Software Development Effort Estimation Using Soft Computing

Sandeep Kad and Vinay Chopra

***Abstract*—Software development effort estimation is a daunting task that is being carried out by software developers as not much of the information about the software which is to be developed is available during the early stages of development. The information that is to be gathered for various attributes of software needs to be subjective which otherwise leads to imprecision and uncertainty. Inaccurate estimation of the software effort and schedule leads to financial loses and also delays in project deadline. In this paper, we present the use of soft computing technique to build a suitable model which improves the process of effort estimation. To do so, various parameters of Constructive Cost Model (COCOMO) II are fuzzified that leads to reliable and accurate estimates of effort. The results show that the value of Magnitude of Relative Error (MRE) obtained by applying fuzzy logic is quite lower than MRE obtained from algorithmic model. By analyzing the results further it is observed that Gaussian Membership Function (gaussmf) performs better than Triangular Membership Function (trimf) and Trapezoidal Membership Function (trapmf) as the transition from one interval to another is quite smoother. Here varying number of COCOMO II inputs are fuzzified with these membership functions. The validation of the experiment is carried on COCOMO public dataset.**

***Index Terms*—Software cost estimation, COCOMO, soft computing, fuzzy logic.**

I. INTRODUCTION

Software development effort estimation is a vital aspect that deals with planning, prediction of amount of time and cost that will be incurred in developing of software project. Controlling the expenses of software development effectively is of utmost importance in today's competitive world [1]. The need for reliable and accurate software development cost predictions in software engineering is a challenging job as it accounts for considerable financial and strategic planning [2]. Despite considerable research and practical experience it is still a formidable challenge to understand and predict what happens in a large software projects. In 1995, Standish Group surveyed over 8,000 software projects for the purpose of budget analysis. It was found that 90% of these projects exceeded its initially computed budget.

Moreover, 50% of the completed projects lack the original requirements [3]. From these statistics, it can be seen how prevalent the estimation problem is. Improving the accuracy of the cost estimation models leads to effective control of time and budget during software development. In order to make accurate estimates and avoid large errors, several cost estimation models have been proposed. Among those techniques, COCOMO is the most commonly used because of its simplicity for estimating the effort in person-month for a project at different stages.

II. SOFTWARE EFFORT ESTIMATION MODELS

Software effort estimation models helps in estimating the amount of effort that needs to be put in to develop the software. However, the process estimation is uncertain in nature as it largely depends upon some attributes that are quite unclear during the early stages of development, but it needs to be carried out as huge investments are involved in building the software [4]. Software effort estimation models divided into two main categories: algorithmic models and non-algorithmic models.

Algorithmic models are based on the statistical analysis of historical data (past projects), e.g. Software Life Cycle Management (SLIM) [5] and COCOMO [6] and Albrecht's Function Point. These models rely upon accurate estimate of size of software in terms of line of code (LOC), number of user screen, interfaces, complexity, etc., at a time when uncertainty is mostly present in the software [7]. Now considering the current technological advancements these algorithmic models are unable to provide a suitable solution. Though these models may be good enough to handle a particular environment but they are not flexible enough to adapt new environment.

The limitations of algorithmic models have led to the exploration of non algorithmic models which are based upon soft computing techniques. Non-algorithmic techniques are based on new approaches such as, Parkinson, Expert Judgment, Price-to-Win and machine learning approaches. The soft computing techniques include methodologies like artificial neural networks, fuzzy logic and evolutionary computations. Due to their inherent nature these techniques are used to handle imprecision and uncertainty [13].

Fuzzy logic with its offerings of a powerful linguistic representation can represent imprecision in inputs and outputs, while providing a more expert knowledge based approach to model building. The first realization of the fuzziness of several aspects of COCOMO was carried out by Fei and Liu. They observed that an accurate estimate of delivered source instruction (KDSI) cannot be made before

Manuscript received July 29, 2012; revised September 19, 2012.

Sandeep Kad is with the Department of Information Technology, Amritsar College of Engineering and Technology, Amritsar, Punjab, India (e-mail: kadsandeep@yahoo.com).

Vinay Chopra is with the Department of Computer Science and Engineering,DAV Institute of Engineering and Technology, Jalandhar, Punjab, India (e-mail: vinaychopra222@yahoo.co.in).

