Journal of The Korea Society of Computer and Information www.ksci.re.kr
Vol. 24 No. 3, pp. 65-75, March 2019 https://doi.org/10.9708/jksci.2019.24.03.065

Reasoning Non-Functional Requirements Trade-off in Self-Adaptive Systems Using Multi-Entity Bayesian Network Modeling

Ahmed Abdo Ali Saeed*, Seok-Won Lee**

Abstract

Non-Functional Requirements (NFR) play a crucial role during the software development process. Currently, NFRs are considered more important than Functional Requirements and can determine the success of a software system. NFRs can be very complicated to understand due to their subjective manner and especially their conflicting nature. Self-adaptive systems (SAS) are operating in dynamically changing environment. Furthermore, the configuration of the SAS systems is dynamically changing according to the current systems context. This means that the configuration that manages the trade-off between NFRs in this context may not be suitable in another. This is because the NFRs satisfaction is based on a per-context basis. Therefore, one context configuration to satisfy one NFR may produce a conflict with another NFR. Furthermore, current approaches managing Non-Functional Requirements trade-off stops managing them during the system runtime which of concern. To solve this, we propose fragmentizing the NFRs and their alternative solutions in form of Multi-entity Bayesian network fragments. Consequently, when changes occur, our system creates a situation specific Bayesian network to measure the impact of the system's conditions and environmental changes on the NFRs satisfaction. Moreover, it dynamically decides which alternative solution is suitable for the current situation.

▸Keyword: Goal model; Self-Adaptive Systems; Non-Functional Requirements Trade-Off; Multi-Entity Bayesian Network

I. Introduction

Software engineers must address both Functional and Non-Functional Requirements (NFRs) during the software development process [1]. Functional requirements represent procedures that a given system will be capable of execution. in a different manner, NFRs are known to define quality attributes for a software system [2, 3], including characteristics such as privacy, security, usability, and other similar aspects related to software quality.

It is noteworthy to mention that NFRs are considered complex due to their diversity and fuzziness. Different types of NFRs include constraints that may not be presentable in a formal way and not defined as clear as they should be. For instance, some constraints such as expected response time and failure provisioning may be related to design implementations that are not acknowledged by the time NFR requirements are specified [4]. Also, interpreting NFRs is a task that relies on a subjective understanding. Possible Tasks and/or Operationalizations for satisficing a given NFR might differ according to each stakeholder's needs. Moreover,

• First Author: Ahmed Abdo Ali Saeed, Corresponding Author: Seok-Won Lee
*Ahmed Abdo Ali Saeed (hmdabdo@ajou.ac.kr), Dept. of Computer Engineering, Ajou University
**Seok-Won Lee (leesw@ajou.ac.kr), Dept. of Software and Computer Engineering, Ajou University
• Received: 2018. 12. 11, Revised: 2019. 02. 18, Accepted: 2019. 02. 19.
• This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B03034279).

one solution to implement a single NFR may produce synergies and perhaps more important conflicts with another NFR. Currently, several researchers have discussed how NFRs play an important role in runtime adaptation and the need for it to be considered in the different alternative solutions [5].

Recently, the need for dynamic software systems has been growing intensively. There are now systems that can dynamically change their runtime behavior according to the change in their environments. Consequently, dealing with uncertainty in Software Engineering becomes challenging. Such systems are called self-adaptive systems (SASs). Strong research interest in dealing with uncertainty that SAS brings as a first-class concept [6] and SAS must be considered. As in Software Engineering, Non-Functional Requirements (NFRs) are more important than Functional Requirements and can determine the success of the software system.

SASs are operating in dynamically changing environment. Furthermore, the configuration of the SAS systems is dynamically changing according to the current systems context. This means that the configuration that manages the trade-off between NFRs in this context may not be suitable in another. This is because the NFRs satisfaction is based on a per-context basis [7]. Therefore, one context configuration to satisfy one NFR may produce a conflict with another NFR. For example, satisfying a necessary task ('Use of strong Cryptography') to satisfy security NFR might directly affect the satisfaction of NFR performance negatively. Another example is when considering an Internet of Things (IOT) device that has "Send data frequently" and "Conserve battery power" NFRs. Trying to satisfy one NFR will directly affect the satisfaction of the other NFR. Therefore, this relationship between NFRs brings the perception that one NFR cannot be 100% satisfied. Moreover, the measurement of NFRs is complicated to understand due to their vague and subjective manner. The term satisfied was introduced in [8], [9]. Furthermore, NFRs usually conflict with each other making it a challenge when trying to decide better alternatives for satisfaction.

In this paper we propose a novel approach of using goal modeling and system run-time environment to manage the trade-off among multiple NFRs. This study tries to fill the void between NFRs' satisfaction degree and the system goal alternative solutions. We achieve this through identifying the alternative solutions using goal modeling, identifying the NFRs and identifying the system variables together with assumptions that influence the NFRs. Therefore, we decompose our main objective into two major accomplishments:

1. Mapping the system goal model and the system runtime environment's monitored variables together with the system's assumptions into a form of Multi-entity Bayesian Network (MEBN) fragments organized by Multi-entity Bayesian Network (MEBN) MTheories.
2. Monitoring the system runtime environment's monitored variables and assumptions values during runtime and answering the following questions:
    a. Which monitored variable values need to be changed in order to increase NFRs' satisfaction?
    b. Which alternatives that can be used to ensure the satisfaction of all NFRs?

