanism for Reliable Architecture with Focus on Failure Detection

**Eslam Nazemi**

Assistant Professor

Department of Computer engineering Shahid Behesti University, Tehran, Iran  
Email: [eslam.nazemi@sbu.ac.ir](mailto:eslam.nazemi@sbu.ac.ir)

**Tahere Talebi**

Department of Computer engineering Shahid Behesti University, Tehran, Iran  
Email: [Kt.talebi@gmail.com](mailto:Kt.talebi@gmail.com)

**Hadi Elyasi**

Department of E-Banking Services of ICT Sina Bank, Tehran, Iran  
Email: [h.elyasi@sinabank.ir](mailto:h.elyasi@sinabank.ir)

**Abstract**—Many companies are embracing e-commerce, or business conducted on-line over electronically networks, as a means of expanding markets, improving customer service, reducing costs, and enhancing productivity. Like other companies, banks are taking steps to expand the use of computer science technology in their business operations. On the other hand, in recent years, the adoption of e-banking began to occur quite extensively as a channel of distribution for financial services due to rapid advances in IT and intensive competitive banking markets. For a bank that wants to remain in the competition, it's more important that each e-banking service have some features such as reliability, availability, extensibility. So we have represented a new e-banking framework which used a biology concept so called self-healing. In our framework, we presented some additional services respect to self-healing cycle for decrease the failure frequency and consequently increase services reliability.

**Index Terms**—E-Banking, Self-Healing, Availability, Quality of Service.

## I. INTRODUCTION

Majority of companies are looking for common goals such as expanding markets, improving customer service, reducing costs and enhancing productivity. Therefore to reach these concepts, they decided gait to "e-commerce" or business conducted on-line and implement the properties of it in their systems. Banks are taking steps to expand the use of networking technology in their business operations, like other companies. However, banks are beginning to use the electrical technology to deliver traditional banking products in more efficient ways. In addition, some banks have taken the further step of developing new products designed specifically to facilitate e-commerce participation by their customers. Therefore respect to these reasons any banks or financial

institutes desire to trading through e-banking. On the other hand, despite to electronic banking is becoming an important aspect of worldwide commerce; in some countries it hasn't enough acceptability, especially in developing countries. In these countries, the future of this concept appear bright but merchants and customers face some barriers for substantiation it such as less reliable telecommunications infrastructure and power supplies, less access to online payment mechanisms, and relatively high costs for personal computers and Internet access for some banking services.

Respect to Intense competition in the banking industry and their effort for retaining the existing customers as well as attracting new ones enforce them to develop e-Banking in a comprehensive service infrastructure strategy [1]. Many definition have exist for e-banking such as "e-Banking is the delivery of banks' information and services by banks to customers via different delivery platforms that can be used with different terminal devices such as a personal computer and a mobile phone with browser or desktop software, telephone or digital television"[2].

In banking term, e-Banking allows customers to perform many simple banking functions anytime and anywhere. Although banks may wish their customers migrate most of their transactions to the lower-cost e-Banking channels, customers often mix their use of multiple channels in a way not necessarily in line with what the banks may want. In customers' view point; it's so important that banking operations to be available round the clock, 365 days in a year despite planned or unplanned outages. Many banks have geographically dispersed data centers that need information across the centers. To achieve this, banking platforms must be robust enough to provide "five nines" of availability or the system should be available for 99.999% of the time in a year. Five nines mean that a user is ensured practically no loss of service due to hardware or software errors, or any downtime for upgrades or hardware maintenance.

Several previous studies show that the availability of new service channels or different service infrastructure change consumers' behaviors. For example, [3] has shown that consumers' adoption of e-Banking motivates them to open more accounts and increase their overall demand for banking services, suggesting that there exists some strategic synergy among different channels that banks can explore further to obtain a competitive advantage[1].

As a result of the electronic banking system addresses several emerging trends: customers' demand for anytime, anywhere service, product time-to-market imperatives and increasingly complex back-office integration challenges. Some properties have special importance degree in electronic banking are: Security, privacy, Convenience, design, content, speed, fees and charges, accessibility, reliability, QoS<sup>1</sup>, etc. moreover there are some factors that cause to customers satisfaction such as satisfaction, loyalty, WoM<sup>2</sup>. In this paper we focus on two mentioned parameters of adopting e-Banking which are accessibility and quality of services.

