Object-Oriented Full Function Point Analysis: An Empirical Validation

[1,]Sheeba Praveen, [2,]Dr. Rizwan Beg
[1,]Dept. CSE,Integral University Lucknow ,U.P
[2,] Dept .CSE & IT,Integral university Lucknow,U.P.

Abstract

this research work focuses on validation work of my proposed work to determine the functional size of real time application at early stage. This paper will describe how to calculate the functional size of real time system using my proposed model that is Object Oriented Size Estimation Model for Real Time Application [1]. Here in this paper I am validating my proposed model with the help of Real Time System. I am taking Human Emotion detection which is real time software and applying OOFFPA on it. OOFFPA metrics of my proposed model will calculate the size in terms of Function Point of this HED software and after calculating the size I compare this size with the size which is calculated by other metrics. Comparisons will prove that OOFFP metrics of my proposed model is best for size measurement of real time system.

Keywords: OOFFPA, OPA, FPA, OO, TFPCP, FFP, OOSE, UML, OOFP, SC, AS, AG, GN/SP, MX, OOMC, OOTC, OOCC, OOUCG, OORCG, OOCECE, OOCECX, OOCICR, OOCICW, OOEI, OOEO, OOEQ, OOILF, OOEIF, OOFC.

1. INTRODUCTION

This research is focuses on the cost estimation of Real Time System and for that I choose “Human Emotion Detection (HED)” which is a Real time system. This case study will help to validate my proposed work. In this case study I am using OOFFP metrics and Object oriented development procedures to estimate the size and cost of HED real time software.The HED can be applied in robotics in AI. The Robot can detect the human emotion and act accordingly the situation. This code can be interfaced with the help of embedded system using some programming language.
The camera will act as a sensor which captures the image of human and analyses the image on the basis of his/her facial expression. It detect the emotion i.e. Happy, Sad, Strange, Normal, Surprise etc. means act accordingly the situation.
Scenarios of HED are following:

1. Take an image as an input
2. Then apply skin color segmentation on an image
3. Find the largest connected region
4. Check the probability to become a face of the largest connected region
5. Convert the RGB image into Binary image
6. To detect that the image is human face or not
7. Separate the Eye and Lips from the image
8. Convert the Eye and Lips into Binary image
9. Apply the Bezier Curve on Eyes and Lips
10. Detect the human Emotion From the image

2. OBJECT ORIENTED DEVELOPMENT LIFE CYCLE

A. Object Oriented Design

In this research I follow the object oriented development life cycle to estimate size and cost of project [].

1. Object modeling Techniques
2. Dynamic modeling techniques
3. Functional modeling techniques

All three object oriented design techniques and associated models (D) are used in the proposed Size estimation model. These models help to estimate the size of the project. So In this research I am applying Management Function count and Control function count on all three models of object oriented design.

4.2.1 Unified Modeling Language

The Unified Modeling Language (UML) defines a large number of different diagrams. They are divided into following three categories: Static structure diagrams, Behavior diagrams and Implementation diagrams. In order to calculate the function point from the above diagrams, we use the sequence diagrams and class diagrams. Because these diagrams includes the information about all functions and data manipulated in the system. All these UML diagram are explained in my proposed paper” Full Functional size measurement Model applied to UML-based Real Time Application” [1].

3. PROPOSED RULES FOR OOFFPA FUNCTION POINT

Aim is to calculate the Unadjusted Function Point. Here I am proposing the following five steps to apply OOFFPA to the requirements/design specifications (class diagrams and sequence diagrams) based on the UML.

Step1 Determine the Type of Function Point Count
Step2 Identify the Counting Boundary
Step3 A. Count Data Function Types
Step4 Count Transactional Function Types
Step5 Count new Control Data Function Type
Step6 Count new Control Transactional function Type
Step7 Calculate Unadjusted Function Point Count
Step8 Determine value adjustment factor
Step9 Calculate Adjusted Function point

B. **Count Data Function Types**

a. Object Oriented ILF Complexity and Contribution (OOILF):

The OOILF steps are as follows:
1. Rate the OOILF complexity.
2. Translate the complexity to unadjusted function points.
3. Calculate the OOILF contribution to the total unadjusted function point count.

