

A Fuzzy Approach for Evaluation of Maintainability of Object Oriented Software System

Sanjay Kumar Dubey
Amity University
Sector-125, NOIDA
Uttar Pradesh, India

Prof. Ajay Rana
Phd, Amity University
Sector-125, NOIDA
Uttar Pradesh, India

ABSTRACT

The demand for efficient software system is increasing day by day. Maintainability is considered as an important quality factor for developing the efficient software system. Recent trends show that mostly software systems are using object-oriented technique to develop the quality software products. Object-oriented approach enhances the maintainability of software system. In literature there are no well defined criteria to evaluate maintainability. This paper proposes a fuzzy model to quantify maintainability of object-oriented software system. The model takes object-oriented projects and evaluates its maintainability. The value obtained by fuzzy model is validated by using analytical hierarchy processing technique.

General Terms

Software Engineering

Keywords

Maintainability, AHP, Object-oriented, Model, Fuzzy, Metric

1. INTRODUCTION

A software product requires a number of measures to be taken into account for its designing. The most important measure that must be considered in any software product is its design quality [1]. Among all the quality criteria, software maintainability is broadly accepted as a highly significant quality criterion in the economic success of engineering systems and products. There is a need for software engineers to understand how various components of a design interact in order to maintain and enhance the reliability of software during maintenance. Maintenance of software is one of the most expensive and resource requiring phase of the software development process. Statistics from various organizations shows that 40% to 80% of the development expenditure on the average software is spent in the 'maintenance' phase in which bugs are fixed, features are enhanced, and the software is updated to keep pace with changing domain requirements [2], [3]. Thus maintainability evaluation is an essential component of modern software development life cycle. Evaluation of software maintainability, if done accurately, can be useful in aiding decision making related to the software, efficiency of the maintenance process, comparing productivity and costs among different projects, allocation of resource and staff, and so on. This minimizes the future maintenance effort [4]. Assessing maintainability of a system is a difficult process as many contradictory criteria must be considered in order to reach a decision [5]. Hence a layered approach is used to evaluate software maintainability [6]. In this approach, fuzzy evaluation method in combination with Analytic Hierarchy Process (AHP) is utilized to handle problems involving multiple indices based on quantitative procedural information

to get the qualitative results. AHP [7] is used since it helps to capture both subjective and objective evaluation measures, providing a useful mechanism for checking the consistency of the evaluation measures and suggested alternatives thus reducing bias in decision making.

The study has been conducted in object-oriented paradigm. This is due to the fact that the primary purpose of object-oriented design is to improve software quality criteria such as maintainability, reliability, usability, etc by managing software complexity. The logical complexity of the source code has a strong correlation to the maintainability of the resultant software [8], [9]. Reducing the software development and maintenance costs is the main objective of object-oriented design. In order to facilitate the analysis and evaluation of maintainability of an object-oriented system, Chidamber and Kemerer (CK) metrics [10] have been used. CK metrics are design complexity metrics that aid in identifying certain design and code characteristics in object-oriented software which in turn helps in assessing external software qualities such as software defects, testing, and maintenance effort [11]. Hence the main objective of this paper is to evaluate software maintainability by using fuzzy layered evaluation method in combination with Analytic Hierarchy Process.

2. FACTORS AFFECTING MAINTAINABILITY

For evaluating the maintainability of object oriented system, five factors are taken viz. complexity, class, coupling, inheritance and number of children. These factors are chosen since they are the design complexity factors and show more impact on the maintainability of object-oriented software system. Brief outlines of all these factors are shown below:

Complexity

By software complexity we mean the difficulty to preserve, modify and comprehend the software.

Class

A class is a basic unit of OOP and it can be portrayed as a set of objects that includes same methods, attributes and relationships.

Coupling

Coupling means the interdependency between different components or functions. Coupling is the measure of interconnections among the modules in a software structure.

Inheritance

Inheritance is defined as classes having same methods and operations based on hierarchy. It is a mechanism whereby one object acquires the characteristics from one or more other objects.

Number of Children

Number of Children defines the number of subclasses subordinate to a class in the hierarchy. It indicates the potential influence of a class on design and system.