#### A. Accessibility

Accessibility is defined as the ability of users to access information and services from any terminal which is dependent on many factors. These include the content format; the user's hardware, software and settings; sufficient and confident connections; the environmental conditions and the user's abilities and disabilities [2, 4].

#### B. QoS

There exist several Quality models for Systems, one of the most relevant was established in [5] classifying the quality in a structured set of characteristics and sub-characteristics as follows:

- **Functionality:** Suitability, Accuracy, Interoperability, Compliance, Security
- **Reliability:** Maturity, Recoverability, Fault Tolerance
- **Usability:** Learnability, Understandability, Operability
- **Efficiency:** Time Behavior, Resource Behavior
- **Maintainability:** Stability, Analyzability, Changeability, Testability
- **Portability:** Installability, Replaceability, Adaptability, Conformance

If a system observes Service quality conditions, then it can reach maximizing customer satisfaction. So the underlying motive is that satisfied customers yield greater profits. It seems in Chain Reaction of Customer Satisfaction which shown in "Fig. 1".

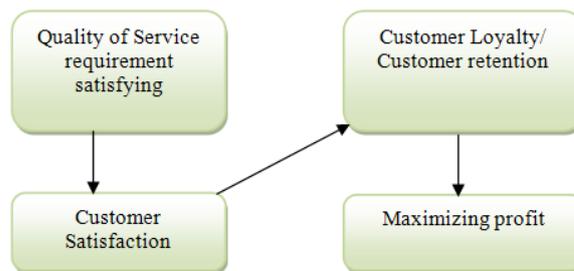


Fig. 1. Chain Reaction of Customer Satisfaction

For covering quality of services' requirement, we used a biologist concept so called self-healing which introduce in next section.

## II. SELF-HEALING

Self-healing is a phrase applied to the process of recovery<sup>3</sup>, motivated by and directed by the patient, guided often only by instinct. Self-healing is a mechanism that detect and repair problems of functionality and services offered by electronically systems. In some systems which based on single components and services, it's usually easy to test and analyze, In contrast complex systems suffer from unexpected integration problems, can lead to subtle run time problems difficult to prevent and repair. Self-healing mechanisms focus on detecting and overcoming such problems. As a result of, self-healing has evolved as a property of IT systems that enables such systems to diagnose themselves and react to faults such that a satisfactory mode of operation is restored after the occurrence of a fault or a failure. So we use bellow cycle for recover failures with minimum dependency to expert person "Fig. 2" shows a self-healing cycle.

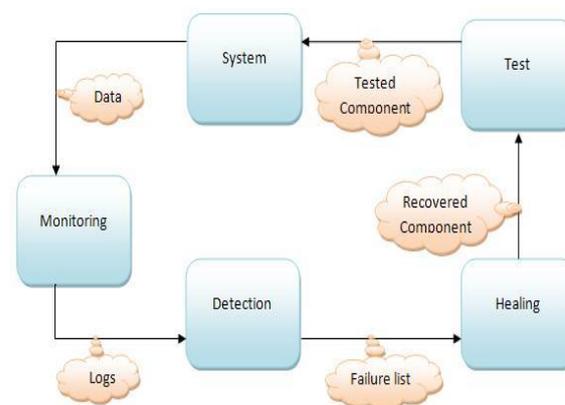


Fig. 2. Self-Healing Cycle

## III. SELF-HEALING ON E-BANKING SERVICES

Respect to high rate of banking transaction, it's essential on e-banking concept that software and

<sup>1</sup> Quality of service

<sup>2</sup> Word of Mouth

<sup>3</sup> generally from psychological disturbances, trauma, etc

hardware practical Throughput should be enough. So system could be achieved to maximum of availability and therefore customers never lose their confidence to system. Because it's more important for them to on every referral to every banking service, it's working correctly.