1. Rate Ooilf Complexity:

Rate the complexity of the OOILF using the following complexity matrix.

RER/DET	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RETs	Low	Average	High
6or more RETs	Average	High	High

Table1. OOILF

2. Translate OOILFs :

The following table translates the external inputs' functional complexity to unadjusted function points. Low=7, Average=10, High =15.

3. Calculate Ooilf Contribution:

The following table shows the total contribution for the OOILF function type.

Type	ILF	SR	ILF OOFP	SR OOFP	Total OOFP
SC	20	7 (L)	20*7	140+60	200
AS	14	7 (L)	14*7	98+60	158
AG	14	7 (L)	14*7	98+60	158
GN	7	10 (A)	7*10	70+60	130
MX	7	10 (A)	7*10	70+60	130

Table2. OOILF

b. Object Oriented EIF Complexity and Contribution (OOEIF):

The OOEIF steps are as follows:
1. Rate the OOEIF complexity.
2. Translate the complexity to unadjusted function points.
3. Calculate the OOEIF contribution to the total unadjusted function point count.

1. Rate Ooeif Complexity:

Rate the complexity of the OOEIF using the following complexity matrix.

RET/DET	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RETs	Low	Average	High
6or more RETs	Average	High	High

Table3. OOEIF

2. Translate Ooilfs :

The following table translates the external inputs' functional complexity to unadjusted function points. Low=5, Average=7, High =10.

3. Calculate Ooeif Contribution:

The following table shows the total contribution for the OOEIF function type. As there are 12 concrete methods in the model, service requests contribute 12 * 5 = 60 OOFPs (SR OOFP). The Value 7 is rated as Low and it is weighted 4.

Type	EIF	SR	ILF OOFP	SR OOFP	Total OOFP
SC	8	5 (L)	8*5	40+60	100
AS	7	5 (L)	7*5	35+60	95
AG	7	5 (L)	7*5	35+60	95
GN	5	10 (A)	5*10	50+60	110
MX	4	10 (H)	4*10	40+60	100

Table4. OOEIFs

C. **Count Transactional Function Types**

a. Object Oriented External Inputs Complexity and Contribution (OOEI):

The OOEI steps are as follows:

1. Rate the OOEI complexity.
2. Translate the complexity to unadjusted function points.
3. Calculate the OOEI contribution to the total unadjusted function point count.

1. Rate Ooei Complexity:

Rate the complexity of the OOEI using the following complexity matrix.

RER/DET	1 to 5 DET	6 to 19 DET	20 or more DET
0 to1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

Table5. OOEIs Complexity Rate

2. Translate OOEIs :

The following table translates the external inputs' functional complexity to unadjusted function points. Low=3, Average=4, High =6. The following table shows the functional complexity for each OOEI.

OOEI	FTRs	DETs	Functional Complexity
Browse Image	1	4	L
Apply Skin colour	1	3	L
Largest connected Region	2	10	A
RGB Image	2	8	A
Binary Image	1	4	L
Image Lip	0	2	L
Image Eye	0	2	L
Image Eyebrow	0	2	L
Bezier Curve	1	3	H

Table6.OOEIs

3. Calculate OOEI Contribution:

The following table shows the total contribution for the OOEI function type.

Total no. of OEI Function Type	Functional Complexity	Total Complexity	Total function type
8	5(L)	5*3	15
	2(A)	2*4	8
	1(H)	1*6	6
	Total	+	29

Table7. Total OOEIs

b. Object Oriented External Output Complexity and Contribution (OOEO):

The OOEO steps are as follows:

1. Rate the OOEO complexity.

2. Translate the complexity to unadjusted function points.
3. Calculate the OOEO contribution to the total unadjusted function Point count.

1. Rate Ooeo Complexity:

Rate the complexity of the OOEO using the following complexity matrix.

RER/DET	1 to 5 DET	6 to 19 DET	20 or more DET
0 to1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

Table8. OOEOs Complexity Rate

2. Translate Ooeos :

The following table translates the external inputs' functional complexity to unadjusted function points. Low=4, Average=5, High =7.

The following table shows the functional complexity for each OOEO.

