Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Fuzzification of Complexity Matrix to Calculate Function Points

Sonia Juneja*
Department of CSE/IT , HCE, DCRUST
India.

Preeti Rana
M.Tech (CSE)student , HCE, DCRUST
India.

***Abstract*— Function Point Analysis is a widely used estimation method for measuring software size which can be further used to estimate the effort and cost required to design the software. During the traditional point counting process, each function is classified according to its relative functional complexity. The paper uses the concepts and properties from fuzzy set theory to extend FPA to FFPA (Fuzzy Function Point Analysis). Fuzzy theory is capable of building a formal quantitative structure which can emulate the imprecision of human knowledge. With the function points generated by FFPA, derived values such as costs and terms of development can be more precisely determined.**

Keywords— Fuzzification, Membership Functions, Complexity Matrix, Translation Table, Newton Interpolation formula

I. INTRODUCTION

The method of Function Point Analysis was developed by Allan Albrecht to help measure the size of a computerized Business Information System. Such sizes are needed as a component of the measurement of productivity in system development and maintenance activities and as a component of estimating the effort needed for such activities .Software productivity can be defined as a measure of the rate at which individual engineers involved in software development produce software and associated documentation. Not quality-oriented although quality assurance is a factor in productivity assessment. Essentially we want to measure useful functionality produced per time unit. There are two productivity measures which are commonly used

a) Size related measures based on some output from the software process. This may be lines of delivered source code, object code instructions, etc.
b) Function-related measures based on an estimate of the functionality of the delivered software. Function-points are the best known of this type of measure.

Allan Albrecht has described a method known as "Function Point Analysis" for determining the relative size of a system based on these two factors [1].

a) *information processing size*, that is some measure of the information processed and provided by the system.
b) A *technical complexity factor*, that is a factor which takes into account the size of the various technical and other factors involved in developing and implementing the information processing requirements.

The method has gained widespread acceptance in the business information systems community, for system size assessment as a component of productivity measurement, when system development or maintenance and enhancement activities are completed. Where historic productivity data are available, the method can also be used as an aid in estimating man-hours, from the point where a functional requirements specification is reasonably complete. Estimating the size of software has always been something more complicated, because of the complexity of the software itself, the lack of historical experience, the lack of estimate tools and some human error conducing that the size of software estimation is often far from the actual situation. Many studies have already proposed extending FPA, such as FFP (FFP, 1997), mainly seeking to obtain higher precision in the point assessment of systems of greater algorithmic complexity, such as real-time systems, embedded systems, and communications systems, among others. Other extensions are specific to enhancement projects. Maya (Maya, 1996), for instance, suggests altering the structure of FPA for a purer evaluation of small functional increments. This work proposes the utilization of concepts and properties taken from fuzzy set theory in order to extend FPA into FFPA (Fuzzy Function Point Analysis). This solves the problem of uncontinuity caused by complexity grade division with IFPUG Function Point Analysis. This work is organized in the following manner: section 2 describes the IFPUG Function point Analysis ; section 3 focuses on fuzzy set theory, the underlying basis of the proposed model; section 4 succinctly describes fuzzified process of calculating Function Points , proposing membership functions, extending complexity matrix and generating revised Translation Table Section 5 describes the Defuzzification Process and section 6 shows experimental study and section 7 offers the fundamental conclusions of the work.

II. TRADITIONAL FPA

IFPUG Function Point Analysis [2] proposed by the International Function Point User Group (IFPUG) though inheriting and developing Albrecht's Function Point Analysis [3]. is a software size estimation method which is based on the system

function. FPA begins with the decomposition of a project or application into its data and transactional functions. The data functions represent the functionality provided to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions describe the functionality provided to the user in relation to the processing of data by the application. IFPUG defines five function types[4] that can be counted from early life cycle documents and then stipulates a linear weighting scheme for each of the five function types depending on their complexity. The five function types are classified as follows:

1. An external input is any elementary process of an application that processes data or control information that enters from outside the boundary of the application.
2. An external output is an elementary process of an application that generates data or control information that exits the boundary of the application.
3. An internal logical file is a user identifiable group of logically related data or control information maintained through an elementary process of the application.
4. An external interface file is a user identifiable group of logically related data or control information references by the application but maintained within the boundary of a different application.
5. An external inquiry is an elementary process of the application that is made of an input output combination that results in data retrieval. The input side is the control information that defined the request for data. The output side contains no derived data.