10.7763/IJMLC.2012.V2.186

starting a project, and it is unreasonable to assign a determinate number for it [8]. Jack Ryder investigated the application of fuzzy modeling techniques to two of the most widely used models for effort prediction; COCOMO and the Function-Points models respectively [7]. Idri, Abran and Kjiri applied fuzzy logic to the cost drivers of intermediate COCOMO model [9]. Musilek et al. presented the application of fuzzy logic to represent the mode and size as input to COCOMO model. They presented a two-stage implementation called simple F-COCOMO model and augmented F-COCOMO model. Ahmed et al. fuzzified the two parts of COCOMO model i.e., nominal effort estimation and the adjustment factor. They proposed a fuzzy logic framework for effort prediction by integrating the fuzzified nominal effort and the fuzzified effort multipliers of the intermediate COCOMO model [10]. Hodgkinson and Garratt represented that estimation by expert judgment was better than all regression based models. A marriage between neural networks and fuzzy logic, is named Nero-fuzzy, was introduced into cost estimation. Nero-fuzzy systems can take the linguistic attributes of a fuzzy system and combine them with the learning and modeling attributes of a neural network to produce transparent, adaptive systems [2].

Thus it can be summarized from the previous research that all soft computing based techniques lack in one aspect or the other and still there is lot of uncertainty in deciding that what soft computing based prediction technique should be applied to which prediction problem. In this paper a fuzzy logic based COCOMO II model is proposed to overcome the problem of imprecision and uncertainty.

III. COCOMO Framework and Fuzzylogic

COCOMO 81 model is a regression based model derived by collecting data from large number of software projects [6]. It is considered to be one of most cited, best known and most plausible effort and cost prediction model [11]. Though it was one of the stable models of its time but it had number of drawbacks. It does not cope up with the current development environment like RAD and 4GL etc., thereafter COCOMO II was published that overcomes most of the drawbacks of COCOMO 81. COCOMO II comprises of three models [12]:

Application Composition Model - It is used during early stages of development and is suitable with GUI builder tools. It makes use of object points.

Early Design Model - It is used when not much information about the project is available and only rough estimates are needed. It uses few cost drivers and is based upon KSLOC and unadjusted function points.

Post Architecture Model - It is used when top level design of project is complete and detailed information about project is available. It makes use of all 17 cost drivers and 5 scale factors. It is given by:

$$\text{Effort} = A * [\text{Size}]^B * \prod_{i=1}^{17} \text{Effort Multiplier}_i \quad (1)$$

$$(\text{ where } B = 1.01 + 0.01 * \sum_{j=1}^{5} \text{Scale Factor}_j)$$

'A' is multiplicative constant and Size is the size of project measured in KSLOC/Function Points/Object Points.

Fuzzy Logic is a mathematical tool for dealing with uncertainty and imprecision. It is a theory of unsharp boundaries and is used to solve problems that are too complex to be understood qualitatively [13]. It consists of four main components:

Fuzzifier- It converts the crisp input into a fuzzy set. Membership Functions are used to graphically describe a situation.

Fuzzy Rule Base- It uses if-then rules.

Fuzzy Inference Engine- A collection of if -then rules stored in fuzzy rule base is known as inference engine. It performs two operations i.e. aggregation and composition.

Defuzzification- It is the process that refers to the translation of fuzzy output into crisp output.

IV. Proposed Research Methodology

It is important to stress that uncertainty at the input level of COCOMO model results in uncertainty at output [14]. COCOMO II comprises of size, cost drivers and scale factors input and effort as output which is measured in person months (PM).The problem with software effort estimation is that it largely depends upon single values of size, cost drivers and scale factors. The size of the project is estimated based upon previously completed projects that are somewhat similar with the current project. Also cost drivers and scale factors need to have through assessment rather than assigning a fixed numeric value. To overcome this situation it would be better to represent these inputs in the form of fuzzy sets, in which interval values are used that can be represented using variety of membership functions like triangular, trapezoidal or gaussian. The fuzzy based COCOMO II model is shown in Fig. 1.

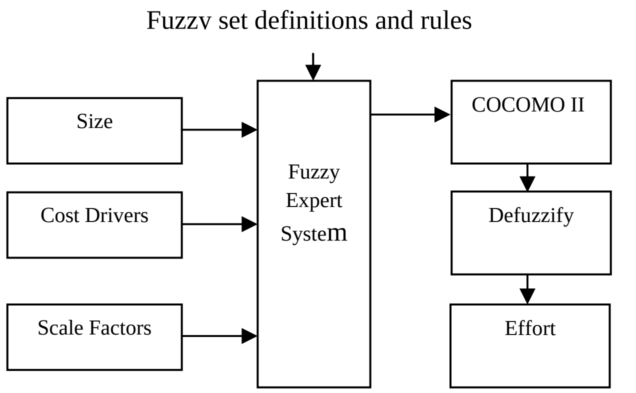

Fig. 1. Fuzzy COCOMO model

All the input variables in COCOMO II are changed to fuzzy variables using fuzzy sets for each linguistic value such as very low, low, nominal, high, very high etc. as applicable to each cost driver and scale factor. Rules are developed as cost driver in the antecedent part and corresponding effort multiplier in the consequent part. Similarly scale factors are also fuzzified. The case of programmer capability (pcap) cost driver is discussed as sample. Fuzzification of programmer capability is based upon COCOMO II 2000 Calibrated Post-Architecture model values (in Tables I and II) are shown in Fig. 2 and Fig. 3 as sample.