II. Related Works

Previous works based on Bayesian networks have been used to enable reasoning and making decisions over probabilistic models. Fenton et al [10] used Bayesian networks in a way to predict and reason about the satisfaction of NFRs as multi-criteria decision making. However, the Bayesian networks by itself does not provide direct modeling for dynamic systems [11]. Filieri, A et al [12] used Discrete-Time Markov Chains (DTMC) to deal with the impact of the changes in the environment and the uncertainty that brings on the compositions of services, the quality properties, and QoS of the service-based applications. Portinale et al [13] used dynamic Decision Networks (DDNs) for Fault Detection, Identification, and Recovery in autonomous systems. Bencomo et al [14] used dynamic Decision Networks (DDNs) to measure the satisfaction degree of NFRs in self-adaptive systems and deal with uncertainty related to the vague nature of the non-functional requirements.

MEBN was first introduced by Laskey, K. B [15]. Which is a theory combining expressivity of first-order logic principles and probabilistic reasoning of Bayesian networks. Since then, MEBN has been used in various domains including, maritime domain awareness [16], predictive situation awareness [17]. The other domains also include, Dynamic firewall [18], Feature-oriented adaptive system [19], predicting risk assessment for security requirements [20], detecting abnormal behavior of an insider [21] and Fraud Detection [22]. A review of probabilistic reasoning methods used in automated driving situational awareness identified MEBN with a fuzzy extension. It defined it as the most broadly capable approach for probabilistic first-order logic modeling in automated driving domain [23].

MEBN has been used meticulously to overcome several limitations of Bayesian networks. However, there have been no study that focuses on using MEBN to deal with the Non-Functional requirements trade-off problem especially focusing in self-adaptive systems.

III. Proposed Approach

1. Approach overview

Figure 1 introduces our approach towards reasoning and selecting the suitable solution for any given situation.

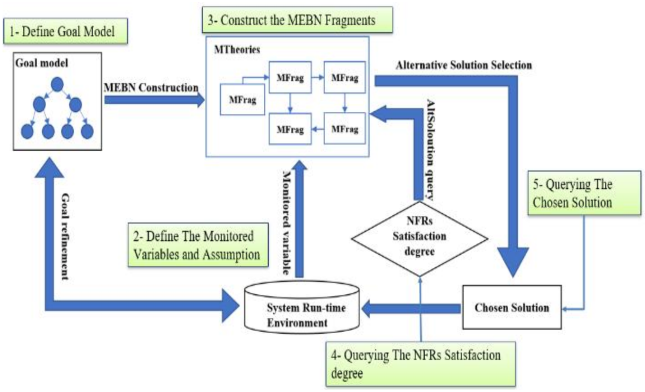

Fig. 1.Proposed Approach

Given the system run time monitored variables and assumptions values. The approach contains five main components. These comprise of Goal model; System runtime environment; MEBN fragments organized by MTheories; NFRs satisfaction degree and chosen alternative solution together with the relations between these components.

2. Approach main Component

2.1. Goal Model

The goal model plays an important role in our approach as gives us a clear representation of the system goal and the alternative solutions that can be used to realize the goal. Furthermore, the goal model gives us the contribution of each alternative on the different NFRs associated with. These details will be used to construct the MEBN fragments (MFrags) together with the information that will be defined in the System run-time Environment component.

2.2. System run-time Environment

In the System run-time Environment, we define the system monitored variables and assumptions that have an effect of the NFRs satisfaction degree. The monitored variables and assumptions will be used as a metric that will be used to measure the satisfaction degree of the NFRs. The system monitored variables and assumptions values are continuously changing during the system run time. Therefore, the satisfaction degree of the NFRs will change accordingly.

2.3. MEBN MTheory

MTheory organizes the MEBN fragments that have been generated from the goal model and the System run-time Environment. Each fragment represents a small knowledge about the system.

2.4. NFRs Satisfaction degree

Monitors the changes in the system environment. By monitoring the system monitored variables and assumptions values during the system run-time. Therefore, if the values changed a Situation Specific Bayesian Network (SSBN) will be generated to query about the effect of the changes on the NFRs' satisfaction degree.

2.5. The chosen Solutions

The Chosen Solution stores the different architecture for the different alternative that realizes the system goal. These architectures will be applied to the running system when chosen as a suitable solution for a given situation.

3. Approach Methodology

In this Section, we will define the steps that need to be followed in order to apply the proposed approach to a give system. The steps are as follow:

3.1. Define the goal model

A. What is the system goal?
B. What are the alternative solutions that realize the system goal?
C. What are the system's NFRs?
D. What are the contribution links that each alternative solutions have on the different NFRs?