Data function includes Internal Logical Files (ILF) and External Interface Files (EIF). Transactional function includes External Inputs (EI), External Outputs (EO) and External Query (EQ). The data functions' relative functional complexity is based on the number of Data Element Types (DETs) and the number of Record Element Types (RETs). The transactional functions are classified according to the number of File Types Referenced (FTRs) and the number of DETs.

After identifying the functions, each needs to be classified as either low, average ,or high in accordance with its relative functional complexity, which is expressed by a certain value in points, depending on the function.

Upon completing a point assessment to all functions, the application is then adjusted in accordance with the general characteristics of the system, which evaluates the general functionality of the application.

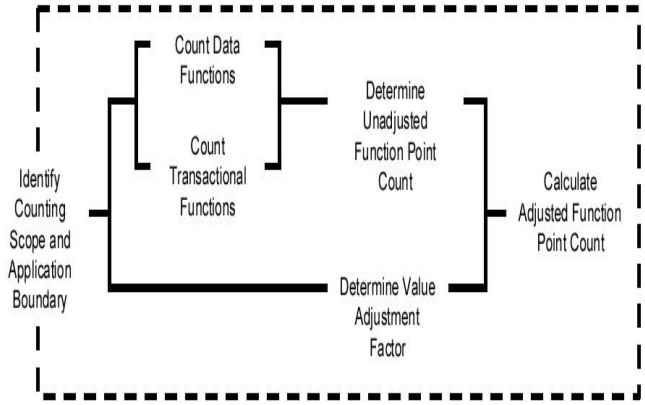

Fig 1 Process of calculation of Traditional Function Point

III Fuzzy Logic & Fuzzy Set Theory

Since its foundation by Zadeh[5,6] in 1965, Fuzzy Logic (FL) has been the subject of important research investigations. During the early nineties, fuzzy logic was firmly grounded in terms of its theoretical foundations and application in the various fields in which it was being used, such as robotics, medicine, and image processing.

A fuzzy set is a set with a graded membership function, μ, in the real interval [0, 1]. This definition extends the one of a classical set where the membership function is in the couple $\{0, 1\}$. Fuzzy sets can be effectively used to represent linguistic values such as *low*, *young*, and *complex*, in the following two ways [6]:

- Fuzzy sets that model the gradual nature of properties, i.e., the higher the membership μ that a given property x has in a fuzzy set A, the more it is true that x is A. In this case, the fuzzy set is used to model the vagueness of the linguistic value represented by the fuzzy set A.
- Fuzzy sets that represent incomplete states of knowledge. In this case, the fuzzy set is a possibility distribution of the variable X, and consequently, is used to model the associated uncertainty. When considering that it is only known that x is A, and x is not precisely known, the fuzzy set A can be considered as a possibility distribution, i.e., the higher the membership x' has in A, the higher the possibility that x = x'.

The representation by a fuzzy set is more advantageous than the other two approaches, because:

- It is more general,
- It mimics the way in which the human-mind interprets linguistic values, and

- The transition from one linguistic value to a contiguous linguistic value is gradual rather than abrupt.

IV Fuzzy Function Point Analysis

As described in Traditional Process of calculation of Function Point, the complexity of a function (data or transactional) is characterized by one of the following linguistic terms (complexity grades): "low", "average", or "high". The attribution of these terms is determined by the functional complexity matrix of each function component. For example, Table I shows the complexity matrix of ILF, whose terms "low", "average", and "high" complexity correspond to 7, 10, and 15 function points, respectively as shown in Table IV. Similarly Table II represents the complexity Matrix of EI and Table III represents the Complexity matrix of EO & EQ.

