



Predicting Maintainability of Autonomous Software Systems

Taranjeet Singh, Avadhesh Kumar

School of Computing Science & Engineering, Galgotias University, Greater Noida,
Uttar Pradesh, India

Abstract— *Advances in science and technology have lead to the development of approaches based on biological systems such as human body. One such approach is autonomic computing it is inspired from Autonomous Nervous Systems. Systems based on this approach can manage them self with the high level guidance from human experts. Although these systems have freed humans from maintenance and other complexities but still we can't predict their degree of maintenance because there can be infinite unpredictable environmental conditions. Autonomic computing approach has been used for developing quality software systems. These software systems are injected with self properties of autonomic computing, so they are called Autonomous Software Systems (ASS) or Autonomic computing based software systems. In this paper taking maintainability as a prime concern form available literature we have identified self properties which strongly affect its maintainability. Further using precious opinions from the experts of the domain and fuzzy logic technique a model is proposed which is capable of predicting maintainability of ASS.*

Keywords— *Autonomic computing, self properties, ISO 9126 quality factors, fuzzy logic and Fuzzy Inference System*

I. INTRODUCTION

Autonomic computing approach is inspired from autonomous nervous systems e.g. Human Body and was coined by Paul Horn in 2001 [1]. These systems can run and work with minimum human interference and tend to remove the complexity of IT environment. Developing software systems based on autonomic computing requires a team of skilled and experienced requirement engineers, software developers, designers, policy managers etc. Some autonomic software systems have already been developed such as DB2 Configuration Advisor, AutoAdmin, Oceano, Automate, Recovery Oriented Computing, AntHill, IBM Tivoli etc [2] [3]. These software systems are currently being used in industries. But still we don't have clear idea of maintenance of such software systems. These software systems are self managing systems with self properties: *self configuring, self healing, self optimizing, self protecting, self aware, context aware, anticipatory* and *openness* [4] [5]. Many researchers later provided with more self properties these properties are self anticipating, self adapting, self critical, self defining, self destructing, self diagnosis, self governing, self organized, self recovery, self reflecting, self simulation, self stabilizing etc[6]. Autonomic software system must have ability to configure or reconfigure itself dynamically according to changing environmental conditions and their quality does not degrade. It can discover any malfunctioning and heal itself accordingly in two modes proactive and reactive like biological systems do. It must find a way to optimize its resources and tune itself according to IT environment. It should detect any malicious or security attack and must protect itself from it. These systems must not only aware of itself, its internal state, its components, its resources, and its business policies but also aware of its context or environment and also way of reacting if any change occurs internally or externally. They must be open i.e. they can run in multiple heterogeneous environments easily. Such systems should anticipate themselves while keeping their complexity hidden from the users. These software systems should be self adapting so they should adapt and respond to changes in its state or external operating environment. The environmental conditions of autonomic software systems can change in unpredictable ways so these systems are difficult to maintain as these systems can adapt in many other ways which even human could not predict even during developing and designing them. They should be maintainable enough so that their quality does not degrade while performing a high level of modifications and changes in their environment. If their quality degrades while performing maintenance then it will result in huge loss. Figure 1 shows the major and minor properties of autonomic computing. Architecture of autonomous systems includes five major building blocks; these building blocks are *autonomic manager, knowledge source, touch points, manual manager and enterprise service bus*. These building blocks perform manageability operations on the managed resource and make self management possible. Referring IBM white paper [7] these building blocks are explained under.

- **Autonomic manager:** It manages the managed element using MAPE or control loop. This loop consists of four major functions, these functions are *monitor, analyze, plan and execute*. Monitor function monitors the managed element or managed resource and it collects, aggregate, filter and store data collected from the managed resource [8].
- **Knowledge source:** It can be implemented using registry, database, dictionary or any other repository. Knowledge stored in knowledge source. Knowledge can be of particular type of data, syntax, semantics or

policy knowledge [7]. It is shared among various autonomic managers for achieving self management on the managed resource.

- **Touchpoints:** This block exposes the states of the managed resource for performing management operations on it. Autonomic manager uses manageability interface to communicate with touch points. Touchpoints exposes states of managed resource like properties such as configuration, metrics etc and relationships like hosts, users etc. Manageability interface is organized in two interfaces these are sensors and effectors, it also decreases the complexity of the system by providing interface to the autonomic manager [7].
- **Manual managers:** These managers are implemented through user interface. If IT expert observes any situation within the system he/she expert can perform some management functions manually using managers. These managers provide interface to IT experts for performing some management operations [7].
- **Enterprise Service Bus (ESB):** It is used for integrating other building blocks by providing interactions among them. Various building blocks of autonomic systems can be connected using enterprise service bus. Role that enterprise service bus plays are aggregating multiple manageability mechanisms for single management resource, enabling autonomic manager to manage many touch points, enabling many autonomic managers to manage a touchpoint and enabling many autonomic managers to manage many touch points [7].

SELF PROPERTIES OF AUTONOMIC COMPUTING	
MAJOR PROPERTIES	SELF CONFIGURING, SELF HEALING, SELF OPTIMIZING and SELF PROTECTING
MINOR PROPERTIES	SELF AWARE, OPEN, CONTEXT AWARE and ANTICIPATORY

Figure 1: Self properties of autonomic computing system [5]