When using self-healing cycle on a system, SOA<sup>4</sup> has the best feedback between current architectures. Because each part of the system considered as a service, so managing and failure detection will be easier than other architecture such as CBA<sup>5</sup> or AOA<sup>6</sup>. But e-banking model of more than half of Iranian Banks based on centralized banking which system elements managed as centralized respect to security importance degree. It's better that upgrade these models to SOA, but because they use CBA model, we focused on self-healing on centralized architecture. "Fig. 3" has shown the bank schema as follow:

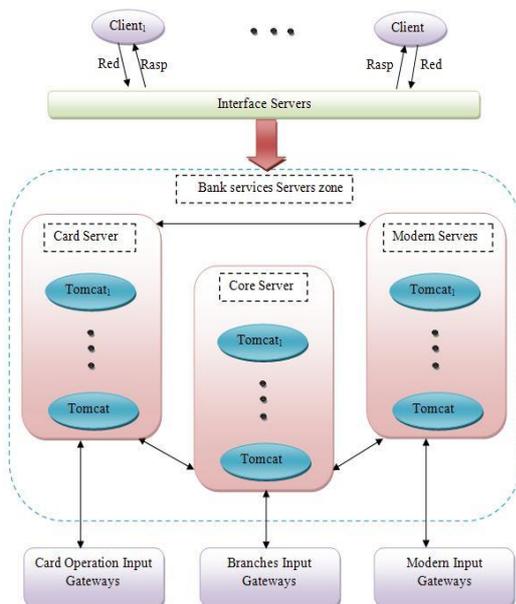


Fig. 3. E-Banking Schema

So according to the top schema, we can classify failures as "Fig. 4" Respect to this classification, failures which have occurred on banking service divided to two categories:

- **Business failure:** failures which occur for users mistake or business restricts on transaction time. According to ISO-8583 standards, these failures are listed in appendix 1.
- **System failure:** some failures that fall out on Network infrastructure, switches, connection to centralized system or connection to other channel such as SHETAB<sup>7</sup> and interface switches. These are listed as appendix 2.

<sup>4</sup> Service Oriented Architecture

<sup>5</sup> Component Based Architecture

<sup>6</sup> Aspect Oriented Architecture

<sup>7</sup> inter-banking information communication network in Iran

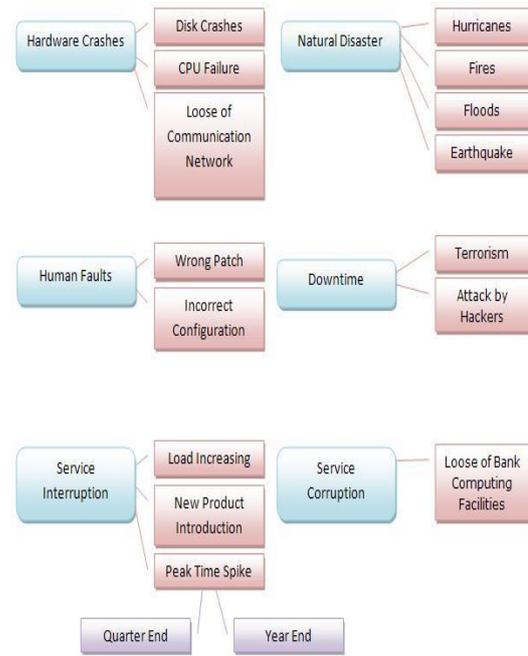


Fig. 4. Failure Classification

Note that knowledge of customers on e-banking in developed countries is more than developing countries. So business failures have more frequency than system failures. TABLE 1 has shown Sina-Bank transaction status between 21 June 2012 and 21 July 2012.

Table 1. Sina-Bank Transaction Status Between 21 June 2012 And 21 July 2012.

Response Code	Response Text	Transaction Count
6	Error	27
12	Invalid Transaction	61977
25	Unable to locate record on file	13756
30	Format Error	985
31	Bank not supported by switch	36
68	Response received too late	57832
77	Slm-Invalid business date	9881
80	Slm-Transaction denied	14774
84	Slm-Issuer down	284640
87	Pin key sync error	2138
90	Cut off is in progress	35799
91	Issuer or switch is inoperative-decline	301366
93	Transaction Cannot be complete	41220
94	Duplicate transmission	903
96	System malfunction	49493

Respect to this table, we define availability as ratio of the time of service to total time. Mentioned definition is accordance with the concepts which expressed on section I.A. Therefore availability is essential condition for failure validation. For example inter-banking switches couldn't comprehend that they are out of service. So if switches availability is higher, then presented statistics from switches view point have greater creditability. Availability computes on three indexes: internal, issuer, acquire that the result of them shown in Table II. It is noteworthy that our goal isn't only decreasing the failures that lead to banking service be unavailable.