3. METRICS USED FOR EVALUATING MAINTAINABILITY

Maintainability evaluation into object-oriented paradigm uses fuzzy layered technique [6]. In order to do so, Chidamber and Kemerer (CK) software metrics [10] have been used. These metrics are aimed at assessing the design of object-oriented system rather than implementation. This makes them more suited to object-oriented paradigm as object-oriented design puts great emphasis on the design phase of software system [12]. The CK metric suite consists of six design complexity metrics- WMC, DIT, NOC, CBO, RFC and LCOM. Except for LCOM, all these metrics can be used as maintainability predictors as LCOM is uncorrelated with the maintainability of the software [13]. The CK metrics (except LCOM) are briefly described as follows [10]:

3.1 WMC (Weighted Methods per Class)

It is a weighted sum of all the methods defined in a class. It measures the complexity of a class. It also predicts how much time and effort is required to develop and maintain the class. High WMC indicates greater complexity and hence low maintainability.

3.2 DIT (Depth of Inheritance Tree)

It is the length of the longest path from a given class to the root class in the inheritance hierarchy and is measured by the number of ancestor classes. So this metric calculates how far down a class is declared in the inheritance hierarchy. High DIT indicates greater design complexity and more fault-proneness.

3.3 NOC (Number of Children)

It is equal to the number of immediate child classes derived from a base class. High NOC means greater level of reuse, more effort required for testing, more complexity and fault-proneness.

3.4 CBO (Coupling Between Objects)

For a class, CBO is measured by counting the number of other classes to which it is coupled. Coupling is a measure of interdependence of two objects. Two classes are coupled if methods of one use methods and/or instance variables of the other. High CBO indicates complex design, decreases modularity, and complicates testing of the class.

3.5 RFC (Response for a Class)

It is the count of all the methods which can potentially be executed (directly or indirectly) in response to a message to an object of that class or by some method in the class. (This includes all methods accessible within class hierarchy). High RFC means more effort required for testing, greater design complexity and fault-proneness.

The values of all the above metrics are inversely proportional to the maintainability of a system [14].

4. FUZZY APPROACH FOR MAINTAINABILITY EVALUATION

4.1 Proposed Model

There are various methods for maintainability measurement [15] but none of them was exact approach. Thus we propose a fuzzy model approach for maintainability measurement of an object oriented system. Fuzzy logic is a captivating field of

research these days as it considers the fuzzy value instead of binary values. The benefit of using fuzzy logic is that the fuzzy logic models can be built even with little or no data. In this paper, we propose a fuzzy model to measure maintainability. Fuzzy logic is used because maintainability depends on various factors. These factors are fuzzy in nature.

4.2 Working of the model

In this model we have taken five inputs as complexity, class, coupling, inheritance and number of children to provide a crisp value of maintainability using rule base. Fuzzy Inference System (FIS) uses fuzzy logic to map the input to output. Mamdani fuzzy inference method is used.

After the fuzzification process is completed, we take the fuzzy sets for output variable that requires defuzzification. For defuzzification the input will be a fuzzy set and output will be a singleton value. The centroid method which gives center of area under curve is most commonly used for defuzzification.

There are many types of membership functions but for simplicity we have used triangular membership function.