3.2 Define the system run-time monitored variables and assumptions

A. What are monitored variables that have an effect on the NFRs satisfaction degree?
B. What are the system assumptions?
C. Which monitored variable influences which NFR?
D. Which assumption has an effect on which NFR?

3.3 Construct the MEBN Fragments

Given the Define goal model and the system monitored variables and assumptions, we construct the MEBN fragments as in Section 3.4. These fragments then will be used to reason about the satisfaction degree of the different NFRs. We need to do the following:

A. Construct the monitored variables MFrags.
B. Construct the assumptions MFrags.
C. Construct the NFRs MFrags.
D. Construct the alternative solutions MFrag.
E. Specify the context, input and resident nodes in each MFrag.

3.4 Querying the NFRs Satisfaction degree

Monitors the changes in the system environment. By monitoring the system monitored variables and assumptions values during the system run-time. We need to query about the following:

A. What are the current monitored variables and assumptions values?
B. Did the values change?
C. The changed values have an effect on which NFRs?
D. What is the effect of this changed values on the NFRs satisfaction degree?

3.5 Querying the Chosen Solution

Monitors the NFRs' satisfaction degree. Therefore, if the NFRs' satisfaction degree is not within the defined limit a Situation Specific Bayesian Network (SSBN) will be generated to query about the alternative solution. We need to query about the following:

A. What are the current NFRs satisfaction degree?
B. Are the NFRs satisfaction degree within the defined limit?
C. Given the current NFRs satisfaction degree, what is the suitable alternative solution?

4. Mapping Goal model and System run-time to MEBN

The goal model is given in the form of four-tuple, the tuples will be used to create MEBN fragments. As well as the assumption and monitored variables that can be observed via sensors from the system's run-time environments. The assumptions and the monitored variables are used as metrics and measures for NFRs satisfaction degree. First, for each assumption and monitored variable, a mapping MFrag will be created. Each MFrag has a resident node with a probabilistic value. For example, "Response time" can have low, medium, and high. values which influence performance NFR. Next, for each NFR listed in the goal model, a mapping MFrag will be created. Each MFrag has a resident node with satisfaction range probabilistic value (Low, Medium, and High) or (True and False) of the NFR together with an input node reference to the assumption or monitored variables MFrag. For example, given the current value of the response time, and other defined variables as metrics and measures for performance NFR. This helps us to answer the question: "which variable values need to be changed in order to increase the performance satisfaction"? And same for other NFRs. Then for the alternative solutions, a mapping MFrag will be created. These alternatives correspond to the solution decisions in the system. This MFrag's resident node has probabilistic values for using these alternatives, and its input node reference to the different NFRs MFrag. For Example, given the satisfaction degree of both performance and cost NFRs, we can answer the question: "what alternative that can be used to ensure the satisfaction of both performance and cost NFRs"? In order to decide what variable needs to be changed to keep both NFRs in an acceptable level of satisfaction. Once the MEBN MFrags are defined, we must define their probabilities and relationships in a collection of variables nodes within the MFrage architecture. The probability distribution and the relationships between them are based on the domain knowledge and heuristic observation.

Figure 2 shows the Network of Main Components used to create the MEBN MFrags and then used for reasoning about the NFRs trade-off.

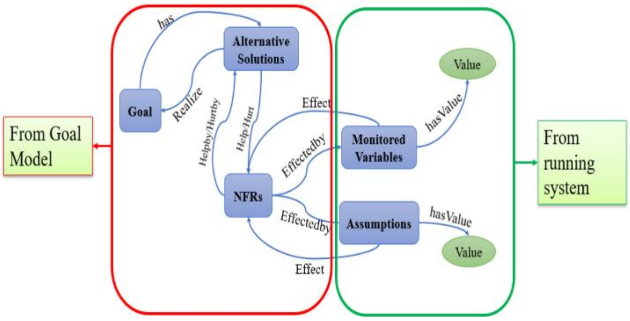

Fig. 2.Network of Main Components to Construct MEBN Fragments

The figure shows that each goal has more than one alternative solution that can be realized with. Each alternative solution has one or more NFR associated with, the effect that the alternative has on the NFRs is represented by the contribution like which can be; make (++), help (+), hurt (-) and break (--). The monitored variables and assumptions have values that can be changed in the system run-time and influence the NFRs satisfaction degree.

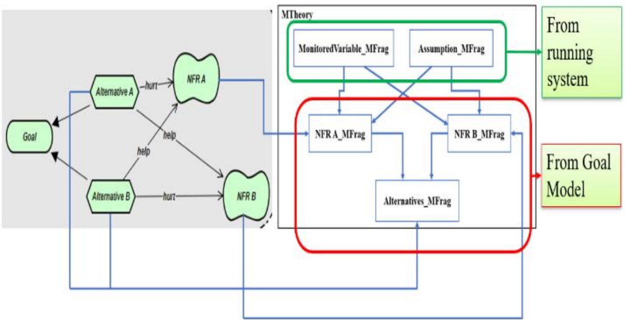

Fig. 3.MEBN Construction

Figure 3 shows the mapping between the goal model and the system run-time into MEBN fragments. As in the Figure, the list of the alternative solutions will be represented in one MEBN MFrag. In addition, each NFR in the list of the NFRs associated with each alternative will be represented in MFrag. The NFRs and the alternative solutions directly extracted from the goal model. As from the system runtime, each of the monitored variable and assumption will be represented in a MFrag.