II. LITERATURE REVIEW

In 2001 IBM's senior vice president of research introduced a new approach for managing the growing complexity of IT environment; this approach was called Autonomic computing. Till now lot of research has been done in this domain and various academic oriented and industrial projects have been developed and deployed. Horn [1] shows how the complexity of current IT environment is rising and unmanageability is increasing day by day. For solving these issues he gave approach called autonomic computing which is inspired from biological systems like human nervous system. Eight characteristics of autonomic computing are described in the paper. Still a problem was pointed out that how to design architecture for autonomic computing systems. Prashar and Hariri [2] present overview of the autonomic computing approach and the challenges it is going to face in the IT environment, paper presented eight characteristics more clearly, these characteristics are self aware, self configuring, self healing, self optimizing, self protecting, context aware, open and anticipatory. They provide architectural view of the autonomic systems and various applications based on the approach. Challenges like conceptual, architectural, middleware and application challenges are introduced. Salehie and Tahvaldiri [5] proposed categories of complexity in IT systems. Paper presents an overview of autonomic computing and discusses major applications and projects that have already been developed. Research issues and challenges from both theoretical and practical point of view are also outlined by authors. Relationship between quality factors and characteristics of autonomic computing is also specified. Dheraj and Sharma [9] tried to explore relationship between complexity and maintainability of autonomic systems and they proposed a model which can assess maintainability of autonomic systems using fuzzy logic, taking interface complexity, interaction complexity and function point per autonomic feature as inputs for mamdani's fuzzy inference systems complexity is estimated. Our proposed model is different from the one proposed by Dheraj and Sharma [9], as our model uses characteristics of ASS which effects its maintainability then experts are used for giving their opinions after we have received opinions form them a model is develop using fuzzy logic and inputs taken by us are the characteristics of ASS affecting its maintenance. Nami and Sharifi [10] commence with challenges in managing and maintaining large scale distributed computing systems and increasing complexity. For managing growing complexity idea of autonomic systems is specified. This survey paper tends to explain autonomic computing approach by providing definitions, characteristics, architectural details, challenges and issues and effects on quality factors. Zhu et.al [11] addresses various maintainability issues in designing of autonomic systems. They tend to identify and discuss various categories of autonomic element patterns and their relationship and various maintainability concerns. This paper tries to sort out various maintainability issues for autonomic element patterns using book store examples. Muller et.al [12] provides overview of autonomic computing and feedback control for autonomic computing. With the architecture of autonomic element patterns and reference architecture of autonomic system they tend to present various research challenges for autonomic systems these challenges are model construction, managing and leveraging uncertainty, making control loops explicit and characterizing architectural patterns and analysis framework. Sahadeva et.al [13] demonstrates a case study for minimizing maintenance cost by injecting autonomic manager as self maintenance system. They analysed scenarios of

problems in maintenance of software and tend to establish a new paradigm of software development using autonomic computing concepts. Case study of library management system is provided where autonomic manager is developed in software life cycle. Injecting of autonomic manager decreases overall maintenance cost. Ahuja and Dangey [14] introduce autonomic computing for dealing with increasing dynamism, growing complexity and uncertainty. Paper also provides how autonomic computing approach is evolved and architecture of autonomic computing systems. They discuss some of issues and challenges that autonomic computing systems may face in SDLC for their development, these challenges and issues are problem definition and specification, designing, implementation, policy management, testing and post implementation and maintenance challenges. Future scopes of autonomic computing approach are also discussed in the paper in the next section fuzzy logic is explained briefly.

III. FUZZY LOGIC

Soft computing is different from conventional hard computing; it is used to model very complex problems which has solutions that are unpredictable and uncertain. Soft Computing consists of various techniques such as fuzzy logic, artificial neural networks, genetic algorithms and many more [15]. In this paper fuzzy logic has been used for predicting maintainability of autonomic software systems as fuzzy logic can model problems which are vague, uncertain and ambiguous. The developed model is shown in section 5 of this paper. It is a multivalve logic that can be used for approximate reasoning. Fuzzy logic is approach which mimics the way human brain thinks and solves the. Fuzzy Logic contains fuzzy sets these sets are different from those conventional sets as they contains values as words or sentences which a conventional set is unable to contain [16]. These sets contain values which are imprecise and vague. Fuzzy sets define linguistic variables which are then converted into if then rules. The “if” part is known as antecedent or premise and the “else” part is known as consequent or conclusion. The output of the fuzzy logic is represented by a number which ranges from 0 to 1. In fuzzy logic FIS (Fuzzy Inference process) is used obtain output from a given input here mapping of input to output is done. This process comprises into five parts. First is Fuzzify input variables, here inputs of fuzzy are represented into a degree of membership between some range of number eg.0&1.In second part fuzzy operators are applied to the inputs or antecedent .There are two fuzzy operators if min or AND operator and max or OR operator Rules are made according using these operators. In third part Implication is applied from antecedent to consequent. Input of implication process is a single number which is output given by antecent. This method either truncates the output of fuzzy set or scales it. Then Aggregation of all the outputs is done in fourth part. In the last part called Defuzzification where centriod is calculated of aggregated fuzzy set and final result of Fuzzy Inference process is received. There are two methods of applying fuzzy Inference process i.e. Mamdani and Sugeno method .In this paper we have applied Mamdani’s method for predicting maintainability of the autonomic computing software systems. Certain factors are taken as inputs which effects the maintainability of such systems, these factors are the characteristics of autonomous software systems. A model is proposed to measure maintainability of Object oriented system is using fuzzy logic by Gill and Sharma [17].