Table IV known as Translational table presents the value of function points for the terms of "low", "average" and "high" to each FPA function.

	1-4 DET	5-15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTRs	Low	Average	High
>=3 FTRs	Average	High	High

Table I: Complexity matrix of ILF/ELF

	1-19 DET	20-50 DET	51 or more DET
1 RET	Low	Low	Average
2-5 RET	Low	Average	High
>=6 RET	Average	High	High

Table II: Complexity matrix of EI

	1-5 DET	6-19 DET	20 or more DET
0 to 1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
>= 4 FTRs	Average	High	High

Table III : Complexity matrix of EO/EQ

Term	ILF	EIF	EI	EO	EQ
Low	7	5	3	4	3
Average	10	7	4	5	4
High	15	10	6	7	6

Table IV : Translation Table

The weakness in the IFPUG FPA is that it can not accurately make the translation from element type number and complexity grade into function points. For example: Assuming there is an ILF with 1 RET and 19 DETs (function f1), it has a "low" complexity, which is translated to 7 function points. Under the same standard, another ILF with 1 RET and 50 DETs (f2) is also classified as "low" complexity with 7 function points. However, an ILF with 1 RET and 51 DETs (f3) is classified as "average" complexity with 10 function points. Even if there is a disparity of 31 DETs between f1 and f2, they have identical number of points. Nevertheless, there is only one DET's disparity between f2 and f3, but they have different number of points. If this situation happens in the same project, the final resulting measurement will not correspond to a sufficiently accurate number of function points, that is, the number of function points will not represent the functional complexity of the system.

The solution[7] to this problem is the fuzzification of process of calculation of Function Points.The complete process consists of 4 stages:

Stage 1 Fuzzification of Complexity matrices

Stage 2Extending the complexity matrices for new linguistic terms

Stage 3 Determine the complexity grade of extended linguistic terms

Stage 4 Deffuzification Process of Fuzzy function point values.

IV.A Fuzzification of complexity matrices & extending them to new linguistic terms

The trapezoidal membership function are most widely used for representation of complexity grades. A trapezoid fuzzy number can be represented by *N* (*a*, *m*, *n*, *b*),whose membership function is expressed as follows:

$$\mu_N(x)= \begin{cases} 0, & \text{for } x<a \\ (x-a)/(m-a) & \text{for } x\epsilon[a,m] \\ 1 & \text{for } x\epsilon [m,n] \\ (b-x)/(b-n) & \text{for } x\epsilon[n,b] \\ 0, & \text{for } x<a \end{cases} \quad (1)$$

Fig. 2 shows the distribution situation of intermediate trapezoid membership function. The values *a* and *b* indicate the lower and upper limits of larger base of the trapezoid respectively, where $\mu_N(x)=0$. The values *m* and *n* indicate the lower and upper limits of smaller base of the trapezoid respectively, where $\mu_N(x)=1$. The values k indicate the midpoint of line segment *am*, where $\mu_N(x)=0.5$.

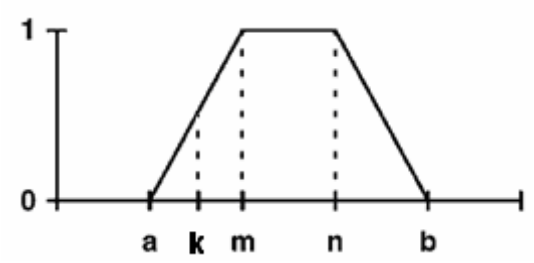

Fig 2: Trapezoidal membership functions

While fuzzification the two fuzzy sets intersect at pot where two fuzzy sets have membership grade = 0.5.

The problem with traditional FPA as specified in section 3 is solved by fuzzification process but it becomes more critical as the number of RETs and the number of FTRs increase in the data functions and in the transactional functions, respectively. Thus in addition to fuzzification, a new interval of high complexity was added to data and transactional functions that called for at most an interval of average complexity. For the same reason, a new interval of very high complexity was added for the remaining functions, applying the modifier very to the linguistic term high.