We then define the relation between the MFrags. Monitored variables and assumptions MFrags could have an effect on the one or more NFRs MFrags. These relations are defined when constructing the MEBN MFrags. On the other hands, the relation between the NFRs MFrags and the alternative solutions MFrag is extracted from the goal model.

IV. Evaluation

We used the evaluation methodology recommended in [24] [25] to evaluate our proposed approach. First, we specify the study questions followed by the study propositions. Then the study propositions were supported by unit of the analysis. Finally, the linking between the study propositions and the unit of analysis.

1. Theoretical Evaluation of Proposed Methodology

This research purpose is to study the process to manage the trade-off of NFRs considering the variability of the alternative solutions of the system goals. to precisely discuss How and Why this approach will help to accomplish the research goal.

1.1 Study Questions

Through this study, the study questions should be answered, to justify and provide a legitimacy to the proposed research methodology. The Study Questions are shown in Table 1.

Table 1. Study Questions

Study Questions
When the system run-time monitored variables and assumptions values changes, How the changes affect the satisfaction of the NFRs?
What is the suitable alternative solution the system use, given the current variables and assumptions values as well as the satisfaction level of the NFRs? and why?

1.2 Study Proposition

Study propositions are discovered and derived from the case study questions which should be examined. Case study propositions are the declarations that need to be investigated to justify the established study source of information that measures the achievement of study proposition. The unit of analysis that we use in questions The general proposition (GP) to accommodate the study questions is: “The proposed methodology can accomplish its research goals as it defines specific guidelines to gather and analyze the requirements in various contexts .The study propositions are listed in Table 2.

Table 2. Study Propositions

General Proposition	Specific proposition
GP1. The proposed approach helps the system engineers to build the goal mode and define the system variables.	SP1.1 By building the right goal model with the detailed alternatives and the detailed contributions these alternatives have on the different NFRs, the construction of the MEBN

	fragments will right accordingly.
	SP1.2 By clearly defining the right system monitored variables and its assumptions that have an impact on the NFRs satisfaction degree, the construction of the MEBN fragments will right accordingly.
GP2. The proposed approach helps the system to measures the satisfaction degree of the NFRs.	SP2.1 The proposed approach keeps monitoring the system run-time monitored variables and assumptions values through time and alert when they are changed.
	SP2.2 The proposed approach notifies when monitoring the system run-time monitored variables and assumptions and generate a query to measure the impact of these changes on the NFRs satisfaction degree.
GP3. The proposed approach helps the system to choose the suitable alternative solution to use.	SP3.1 The proposed approach keeps monitoring the NFRs satisfaction degree values through time and alert when they are changed.
	SP3.2 The proposed approach notifies when the NFRs satisfaction degree values and generate a query to measure the impact of these changes on the what alternative to use.

1.3 Unit of Analysis

Unit of analysis is the set of selected resources to be examined during the experiment process and used as evidence to support research hypothesis. The Unit of Analysis are listed in Table 3.

Table 3.Unit of Analysis

Unit of Analysis	Definition
Goal Model	The goal model gives us a clear representation of the system goal and the alternative solution that can be used to realize the goal. Furthermore, the goal model gives us the contribution of each alternative on the different NFRs associated with. These details will be used to construct the MEBN fragments (MFrags) together with the information that will be defined in the System run-time Environment component.
System run-time environment	In the System run-time Environment, we define the system monitored variables and assumptions that have an effect of the NFRs satisfaction degree. The monitored variables and assumptions will be used as a metric that will be used to measure the satisfaction degree if the NFRs. The system monitored variables and assumptions values are continuously changing during the system run time. Therefore, the satisfaction degree of the NFRs will change accordingly.
MTheory	MTheory organizes the MEBN fragments that have been generated from the goal model and the System run-time Environment. Each fragment represents a small knowledge about the system.
NFRs satisfaction degree	Monitors the changes in the system environment. By monitoring the system monitored variables and assumptions values during the system run-time. Therefore, if the values changed a Situation Specific Bayesian Network (SSBN) will be generated to query about the effect of the changes on the NFRs' satisfaction degree.
Chosen solutions	The Chosen Solution stores the different architecture for the different alternative that realizes the system goal. These architectures will be applied to the running system when chosen as a suitable solution for a given situation.

1.4 Linking Data

To link the generated unit of analysis and study propositions, connecting both the objects is important.