IV. FACTORS EFFECTING MAINTAINABILITY OF AUTONOMOUS SYSTEMS

ISO 9126 provides six quality factors for measuring the quality of software systems these factors are functionality, reliability, usability, efficiency, maintainability and portability [18]. These factors are subdivided into sub factors. According to ISO 9126 maintainability quality factor have four sub factors these are analyzability, changeability, stability and testability [19]. Maintenance of autonomic software systems can’t be ignored; therefore we developed a model based using soft computing for predicting maintainability of autonomic software systems. The characteristics of autonomic software systems can be mapped with these quality factors according to their functionality. This mapping can help us to describe the characteristics of autonomous software system and define them more efficiently. Relationships between characteristics of autonomic systems and quality factors are shown by various researchers [5] [10] [20]. Maintainability can be mapped to self configuring, self healing, self optimizing, anticipatory and self adapting characteristics of autonomic computing. Self configuring is a capability of system to reconfigure itself according to changing environmental conditions dynamically, reconfiguring can be installing, updating, integrating the software and its entities itself by the software system without human interruption or little interruption thus self configuring helps in providing maintainability to the autonomous software systems. Self healing is the property to detect and to recover itself from any disruptions; system can detect failed components and fix them. Major objective of self healing is to maximize availability, survivability, maintainability and reliability of the system. Self optimizing is also called self tuning or self adjusting; such systems can tune its performance and resources in different environmental conditions. It provides efficiency, maintainability and functionality to the autonomic software systems. These systems can anticipate themselves by optimizing their resources and keeping their complexity hidden from the users. They can detect all the devices in their environment. This characteristic provides maintainability and efficiency to them. Self adapting also provides maintainability and reliability to autonomic software systems as the environment of these systems can change in unpredictable ways so they should be capable of adapting any change in their internal and external environment, and can maintain their elements accordingly. Other characteristics such as self adjusting, self diagnosing, self recovery, self destruction which are given by sterritt et.al [6] plays minimum role in maintainability of such software systems. So these characteristics can be ignored as they have minimum effect in maintenance of autonomic computing based software system. So self configuring, self healing, self optimizing, anticipatory and self adapting are characteristics which are useful in maintenance of such systems [20]. Next section explains methodology used for predicting maintainability of ASS.

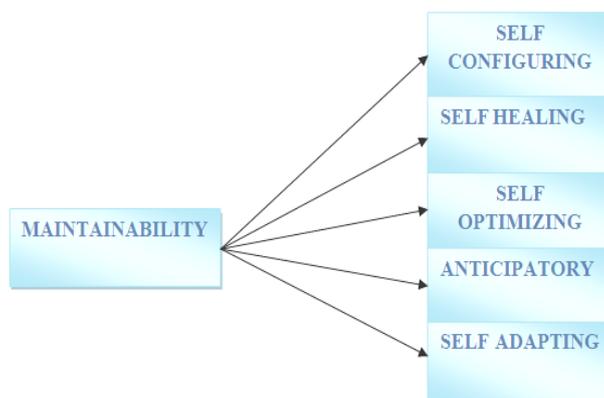


Figure 2: Relationship between maintainability (ISO 9126 quality factor) and self properties of Autonomic system [10]

V. METHODOLOGY USED

Major objective of our work is to estimate maintainability of ASS firstly characteristics that affect its maintainability are identified and then an Expert’s opinion form is developed using these characteristics. SC (self configuring), SH (self healing), SO (self optimizing), AN (anticipatory) and SA (self adapting) are taken as inputs and MN (Maintainability) is taken as the output of the expert’s opinion form. After identifying inputs and outputs, these inputs are divided into functions like SC, SH and AN are divided into L (low), M (medium) and H (high) other inputs SH and SA are divided into S (satisfactory) ,G (good) and VG (very good). Output MN is divided into five functions i.e. VL (very low), L (low), M (medium), H (high) and VH (very high). For ASS with all five properties that effect maintainability 243 possible combinations are made of the inputs as shown in Figure 3. After all possible combinations are developed from the inputs this form is given to Expert’s of the domain. These experts include industrial, researchers and industrial experts working or have experience in autonomic computing domain. Suppose any ASS is developed or being developed these experts will fill the expert opinion form for estimating its maintainability. Suppose five experts filled this form in a combination we have H, M, S, M, and G as input functions. Experts started filling the output function for this combination, expert opinions were M, M, H, M and H respectively for this combination. Then we need to take average of the opinions given by experts and average of this combination will be M. So M will be considered as final result for this combination. Similarly whole Expert’s Opinion Form will be filled by taking opinions and then taking average values for the function as final value. Once Expert opinion form is developed and filled by experts for any ASS.

INPUTS						OUTPUTS
	SC	SH	SO	AN	SA	MN
1	L	L	S	L	S	VL
2	L	L	S	L	G	VL
3	L	L	S	L	VG	L
4	L	L	S	M	S	VL
5	L	L	S	M	G	L
6	L	L	S	M	VG	L
7	L	L	S	H	S	L
8	L	L	S	H	G	M
9	L	L	S	H	VG	M
10	L	L	G	L	S	VL
11	L	L	G	L	G	L
12	L	L	G	L	VG	M
13	L	L	G	M	S	L
14	L	L	G	M	G	M
15	L	L	G	M	VG	M
16	L	L	G	H	S	L

Figure 3: Sample of Expert’s Opinion Form

It is necessary to use a model as opinion form alone is not enough to estimate maintainability of the system. For that purpose model is proposed, next section shows thee model is used for predicting maintainability of the ASS. As shown in Figure 3 the model based on all the five inputs (SC, SH, SO, AN and SA) and the average of output (MN) given by

experts for each possible combination. The entire working from selecting ASS to predicting its maintainability is shown using steps below.

1. Select any ASS whose maintainability is to be predicted.
2. Identify all the characteristics that are incorporated into the ASS, form this list of characteristics choose only those characteristics which affect their maintenance (i.e. self configuring, self healing, self optimizing, anticipatory and self adapting characteristics whichever is present).
3. Taking the characteristics affecting maintainability that are present in ASS as input and maintainability as output, divide these characteristics into functions and taking all the combinations develop Expert's Opinion Form as shown in Figure 3. Suppose if only two characteristics are present then divide them into three functions each, make all the combinations from these characteristics only.
4. Give Expert's Opinion Form to the expert's of the domain of autonomic computing for their precious opinions. These expert's can industrial experts (software designers, software developers, maintainers, project managers etc) as well as academic expert's (researchers, experienced professors).
5. After getting opinions from these experts take the average of output given by them and mark the average as final output. Suppose five experts gave their opinions as VL, L, L, VL and L then average would be L.
6. Using Expert's opinion form take inputs and output for Fuzzy inference system (refer figure 4), give appropriate membership functions and design Fuzzy rule base through MATLAB.
7. Fuzzy Rule viewer is formed trained it for predicting maintainability of ASS.