Table 2. Service Availability on Sina Bank

Service Mode	Availability	Out of service time	Failure percent	Number of transactions
Internal	99.92	31:21	0.22	802722
Issuer	99.32	299:20	0.11	3022239
Acquire	99.25	333:29	4.09	4213461

According to statistics of Table 2, the aim of this paper will be increasing availability by minimize the system failures. These failures identified by expert persons and they will eliminate these detected failures. Afterward when they tested healed failure components and sure about their health practice, returns them to system cycle. Table 3 is shown the result of transaction monitoring on mentioned period which involves number and percent of transactions that lead to faults.

Table 3. Approved VS Not Approved Transaction Status

Transactions	Count	Percent (of all)	Percent(from all not approved)
Approved transactions	20050215	81.9%	-
Not Approved Transaction	4413908	18%	-
Not Approved Transactions(Business Faults)	3594861	14.7%	81%
Not Approved Transactions(System Faults)	819047	3%	18%
All transactions	24464123	-	-

Table 3 is expressed that only 3% of transactions lead to system fault that some of them can be a failure. Nearly 0.02 of these transactions are in issuer mode that we never need to know what are these; because these faults have to considered by issuer banks and we are going to eliminate faults that maybe lead to failures in acquire and internal mode. Thus using by Table I and expert persons knowledge, we have established a small failure table as follows:

Table 4. A Sample Of Fault Table.

failure code	fault code	offset	fault description	healing solutions
1	68	1	Stop one of network hardwires	Found defective part and repair or change it
2	93	1	Stop one of terminal hardware	Repair or change the defective part
3	93	2	Could not fetch customer info from database	The customer database is down and you should stop/start it
4	96	1	Could not send info to banking service layer	Stop/start the interface server
5	96	2	Could not receive info from banking service layer	Stop/start the tomcat which has problem

Afterward we are described self-healing steps in our case study on the following sections.

A. Monitoring

Note to architecture was mentioned, there are differential methods to collect needed data for detection phase. Respect to problem model, we have used a method that collect data from system using by definition of some metrics; at the first we identify system attributes and store them into a database that is called Attribute Database. Afterward, we define some metrics according to system requirement and knowledge of expert persons. We are called them System Metrics and used attributes from attribute database for metric definition. For using these metrics on portions of system, it's needed to define some Metric Protocols.

$$\{M|M \in \text{Metrics AND } \exists A \in \text{Attributes AND } \exists C \in \text{Conditions}:(A:C)\}$$

$$\{p|p \in \text{Protocols AND } \exists M \in \text{Metrics AND } \exists sc \in \text{sys}_{cpnt}:(m, \{sc\})\}$$

For these values, we have:

$$\text{Attributes: } \{a_1, a_2, \dots, a_n\}$$

$$\text{Conditions: } \{c_1, c_2, \dots, c_m\}$$

$$\text{Metrics: } \{M_1: \{a_1:c_1; a_2:c_2; a_3:c_3\}, \dots, M_l: \{a_2:c_6; a_{n-4}:c_4; a_n:c_9\}\}$$

$$\text{Sys}_{cpnt}: \{sc_1, sc_2, \dots, sc_k\}$$

Which  $c_1, c_2, \dots, c_m$  are conditions that assign to some attributes and  $\text{sys}_{cpnt}$  is the list of system components that metrics should apply on them. Then protocols have been defined as bellow.

$$\text{Protocols: } p_1(m_1, \{sc_3, sc_{11}, sc_{k-1}\}), p_2(m_4, \{sc_1, sc_8, sc_k\}), \dots, p_2(m_{1-1}, \{ \})$$

For each protocol, if system components set be null, so indicates that the slightly metric should be apply for all components in system. Respect to our banking system, we can map mentioned elements as following:

Attributes:

- Server response time for sand packets
- Database response time
- The accuracy of the computational performance of banking services
- Send interest deposit message to customers

Metrics:

- Server response time for sand packets should not be higher than 2 seconds
- Database server response time should not be higher than 1 second,
- Tell-Bank service should not be assign to a legal customer
- Every cell phone number have to assign to only one customer
- Send an interest deposit message with amount less than 1,000,000 R

**Metric Protocols:**

- (Every cell phone number should assign to only one customer :{Mobile bank on USSD basis})
- (Server response time for sand packets should not be higher than 2 seconds :{ }),

(Send an interest deposit message with amount less than 1,000,000 R :{CHAPAR<sup>8</sup>, message creator monitoring service})