Table 4.Linking Proposition with Unit of Analysis

Code	Description	Unit of Analysis
SP1.1	The goal model needs to be created clearly as the information the goal model provides used by the approach to constructing the MEBN. Therefore, we create the goal model or will be given by the systems engineers.	Goal Model
SP1.2	AS well as the system monitored variables and assumptions, we define them and the relations that have on the NFRs to construct the MEBN.	System run-time environment
SP2.1	The system monitored variables and assumptions values are continually changes, therefore, the system keeps monitoring them and notify when changes happen.	System run-time environment, MTheory
SP2.2	The system monitored variables and assumptions that have an impact on the NFRs satisfaction degree therefore if changes happened the system generate a Situation Specific Bayesian Network (SSBN) to query about the effect of the changes on the NFRs' satisfaction degree.	System run-time environment, MTheory, NFRs satisfaction degree
SP3.1	The NFRs satisfaction degree values are also continually changes, therefore, the system keeps monitoring them and notify when changes happed.	NFRs satisfaction degree, MTheory
SP3.2	The NFRs satisfaction degree have an impact on choosing the alternative solution that the system will use, therefore, if changes happened the system generate a Situation Specific Bayesian Network (SSBN) to query about the alternative solution for the current situation.	NFRs satisfaction degree, MTheory, Chosen solutions

2. Results

In order to test our approach, we generated different situations with different values for both of the monitored variables and assumptions. Given these values, the system used them to measure the impact of them on the NFRs satisfaction level. Furthermore, to decide what is the suitable alternative solution to use.

2.1 Robot Vacuum Cleaner

2.1.1 Robot Vacuum Cleaner overview

The goal for robot vacuum cleaner is to clean apartment. This goal can be realized by one of the alternatives; "clean at night" or "clean when empty", each alternative has a different effect on the NFRs "avoid tripping hazard" and "minimize energy cost", which means that satisfying one NFR may hurt the satisfaction of the other as in Figure 4.

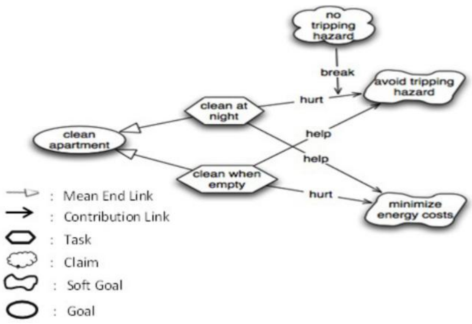

Fig. 4.Robot Vacuum Cleaner Goal Model [26]

2.1.2 Applying the methodology on the Robot Vacuum Cleaner

Table 5. Methodology On the Robot Vacuum Cleaner

Step	Description
Define the goal model	The goal: clean apartment. The alternatives: "Clean At night" and "clean when empty". The NFRs: "avoid tripping hazard" and "minimize energy cost". The contribution links: "Clean at night" hurts "avoid tripping hazard" and help "minimize energy cost" but "clean when empty" helps "avoid tripping hazard" and hurt "minimize energy cost."
Define the system run-time monitored variables and assumption	The monitored variables: The level of energy consumption which influences the "minimize energy cost" NFR. The assumption: The availability of tripping hazard which influences the "avoid tripping hazard" NFR.
Construct the MEBN Fragments	Constructed MFrag of the monitored variable level of energy consumption. Constructed MFrag of the assumption availability of tripping hazard. Constructed MFrags of the NFRs. Constructed MFrag of the alternative solutions.
Querying the NFRs Satisfaction degree	Monitoring the changes in the value of the availability of tripping hazard (Tripping_MFrag) and the level of energy consumption (EnergyRang_MFrag) during the system run-time.
Querying the Chosen Solution	Monitoring NFRs' satisfaction degree. Therefore, if the NFRs' satisfaction degree is not within the defined limit a Situation Specific Bayesian Network (SSBN) to query about the alternative solution.

2.1.3 Results

Table 6 shows some of the situations that have been generated and the associated alternative solution with each situation. In situation 0, the tripping availability is 25% True and 75% False while the energy consumption range is (20% Low, 50% Medium and 30% High), this makes the satisfaction of both avoid tripping and "minimize energy consumption" 65% and 52% True. Given these values the chosen alternative for this situation is to "clean at night" to increase the satisfaction level of "minimize energy consumption" NFR. Situation 1 has a tripping availability is 100% True and 0% False while the energy consumption range is (20% Low, 50% Medium and 30% High), this makes the satisfaction of both avoid tripping and "minimize energy consumption" 20% and 52% True. Given these values, the chosen alternative for this situation is to "clean when empty" so as to increase the satisfaction level of "avoid tripping" NFR. Finally, situation 2 has a tripping availability is 100% True and 0% False while the energy consumption range is (100% Low, 0% Medium and 0% High). This makes the satisfaction of both "avoid tripping" and "minimize energy consumption" 20% and 95% True. Given these values the chosen alternative for this situation is to "clean when empty" as a way to increase the satisfaction level of "avoid tripping" NFR.

