COCOMO COST MODEL USING FUZZY LOGIC

ALI IDRI AND ALAIN ABRAN

Research Lab. in Software Engineering Management,
Department of Computer Science,
UQAM, C.P. Box. 8888, Centre-ville Postal Station
Montréal, Québec, Canada H3C 3P8
E-mail: idri@ensias.ac.ma
alain.abran@uqam.ca

Laïla KJIRI

Laboratoire Génie Informatique
Equipe Génie logiciel
ENSIAS, BP. 713, Agdal,
Rabat, Morocco
E-mail: kjiri@ensias.ac.ma

Abstract

When the COCOMO (Constructive Cost Model) was published at the beginning of the eighties, fuzzy logic was not grounded on solid theoretical foundations. This was not been achieved until Zadeh and others did so in the nineties. Thus, it is not surprising that some of the concepts defined or used in COCOMO are somewhat incompatible with the fuzzy logic. In our work, we investigate the issue of the compatibility of COCOMO with the fuzzy logic.

In software metrics, specifically in software cost estimation, many factors (linguistic variables in fuzzy logic) such as the experience of programmers and the complexity of modules are measured on an ordinal scale composed of qualifications such as 'very low' and 'low' (linguistic values in fuzzy logic). In our work, we study the COCOMO'81 model, specifically its intermediate version. Our work is still applicable to the COCOMOII.

Keywords: *Software Engineering, Software Cost estimation , COCOMO, Fuzzy Logic.*

INTRODUCTION

Estimating the work-effort and the schedule required to develop a software system is one of the most critical activities in managing software projects. In order to make accurate estimations and avoid gross estimation errors, several techniques are used within an organization. The most popular techniques (at least in the literature) use the algorithmic models such as COCOMO [3, 4, 5], IBM-FSD [15], PUTNAM-SLIM[14], SPQR[10] and Function Points Analysis [1, 12]. This paper looks at the COCOMO'81 model, particulary its intermediate version. The reasons why we choose the intermediate, rather than the simple or the detailed version, are as follows:

- It is the most widely used version
- The accuracy of an estimation (measured in terms of Relative Error) obtained with the intermediate version is substantially greater than that obtained with the simple version, and very similar to that obtained with the detailed version (see Table 1).
- The COCOMO software project database (see [3], pp.496) allows only the validation of the simple or the intermediate version
- The simple version does not take into account enough cost drivers (only two) to validate our approach.

	COCOMO Version		
	Simple	Intermediate	Detailed
Relative Error (RE≤20) (%)	25	68	70

Table 1. A comparison of the accuracy of the 3 versions of COCOMO

Since it was first published, COCOMO'81 has been the subject of important studies aimed at to calibrating, enhancing or reformulating the initial version [2, 4, 5, 7, 8]. Beside of these important issues, we address the first application of the fuzzy logic to the COCOMO'81 cost model. Because many of the concepts handled in the COCOMO'81 model do not take into account this, we propose solutions to make it obey the progressively emerging fuzzy set theory.

This paper is organized as follows. In the first section, we briefly outline the principle of the intermediate COCOMO'81 model. In the second section, we explain why and how the fuzzy logic could be applied to the COCOMO'81 cost model. The third section describes the validation and analysis of the results obtained from our application. A conclusion and an overview of future work conclude this paper.

1. Intermediate COCOMO

The intermediate COCOMO'81 model is an extension of the basic version which allows estimations to be generated using only software size and the project mode[3]. In addition to these two attributes, the intermediate version takes into account 15 other cost drivers which are generally related to the software environment. The work-effort estimation formula is then:

> where MM_{est} is the Man-Months required for development, and *SIZE* is the code size measured in KDSI,
>
> A and B are constants which are specific to each project mode (organic, semi-detached or embedded),
>
> $$MM_{est} = A \times SIZE^{B} \prod_{i=1}^{i=15} C_{ij} \qquad j = 1..k_i \qquad (1)$$
>
> *and* C_{ij} the effort multiplier associated with the j^{th} selected rating for the i^{th} cost driver attribute (Table 2).

2. Why and How must Fuzzy Logic be applied to the COCOMO model?

Each cost driver in the intermediate COCOMO'81 model is measured using a rating scale of six linguistic values: 'very low', 'low', 'nominal', 'high', 'very high', 'extra-high'. The assignment of linguistic values to the cost drivers uses conventional quantization where the values are intervals (see [3], pp. 119). For example, the DATA cost driver is measured by the following ratio:

$$\frac{D}{P} = \frac{Database\ size\ in\ bytes\ or\ characters}{\mathrm{Pr}ogram\ size\ in\ DSI}$$

Then, a linguistic value is assigned to the DATA according to the following table:

Low	Nominal	**High**	Very High
D/P<10	10≤D/P<100	100≤D/P<100	D/P≥1000

Table 3. DATA cost driver ratings.