OOEO	FTRs	DETs	Functional Complexity
Valid Face	0	6	H
Connected	2	4	A
Happy emotion	1	3	L
Fear Emotion	1	5	L
Surprise Emotion	1	7	L
Sadness Emotion	1	4	L
Anger Emotion	1	6	L
Disgust Emotion	1	2	L

Table9.OOEOs

3. Calculate Ooeo Contribution:

The following table shows the total contribution for the OOEO function type.

Total no. of OOEO Function Type	Functional Complexity	Total Complexity	Total function type
6	4(L)	5*4	16
	2(A)	2*5	10
	0(H)	1*7	0
	Total	+	26

Table10. Total OOEIs

c. Object Oriented External Inquiries Complexity And Contribution (OOEQ)

The OOEQ steps are as follows:

1. Rate the OOEQ complexity.
2. Translate the complexity to unadjusted function points.
3. Calculate the OOEQ contribution to the total unadjusted function point count.

1. Rate Ooeq Complexity:

Rate the complexity of the OOEQ using the following complexity matrix.

RER/DET	0 to 5 DET	6 to 19 DET	20 or more DET
0 to1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

Table11. OOEQs Complexity Rate

2. Translate Ooeqs :

The following table translates the external inputs' functional complexity to unadjusted function points. Low=3, Average=4, High =6.

OOEQ	FTRs	DETs	Functional Complexity
Fear samples	1	2	L
Surprise samples	1	1	L
Sadness samples	1	4	L
Anger samples	1	6	L
Disgust samples	1	4	L
Happy samples	1	3	L

Table12.OOEQs

3. Calculate Ooeqs Contribution:

The following table shows the total contribution for the EI function type.

Total no. of OEI Function Type	Functional Complexity	Total Complexity	Total function type
5	5(L)	5*3	15
	0(A)	0*4	8
	0(H)	0*6	6
	Total	+	29

Table13. Total OOEQs Complexity Rate

The following table shows the total contribution for the OOEI, OOEO and OOEQ function type

Function Type	Total Function	Low	Average	High	Total function type
OOEI	9	6*3	2*4	1*6	32
OOEO	8	6*4	1*5	1*7	26
OOEQ	6	6*3	0*4	0*6	18
+					76

Table14.Total Function type

D. **Count new Control Data Function And Transactional Function Type**

a. Object Oriented Full Function Points (OOFFP) new Control and Transactional Function Count

OOFFP new function types	**Description**	**No. of sub-processes**
OOUCG	Data updated by the application	8
OORCG	Data not updated by the application	10
OOECE	Incoming external message	1
OOECX	Outgoing external message	1
OOICR	Referred attribute in an elementary action	7
OOICW	Update attribute in an elementary action	15
	Total	42

Table15.Total new control and Transactional Function count.

Low=2, Average=3, High =5(for NCDFC).
Low=3, Average=4, High =6(for NTFC)

Function Type	Total function	Low	Average	High	Total function type
OOUCG	8	6*2	2*3	0*5	18
OORCG	10	6*2	3*3	1*5	26
OOECE	1	1*3	0*4	0*6	3
OOECX	1	0*3	1*4	0*6	4
OOICR	7	3*3	3*4	1*6	27
OOICW	14	7*3	7*4	0*6	49
					127

Table16.Total New function Type

E. **Calculate UnAdjusted Function Point Count (UFP)**

The following table shows the contribution of the application functionality to the unadjusted function point count.

Function Type	Total functional Complexity
OODFC	1341
OOTFC	76
NCDFC & NTFC	42
Total	1459

Table17. Total UFP of OOFFP

F. **Procedures to Determine the VAF**

The following steps outline the procedures to determine the value adjustment factor (chapter3).

1. Evaluate each of the 14 general system characteristics on a scale from zero to five to determine the degree of influence (DI).

2. Add the degrees of influence for all 14 general system characteristics to produce the total degree of influence (TDI).

3. Insert the TDI into the following equation to produce the value adjustment factor.
VAF = (TDI * 0.01) + 0.65
For example, the following value adjustment factor is calculated if there are three degrees of influence for each of the 14 GSC descriptions
(3*14).
VAF = (42 * 0.01) + 0.65
VAF = 1.07

G. **Calculate adjusted Function point count(AFP)**

Using the complexity and contribution counts for this example, the development project count is shown below. The value adjustment factor (VAF) for this example is 1.07.