2.2. RDM

2.2.1 RDM Overview

RDM is a data protection technique that stores copies of data at physically isolated locations to protect data against loss, unavailability, or corruption. As in Figure 5, in RDM goal model, multiple NFRs have been listed and includes "Minimize Operational costs", "Maximize Performance" and "Maximize Reliability". These NFRs are associated with different alternatives. Each alternative provides a different level of data protection, performance, reliability and cost shown by the contribution link between the alternatives and their associated NFRs. In this study we considered the goal "Select Topology" and it's two different alternatives "Use MST Topology" and "Use Redundant Topology".

Table 6. *Vacuum Cleaner Different Situation Result*

MFrag / Situation	Tripping MFrag		EnergyRange MFrag			AvoidTripping MFrag		MinEnergy MFrag		Strategy	
	True	False	Lo	Me	Hi	True	False	True	False	Clean At Night	Clean When Empty
0	25%	75%	20%	50%	30%	65%	35%	52%	48%	60%	40%
1	100%	0%	20%	50%	30%	20%	80%	52%	48%	49%	51%
3	100%	0%	100%	0%	0%	20%	80%	95%	5%	37%	63%

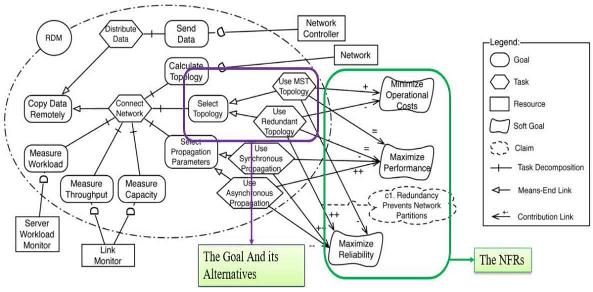

Fig. 5. RDM Goal Model [27]

2.2.2 Applying the methodology on the RDM

Table 7. Methodology on the RDM

Step	Description
Define the goal model	The goal: Select topology. The alternatives: Use MST Topology and Use Redundant Topology. The NFRs: Minimize Operational costs, Maximize Performance and Maximize Reliability. The contribution links: Use MST topology help Minimize operational cost, hurt maximize performance and hurt maximize reliability. Use redundant topology hurt minimize operational cost, help maximize performance and make maximize reliability.
Define the system run-time monitored variables and assumption	The monitored variables: The Number of Concurrent Connection (NCC) which influences the Maximize Performance NFR and Cost range (CR) which influences the Minimize Operational costs NFR. The assumption: The Redundancy Prevent Network Partition which influences the Maximize Reliability NFR.
Construct the MEBN Fragments	Constructed MFrags of the monitored variables NCC and CR. Constructed MFrag of the The Redundancy Prevent Network Partition. Constructed MFrags of the NFRs. Constructed MFrag of the alternative solutions.
Querying the NFRs Satisfaction degree	Monitoring the changes in the value of the Number of Concurrent Connection, cost range and the Redundancy Prevent Network Partition assumption during the system run-time.
Querying the Chosen Solution	Monitoring NFRs' satisfaction degree. Therefore, if the NFRs' satisfaction degree is not within the defined limit a Situation Specific Bayesian Network (SSBN) to query about the alternative solution.

2.2.3 Results

Table 8 shows some of the situations that have been generated and the associated solution with each situation. Situation 0 is the situation. When the evidence shows that the cost value had been reduced to low with 100% in situation 1, while the performance and assumption value are still having the same values. The satisfaction of both "maximize the performance" and "maximize reliability" did not change. On the other hand, the satisfaction of "minimize the operational cost" increased to 95%. As a result, the selected topology did not change but also increased the confidence on choosing it as the suitable solution. In situation 3, the cost value increased to 100% high, while the performance and assumption values are unchanged, the satisfaction of both "maximize the performance" and "maximize reliability" did not change. On the other hand, the satisfaction of "minimize the operational cost" decreased to 90%. Therefore, the used topology changed to "UseMST" in order to increase the satisfaction level of "minimize the operational cost". On Situation 4, the cost value increased to (45% high, 30% medium and 25% high), the performance value changed to 100% low and assumption value is still the same value. The satisfaction of both "maximize the performance" and "minimize the operational cost" changed to 95% and 63.25%. Therefore, the used topology did not change so as to increase the satisfaction level of "minimize the operational cost". For Situation 5, the cost value increased to 100% low, the assumption value changed to 100% false and the performance value increased to 100% high. The satisfaction of "minimize the operational cost" changed to 95% true while the satisfaction of "maximize the performance" and "maximize reliability" changed to 10% true for both. As a result, the selected topology changed with 92.24% confidence regarding choosing it as the suitable solution from "UseMST" to "Use Redundant Topology". This allows for the increase satisfaction level for both "performance" and "maximize reliability". Finally, in situation 6, the cost value changed to medium with 100%, the performance value changed to medium with 100%, and the assumption that the current network is not redundant with 100%, all these current evidence changes the selected topology to "Use Redundant topology" with 71%, to increase the redundancy.