When our definitions have been completed, we establish some components that have involved multiple methods for using these protocols. We are called them Environmental Components that have duty to collect data from system and decision about which of these data can be useful as for defended metrics that used on protocols, will done on next phase. Environmental components execute respect to several plans and organization policies on specified periods. For example we can define following components:

Component name	method	Params	Description
Customer	Customer_Info( )	CID	Get a test customer Info by his/her ID
Customer	Customer_Info( )	IDNO	Get a test customer Info by his/her IDNO
Customer	Customer_Deposit_Info()	CID	Get Deposit info for a test customer by his/her ID
Customer	Customer_Deposit_Info()	DNO	Get Deposit info for a test customer by his/her Deposit number
Message	Send_Interest_Deposit()	CID	Send a message with amount less than 1,000,000R for a test customer
Network	Telnet()	Node_IP	Broadcast telnet packet for nodes and get their response

So while data gathering is completed, data logs are transferred to detection phase.

**B. Detection**

The data collected from the previous phase are interpreted by different methods and identified critical cases. V-Cure apply a database consist of system normal status for each components and compare monitoring results with these status and identify suspicious cases which can be lead to failure. But respect to the scale of Banking system, we apply metric protocols. So after detect system parts which have conditions such as failure condition, we should apply healing methods for them. But note to quantity of detected cases and low importance of most of them, it's necessary to cluster these cases and then clusters with high degree importance are selected for applying recovery action.

<sup>8</sup> Message manager service

$$\{C|C \in Clusters \text{ AND } \exists A, B \in Attributes : (C:A, B)\}$$

It should be mentioned that clustering weighting coefficient is determined by expert persons. For our case study, we can cluster monitored evidence as following:

- $C_1$ : calculating interest of long term deposits
- $C_2$ : time delay on send and receive messages, time delay on Database connection
- $C_3$ : send message for interest deposit less than 1000000 to customers who are one member of group with limited conditions

After mapping monitoring data to respective cluster, clusters with higher importance degree transfer to healing phase to do recovery action on them.

**C. Healing**

After failure identification on detection phase, it's the turn of healing on these diagnosis failures. Respect to previous sections, we can find feasible solution using by failures table and the component can be healed. Hence, different approaches to achieve this goal are available which some of them is listed below. Most of the already existing self-healing solutions differ in the intercession mechanisms that they use. So, there are some failure recovery techniques that listed in following:

• **Redundancy Techniques**

[Self-Healing and Recovery Methods and their Classification] Some self-assembly mechanism based on an agent entity that replicates components to replace dead neighbors and enables recomposition of entire structures. Another strategy inspired from biology is providing the system with the ability of replicating cells in excess to combat external intrusions [8]. One of the techniques used in Recovery-Oriented Computing (ROC) consists on isolating faulty components and replacing them with redundant ones [6].

• **Architecture Models and Policies**

Some component-based frameworks support interchangeable architectural styles to suit performance deviations. Two examples are Rainbow [7] and Madam [8] Rainbow statically associates a set of action rules for each of the pre-identified failure causes. Madam uses some utility functions to select the most suitable architectural variant to repair the fault. You can see another kind of architecture models which represented by researcher on [9-12].

• **Component Micro-Rebooting**

Faulty modules are micro-rebooted independently and automatically to avoid fault propagation whenever they are suspected of not functioning properly. The efficiency of this technique resides on the fact that re-starting single components takes less time than rebooting the whole system [13, 14].

There are some other recovery action such as *SOA-based Process Reorganization* [15], *voting methods for*

*healing/Byzantine agreement* [16], etc. Respect to our model, we use a rainbow model to recover faults. Before any action we should create a fault table like as Table IV. If there isn't any solution for a detective failure, the system can notify it to expert person. Then s/he can add it to fault table so system can handle this failure if it occurred later.

#### D. Test

After recovery action on component leads to failure, we use some test scenario to ensure that there is no negative impact on other component. If the test result is ok, recovered component will be returned to system; otherwise system should notify system supporter.

Respect to this platform, system can run a recovery action about every fault that leads to failure and by this solution; there is no need for an expert person for some failures which can recover failure using self-healing cycle.

## IV. COMPARISON

On this section, we compare our case study model<sup>9</sup> in both moods<sup>10</sup>. Now suppose defect elimination that used by acquire and internal systems, used by issuer systems too. So failures will be decrease on both side and we can have a system with high availability capability. If we investigate banking system according to faults table on self-healing architecture, thus comparing the number of failure is shown in Fig. 5.