TABLE I: PCAP COST DRIVER RANGE DEFINED IN TERMS OF PERCENTILES

Very Low	Low	Nominal	High	Very High
15th	35	55	75	90

TABLE II: PCAP EFFORT MULTIPLIER RANGE DEFINED

Very Low	Low	Nominal	High	Very High
1.34	1.15	1.00	0.88	0.76

The proposed fuzzy based software effort estimation model rules contain linguistic variables related to the project. The rule base for fuzzy inference system (FIS) make use of connectives 'and/or' for COCOMO input variables to form number of rules. For fig. 2 and fig. 3 following rules are formed:

If PCAP is very low then EFFORT increases significantly
If PCAP is low then EFFORT is increased
If PCAP is nominal then EFFORT is unchanged
If PCAP is high then EFFORT is decreased
If PCAP is very high then EFFORT decreases significantly

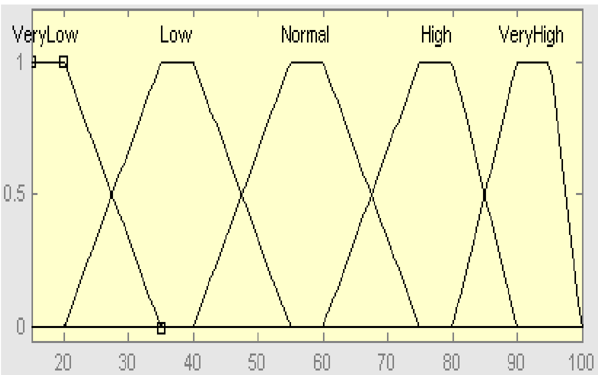

Fig. 2. Antecedent of PCAP cost driver using trapezoidal membership function

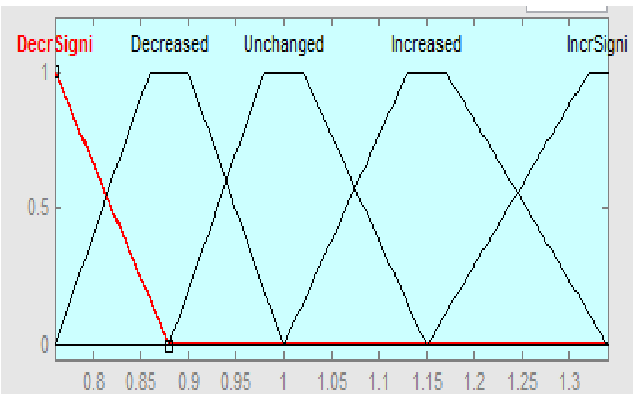

Fig. 3. Consequent of PCAP cost driver using trapezoidal membership function

V. EXPERIMENTAL RESULTS

To evaluate the proposed model a subset of dataset is taken from COCOMO dataset that includes 63 historical projects. Software development effort obtained using COCOMO and efforts obtained by using fuzzy logic on various membership functions like trimf, trapmf and gaussmf are calculated. It is observed that the effort obtained after applying fuzzy logic was closer to actual effort as compared to COCOMO. The parameter used for evaluation of proposed model is MRE and is given by:

$$\mathrm{MRE} = \frac{|\text{Actual Effort} - \text{Predicted Effort}|}{\text{Actual Effort}} \times 100 \tag{2}$$

MRE is calculated for each dataset that is applied to COCOMO and also for the proposed model comprising triangular membership function, trapezoidal membership function and gaussian membership function. It is observed that MRE obtained for the proposed model is quite less as compared to COCOMO. Fig. 4 shows the graphical representation of comparison of MRE.

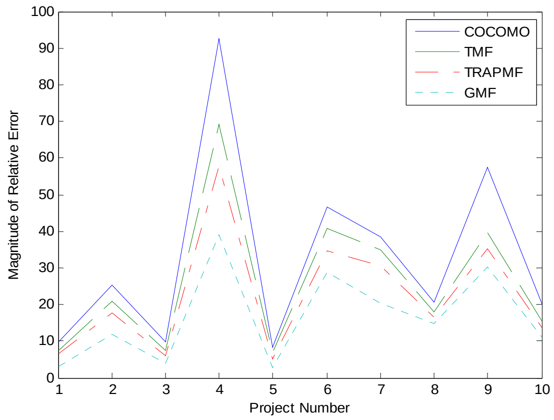

Fig. 4. Comparison of MRE between COCOMO model and proposed fuzzy model using COCOMO dataset.