V. Discussion

We have evaluated the approach in both robot vacuum cleaner and remote data mirroring case studies by following the proposed approach methodology. However, we would like to discuss the proposed approach in the following attributes and constraints.

1. Applicability

The proposed approach and its methodology have been applied to both robot vacuum cleaner and remote data mirroring systems. By looking at the results from both systems, the results showed that the proposed approach

Table 8. Vacuum Cleaner Different Situation Result

MFrag / Situation	MonitoredVariableCost_Frag			Assumption_Flag		MonitoredVariablePerf_Fra		
	Lo	Me	Hi	True	False	Lo	Me	Hi
0	45%	30%	25%	90%	10%	20%	50%	30%
1	100%	0%	0%	90%	10%	20%	50%	30%
3	0%	0%	100%	90%	10%	20%	50%	30%
4	45%	30%	25%	90%	10%	100%	0%	0%
5	100%	0%	0%	0%	100%	0%	0%	100%
6	0%	100%	0%	0%	100%	0%	100%	0%

MFrag / Situation	MinoperationalCost_Frag		MaxReliability_Frag		Maxoerformanance_Frag		SelecterTopology_Flag	
	True	False	True	False	True	False	UseMST	UseRedundant
0	63.25%	36.75%	82%	18%	52%	48%	43.81%	56.19%
1	95%	5%	82%	18%	52%	48%	39.12%	60.88%
3	10%	90%	82%	18%	52%	48%	51.68%	48.32%
4	63.25%	36.75%	82%	18%	95%	5%	71.63%	28.37%
5	95%	5%	10%	90%	10%	90%	7.76%	92.24%
6	60%	40%	10%	90%	60%	40%	28.16%	71.84%

and its methodology can be applied to various systems. Systems that have to select the appreciate alternative solution or system that have to make a decision between multiple conflicting actions.

2. Scalability

This work proposed approach and its methodology used to make a decision between one or more alternative that satisfy specific system goal taking into consideration the NFRs satisfaction degree. Since the proposed approach and its methodology fragmentize the system to small fragments, the proposed approach and its methodology can work well in complex systems. However, the manual creation of the fragments and the relations between them can be labor intensive for a very complex system.

3. Usability

As per the results from both robot vacuum cleaner and remote data mirroring case studies, the proposed approach and its methodology can be very helpful for choosing the suitable alternative solution taking into consideration the trade-off that each alternative has on the different NFRs.

VI. Conclusions and Future Work

In our study, we showed how the goal model and the System running environment are used for constructing the Multi-Entity Bayesian Networks fragments. They are then used for reasoning about the alternative solutions that satisfy the system Non-Functional Requirements as a way to balance the trade-off between them at runtime. The approach helps to resolve the uncertainty in the Self-Adaptive systems environment. Especially the uncertainty related to the NFRs satisfaction and their alternative solutions. Furthermore, our approach determines the suitable solution.

We applied the proposed approach and its methodology to a different two case studies. In the case studies, we showed when the system's monitored variables changes, the satisfaction of the NFRs were affected positively or negatively. Therefore, the current solution may or may not change according to the satisfaction of the NFRs in the current situation.

The proposed approach and its methodology can be used in various systems in order to give these systems the ability to choose between the alternative solutions that satisfy the systems' goals. Taking into consideration the trade-off among the Non-Functional Requirements associated with each alternative.

As for future work, we consider adding ontologies for both the NFRs and system domain. Extracting domain assumptions, rules, and evidence from domain ontology as well as the metrics, measures, and influences on others NFRs form NFRs ontology will help us to build a complete MTheory for every aspect of the system's domain. Also, we consider making automatic generation for MEBN fragments from these ontologies.

REFERENCES

[1] Chung, L. and do Prado Leite, J.C.S., "On non-functional requirements in software engineering," In Conceptual modeling: Foundations and applications , pp. 363-379, Springer, Berlin, Heidelberg, 2009.