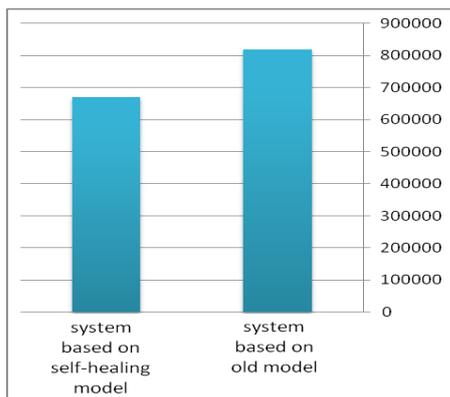


Fig. 5. Comparison between the number of failure before and after using self-healing method

Fig. 6 is shown result at the optimized status. If we consider some other factor then number of detected failures which were healed, will be changed. For example different fault sequence lead to a failure and only some of them was/were identified and used for metrics on detection phase. Therefore remained failure should be identified by expert persons and pattern recognition methods and added them to metrics.

<sup>9</sup> Banking model

<sup>10</sup> Traditional banking model and Self-healing banking model

So respect to existence of these cases, we should define a *handle coefficient* that by using it, number of self-healed failure be considered variable:

$$F_{b-\alpha} \leq F_h \leq F_{b-\beta} \text{ where } \alpha \leq \beta \text{ and}$$

$F_b$ : number of failure before using of self-healing cycle

$F_h$ : number of failure after using of self-healing cycle

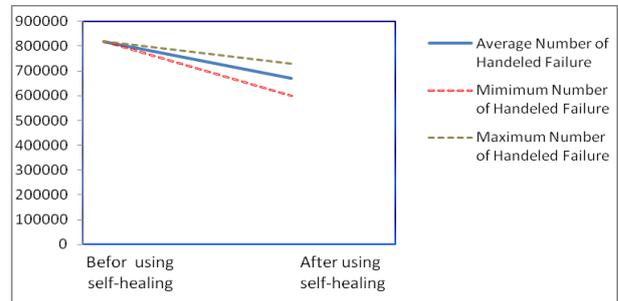


Fig. 6. Changing rate note to handle coefficient

As a result of, despite existence of such intelligent systems, need to expert persons will be sensible.

## V. CONCLUSION AND OUTLOOK

In our work we analyze the banking framework and classify misbehavior of these systems. So we represented a new banking framework based on self-healing which is a psychological terminology. Respect to healing cycle we used some recovery action for failures and the result indicated that we can decrease the time of system return to safe mode. Thus we could have more reliable systems than the systems without using self-healing in customers view point. We are used several metrics and metric protocols for identified system portions that have problem and afterwards we cluster them for give priority to evidences. Thus the system applies recovery action for detected failure. But despite all these, we couldn't omit expert persons from systems. Future work will involve the integration of failure prediction using learning methodology and recovery actions for failures. Also we can increase prediction optimization by knowledge discovery on faults collection.

## REFERENCES

- [1] F. Calisir, C. A. Gumussory, "Internet Banking versus Other Banking Channels: Young Consumers' View," *International Journal of Information Management*, pp 215-221, 2008.
- [2] H. A. Al-Zu'bi, A. M. Ahmad, "E-banking Functionality and Outcomes of Customer Satisfaction: An Empirical Investigation," *International Journal of Marketing Studies* Vol. 3, pp. 50-65 February 2011.
- [3] M. Xue, L. M. Hitt, P. Y. Chen, "The Determinants and Outcomes of Internet Banking Adoption," *Management Science* (Forthcoming), 2011.
- [4] B. Goodwin-Jones, "Emerging technologies – accessibility and web design why does it matter?" *Language Learning and Technology*, pp. 11-19, 2001.
- [5] M. O. Hilari, "Quality of Service (QoS) in SOA Systems: A Systematic Review", 2009.