Also the mean magnitude of relative error (MMRE) is calculated. The values of MMRE using COCOMO is 32.87, whereas 26.03,22.37 and 16.57 are using triangular membership function, trapezoidal membership function and gaussian membership function respectively. It shows that the proposed model has lesser MMRE than COCOMO. In the proposed model gaussian membership function is performing better than triangular membership function and trapezoidal membership.

VI. CONCLUSION AND FUTURE SCOPE

The study reveals that the proposed fuzzy logic based COCOMO II model overcomes the uncertainty in the inputs that is present in the traditional COCOMO and thus improves the accuracy of software effort estimation. By determining more suitable fuzzy rule sets and by deploying technologies like type-2 fuzzy uncertainty can be handled more closely and thus more accurate software effort estimation is possible.

REFERENCES

[1] S. G. MacDonell and A. R. Gray, "A comparison of techniques for software development effort prediction," *International Conference on Neural Information Processing and Intelligent Control Systems, New Zealand*, pp. 869-872, 1997.
[2] A. C. Hodgkinson and P. W. Garratt, "A neurofuzzy cost estimator,"Proceedings of Third International Conference on Software Engineering and Applications, pp. 401-406, 1999.
[3] G. Standish, "The chaos report," *The Standish Group*, 1995.

[4] I. Somerville, *Software Engineering,* 6th ed., Addison–Wesley Publishers Limited, 2001.
[5] L. H. Putnam, "A general empirical solution to the macro software sizing and estimating problem," *IEEE transactions on Software Engineering,* vol. 2, pp. 345-361.
[6] B. W. Boehm, *Software Engineering Economics,* Englewoods Cliffs, NJ, Prentice-Hall, 1981.
[7] J. Ryder, "Fuzzy modeling of software effort prediction," *IEEE* Information *Technology Conference,* pp. 53-56, 1998.
[8] Z. Fei and X. Liu., "f-COCOMO: fuzzy constructive cost model in software engineering," *IEEE International Conference on Fuzzy Systems,* pp. 331-337, 1992.
[9] A. Idri, A. Abrian, and L. Kjiri, "COCOMO Cost Model using Fuzzy Logic," *International Conference on Fuzzy Theory & Technology Atlantic, New Jersey,* 2000.
[10] P. Musilek, W. Pedrycz, G. Succi, and M. Reformat," Software cost estimation with fuzzy models," *Applied Computing Review,* vol. 2, pp. 24-29, 2000
[11] C. L. Martin, J. L.Pasquier, Y. M. Cornelio, and G. T. Augustin, "Software development effort estimation using fuzzy logic: A case study," *IEEE Proceedings of the Sixth Mexican International Conference on Computer Science (ENC),* pp. 113-120, 2005.
[12] B. Boehm, B. Clark, E. Horwitz, R. Madachy, C. Abts, S. Chulani, A. W. Brown, and B. Steece, "COCOMO II model definition manual," *University of South California Center for Software Engineering*, 2000.
[13] M. W. Nisar, J. W. Yong and M. Elahi, "Software development effort estimation using fuzzy logic - A survey," *IEEE International Conference on Fuzzy Systems and Knowledge Discovery,* vol. 1, pp. 421-427, 2008.
[14] M. Jorgenson and D. I. K. Sjoberg, "The impact of customer expectation on software development effort estimates," *International Journal of Project Management,* vol. 22, pp. 317-325
[15] H. K. Verma, and V. Sharma, "Handling imprecision in inputs using fuzzy logic to predict effort in software development," *IEEE International Advance Computing Conference (IACC),* pp. 436-442, 2010.
[16] I. Attarzadeh, and S. H. Ow, "A novel soft computing model to increase the accuracy of software development cost estimation," *IEEE International Conference on Computer and Automation Engineering (ICCAE),* vol. 3, pp. 603-607, 2010.
[17] M. O. Saliu, M.A. Ahmed, and J. AlGhamdi, "Towards adaptive soft computing based software effort prediction," *IEEE Annual Meeting of Fuzzy Information,* vol. 1, pp. 16-21, 2004.
[18] J. M. Mendel, "Fuzzy logic system for engineering: A tutorial," *IEEE Transactions on Neural Networks,* vol. 11, pp. 748-768, 2000.
[19] K. Strike, K. E. Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Transactions on Software Engineering,* vol. 27, 2001.