In the functional complexity matrix of all function components, each complexity grade T_i *low average high and very high(extended)* will generate a corresponding trapezoid fuzzy numbers *N* (*a*, *m*, *n*, *b*) . The value k_i is the lower limits of complexity grade *i* in the complexity matrix. The value n_i is the average value of m_i and m_{i+1}, and it must be an integer with rounding off. The values n_{i-1} and m_{i+1} are assigned to a_i and b_i ,respectively.

The different values of a, m, n and b for different fuzzy sets is calculated using following relations

a) $m_i + m_{i+1} = 2n_i$ (2)

b) $n_i + m_{i+1} = 2ki_{+1}$ (3)

Using equations (2) &(3) one can generate the fuzzy sets for the data and transactional functions as shown below

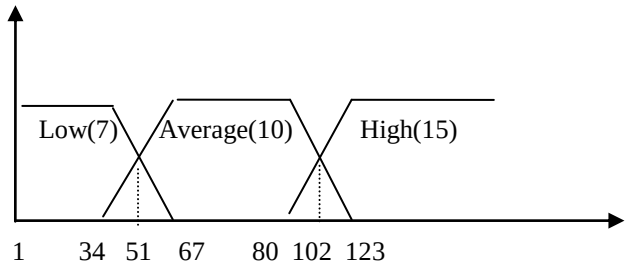

Fig 2:Fuzzy sets for ILF when RET = 1

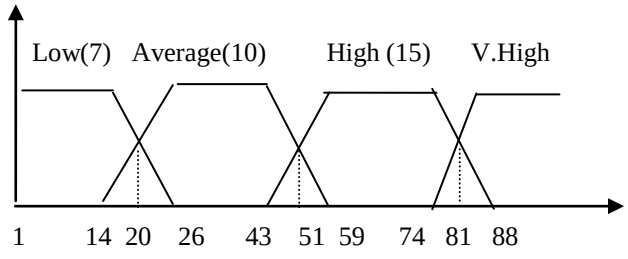

Fig 3:Fuzzy sets for ILF when RET = 2-5

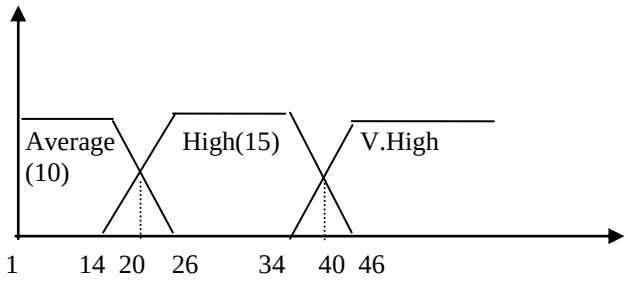

Fig 4:Fuzzy sets for ILF when RET = 6 or more

The membership functions for EIF are same as ILF only difference is there for complexity grades which is 5 for LOW, 7 for average and 10 for high as shown in Table no. IV

