An Evaluation Method for Self-Adaptive Systems

Ali Farahani
Department of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
a_farahani@sbu.ac.ir

Giacomo Cabri
Department of Physics, Informatics and Mathematics
University of Modena Reggio Emilia
Modena, Italy
giacomo.cabri@unimore.it

Eslam Nazemi
Department of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
nazemi@sbu.ac.ir

Alireza Rafizadeh
Department of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
a_rafizadeh@mail.sbu.ac.ir

***Abstract*— nowadays, change is an important aspect of the world. Complexity and change in requirements and environments bring us to Autonomic Systems as a solution. Like any other kind of software system, better implementation of a system needs a proper evaluation method for system implementation. To evaluate a Self-Adaptive system, appropriate quality factors are needed for evaluation. This research tries to use non-adaptive system's evaluation methods to evaluate Self-Adaptive ones. The qualitative factors for self-adaptive systems have been extracted from a literature review (as Self-Adaptive System Qualitative Factors or for abbreviation SAQFs). Hence, there is no explicit or even implicit way for measuring most of the SAQFs, This research has tried to measure them through some measurable Qualitative Criterion. These Qualitative Criteria (we call them QCs) consist of some self-adaptive systems attributes and also some non-adaptive systems' qualitative factors. A map between these SAQFs and software system's QCs (which are more measurable) have been introduced. For each QC, sufficient metrics for measuring could be dug up based on the problem context. For better knowing about the influence of qualitative factors on each other, a prerequisite and post-requisite graph from relations among SAQFs have been introduced. This relational graph shows the importance and impact of each factors measurement on measuring the systems from self-adaptive viewpoint. For evaluating the method, we have proposed a questionnaire to experts about the model the correctness of these impacts and influences have been verified. In addition, a case study on a system in changing environment evaluated with proposed method and the applicability of the method have been reviewed.**

Keywords—self-adaptive system; Autonomic computing; evaluation method; qualitative factors; qualitative criteria

I. Introduction

Recently there are vast researches on autonomic systems and autonomous systems. This viewpoint is growing in all fields. Self-adaptive and self-management are two main subsets of autonomous systems. Self-adaptive systems sense changes inside and outside themselves and adapt to those changes. Since these procedures are based on information about user requirements, resources and environmental conditions, the adaptation has a better result. The main goal of these systems is to reduce human's inference in the system and make systems autonomous. After implementation of self-adaptive systems, their evaluation is vital to check their efficiency and effectiveness. There always has been a conflict between fans and opponents of self-adaptive systems. The opponents are always suspicious about their effectiveness and efficiency. Also as a solution for remediating the fault which occurs because of systems' environment's changes, the user must know about the effectiveness of the solution and a way for measuring adaptation. This paper brings forward an evaluation method for measuring the level of the adaptation. This method could ease the decision making about either using of autonomic computing as a solution for facing the changes in system's requirements and environment or remediating the systems from its fault with another method.

II. Background and Related Work

A. Self-Adaptive Systems

Self-adaptive system is a concept which has been introduced firstly by IBM at 2001 as a solution for better reaction to the growing complexity and environmental changes in computer systems [1]. These systems adapt themselves to their changing environment, using entrance samples that are accurately classified and are available at runtime [2]. They are able to adapt themselves without any kind of interoperability request to users [3]. In a maturity model for these systems (proposed by IBM), autonomy is in a higher level than self-adaptive systems [3][4]. Also, this concept could be considered similar for closing the open loop of the software system into the closed loop which makes them capable of reacting to the changes in themselves and even in their environment [5]. This loop has was a control loop. IBM proposed a reference control loop to achieve autonomic computing. This loop is mostly called MAPE-K.

Main attributes of an autonomous system include four main properties: self-configuration, self-healing, self-optimization and self-protection.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

In so many researches the words "self-adaptive" and "autonomic computing" have been used interchangeably in this fields [1]. We use them interchangeably too.

These properties could be implemented at different levels and with different influence on systems characteristics or specifications. In other words, self-* properties could be implemented with narrow effects on a system to a deep change even on systems architecture. The initial try of a system to become an autonomic system could be defined as supporting self-awareness, self-monitoring and context-awareness as main properties of the self-adaptive system. Some of other optional attributes, like openness, are also at this level [5].

Knowing whether the systems is in the initial level of being self-adaptive or whether it could be categorized as a fully adaptive system is not easy. There are some tries about measuring the state of a system in the path to being adaptive which were reviewed in the following section.