Using these steps a generic model can be proposed for prediction of maintainability of any autonomic computing based software system. The next section explains how the model is proposed using Expert's opinion form in fuzzy logic.

VI. PROPOSED MODEL

When expert opinion form is filled then next step is to form a model which would use inputs and output of the form filled by experts. Since Experts Opinion Form alone is not capable for maintainability prediction so for purpose that a model is to be proposed. We used fuzzy logic for proposing the desired model through MATLAB R2013a. As fuzzy logic is one of simplest technique of soft computing, easy to understand, data which we have is imprecise and based on natural language. There are three operators used in fuzzy logic AND, OR and NOT. In this model AND operator is used for performing logical AND of the antecedents. Fuzzy Inference Process (FIS) is used for mapping between given inputs and outputs by this mapping a medium is made for taking decisions using fuzzy logic. FIS process consists of five basic steps these steps are fuzzification of input variables, applying operators to antecedent, implication of antecedent to consequent, aggregation of all the consequents and last step is defuzzification. In the two FIS systems we have used mamdani's method. Figure 4 shows the FIS systems and inputs and outputs for the process. Inputs and outputs taken for the model are the same used for Expert's Opinion Form.

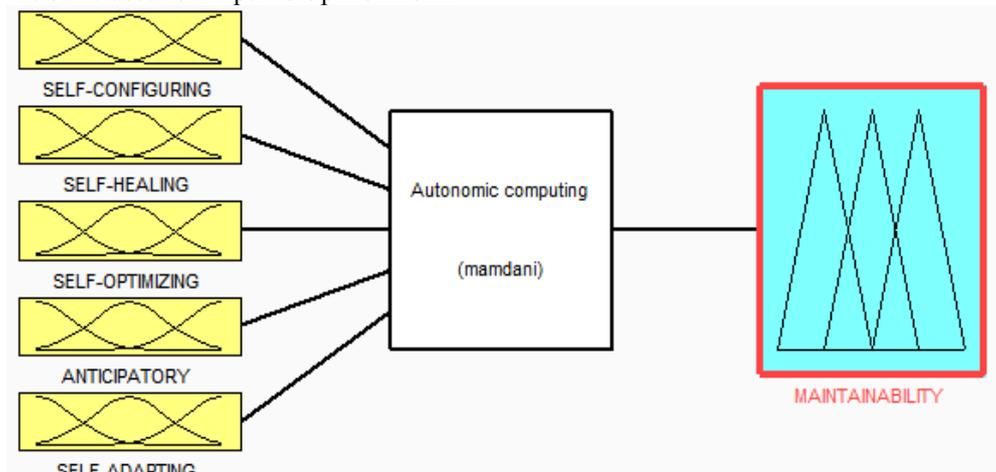


Figure 4: Fuzzy Inference Systems

These inputs are the self properties of the autonomic systems that affect their maintenance. The model is proposed for the autonomic software system having all the five self properties i.e. self configuring, self healing, self optimizing, anticipatory and self adapting. As shown in the FIGURE 4 for FIS mamdani's model is used with output as MAINTAINABILITY and inputs as SELF-CONFIGURING, SELF HEALING, SELF- OPTIMIZING, ANTICIPATORY and SELF-ADAPTING. Maintainability in this model estimates how easily autonomic software system can modify its components and itself according to unpredictable environment and internal conditions. Membership functions of all the inputs and outputs are taken same as the functions taken for inputs and outputs for the Figure 3. Figures 5, 6, 7, 8 and 9 show the membership functions of all the inputs of fuzzy inference systems for predicting maintainability of ASS incorporated with all the five characteristics. Details of all the five inputs with their graphical representations are shown in these figures below.

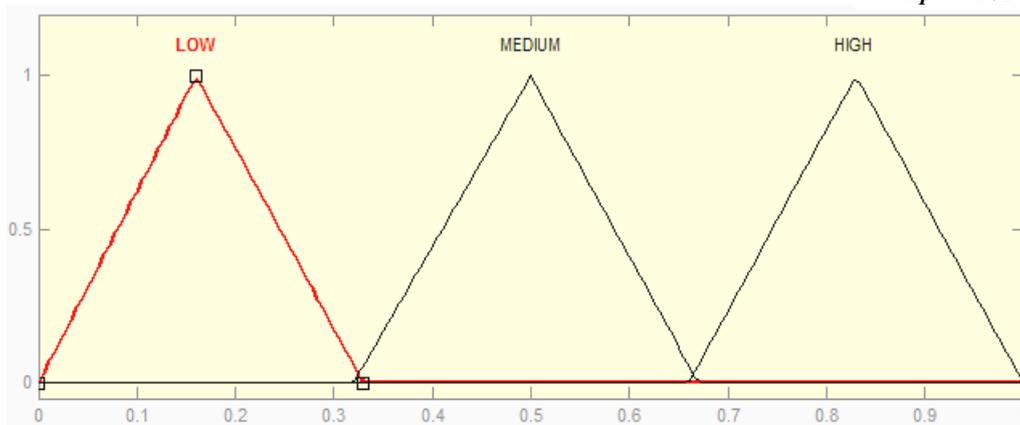


Figure 5: Membership functions of input SELF-CONFIGURING

Each input variable is defined in terms of four parameters i.e. name, type, range and display range, each input has three membership functions similar to functions given in Figure 3. Every membership function of each input has three parameters. Trimf indicates that the membership functions are in triangular form. Consider the input variable named as SELF-CONFIGURING which is shown in figure 5.

Further details of this input are given below.