- [6] Berkeley/Stanford, "Recovery-Oriented Computing (ROC)," <http://roc.cs.berkeley.edu>, 2008
- [7] D. Garlan, S. W. Cheng, A. C. Huang, B. Schmerl, P. Steenkiste, "Rainbow: Architecture-based self adaptation with reusable infrastructure," *IEEE Computer*, vol. 37, pp. 46-54, October 2004.
- [8] J. Floch, S. Hallsteinsen, E. Stav, F. Eliassen, K. Lund, and E. Gjørven, "Using architecture models for runtime adaptability," *IEEE Software*, vol. 23, pp. 62-70, 2006.
- [9] E. M. Dashofy, A. v. Hoek, R. N. Taylor, "Towards architecture-based self-healing systems," in *WOSS'02: Proceedings of the first workshop on Self-healing systems*, New York, NY, USA, pp. 21-26, 2002.
- [10] M. E. Shin, J. H. An, "Self-reconfiguration in self-healing systems," in *EASE'06: Proceedings of the Third IEEE International Workshop on Engineering of Autonomic & Autonomous Systems*, Washington, DC, USA, pp. 89-98, 2006.
- [11] S. Montani, C. Anglano, "Achieving self-healing in service delivery software systems by means of case-based reasoning," *Applied Intelligence*, vol. 28, pp. 139-152, 2008.
- [12] M. M. Fuad, M. J. Oudshoorn, "Transformation of existing programs into autonomic and self-healing entities," in *ECBS'07: Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, Washington, DC, USA, pp. 133-144, 2007.

### Authors' Profiles



**Eslam Nazemi** was born in Sarab, Iran, in 1954. He got the BSc. degree in Applied Mathematics and Operational Research from School of Planning and Computer Application, Tehran, Iran in 1977, The MSc. degree in Both System Engineering and Economics in 1987 and 1996, and PhD. in Industrial engineering and Information technology in 2005, Iran.

He was the faculty Member from 1978 in School of Planning and Computer application and then from 1986 to the present, he has been with the Computer engineering Faculty at Shahid Beheshti University (SBU), Tehran, Iran. He was deputy of graduate and education affairs and now is the manager of informatics development of education in SBU. He is an Assistant Professor of Computer Engineering Department. His main fields of research are Self-\* Software Engineering, Large Scale Software Development, Search engines, Web Mining, and Self- Adaptive Software quality. He has authored and coauthored more the 90 papers in Journals and Conferences and has 10 books on mathematics, project management, software engineering, Software Quality and game theory.



**Tahere Talebi** was born in Qaemshahr, Iran, in 1986. She got the BSc. degree in Computer Engineering from Islamic Azad University of Sari, Iran on 2010, The MSc. degree in Software Engineering on 2014 from Shahid Beheshti University (SBU), Iran. She was a member of Self-\* Group in SBU and an employee of MA Insurance

Company.



**Hadi Elyasi** was born in Tehran, Iran, in 1985. He got the BSc. degree in Computer Engineering from Islamic Azad University of Sari, Iran on 2009, The MSc. degree in Software Engineering on 2011 at Isfahan University of Technology, Iran. He was worked in some companies with financial business as a web designer and now he is an employee of SINABANK as Test unit developer, android and iOS.

Appendix 1. Business Failures

Response Code	Response Text
2	Refer To Card Issuer Special Conditions
13	Invalid Amount
14	Invalid Card Number
17	Customer Cancellation
20	Invalid Response
33	Expired Card – Pick Up
34	Suspected Fraud – Pick Up
36	Restricted Card – Pick Up
38	Allowable PIN Tries Exceeded – Pick
42	No Universal Account - Decline
51	No Sufficient Funds - Decline
53	No Saving Account - Decline
55	Incorrect PIN - Decline
57	Transaction Not Permitted To Card Holder – Decline
58	Transaction Not Permitted To Terminal - Decline
61	Exceeds Withdrawal Amount Limit - Decline
63	Security Violation
65	Exceeds Withdrawal Frequency Limit - Decline
66	Card Acceptor Call Acquirers Security - Decline
75	Allowable Number Of PIN Tries Exceeded – Decline
76	Slm - Invalid Interchange Amount
78	Slm - Deactivated Card
79	Slm - Invalid Account
92	Financial Inst Or Net Not Found For Routing – Decline

Appendix 2. System Failures

Response Code	Response Text
6	Error
12	Invalid Transaction
25	Unable to locate record on file
30	Format Error
31	Bank not supported by switch
68	Response received too late
77	Slm-Invalid business date
80	Slm-Transaction denied
84	Slm-Issuer down
87	Pin key sync error
90	Cut off is in progress
91	Issuer or switch is inoperative-decline
93	Transaction Cannot be complete
94	Duplicate transmission
96	System malfunction