B. Related work

In the past years, there had been some efforts on the evaluation of self-adaptive systems. Each of these methods focuses on different aspects of evaluation. Most of them are about the perception of the author about self-adaptive systems, some definitions and providing some quantitative attributes. None of them has provided metrics for self-adaptive systems' evaluation. In some cases, they have used empirical evaluation and there is no study on system attributes. The main paths which had been taken by a researcher for evaluating self-adaptive systems will be presented with at least one of the best research on each paths.

1) Evaluation through Architecture view

In a self-adaptive architecture, it is not necessary that all components act in an adaptive way, but the result of their behavior should be adaptive. There are some different architectures for self-adaptive systems. In [6] architecture is based on monitoring and tries to change structure and connections of its components to increase efficiency against environmental changes, by constant monitoring of the environment and understanding its characteristics. Better reacting to the changes need to understand the environmental changes better [6]. Probes are the monitoring tools in this architecture. For example, a probe detects size of imported file and the system specifies appropriate action on that file. Gauges are the next step of sensing the changes. In fact, gauges control connections between Architecture Manager in adaptation level and middle components which are responsible for controlling probes. Gauges gather information of probes to have a higher level monitoring on architecture. Whenever an attribute is changed in architecture model, a change plan is generated. This plan provides changes on components and their connections to realize the changes in the behavior. J. A. McCann's research about evaluation through architectural view and its monitoring part has close concepts with RAINBOW infrastructure [7] and it brings into mind that this method of evaluation could be used on the works which are similar in their architectural view. Comparison between this kind of evaluation and the kind which had been used in RAINBOW research could help this research to conclude about its applicability.

Non-functional requirements of a software system take effect from its architecture [8]. Any software should satisfy some non-functional requirements, based on requirements in its design. J. A. McCann et al. mentioned that self-adaptive systems' should satisfy the following metrics [6]. These metrics could be considered as non-functional requirements of systems too. Mentioned metrics consist of: Cost, Flexibility, Robustness, Degree of Autonomy, Adaptability and Stability.

There is a research [9] in which the influence of the architectural attributes on the quality of the software have been presented. It shows that we should think about the architectural attributes and tail their influence on the quality of the systems. For example, one of the main attributes of the systems could be named as protecting itself from outside changes. This attribute could be mapped to self-protecting properties (as a self-adaptive property). So these properties also could be considered as system's requirements.

2) Evaluation based on self-adaptive scenarios

According to Liu and et al [10], a self-adaptive system can be evaluated using self-adaptive scenarios and in a quantitative way. The author considers the architecture of self-adaptive systems as a set of components and their connections (nothing more). Self-adaptive systems have been defined as: the ability to produce systems that can resolve their environmental requirements without human interference [10].

Different stakeholders have different opinions about the ability of adaptation. For example, adaptation from the viewpoint of an end-user is that a new function can be added to system, but architect supposes it as adapting to different operating systems. In a self-adaptive architecture, components and their connections should have proper feedback against the changes. The research shows this feedback as SAAction (Self-Adaptive Action). After recognizing the requirements, a tree is going to be shaped with Adaptability in root and changeable requirements in leaves. Relation of these attributes is as "Or" and "And". The author has used two metrics: IOSA (Impact On the Software Architecture) and ADSA (Adaptability Degree of Software Architecture)

IOSA is the amount of the impact of different factors on components of the architecture and their connections. If ADSA is equal to 1, it means the architecture is adaptive in all aspects and if it is equal to 0, it means there is no adaptation. This research is a way to measure the systems adaptation level in a variety from non-adaptive systems to a pure adaptive system. This could be translated into a concept of measuring the level of adaptation in other viewpoints for measuring the adaptation level. For example the adaptation level in [11] is an amount of recovery from failure or elevating the customer's experiences of using the system. These measurements could not fit into the value from 0-1 and they are priorities value.