[Input 1]

Name= 'SELF-CONFIGURING'

Range= [0 1]

Number of membership functions = 3

MF1= 'LOW'; 'trimf'; [0 0.16 0.33]

MF2= 'MEDIUM'; 'trimf'; [0 0.16 0.33]

MF3= 'HIGH'; 'trimf'; [0 0.16 0.33]

Second input of fuzzy inference systems is named as SELF OPTIMIZING is shown in figure 6 it also consists of three membership functions LOW, MEDIUM and HIGH. Further details of input 2 are given below.

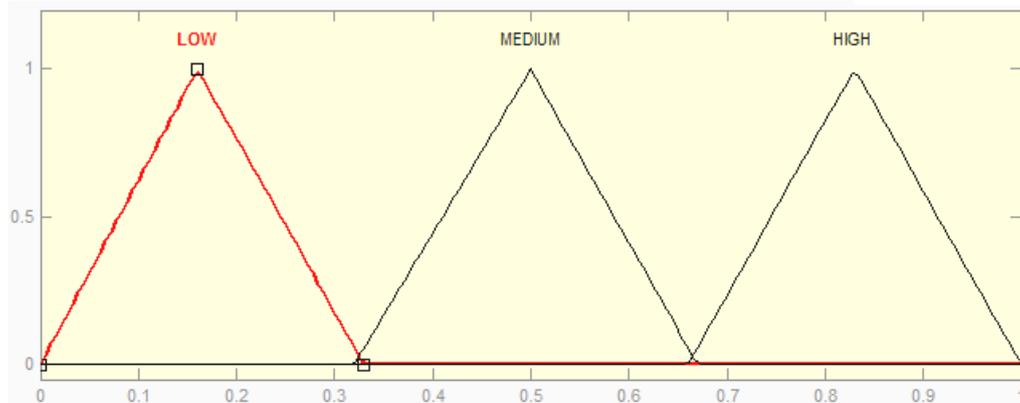


Figure 6: Membership functions of input SELF-HEALING

[Input 2]

Name= 'SELF-HEALING'

Range= [0 1]

Number of membership functions = 3

MF1= 'LOW'; 'trimf'; [0 0.16 0.33]

MF2= 'MEDIUM'; 'trimf'; [0 0.16 0.33]

MF3= 'HIGH'; 'trimf'; [0 0.16 0.33]

Third output is SELF-OPTIMIZING which also contains three membership functions similar to functions taken in Expert's Opinion Form. These functions are SATISFACTORY, GOOD and VERYGOOD. Details of membership functions for input SELF-OPTIMIZATION are shown below and are graphically shown in figure 7.

[Input 3]

Name= 'SELF-OPTIMIZING'

Range = [0 1]

Number of membership functions = 3

MF1= 'SATISFACTORY'; 'trimf'; [0 0.16 0.33]

MF2= 'GOOD'; 'trimf'; [0 0.16 0.33]

MF3= 'VERYGOOD'; 'trimf'; [0 0.16 0.33]

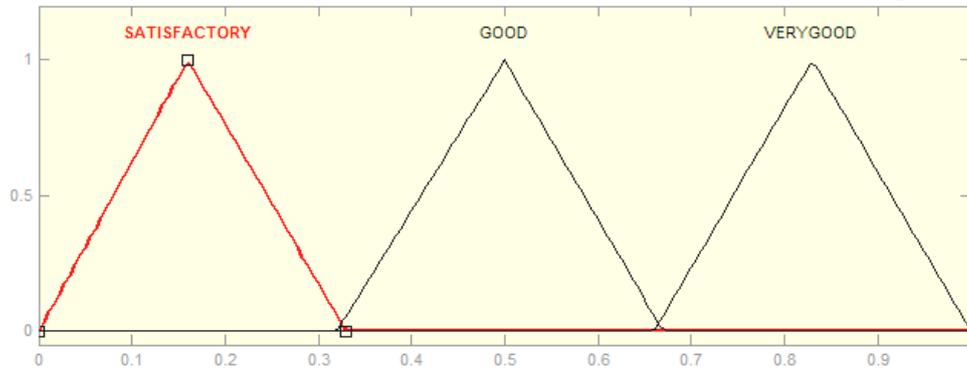


Figure 7: Membership functions of input SELF-OPTIMIZING

As shown figure 8, input 4 (ANTICIPATORY) containing three membership functions namely LOW, MEDIUM and HIGH.

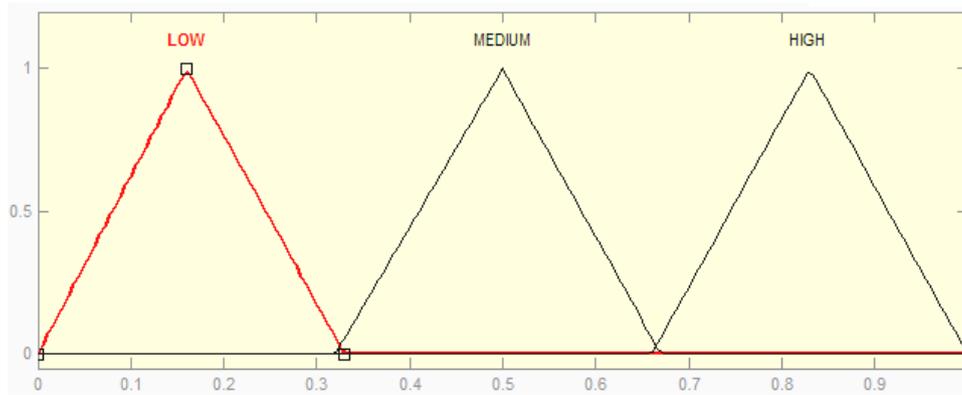


Figure 8: Membership functions of input ANTICIPATORY

Further details of this input are given below.