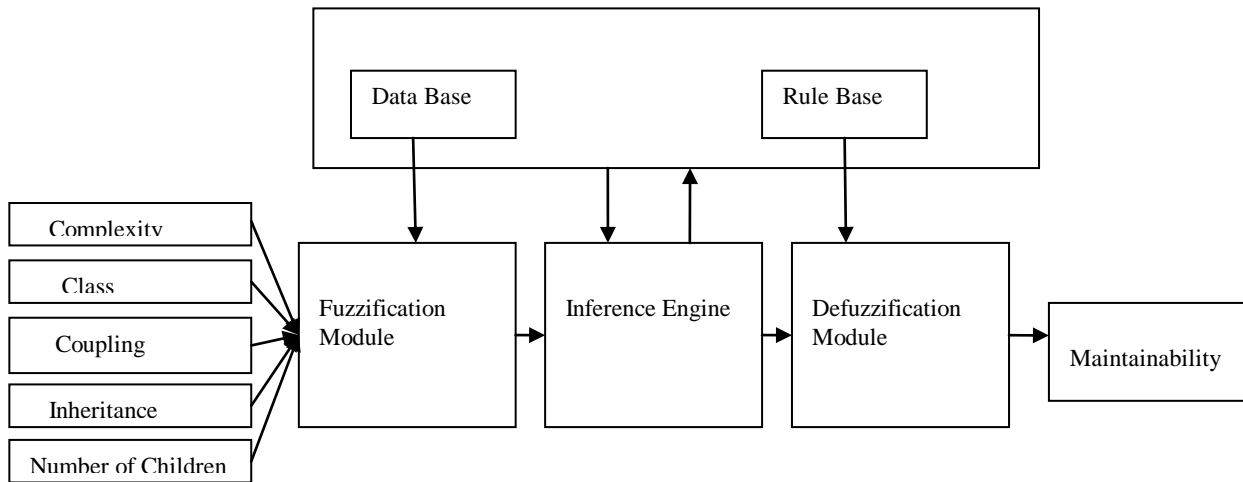


Figure 1. Proposed Fuzzy Model

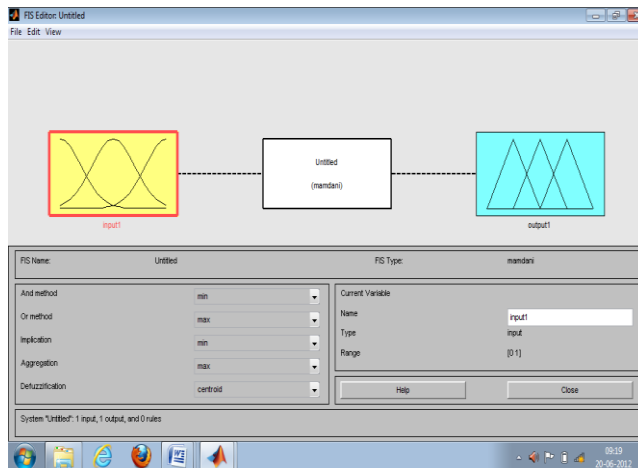


Figure 2. Inputs and Outputs of Fuzzy Model

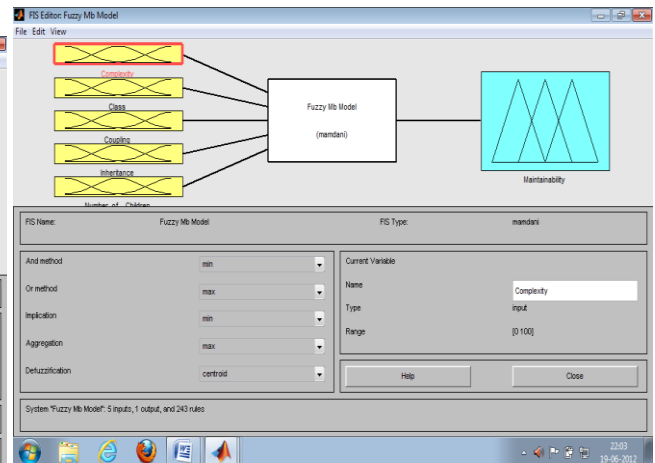


Figure 3. Membership function for complexity

4.3 Membership Function for Inputs and Output

For measuring maintainability of an object oriented system we have considered five inputs- complexity, class, coupling, inheritance and number of children. These are shown in figure 3, 4, 5, 6, 7. We have taken three membership functions –low, medium and high for each input. These inputs are taken on an interval of [0,100].