3) Empirical evaluation of self-adaptive systems

There are two types of empirical evaluation methods: *user model* and *user adaptation model* [12]. In the user model, connection between user and system is stateless. It means the user should provide the system with a complete set of information for each function he/she desires to do. Differently from user model, the system can predict future actions of a user based on her interaction with the system in the user adaptation

TABLE I. APPROACHES PROPOSED FOR EVALUATING SELF-ADAPTIVE SYSTEMS

Methods	*Evaluation goal*	*Imp.*	*Research scope*	*Evaluation metrics*	*Qualitative model*	*cons*	*pros*
Villegas [13]	6 properties at managed level and 4 properties at manager level	No	Limited to a specific self-adaptive model	There is no evaluation metrics	No	No Qualitative Model, No Implementation, Specific for an adaptive model, No Evaluation Metrics	Good for a specific adaptive model, Has some Evaluation Goal
McCann [6]	4 qualitative attribute (cost, level of self-adaptive, flexibility and robustness)	No	No limit, unless there is a way for measuring evaluation goals	There is no evaluation metrics	No	No Qualitative Model, No Implementation, No Evaluation Metrics	Good for all adaptive model, Has some Evaluation Goal
Chin [12]	Applicability, efficiency and influence from others	No	Specific for Use case models	Empirical	No	No Qualitative Model, No Implementation, No Evaluation Metrics, Just Empirical not Theoretical	Good for just Use Case model, Has an Evaluation Goal, Has Empirical Evaluation Metrics
Liu [10]	Users adaptive scenario	No	No limit unless there is a scenario	Mathematical evaluation	No	No Qualitative Model, Just based on Scenario and has no explicit Evaluation goal	Good for all adaptive model, Mathematical Evaluation

model. In the empirical evaluation, the focus is on effectiveness, efficiency and reusability. As mentioned before, this type of evaluation is appropriate for systems that are in direct engagement with users.

Empirical evaluation is inherently restricted. One of the obvious restrictions of this method is that it is so user dependent, i.e. inputs from users is an inseparable part of the evaluation process. But as we know, in pure self-adaptive systems (which do not consider user as a source of change in system) there is no need for user imported data. So this kind of user dependency is going to narrow the desired scope of systems which could be evaluated for their self-adaptive properties. Furthermore, the layered nature of this method is also a restriction. Also empirical methods are very suitable for detecting defects in design time and wrong assumptions.

Evaluation of self-adaptive systems and their cons and pros could be summarized in Table I. So the proposed method should cover some of these unresolved problems. The researched were planned to have no limitation for evaluation scope and to present an evaluation metrics.

III. PROPOSED METHOD

For better understanding the proposed method, we first propose a brief definition of keywords and concepts which are going to be used in the describing of the proposed model.

A. Definitions

Self-adaptive attributes: self-* properties are well-known term in autonomic computing. Self-* properties covers so many others like self-healing, self-optimizing and etc. for simplification and unification of terminology we call them self-* *attributes*. In general, also they have been called self-adaptive attributes and adaptive attributes. This interchangeable use was mentioned before in [5]. Examples of self-adaptive attributes are: Self-awareness, Self-healing, Self-Optimizing.

Self-Adaptive system's Qualitative Factors: Qualitative factors (Self-adaptive system's Qualitative Factors as a complete term in this paper) are factors about the quality of self-adaptive systems. By using these factors we could measure the systems quality from the viewpoint of self-adaptation. We call them SAQF for abbreviation. Some of the main and measurable (even by the other Qualitative Criterion) Self-adaptive attributes could be found as SAQFs. Examples are: Robustness, Consistency, Termination. In proposing a method there is the need for a starting point, which shows what factors or attributes are planned to be measured. Also considering that it is better to measure the system's quality at the design phase and before implementation (it will ease the adapting the system to preferred quality). Documents of analysis and design phases of software production cycle provide so much information about system behavior in runtime. The origin of non-functional requirements is in architecture which is a result of design phase in software production cycle [8]. For this purpose, architectural techniques and styles are used in software architecture. Having some metrics, concluded from architectural techniques, we can have a proper verification tool in this phase. If these non-functional requirements for the self-adaptive system could be found will show which QCs are important for measuring the self-adaptive system's quality.

Qualitative Criterion: these are the criteria for measuring the quality of the software systems. Each one of them could come from different viewpoints (not always self-adaptation). We call them QC for abbreviation. These are going to be used as sinks for a map between SAQF and them. So there is going to be a map between each one of the SAQFs and one or more of the QCs. Examples are: Anticipation, Performance, Modification, Safety.

For better drawing the difference between QCs and SAQFs, it is worth mentioning that, for instance, "Safety" is QCs but we do not consider it as SAQF. Because it has neither direct usage as a qualitative factor in the literature review of self-adaptive system's evaluation nor in non-functional requirement of self-adaptive systems. But "anticipation" has been considered as a SAQF and also as QC too. Because it is extracted from literature review as Qualitative Factor for self-adaptive systems and also has influence on "Failure avoidance" (which is an SAQF).