[Input 4]

Name= 'ANTICIPATORY'

Range= [0 1]

Number of membership functions = 3

MF1= 'LOW'; 'trimf'; [0 0.16 0.33]

MF2= 'MEDIUM'; 'trimf'; [0 0.16 0.33]

MF3= 'HIGH'; 'trimf'; [0 0.16 0.33]

Fifth output is SELF-ADAPTING which also contains three membership functions similar to functions taken in Expert's Opinion Form for input SA. These functions are SATISFACTORY, GOOD and VERYGOOD. Details of membership functions for input SELF-ADAPTING are shown below and are graphically shown in figure 9.

[Input 5]

Name= 'SELF-ADAPTING'

Range= [0 1]

Number of membership functions = 3

MF1= 'SATISFACTORY'; 'trimf'; [0 0.16 0.33]

MF2= 'GOOD'; 'trimf'; [0 0.16 0.33]

MF3= 'VERYGOOD'; 'trimf'; [0 0.16 0.33]

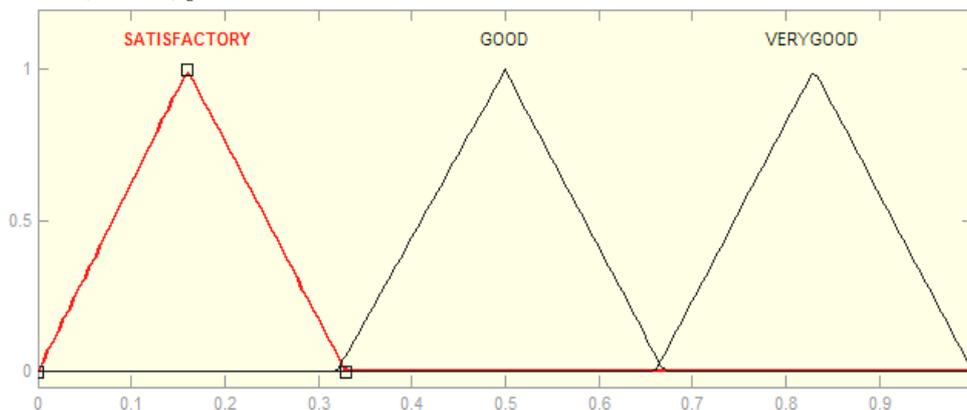


Figure 9: Membership functions of input SELF-ADAPTING

We have five inputs namely SELF-CONFIGURING, SELF-HEALING, SELF OPTIMIZING, ANTICIPATORY and SELF-ADAPTING which are shown and details are all explained above. Besides these inputs we have output named as MAINTAINABILITY which contains five membership functions named as LOW, VLOW, MEDIUM, HIGH and VHIGH. Figure 10 shows the membership functions for output variable MAINTAINABILITY and details of output membership function are given below.

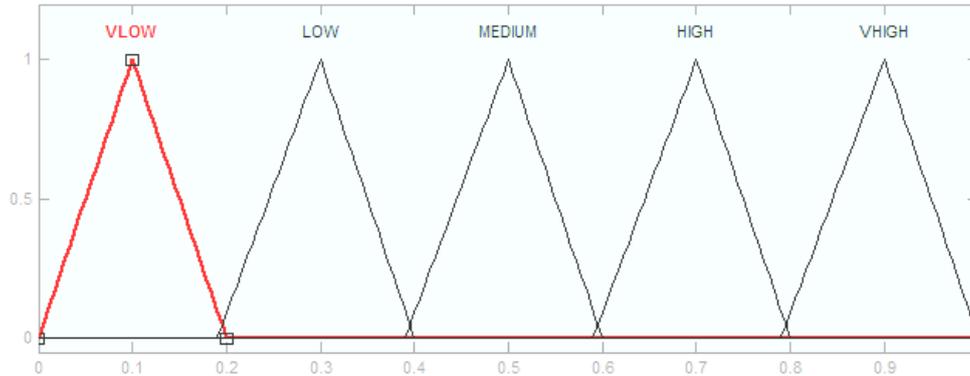


Figure 10: Membership Functions of output MAINTAINABILITY

[Output 1]

Name= 'MAINTAINABILITY'

Range= [0 1]

Number of membership functions = 5

MF1= 'VLOW'; 'trimf'; [0 0.1 0.2]

MF2= 'LOW'; 'trimf'; [0.9 0.3 0.4]

MF3= 'MEDIUM'; 'trimf'; [0.39 0.5 0.6]

MF4= 'HIGH'; 'trimf'; [0.59 0.7 0.8]

MF5= 'VHIGH'; 'trimf'; [0.79 0.9 1]

Generally with m inputs and n membership functions n^m rules can be generated. In our fuzzy inference system we have five inputs (SELF-CONFIGURING, SELF-HEALING, SELF-OPTIMIZING, ANTICIPATORY and SELF-ADAPTING) and one output (MAINTAINABILITY). Each of the input variables contains three membership functions so in total we have 3^5 (i.e. 243) fuzzy rules. Format of fuzzy rule base is as follows.

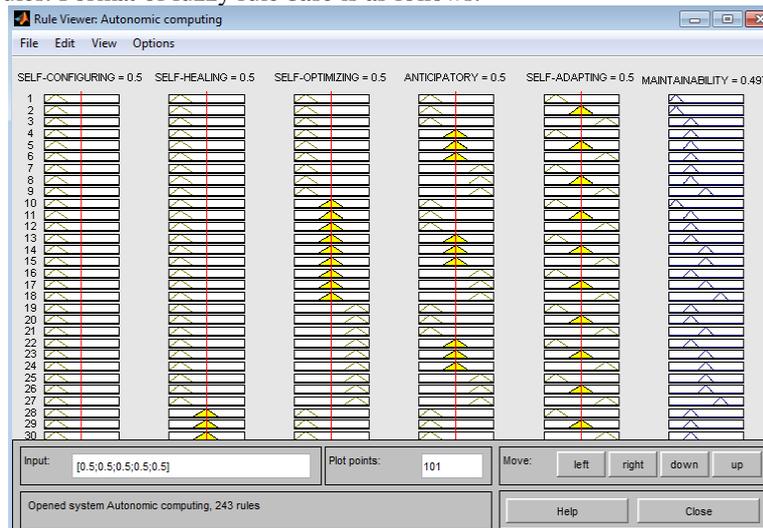


Figure 11: Graphical Representation of Fuzzy Rules

IF (SELF-CONFIGURING is LOW/MEDIUM/HIGH) and (SELF-HEALING is LOW / MEDIUM /HIGH) and (SELF-OPTIMIZING is SATISFACTORY/GOOD/ VERYGOOD) and (ANTICIPATORY is LOW/MEDIUM/HIGH) and (SELF-ADAPTING is SATISFACTORY/GOOD/VERGOOD) then (MAINTAINABILITY is VLOW/LOW/MEDIUM/HIGH/VHIGH).