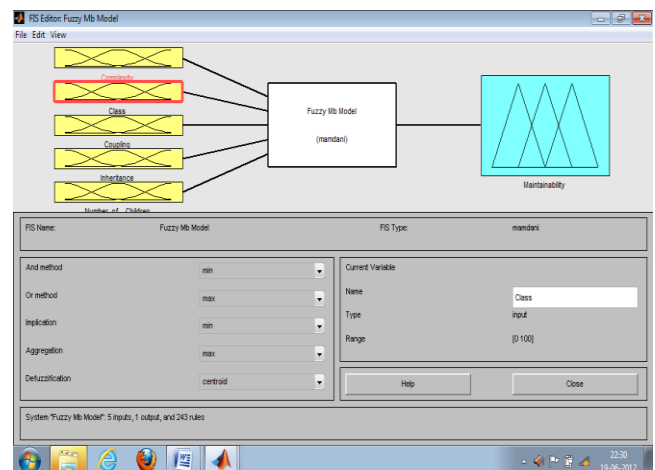


Figure 4. Membership function for class

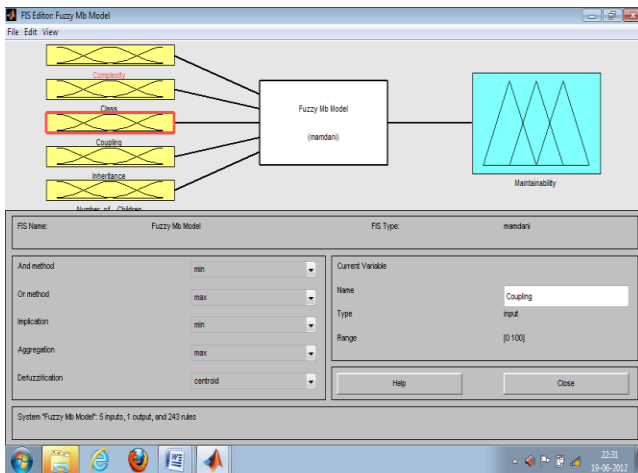


Figure 5. Membership function for coupling

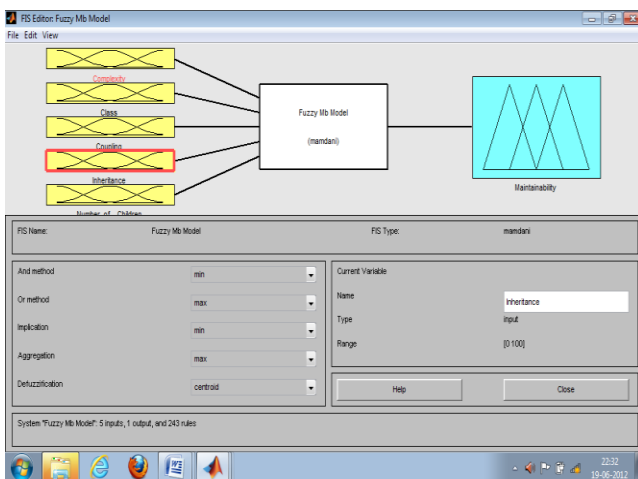


Figure 6. Membership function for inheritance

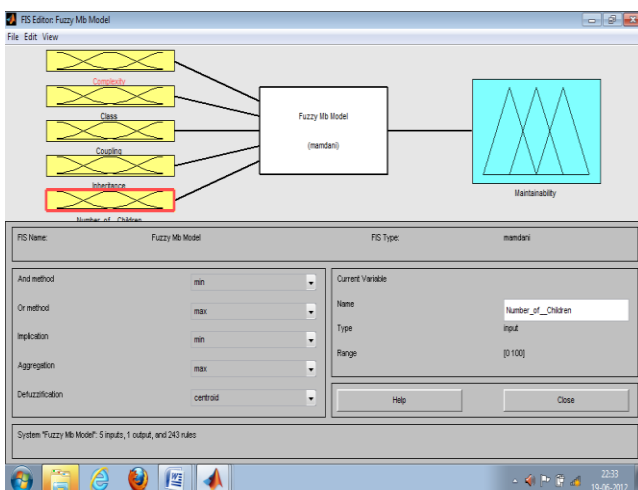


Figure 7. Membership function for number of children

For the output (maintainability) we have taken five membership functions –very low, low, medium, high and very high. The range for this is also taken from [0,100]. This is shown in the figure 8.

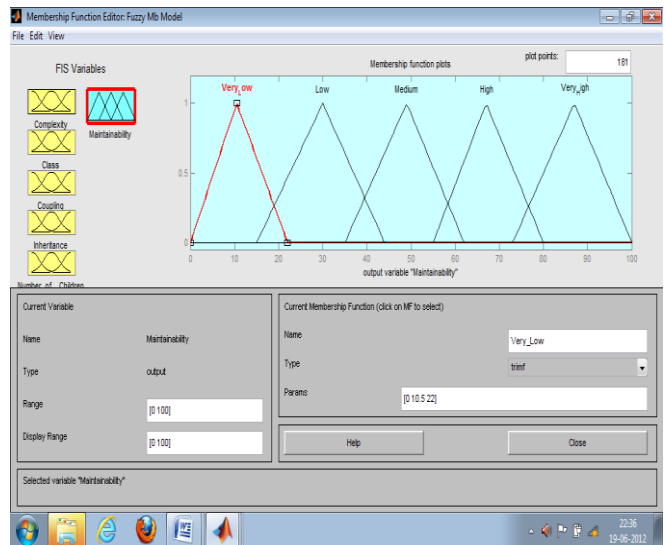


Figure 8. Membership function for maintainability

4.4 Knowledge Base and Evaluation Process

In order to measure maintainability of a software system, all the five inputs (complexity, class, coupling, inheritance and number of children) are integrated with the help of fuzzy model. Each of these inputs contains three terms- Low, Medium and High. Thus by integrating and forming different combinations for all the inputs we get 243 rules. In general terms if there are x inputs with y terms each then total number of rules R formed will be $y*y*y....x$ times. Thus $R=y^x$