So, no project can occupy more than one class. If D/P is equal to 9.99, then the DATA of the project is rated 'low'. If D/P is equal to 10.01, then the DATA is rated 'nominal'. This is a serious problem in that it can lead to a great difference between the estimations of two analogous projects. Let us suppose that this case occurs for all cost drivers of two projects P_1 and P_2:

- For the cost drivers whose effort multipliers are increasing, P_1 comes right before the lower limit of 'high' and P_2 comes right after this limit,
- and for these whose effort multipliers are decreasing, P_1 comes right after the lower limit of 'nominal', P_2 comes right before this limit.

			Rating			
Attribute	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1/66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	

Table 2. The 75 effort multipliers used in intermediate COCOMO

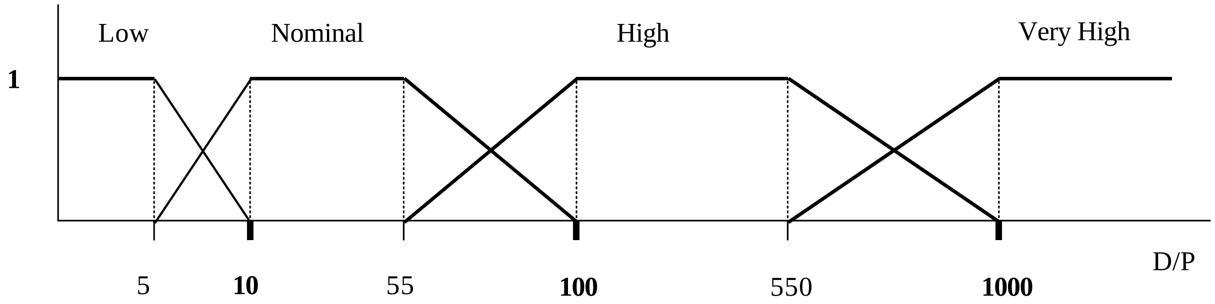

Figure 1. Membership function of fuzzy sets defined for the DATA cost driver

If P_1 and P_2 have the same nominal effort (MM_{nom} is the first part of formula (1): $A \times size^B$), say 15MM, then the adjusted effort for P_1 is unchanged (15MM), but for P_2 it is 52MM!

This problem is caused by the use of conventional quantization where the values are intervals. So, the transition from one interval to a contiguous interval is abrupt rather than gradual. In our work, we use fuzzy sets rather classical intervals to represent the linguistic values ('very low', 'low', etc). The advantages of this over quantization are as follows:

- they are more general
- they mimic the way in which humans interpret linguistic values
- The transition from one linguistic value to a contiguous linguistic value is gradual rather than abrupt.

For example, in the case of the DATA cost driver, we define a fuzzy set for each linguistic value with a trapezoid-shaped membership function μ (Figure 1). For the other cost drivers of the intermediate COCOMO'81 model, we proceed in the same way as for DATA. Among its 15 cost drivers, the four factors RELY, CPLX, MODP and TOOL are not studied because these relative descriptions are insufficient. Thus, we have defined the fuzzy sets corresponding to the various associated linguistic values for each cost driver.

3. Evaluation

In the next step, we evaluate the intermediate COCOMO'81 model using formula (1) and effort multipliers obtained from fuzzy sets (F_C_{ij}) rather than from the classical C_{ij}. F_C_{ij} is calculated from the classical C_{ij} and the membership functions μ defined for the various fuzzy sets associated with the

$$F\_C_{ij} = F(m_{A_1}^{V_i}(P)........m_{A_j}^{V_i}(P), C_{i1}...........C_{ij})$$

cost drivers:

$$F\_C_{ij} = \sum_{j=1}^{k_i} m_{A_j}^{V_i}(P) \times C_{ij} \qquad \textbf{(2)}$$

For simplicity, we take F as a linear function:
where the $m_{A_j}^{V_i}$ is the membership function of the fuzzy set A_j associated with the cost driver V_i.

This evaluation consists in comparing the accuracy of the estimated with actual values[3]. Exactly, as in COCOMO, the following five quantities are used for the appreciation of the degree of accuracy:

- the percentage of projects that have a Relative

  $$RE_i = RE(P_i) = \left| \frac{MM_{est}^{P_i} - MM_{act}^{P_i}}{MM_{act}^{P_i}} \right| \times \mathbf{100}$$

  Error under 20 (pred(20)):

  where MM^{Pi}_{est} is the estimated work-effort for project P_i, and MM^{Pi}_{act} is the actual work-effort for project P
- Min RE_i
- Max RE_i
- Mean RE_i
- Standard deviation of RE_i

Because the original COCOMO'81 database contains only the effort multipliers, our evaluation will be made on three artificial datasets deduced from the original COCOMO'81 database. These artificial datasets contain the real values that are necessary to determine the $m_{A_j}^{V_i}(P)$ of the formula (2). For example, the DATA cost driver for the fifth project in the COCOMO'81 database is declared 'low'. Thus, the randomly generated singleton value for the fifth project in each dataset is between 0 and 10. The following table shows the results obtained for the accuracy of the 'fuzzy' intermediate COCOMO'81 model compared to the original intermediate COCOMO'81.