B. Method

In the proposed method, we have three phases for verification process: first phase is to recognize qualitative factors (contain non-functional requirements, too) necessary for self-adaptive software systems. These requirements are extracted from multiple researches on self-adaptive systems' verification. We call them self-adaptive system's qualitative factors (SAQFs) and place them in the first level of our qualitative model. The ssecond phase is to create a relational model between these SAQFs and QCs, which could be used for measuring them. The third phase is to present the impact and influence of these SAQFs and QCs on each other and themselves. This model should be similar to MACCAL and ISO 1926 which are qualitative models for software systems[14].

The result of this proposed method will be a set of important SAQFs for measuring the self-adaptive systems quality and a relational model for mapping these SAQFs to measurable QCs (for which there are some proposed evaluation metrics). So by continuing this path from SAQFs to QC and then evaluation metrics and the help of knowing the impact and influence of each QCs (could be extracted from model) the quality of the system from self-adaptation viewpoint could be measured.

Some researches state that self-adaptive systems' architecture is a combination of non-adaptive systems' architecture and components with self-adaptation loop [6]. This idea supports keeping the components of self-* outside of the other components (which are responsible for system's main task). With this definition in mind, we can have a mapping between self-adaptive systems' architecture and qualitative

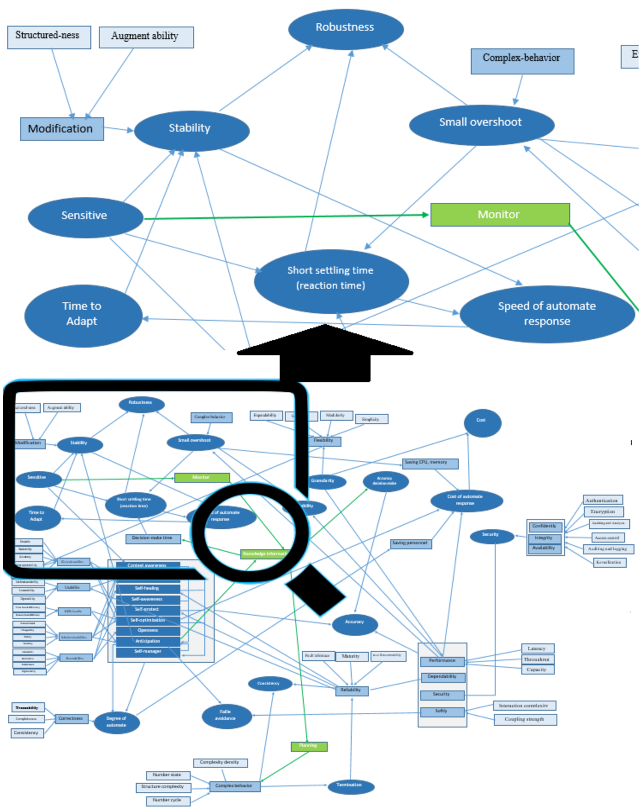

Fig. 1. Relational model between qualitative factors and qualitative criterion and their sub-attributes

TABLE II. SELF-ADAPTIVE QUALITATIVE FACTORS

SAQF	*Description*	*Ref*
Stability	The time it takes a system to learns its environment. This happens when a change is happened in the environment and after self-adaptive activities, system percepts the new environment state. On the other hand, stability comes with changeability, i.e. the more changeable the system is, the more stability comes along.	[15]
Short settling time	The time it takes a system to reach a desired state. The more this time is, the more system has probability of instability.	[16] [13]
Time to adapt	The time it takes the requested changes to be applied.	
Accuracy	This attribute determines how much the adaptation goals are met.	[17]
Small overshoot	It is better to keep the amount of used resources under a specific threshold. The time of transition should be short to reach stability.	[18] [13]
Robustness	The managed system should stay stable, have accuracy, and have short deployment time.	[19] [13]
Termination	To run system without deadlock. The termination guarantees that desired control process, even if system is not in desired state, is limited and its execution will terminate.	[20]
Consistency	The aim of this attribute is to adapt structural integration of system behavior, after an adaptation process.	[20]
Scalability	Refers to the ability of system to support increase even in systems scale or changes with reasonable amount of resources.	
Security	This attribute is the degree of security in a self-adaptive system.	[21]
Sensitivity	This attribute is the degree of system sensitivity towards itself and its environment.	
Granularity	Granularity will effect flexibility of systems (lower granularity will result in more flexible system and ease of change in system) and also will affect performance (make the changes time/resource consumer.	
Failure avoidance	The ability of system to predict future defects.	
Degree of autonomy	The level of systems reaction (from simple reaction to make a deep change in system's structure) to the changes. It is based on systems perception of environment and ability to predict the upcoming situation.	[5]
Cost	Including saved costs of human workforce and hardware resources, etc.	
Cost of Automate Decision	The cost which have to made by systems components to run automatically a task/decision for making system adaptive	
speed autonomic response	It is the time it takes the system to respond to an occurrence and consists of the time to percept the changes and the time to react.	[22]
Openness	It means system is able to work in any environment and be efficient.	
Anticipation	The ability of system to predict future occurrences.	[5]
Context awareness	The amount of system's awareness of its surrounding environment.	[5]
Self-awareness	The amount of system's awareness of itself.	[5]
Self-* properties (example: self-healing)	The properties of autonomic computing system which could be considered for measuring the level of being autonomic/self-adaptive.	