In our model we have 5 inputs and 3 terms. Hence our total number of rules will be $3^5=243$. For all 243 combinations maintainability is either classified as very high, high, medium, low or very low. A survey is taken from n experts including project managers, software developers, research scholars and maintainability experts to finalize the set of rules are found.

4.5 Metric Values

To find the value of factors we need metrics. For this purpose we have chosen CK metrics. The factor complexity is related with WMC, class is related with RFC, coupling is related with CBO, inheritance is related with DIT and number of children is related with NOC. Value of these metrics is found using analyst4j standalone tool [16]. We have taken out these values from object-oriented software developed in java [17]. Now the obtained metric values are given as input and the crisp value of maintainability is obtained from the proposed model by using MATLAB. The following rule viewer shows the obtained value of maintainability as 29.5.

4.6 Value of Maintainability

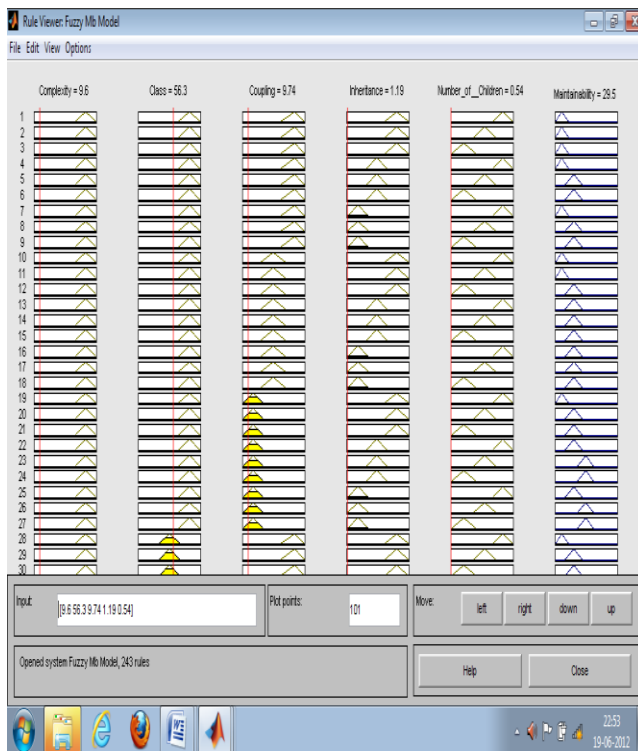


Figure 9. Value of maintainability obtained using MATLAB

4.7 Validation of Proposed Model

The proposed model is validated using standard AHP (Analytic Hierarchy Process) technique given by Saaty [18]. For this technique we first took a survey from the experts of related field, Survey included the factors that affect maintainability keeping in mind the CK metrics. For this we form a square matrix as shown below. Here factors are complexity (Comp), class (Cl), coupling (Coup), inheritance (Inhe) and Number of children (NOC).

	Comp	Cl	Coup	Inhe	NOC	Eigen vector (w)
Comp	1	1/3	1/3	1/7	1	0.055
Cl	3	1	3	1/7	3	0.165
Coup	3	1/3	1	1/9	2	0.093
Inhe	7	7	9	1	7	0.628
NOC	1	1/3	1/2	1/7	1	0.059
Total						1.00

Table 1. Factor values Using AHP Technique

$$\text{Consistency Index (CI)} = (\lambda_{\max} - n) / (n - 1) \quad [n=5]$$

$$= (5.313 - 5) / 4 = 0.078$$

$$\text{Consistency Ratio (CR)} = 0.078 / 1.12 = 0.069 < 0.1$$

for $n=5$ index of consistency=1.12 [18].

Hence, judgments are acceptably consistent.