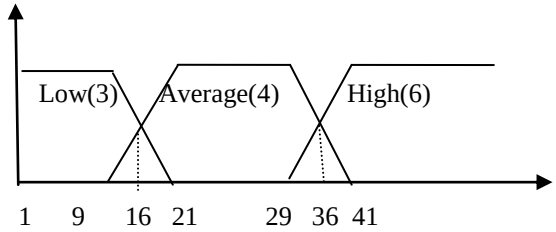

Fig 5:Fuzzy sets for EI when FTR =0 to 1

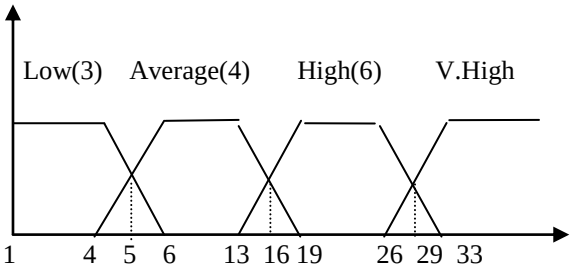

Fig 6:Fuzzy sets for EI when FTR = 2

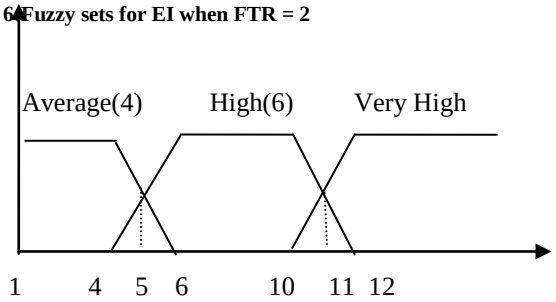

Fig7:Fuzzy sets for EI when FTR = 3 or more

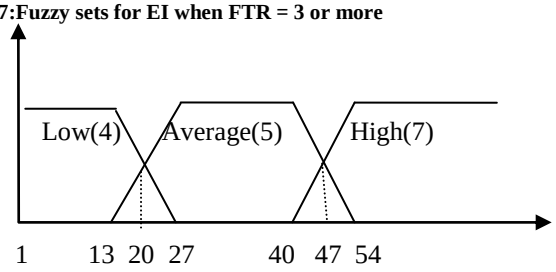

Fig 8:Fuzzy sets for EO when FTR=0 to1

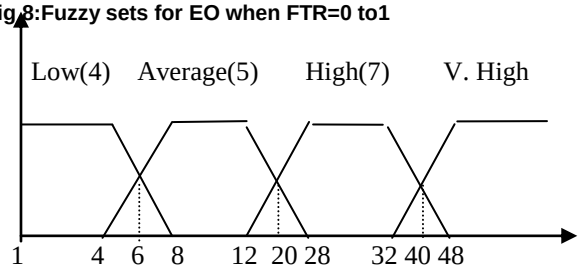

Fig 9:Fuzzy sets for EO when FTR = 2-3

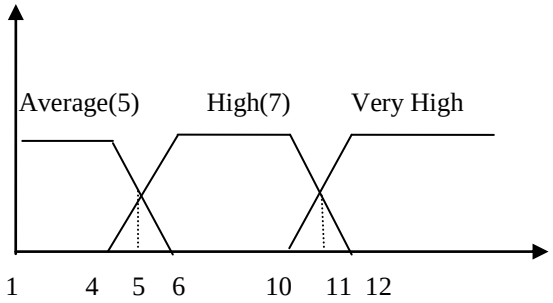

Fig10:Fuzzy sets for EO when FTR = 4 or more

The membership functions for EQ are same as EO, only difference is there for complexity grades which is 3 for LOW, 4 for average and 6 for high as shown in Table no. IV.

IV.B Determine the complexity grade of new linguistic terms

Once the membership functions are defined for all the data and transactional functions next step is to determine the complexity grade of new linguistic term very high added to the existing linguistic terms. This can be done by applying the Newton's divided difference formula which is a special case of binomial interpolation in which the abscissas of the points are equidistant.

Newton's Divided difference backward interpolation formula is given as

$Y=Y_0 +P\Delta Y_0+P(P+1)\ \Delta^2Y_0/2!\ P\Delta Y_0+P(P+1)\ (P+2)\Delta^3\ Y_0/3!+\ldots\ldots$

Where $x=x_0+ph$ and h is the difference between two adjacent abscissa. Applying the definitions above, the values obtained for the fuzzy numbers of very high complexity functions were estimated (Table V). Below, follow the calculations for the data functions ILF:

Complexity	**i**	$\mathbf{y_i}$	Δy_i	$\Delta^2 y_i$
Low	0	7	3	2
Average	1	10	5	-
High	2	15	-	-
Very High	3			

Table V: Table of the Progressive Differences –ILF

Using values $x_0=2, y_0=15, h=1$ we get

$p=(x-x_0)/h=1$. Thus

Y(very High)= 15+5+2=22.