Examples of fuzzy rules are shown below.

1. IF (SELF-CONFIGURING is LOW) and (SELF-HEALING is LOW) and (SELF-OPTIMIZING is SATISFACTORY) and (ANTICIPATORY is LOW) and (SELF-ADAPTING is SATISFACTORY) then (MAINTAINABILITY is VLOW).
2. IF (SELF-CONFIGURING is HIGH) and (SELF-HEALING is LOW) and (SELF-OPTIMIZING is SATISFACTORY) and (ANTICIPATORY is LOW) and (SELF-ADAPTING is SATISFACTORY) then (MAINTAINABILITY is LOW).

3. IF (SELF-CONFIGURING is HIGH) and (SELF-HEALING is HIGH) and (SELF-OPTIMIZING is VERYGOOD) and (ANTICIPATORY is HIGH) and (SELF-ADAPTING is VERGOOD) then (MAINTAINABILITY is VHIGH).

So $3^5 = 243$ rules are designed using fuzzy logic through MATLAB. Figure 11 shows the graphical representation of these rules designed in MATLAB. As it is not possible for the system to show all the 243 rules at once so there are four icons for moving through this rule viewer, these icons are left, right, up and down. These fuzzy rules can be trained for any ASS with all the five self properties that affect maintainability and accordingly maintainability can be estimated. In figure 11 values of all inputs i.e. SELF-CONFIGURING, SELF-HEALING, SELF-OPTIMIZING, ANTICIPATORY and SELF-ADAPTING are 0.5 so the output MAINTAINABILITY is 0.497. In the next section results of some scenarios are shown using this proposed model.

VII. RESULTS

We can train fuzzy rule base according to ASS self properties affecting its maintenance and can predict its maintainability. In this chapter two scenarios are taken, in first scenario the values of each input variables are decreased the so the output variable is also decreased to great extent. In second scenario values of each input variable is increased and output is rise to great extent. In Figure 11 graphical representation of rule base is shown each input of fuzzy inference system has value 0.5 and output MAINTAINABILITY is 0.497. In Figure 12 values of each input of fuzzy inference system is decreased and that also affects output MAINTAINABILITY which is also decreased, as input SELF-CONFIGURING value is 0.273 , SELF-HEALING is 0.283, SELF-OPTIMIZING is 0.273, ANTICIPATORY is set as 0.167 and SELF-ADAPTING is set as 0.233 and output MAINTAINABILITY will decrease to 0.1. Figure 13 shows how increase in input values leads to rise in the output of ASS. In the figure 7.3 SELF-CONFIGURING value is set as 0.776, SELF-HEALING is 0.721, SELF-OPTIMIZING is 0.75, ANTICIPATORY is set as 0.801 and SELF-ADAPTING is set as 0.773 and output MAINTAINABILITY will rise to 0.896.

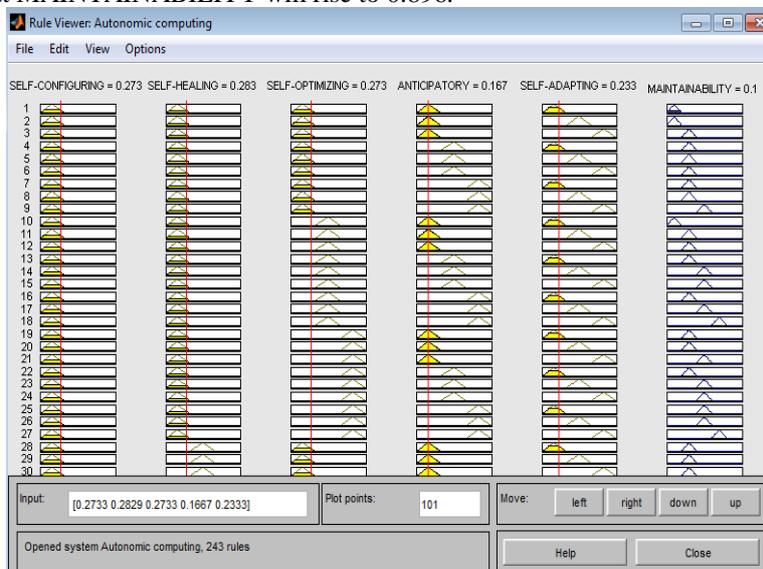


Figure 12: Graphical representation of fuzzy rules indicating decreased in MAINTAINABILITY

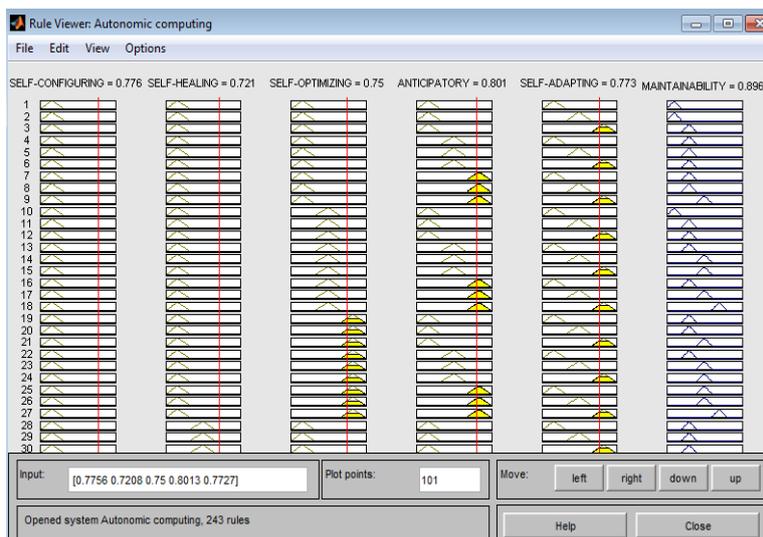


Figure 13: Graphical representation of fuzzy rules indicating increase in MAINTAINABILITY