models, in particular, the introduced SAQFs, which consists of adaptive properties with other adaptive attributes of self-adaptive systems in the first level of this model. We have qualitative criteria (QCs) - that consist of some SAQFs and also non-adaptive system's qualitative attributes - in the middle level. Finally, the third level is verification metrics.

C. *Qualitative Factors of self-adaptive systems (SAQFs)*

As discussed, the important qualitative factors should be derived from important non-functional requirements of self-adaptive adaptive systems. SAQFs with their description have been dug up from the literature review could be found in Table II

D. *Qualitative model of self-adaptive systems*

In this model, we have studied the impact of each non-functional requirement or QCs (also each sub-attribute of each QC) on the SAQFs. Figure 1 shows the way SAQFs are connected and the way QCs effect on SQAFs. Sub-attributes of each QCs is chosen based on the accuracy of qualitative models. Additional to the impact of QCs on SAQFs, the impact of self-adaptation components (autonomic computing's control loop components) on SAQFs are also considered. The impact of each one of self-adaptation components on SAQFs is considered in the relational model and colored in green in Figure 21.

Starting from the connections of Figure 2, we have built a new graph that reports both direct and indirect connections among SAQFs (Figure 3). Direct connections consist of existing connections in relational model and indirect ones consist of connections derived from combining SAQFs and QCs, i.e. for example SAQF "granularity" effects on SAQF "reliability". Therefore, we can conclude that non-self-adaptive qualitative criterion (QC) "granularity" is a prerequisite of (has impact on) adaptive attribute (SAQF) "self-healing" and makes an indirect connection. The relational model is concluded from the relations between SAQFs and QCs. Using these relations, we can extract prerequisites and post-requisite graph. In Figure 3, three kinds of SAQFs are reported. The green ones do not have any prerequisite SAQFs, i.e. are not dependent on any other factors. In a graph and the math terminology, we can consider these SAQFs as sources. The red ones do not have any post requisite, i.e. no other SAQFs needs them to be satisfied. These SAQFs are just affected. In the math terminology, we can consider these SAQFs as drains. The blue nodes have both prerequisite and post requisite SAQFs, i.e. they are affected by other SAQFs and can affect other SAQFs.

The proposed qualitative model, differently from traditional qualitative models like MACCAL and ISO 1926, is a graph. The traditional models were in a tree shape and the hierarchy was from root to leaves. The connections between hierarchies were vertical. If the connection is in both vertical and horizontal directions, we will have graph qualitative model.

The graph qualitative model gives us an order of importance of SAQFs or non-functional requirements (which SAQFs derive from). For example, in our proposed qualitative model, the green nodes have a higher priority than the red ones. Besides, this model is more clear and understandable.

By creating the prerequisite and post requisite graph, the influence and impact of the SAQFs on SAQFs on each other could be extracted.