Same formula can be applied to EIF, EO, EQ and EI to obtain the complexity grade of very high. The resultant revised Translation table is given below as Table VI

Term	**ILF**	**EIF**	**EI**	**EO**	**EQ**
Low	7	5	3	4	3
Average	10	7	4	5	4
High	15	10	6	7	6
Very High	22	14	9	10	9

Table VI: Revised Translational Table

V.C Defuzzification Of Fuzzy Function Point

According to the above method, we get the trapezoid membership function of fuzzy complication matrix. Because the concept of fuzzy membership function is introduced in fuzzy function point analysis, in order to obtain the function points with $\mu_n(x)<1_1$, the fuzzification eliminating process is executed as follows:

$p_d= \mu_n(x)*p_i +(1-\mu_n(x))*p_{i+1}$ *where p_d is defuzzified function point.*

VI Experimental Study

The data for experimental study is taken from [8] and compared with fuzzy system implemented with triangular membership functions [9] .The calculation of Unadjusted Fuzzy Function points for real life application is given in table below.

DET	RET/ FTR	Data/ Transactional Function	FP	FFP [9]Using Triangular membership functions	FFP Using Trapezoidal membership functions
60	RET= 3	EIF	10	8.5	10
75	RET= 3	EIF	10	11	10.18
22	FTR= 1	EO	5	4.3	4.7
10	FTR= 2	EO	5	4.71	5
2	FTR= 1	EQ	3	.80	3
7	FTR= 2	EQ	4	4.23	3.75
16	FTR= 1	EI	4	3.1	3.5
13	FTR= 2	EI	4	3.6	4

Section Vii Conclusions

This work extended standard FPA (Function Point Analysis) to FFPA (Fuzzy Function Point Analysis) , utilizing concepts and properties adopted from fuzzy set theory. By analyzing some of the results obtained through the use of FFPA, we note:

- Through the use of trapezoid-shaped fuzzy numbers for the linguistic terms low, average, and high, functions falling along the border areas of the intervals used receive values with a continuous graduation, without an abrupt change of those values;
- The creation of the linguistic term of very high complexity, pertaining to a parameterized interval through the value of k, which can be adjusted according to the characteristics of organization to better deal with larger systems;

The model proposed uses of a historical base of governmental systems data calculated in FPA to validate its results. This study will also be carried out using other historical bases within other application domains. It is important to point out that FFPA generates the same function points as traditional FPA for projects that have their functions with the complexity classified within non-extended intervals.

Acknowledgement

This is to acknowledge the support of all my friends, colleagues and my teachersfor their guidance. I am also very thankful to my family for their help and support.

References

[1] A. J. Albrecht and J. **E.** Gaffney, "Software function, source lines ***of***code and development effort prediction: A software science validation,"***IEEE Trans. Sofware Eng.,*** vol. **SE-9,** no. 6, pp. 639-647.Nov. 1983.

[2] IFPUG. Function Point Counting Practices Manual, Release 4.1[Z]. International Function Point Users Group,2003.

[3] A. J. Albrecht. "Software Function, Source Lines of Code and Development Efort Prediction: A Software ScienceValidation". IEEE Transactions on Software Engineering,1983(11), 1(6).

[4] Roger S. Pressman, "Software Engineering; A Practitioner Approach", Mc Graw-Hill InternationalEdition, Sixth Edition (2005).

[5] L.A. Zadeh, 2002, From Computing with numbers to computing with words-from manipulation ofmeasurements to manipulation of perceptions, Int. J. Appl. Math.Comut.Sci.,2002, Vol.12,No.3,307-324.

[6] L.A. Zadeh, 1965, Fuzzy Sets, Information and Control, 8, 338-353.

[7] CHEN,CHENG,FANG,OU,"Study of function points Analysis based on Fuzzy Intrepolation",Journal Of Computational Information Systems6:5(2010)1369-1375.

[8] Chuk Yau, Raymond H.L. Tsoi, "Assessing the Fuzziness of General System Characteristics in Estimating Software Size", IEEE, 189-193, 1994.

[9] Harish Mittal, Pradeep Bhatia ," A comparative study of conventional effort estimation and fuzzy estimation based on Triangular Fuzzy Numbers".IJCSS,Volume(1):Issue(4)Pg 36-47