	'fuzzy'/classical intermediate COCOMO'81			
	Database #1	Database #2	Database #3	COCOMO'81
Pred(20) (%)	62.14 / 68	46.86 / 68	41.27 / 68	68 / 68
Min RE_i (%)	0.11 / 0.02	0.40 / 0.02	0.06 / 0.02	0.02 / 0.02
Max RE_i (%)	88.60 / 83.58	3233.03 / 83.58	88.03 / 83.58	83.58 / 83.58
Mean RE_i (%)	22.50 / 18.52	78.45 / 18.52	30.80 / 18.52	18.52 / 18.52
Standard deviation RE_i	19.69 / 16.97	404.40 / 16.97	22.95 / 16.97	16.97 /16.97

Table 4. Results of the evaluation

The results obtained for the three datasets, when applying 'fuzzy' intermediate COCOMO'81, are different. But if we apply intermediate COCOMO'81, the results of evaluation for the three datasets are the same ones as those of the COCOMO'81 database. This implies that the 'fuzzy' intermediate COCOMO'81 tolerates imprecision in its inputs (cost drivers) and consequently it generates more gradual outputs (cost). This graduation is then less sensitive to the changes in the inputs, contrary to intermediate COCOMO'81 that generate the same or significantly different outputs when the inputs are different. This accuracy is then very sensitive to the changes in the inputs.

4. Conclusion

In this paper, we have proposed the use of fuzzy sets rather than classical intervals in the COCOMO'81 model. For each cost driver and its associated linguistic values, we have defined the corresponding fuzzy sets. These fuzzy sets are represented by trapezoid-shaped membership functions. The accuracy is certainly affected by this. There are other possible representations which can be tried, such as the Bell, Gaussian and triangular membership functions. To define a convenient representation, we must study the significance of the various linguistic values in the environment from which the COCOMO'81 database was assembled.

Many other aspects of COCOMO remain incompatible with the fuzzy set theory; unless

- the three modes of a project can be defined by fuzzy sets,
- the KDSI, the size of code source, can be measured by fuzzy numbers.

Our main goal following this investigation is to build a software cost estimation model supporting Soft Computing as defined by Zadeh [17].

BIBLIOGRAPHY

[1] Albrecht, A.J., Gaffney, J.E., 'Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation', *IEEE Transactions on Software Engineering*, Vol. SE-9, No. 6, Nov, 1983, pp. 639-647

[2] Amrane, B., Slimani, Y., 'ALCOMO Model: A statistical reformulation of the COCOMO model', *Information science and Technology*, Apr, 1993.

[**3]** Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, 1981.

[4] Boehm, B.W., Royce, W.W. 'Le COCOMO Ada', *Génie logiciel & Systèmes experts*, 1989.

[5] Boehm, B.W., et *al*., 'Cost Models for Future Software Life Cycle Processes: COCOMO 2.0', *Annals of Software Engineering on Soft. Process and Product Measurement*, Amsterdam, 1995.

[6] Fenton, N., Pfleeger, S.L., *Software Metrics: A Rigorous and Practical Approach*, International Computer Thomson Press, 1997.

[7] Gulezian, R., 'Reformulating and Calibrating COCOMO', *Journal of Systems Software*, Vol 16, 1991, pp.235-242

[8] Idri, A., Griech, B., El Iraki, A., 'Towards an Adaptation of the COCOMO Cost Model to the Software Measurement Theory', In *Proc. ESEC/FSE*, Sep., Zurich, 1997.

[9] Jager, R., 'Fuzzy Logic in Control', Ph.D. Thesis, Technic University Delft, ISBN 90-90008318-9, Dutsh, 1995.

[10] Jones, C., *Programming Productivity*, McGraw-Hill, New York, 1986.

[11] Marwane, R., Mili, A., 'Building Tailor-made Software Cost Model: Intermediate TUCOMO', *Inf. Soft. Techn*., Vol. 33(3), Apr., 1991, pp.232-238

[12] Matson, J.E., Barrett, B.E., Mellichamp, J.M., 'Software Development Cost Estimation Using Function Points', *IEEE Trans. on Soft. Eng*., Vol. 20, No. 4, Apr., 1994, pp. 275-287

[13] Miyazaki, Y., Mori, K., 'COCOMO evaluation and tailoring', In *Proc. Eighth Int. Conf. Soft. Eng*., London, UK, Aug, 1985, pp.292-299

[14] Putnam, L.H., 'A General Empirical Solution to the Macro Software Sizing and Estimation Problem', *IEEE Trans. on Soft. Eng*., Vol. SE-4, no. 4, July, 1978.

[15] Walston, C.E., Felix, A.P, 'A Method of Programming Measurement and Estimation', *IBM Systems Journal*, Vol 16, no. 1, 1977.

[16] Zadeh, L.A., 'Fuzzy Set', Information and Control, Vol. 8, 1965, pp. 338-353.

[17] Zadeh, L. A., 'Fuzzy Logic, Neural Networks, and Soft Computing', *Comm. ACM*, Vol. 37, no. 3, March, 1994, pp.77-84

[18] Zimmerman, H.J., *Fuzzy Set Theory and it Applications*, 2d ed, Kluwer-Nijhoff, 1990.