Considering the relation between each SAQFs with the other SAQFs, QCs and also sub-attributes (which are extracted

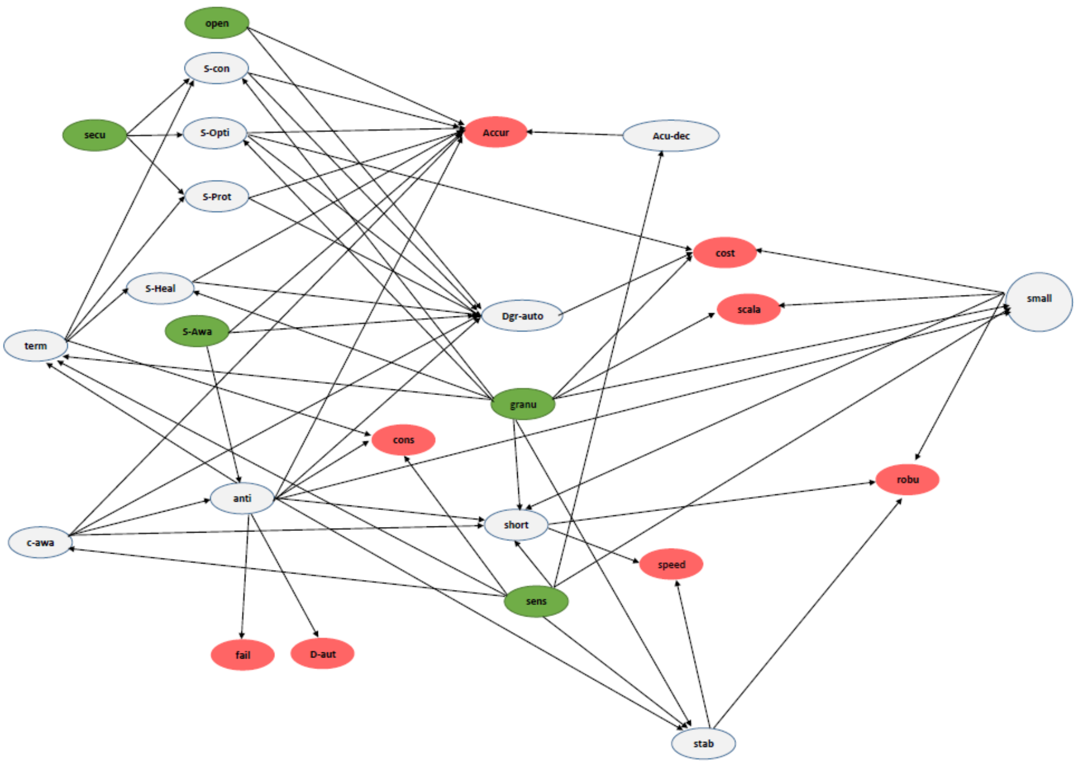

Fig 2. The graph of prerequisites and post-requisites of qualitative factors (with abbreviate names)

for each QC) the table for SAQFs and their related QCs could be extracted. Table 3 shows the SAQFs and their related QCs.

E. Verification Metrics

In order to measure each specific system from self-adaptation point of view, extracting sufficient measuring metrics is going to be the last step. Following specification will help digging up the appropriate verification metrics.

- Verification metrics: for any design metrics, there is an architectural or object oriented solution. The user which are going to use this model should select one of three options below, based on architectural and design documents: “high”, “mediocre” and “low”.
- QCs as verification metrics: each QCs consists of a set of verification metrics.
- SAQFs as verification metrics: each SAQFs consists of a set of verification metrics. The question is: considering complexity of impact of SAQFs on QCs and QCs on SAQFs, which one has more priority. The answer is in self-adaptive system qualitative model (graph model). We determined priority of SAQFs in that model.

TABLE III. MAP BETWEEN SAQFS AND QCS

SAQF	*QCs*
Stability	modification, sensitive, anticipation, performance, integrity, safety, security
accuracy	context awareness, self-configure, self-healing, self-awareness, self-protect, self-optimization, openness, anticipation, performance, security, accuracy decision make
short settling time	decision make time, sensitive, small overshoot, performance
small overshoot	performance, complex behavior
termination	complex behavior, performance
consistency	complex behavior, reliability, integrity, maintainability
scalability	small overshoot, performance
sensitivity	active-sensor
granularity	flexibility, fine/thicker grain
failure avoidance	anticipation, safety
anticipation	efficiently, maintainability, self-awareness, context awareness
robustness	stability, short settling time, small overshoot, safety, availability
speed of automate response	stability, short settling time
degree of automate	correctness, anticipation
security	security
cost	performance, level of automate, small overshoot, self-optimization
level of automate	context awareness, self-configure, self-healing, self-awareness, self-protect, self-optimization, openness, anticipation
accuracy decision make	knowledge information
self-configure	maintainability, usability, functionality, portability
self-healing	maintainability, reliability, availability
self-protect	portability, functionality
self-optimization	efficiently, maintainability, functionality
context awareness	functionality, sensitivity
self-awareness	functionality
openness	portability

IV. EVALUATION OF PROPOSED METHOD

Each method which had been proposed needs to be evaluated. Some methods had been evaluated during an experimental or an empirical research. This method has been evaluated by asking the experts about it by means of a questionnaire and also running it through a case study. The impact of the QCs on SAQFs had been asked about. Afterward the methods had been tested through a case study which had changing requirements. The level of the adaptation has been evaluated with the proposed method in order to know about the applicability of the method.