The metrics values of the project [17] obtained through the analyst4j is multiplied by their corresponding weight values obtained by AHP to get the maintainability of the project. According to this method we got maintainability of project as 25.65. The relation between CK metrics and maintainability is already shown in [14].

Similarly for another project taken from [17] we obtained maintainability as 68.32 from proposed fuzzy model and maintainability as 66.73 from AHP method. Which indicated that proposed model is able to evaluate maintainability of object-oriented software system and can be used by application developers.

5.CONCLUSION

This paper proposes a fuzzy model to the maintainability of object-oriented software system. The inputs for the proposed model are complexity, class, coupling, inheritance and number of children on which maintainability depends. These inputs were determined based on study and using extensive survey. Rule base were generated by expert's knowledge, with 243 rules for evaluating object-oriented software system. The proposed model evaluated the maintainability of two object-oriented software systems. The results are validated by the AHP technique. The results by both the methods are almost same. So, it validates the proposed model. This model will help maintainability practitioners, software developers and researchers to select the best maintainable object-oriented software system when various alternatives are presented before them. In future the model will be more refined by taking consideration of other object-oriented metrics and more number of projects.

6.REFERENCES

- [1] Ragab, S. R., Ammar, H. H., 2010 "Object-Oriented design metrics and tools: a survey", Proc. Of Informatics and Systems (INFOS), pp. 1 – 7
- [2] Glass, R., 2002 Facts and Fallacies of Software Engineering, Addison-Wesley Professional.
- [3] Pressman, R. S., 2005 Software Engineering - A Practitioner's Approach, 7th ed., McGraw Hill.
- [4] Lucia, A. De, Pompella, E., Stefanucci, S., 2005 "Assessing effort estimation models for corrective maintenance through empirical studies," Information and Software Technology, vol. 47, no. 1, pp. 3-15.
- [5] Roger, P. S., 2005 Software Engineering, A Practitioner's Approach, McGraw Hill International Edition.
- [6] Jing, C., Xuyan, L., 2009 "Software Maintainability Metrics Based on the Index System and Fuzzy Method", The 1st International Conference on Information Science and Engineering (ICISE2009), pp. 5117-5120.
- [7] Saaty, T. L., 1990 "How to make a decision: The Analytic Hierarchy Process", European Journal of Operational Research, vol. 48, pp. 9-26.
- [8] Kafura, D., Reddy, R., 1987 "The use of Software Complexity Metrics in Software Maintenance", IEEE Trans. Software Engineering, vol. SE-13, no. 3, pp. 335-343.
- [9] Rizvi, S. W. A., Khan, R. A., 2010 "Maintainability Estimation Model for Object-Oriented Software in Design

- Phase (MEMOOD)", Journal of Computing, vol. 2, issue 4.
- [10] Chidamber, S. R., Kemerer, C. F., 1994 "A Metrics Suite for Object Oriented Design," IEEE Transactions on Software Engineering, vol. 20, no. 6, pp. 476-493.
- [11] Kaur, A., Singh, S., Kahlon, Dr. K. S., Dr. Parvinder, Sandhu, S., 2010 "Empirical Analysis of CK & MOOD Metric Suit", International Journal of Innovation, Management and Technology, vol. 1, no. 5
- [12] Dubey, S. K., Rana, A., 2010 "A Comprehensive Assessment of Object-Oriented Software Systems Using Metrics Approach", International Journal on Computer Science and Engineering, vol. 02, no. 08, pp. 2726-2730.
- [13] Türk, T., 2009 "The Effect of Software Design Patterns On Object-Oriented Software Quality And Maintainability", M. Tech Thesis, Dept. EE. Eng., Univ. Middle East Technical University.
- [14] Dubey, S. K., Rana, A., 2011 "Assessment of Maintainability Metrics for Object-Oriented Software System", ACM SIGSOFT SEN, vol. 36, No. 5.
- [15] Dubey, S. K., Rana, A. and Mridu, 2012 "Analytical Comparison of usability measurement methods" IJCA, volume 39 number 15, pp. 11-18.
- [16] www.codeswat.com/cswat/index.php? last accessed on 23rd May, 2012.
- [17] www.codeproject.com/KB/java/ last accessed on last accessed on 23rd May, 2012.
- [18] Saaty, T. L., 1988 "Multi criteria decision making: the Analytic Hierarchy process", RWS publications, Pittsburgh, PA.