[2] Boehm, B.W., Brown, J.R. and Kaspar, H., "*Characteristics*

of software quality," North-Holland Pub. Co., 1978
[3] Keller, S.E., "Specifying software quality requirements with metrics," System and Software Requirements Engineering, IEEE Computer Society Press Tutorial, 1990.
[4] Roman, G.C., "A taxonomy of current issues in requirements engineering," Computer 4, pp.14-23, 1985.
[5] GOLDSBY, Heather J., et al. "Goal-based modeling of dynamically adaptive system requirements," 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008). IEEE, pp. 36-45, 2008.
[6] ZIV, Hadar; RICHARDSON, Debra; KLÖSCH, René. "The uncertainty principle in software engineering," submitted to Proceedings of the 19th International Conference on Software Engineering (ICSE'97). 1997.
[7] DE LEMOS, Rogério, et al. "Software engineering for self-adaptive systems: A second research roadmap," Software Engineering for Self-Adaptive Systems II. Springer, Berlin, Heidelberg, pp. 1-32, 2013.
[8] Mylopoulos, John, Lawrence Chung, and Brian Nixon. "Representing and using nonfunctional requirements: A process-oriented approach." IEEE Transactions on software engineering, Vol 18, No. 6, pp.483-497, 1992.
[9] Chung, Lawrence, et al. "*Non-functional requirements in software engineering,"* Vol. 5. Springer Science & Business Media, 2012.
[10] Fenton, Norman, and Martin Neil. "Making decisions: using Bayesian nets and MCDA." Knowledge-Based Systems, Vol. 14, No. 7, pp.307-325, 2001.
[11] S. J. Russell and P. Norvig, "*Artificial intelligence: A modern approach*," 2nd ed., ser. Prentice Hall series in artificial intelligence. Prentice Hall, 2003.
[12] Filieri, Antonio, Carlo Ghezzi, and Giordano Tamburrelli. "A formal approach to adaptive software: continuous assurance of non-functional requirements." Formal Aspects of Computing, Vol. 24, No.2, pp.163-186, 2012.
[13] Portinale, Luigi, and Daniele Codetta-Raiteri. "Using dynamic decision networks and extended fault trees for autonomous fdir." 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, pp.480-484, November 2011.
[14] Bencomo, Nelly, Amel Belaggoun, and Valerie Issarny. "Dynamic decision networks for decision-making in self-adaptive systems: a case study." Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp.113-122, May 2013.
[15] Laskey, K.B., "MEBN: A language for first-order Bayesian knowledge bases," Artificial intelligence, 172(2-3), pp.140-178, 2008.
[16] Carvalho, R.N., Costa, P.C.G., Laskey, K.B. and Chang, K.C., "PROGNOS: predictive situational awareness with probabilistic ontologies," In Information Fusion (FUSION), 2010 13th Conference, pp. 1-8, July 2010.
[17] Park, C.Y., Laskey, K.B., Costa, P.C. and Matsumoto, S., "Predictive situation awareness reference model using multi-entity bayesian networks," In Information Fusion (FUSION), 2014 17th International Conference, pp.1-8, July 2014.
[18] Boruah, A. and Hazarika, S.M., "An MEBN framework as a dynamic firewall's knowledge flow architecture," Signal Processing and Integrated Networks (SPIN), 2014 International Conference, pp. 249-254, February 2014.
[19] Lee, H.C. and Lee, S.W., "Decision supporting approach under uncertainty for feature-oriented adaptive system," Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual Vol. 3, pp. 324-329, July 2015.
[20] Lee, S.W., "Probabilistic risk assessment for security requirements: A preliminary study," In 2011 Fifth International Conference on Secure Software Integration and Reliability Improvement, pp. 11-20, June 2011.
[21] AlGhamdi, G., Laskey, K.B., Wright, E.J., Barbará, D. and Chang, K.C., "*Modeling insider behavior using multi-entity Bayesian networks*," 2006.
[22] Carvalho, R.N., Matsumoto, S., Laskey, K.B., Costa, P.C., Ladeira, M. and Santos, L.L., "Probabilistic ontology and knowledge fusion for procurement fraud detection in brazil," Uncertainty Reasoning for the Semantic Web II, Springer, Berlin, Heidelberg, pp. 19-40, 2013
[23] Golestan, K., Soua, R., Karray, F. and Kamel, M.S., "Situation awareness within the context of connected cars: A comprehensive review and recent trends," Information Fusion 29, pp.68-83, 2106
[24] Lee, S.W. and Rine, D.C., "Case Study Methodology Designed Research in Software Engineering Methodology Validation," SEKE, pp. 117-122, 2004.
[25] Yin, Robert K, "*Case study research: Design and methods,"* Sage publications, 2013.
[26] Bencomo, N. and Belaggoun, A., "Supporting decision-making for self-adaptive systems: from goal models to dynamic decision networks," In International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 221-236, Springer, Berlin, Heidelberg, April 2013.
[27] Ramirez, A.J., Cheng, B.H., Bencomo, N. and Sawyer, P., "Relaxing claims: Coping with uncertainty while evaluating assumptions at run time," In International

Conference on Model Driven Engineering Languages and Systems, pp. 53-69, Springer, Berlin, Heidelberg, September, 2012.

Authors

Ahmed Abdo Ali Saeed received the Bachelor of Information Technology from Cairo University, Faculty of Computers And Information, Egypt in 2010 and Master of Computer Engineering from Ajou University, Korea in 2019.

He is interested in Goal model, Self-Adaptive Systems, Non-Functional Requirements Trade-Off and Multi-Entity Bayesian Network

Seok-Won Lee received the BS in Computer Science from Dongkuk University at Seoul, Republic of Korea, MS in Computer Science from University of Pittsburgh at Pittsburgh, Pennsylvania, and PhD in Information Technology from George Mason University at Fairfax, Virginia.

He is currently a Full Professor in the Department of Software and Computer Engineering at Ajou University, Republic of Korea since 2012. His areas of specialization include software engineering with specific expertise in ontological requirements engineering and domain modeling, and knowledge engineering with specific expertise in knowledge acquisition, machine learning and knowledge-based systems.