Three scenarios are taken for ASS having all the five self properties that affect maintainability these scenarios are shown in figure 14. In first scenario values each inputs of FIS are decreased which leads to decrease in systems overall maintainability. In second scenario values of each input of FIS are increased which leads to rise in overall systems maintainability and third scenario is same which we obtain after forming fuzzy rule base in MATLAB.

No.	INPUTS					OUTPUT
	SELF-CONFIGURING	SELF-HEALING	SELF-OPTIMIZNG	ANTICIPATORY	SELF-ADAPTING	MAINTAINABILITY
1.	0.273	0.283	0.273	0.167	0.233	0.1
2.	0.776	0.721	0.75	0.801	0.773	0.896
3.	0.5	0.5	0.5	0.5	0.5	0.497

Figure 14: Maintainability Prediction

VIII. CONCLUSIONS

Maintainability of any autonomous software system can be predicted using this generic model. If the maintainability is not known and software system is deployed to user then it can result in poor performance, increasing overall maintenance cost, increasing overhead of the maintainers and can even lead to failure of the system. When the software system is based on autonomic computing, then it has ability to maintain itself and we know its maintainability before deploying to user than changes can be made easily but if deployed to the user changing and correcting faulty components would be much expensive and overhead will increase to great extent. So this model can be used for knowing maintainability of any autonomous software systems using opinions from experts of the domain. The results of the proposed model are based on scenarios of ASS having all the self properties affecting their maintenance. There is always scope of improvement in research so our future work can be using this model for predicting maintainability of any current ASS which is currently being used in industries or currently being developed by industries. This model can be developed using other soft computing techniques also so using these techniques like genetic algorithms and artificial neural networks for predicting maintenance of ASS can future work.

REFERENCES

- [1] Horn, P., “Autonomic Computing: IBM’s Perspective on the state of information technology”, IBM, 2001.
- [2] Parashar, M., Hariri, S., “Autonomic computing: An overview”, Springer-Verlag Berlin Heidelberg, pno.247-259, 2005.
- [3] X. Li, H. Kang, P. Harrington and J. Thomas, “Autonomic and trusted computing paradigms,” Springer-Verlag Berlin Heidelberg, page 143-152, 2006.
- [4] Ganek, G, A., and Corbi, A., “The dawning of autonomic computing era”. IBM systems journal volume 42, no 1, 4-18, 2003.
- [5] Salehie, M., Tahvildari, L., “Autonomic computing: emerging trends and open problems”, ACM 2005.
- [6] Sterritt, R., Parashar, M., Tianfield, H., Unland, R.,”A concise introduction to autonomic computing”, Advance engineering informatics science, page no 181-187, 2005, ELSEVIER.
- [7] IBM White paper. “An architecture blueprint for autonomic computing”, 2003
- [8] Hubescher, C, M., and Maccan, A, J., “A survey of Autonomic Computing- Degrees, Models and Applications”. ACM Computing Surveys, Volume.40, No.3, Article 7, 1-28, 2007.
- [9] Dehraj, P and Sharma, A., “Complexity Based Maintenance Assessment for Autonomic Agent”. Advances in Computer Science, 221-231, 2015.
- [10] Nami, M, R and Sharifi, M., “Autonomic computing: A new approach”, proceeding of first Asia international conference on modeling and simulation (AMS’07), IEEE, 2007.
- [11] Zhu, Q., Lin, L., Kienle,M,H., and Müller,A,H., “Characterizing Maintainability Concerns in Autonomic Element Design”, ICSM,IEEE, page no 197-206, 2008.
- [12] Muller. H. A, Kienle. H. M and Stege. U., “Autonomic computing now you see it, now you don’t”, Springer Verlag Heidelberg, 32-54, 2009.
- [13] Sahadeva, K., Yadav, K, S., and Sharma,A., “Minimizing the cost of software by injecting autonomic computing elements”, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, page no 685-690, 2012.
- [14] Ahuja, K., Dangey, H., “Autonomic Computing: An emerging perspective and issues”, International conference on issues and challenges in Intelligent computing techniques (ICICT), IEEE, page no 471-475, 2014.
- [15] Zadeh, A, L., “Softcomputing and Fuzzy Logic”, IEEE software, page 48-56, 1994.
- [16] Bih, J., and Madhavan, G., “Paradigm shift — an introduction to fuzzy logic”, IEEE POTENTIALS, page 6-10, 2006.
- [17] Gill, S, N., and Sharma, M., “Application of Fuzzy Inference System to Measure Maintainability of Object Oriented Software”, 13th International Conference on Software Engineering Research and Practice, World Congress in Computer Science, Computer Engineering and Applied Computing, 2015.
- [18] Boegh, J., “A New Standard for Quality Requirements” IEEE SOFTWARE, 57-67, 2008.
- [19] Jung, H., Kim, S., and Chung, C. “Measuring Software Product Quality: A Survey of ISO/IEC 9126”, IEEE Software, 88-92, 2004.
- [20] Salehie, M and Tahvildari, L., “Self-Adaptive software: Landscape and Research Challenges”, ACM Transactions on Autonomous and Adaptive systems, Vol. V, No. N, March, 2009.