A. Questionnaire

In the questionnaire, we asked some experts to express their opinion about the impact of used qualitative criterions on qualitative factors.

Each questionnaire’s work should have been examined for its validity and reliability. For validity the question within the questionnaire have been reviewed and revised by 2 expertise in software engineering and statistics to ensure that the questions had been covered the evaluation purpose.

Also, the Cronbach alpha test has been run over the results in order to make sure about the reliability of the questionnaire results. The experts had been asked about the impact of qualitative criteria on qualitative factors. For example, the relevance and impact of QCs (stability, short settling time, small overshoot and safety) over an SAQF (robustness). Based on 36 results from experts for all the qualitative factors, the results calculated using SPSS software showed that Based on experts opinion, 99% of extracted impact of the SAQFs on QC or vice versa is accepted and only around 1% of the relation between SAQFs and QCs were considered as non-relevant.

For this study, the Cronbach alpha is .829 and it shows that the questionnaire questions and results have a good reliability.

B. Case study

To measure the applicability of the proposed method we measured the level of self-adaptation in a traffic control system through a realistic case study. In the emergency respond system whenever there is a collision, the call center is provided with a report by a phone call. Then the report is recorded and redirected to a free patrol related area. Each area has a number of patrol cars. The patrol should move to the collision spot and determine geographical information of that spot on a map. Each patrol has devices to set information and contact others. Some of characteristics and limitations are as following:

- The bandwidth of TX network for the call center is 1024 kb and 64 kb for each patrol.
- Devices on patrols may be changed or upgraded in the future.

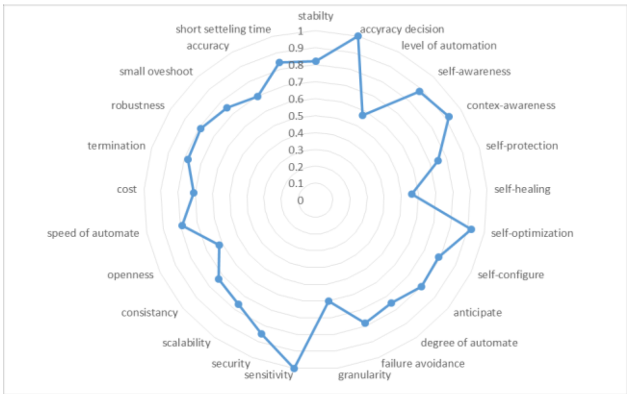

Fig. 2. radar graph for measuring the self-adaptive emergency system (case study)

- There are 20 areas and 10 patrols per area.

The systems have been implemented for simulation. Each qualitative factors have been evaluated by their metrics. Figure 4 shows the results of verification of all qualitative factors. In Figure 4, the relationships between SAQFs foreseen by the qualitative model is respected. For example, having a low openness, self-healing and level of automates causes a low level of accuracy, so the score of these SAQFs are similar and have small diversity. So this method also makes systems developers capable of predicting the results of each architectural decisions on the systems quality. Also, the impact of each SAQFs or QCs on other SAQFs could be predicted base on relevance and impact of SAQFs and QCs on each other even without measuring the factors.

V. Conclusion

In this research, we proposed an evaluation method that, similarly to other evaluation methods for non-adaptive systems, determines some QCs for measuring adaptive attributes, based on qualitative model and adapting architecture and design documents. The main result is providing SAQFs and methods to analyze and verify them in software architecture and design phase, for self-adaptive systems.

A case study showed that the proposed method is applicable in the real world and practical projects. Also there is considerable future work to apply this method to the empirical implementation. Also, using this method on a project about extending of a system which has self-adaptive properties in order to compare the result of this evaluation with the mental satisfaction of the project's stakeholder could be helpful to verify this research in empirical fields. There also could be a DSS (Decision Support System) which has the result of this research as a knowledge and could ease the process of the evaluating a system by providing this method steps as an intelligent questionaries' to complete this research.