AFP = UFP * VAF
AFP = 1459 * 1.07
AFP = 1561.13 or 1561

H. **Assumptions & Results**

Past date indicate that one FP translate into 60 times of code (if an OOP language is to be used) LOCs = 60 * 1561 = 93660 (approximately) Past project have found an average of 3 errors per function point during analysis and design reviews and 4 errors per function point during unit and integration testing. Thus, possible number of errors in analysis and design reviews should be 3*1561 i.e. 4683. At the time of testing
Possible number of errors should be 4*1561 i.e. 6244. Thus total possible number of errors should be 10927.

4. Verification of Results

After implementation it was found that lines of code are 94181, which is more than calculated LOCs (on the basis of FPs in analysis phase) by a value of 1561.
Errors found at the time of analysis and design reviews are 4683 and errors found at the time of testing are 6244. Thus total errors found are 10927 which is more than calculated by a value of 1561.

5.1 COMPARATIVE STUDY OF HED ON DIFFERENT METRICS

Metrics	Count number	Error-proness
LOC	94181	10927
FP	1850	7259
OOFFP	1561	5463.5

Table17. Metrics comparison

5. CONCLUSION

This above table shows that the OOFFP metrics is best for size measurement of real time system. By using this size we can calculate the Productivity and quality of real time system. So OOFFPA helps to increase the performance MIS as well as real time software.

References

[1] Full Functional size measurement Model applied to UML-based Real Time Application is accepted in International Conference on Recent Trends in Control Communication and Computer Technology(RTCCT-2012), Paper ID: RTCCCT-29SEP12-030

URL http://www.interscience.ac.in/Chennai/RTCCCT/index.html

[2] Abran, A., Desharnais, J.-M., Maya, M., St-Pierre, D., & Bourque, P. (1998). Design of Functional Size Measurement for Real-Time Software. Montréal, Université du Québec à Montréal [www document]..

URL http://www.lrgl.uqam.ca/publications/pdf/407.pdf

[3] Bohem, R. (1997). Function Point FAQ. Metuchen,USA, Software Composition Technologies, Inc URL http://ourworld.compuserve.com/homepage/softcom/

[4] Desharnais, J.-M. & Morris, P. (1996). Validation Process in Software Engineering: an Example with Function Points. In Forum on Software Engineering Standards (SES'96), Montreal [www document]..

URL http://www.lrgl.uqam.ca/publications/pdf/104.pdf

[5] Introduction to Function Point Analysis (1998). GIFPA, Issue 2, summer [www document]. URL http://www.gifpa.co.uk/news/News2Web.pdf

[6] International Standards Organisation (ISO) (1991). Information Technology ± Software Product Evaluation ± Quality Characteristics and Guidelines for their Use (ISO/IEC IS 9126). Geneve: ISO/IEC.

[7] Software Metrics - why bother?.(1998). GIFPA, Issue 1, spring

URL http://www.gifpa.co.uk/news/Issue1_ed2.pdf

[8] St-Pierre, D., Maya, M., Abran, A., Desharnais, J.-M. & Oligny, S. (1997a). Full Function Points: Counting Practices Manual. Montréal, Université du Québec à Montréal [www document].

URL http://www.lrgl.uqam.ca/publications/pdf/267.pdf

[9] St-Pierre, D., Maya, M., Abran, A., Desharnais, J.-M. & Oligny, S. (1997b). Measuring the functional size of real-time software. Montréal, Université du Québec à Montréal [www document].

URL http://www.lrgl.uqam.ca/publications/pdf/330.pdf

[10] Fenton, N. E. & Pfleeger, S. L. (1996). Software Metrics - A Rigorous & Practical Approach. London: International Thomson Computer Press.

[11] Functional Size Measurement for Real-Time Software. Montréal, Université du Québec à J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[12] An Empirical Validation of Software Quality Metric Suites on Open Source Software for Fault-Proneness Prediction in Object Oriented System I. S. Jacobs and C. P. Bean, "Fine particles, thin films and