References

[1] Kephart, Jeffrey O., & Chess, D. M. (2003). The vision of Autonomic Computing. IEEE Computer Society, 41–50.

[2] A. Rattani, G. L. Marcialis, F. Roli, "Self adaptive systems: An experimental analysis of the performance over time," In IEEE Workshop on Computational Intelligence in BIiometrics and Identity Management (CIBIM), pp. 36-43, 2011.

[3] B. Jacob, R. Lanyon-Hogg, D. K. Nadgir, and A. F. Yassin, "A practical guide to the IBM autonomic computing toolkit," 2004.

[4] A. Ferrante, A. V. Taddeo, M. Sami, F. Mantovani, j. Fridkins, "Self-adaptive security at application level: A proposal," In ReCoSoC Conferences, 2007.

[5] M. Salehie, L. Tahvildari, "Self-adaptive software-landscape and research challenges," I In M Transactions on Autonomous and Adaptive Systems, vol. 4, pp. 14-56, 20010

[6] J. A. McCann, M. Huebscher, "Evaluation issues in autonomic computing," In Conference of Grid and Cooperative Computing, pp. 5107-608, 2004.

[7] Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., & Steenkiste, P. (2004). Rainbow: Architecture-Based Self Adaptation with Reusable Infrastructure. IEEE Computer, 37(10), 46–54.

[8] Paech, Barbara, et al. "Functional requirements, non-functional requirements, and architecture should not be separated œA position paper." (2002).

[9] Khayami, R., Towhidi, A. and Ziarati, K., "Measurable Quality Characteristics of a software System on Software Architecture Level", International Review on Computer and Software (IRECOS), Vol 3, No.3, 234-239, May 2008

[10] Xia Liu and Qing Wang, "Study on application of a quantitative evaluation approach for software architecture adaptability," in Proceedings of the 5th International Conference on Quality Software (QSIC 2005), pp. 265-272,2005.

[11] Cheng, S.-W., Garlan, D., & Schmerl, B. (2009). Evaluating the effectiveness of the Rainbow self-adaptive system. 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, 132–141.

[12] David N. Chin. Empirical evaluation of user models and user-adapted systems. User Modeling and User-Adapted Interaction, 11(1-2):181–194, 2001

[13] N. M. Villegas, G. Tamura, R. Casallas, "A framework for evaluating quality-driven s self-adaptive software systems," In SEAMS '11 Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 80-810

[14] McCall, J. a., Richards, P. K., & Walters, G. F. (1977). Factors in Software Quality. At'l Tech. Information Servicel, 1, 2 and 3(ADA049055), 689–1699.

[15] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B. Rochwerger. Oc´eano - SLA based management of a computing utility. In 7th IFIP/IEEE International Symposium on Integrated Network Management, pages 855–868, 2001

[16] J. White. Simplifying autonomic enterprise Java Bean applications via model-driven development: a case study. The Journal of Software and System Modeling, pages 601–615, 2005

[17] V. Cardellini, E. Casalicchio, V. Grassi, F. Lo Presti, and R. Mirandola. QoS-driven runtime adaptation of service oriented architectures. In ACM SIGSOFT Symposium on the Foundations of Software Engineering, pages 131–140, New York, NY, USA, 20010. ACM.

[18] S. Parekh, N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, and J. Bigus. Using control theory to achieve service level objectives in performance management. Real-Time Systems, 23:127–141, 2002.

[19] J. Dowling and V. Cahill. Self-managed decentralized systems using K-components and collaborative reinforcement learning. In 1st ACM SIGSOFT Workshop on Self-Managed Systems, pages 310–43, New York, NY, USA, 2004. ACM

[20] H. Ehrig, C. Ermel, O. Runge, A. Bucchiarone, and P. Pelliccione. Formal analysis and verification of self-healing systems. In Fundamental Approaches to Software Engineering, volume 6013 of LNCS, pages 1310–153. Springer, 2010

[21] M. Gudemann, F. Ortmeier, W. Reif, "Safety and dependability analysis of self-adaptive system," In Second International Symposiumon Leveraging Applications of Formal Methods, Verification and Validation, pp. 177-184, 2006

[22] Evaluation and Comparison of Decentralised Autonomic Computing Solutions, Tom De Wolf, Tom Holvoet, Report CW437, March 2006