An Approach for OO Software Size Estimation Using Predictive Object Point Metrics

Shubha Jain, Vijay Yadav
Department of Computer Science and Engineering
Kanpur Institute of Technology,
Kanpur, India.
shubhj@rediffmail.com, vijayyadavuiet@gmail.com

Prof. Raghuraj Singh
Computer Science and Engineering Department
Harcourt Butler Technological Institute,
Kanpur, India.
raghurajsingh@rediffmail.com

Abstract **– For estimating software, system size is the main parameter of the system development effort. It affects substantially on accurate estimation of effort of development. The Predictive Object Point (POPs) input gives an estimate of the size of the software for which the estimation is required. POPs are a metric suitable for estimating the size of object oriented software, based on the behaviors that each class is delivering to the system along with top level inputs describing the structure of a system. However there is no real mapping of Source Lines of Code (SLOC) to POPs exists. This paper is an attempt to map the Predictive Object Point Metrics with software size which may help in further prediction of effort. This may also help in estimation of cost as well as schedule measurement of an OO system. The proposed method of mapping between POP and software size has been empirically investigated. KLOC has been estimated in terms of EKLOC through POP count using the linear regression equation. The results are presented here to show that how POP Count may be mapped to corresponding software size (KLOC) of an object oriented system.**

Keywords— Estimated KLOC, KLOC, Object-Oriented Measurement, Predictive Object Point, SLOC, Software Metrics, Software Sizing, Software Estimation Technique

NOMENCLATURE

KLOC- Kilo Lines of Code
POP - Predictive Object Point
SLOC - Source Lines of Code
EKLOC- Estimated KLOC using Predictive Object Point
FP – Function Point

I. INTRODUCTION

Software sizing is a process in software engineering that is used to estimate the size of software. It plays a major role in software engineering and is one of the key factors that potentially affect the cost and time of the software project [3, 4, 5, 6, 7]. There are number of sizing metric being introduced such as Source Lines of Codes (SLOC), Function Points (FP) [8], Predictive Object Points (POP) [2] etc among them POP is considered to be the better indicator of size of object oriented than FP [8, 9] and thus can be used for the evaluation of effort which will in turn may be used to find cost as well as schedule of a software project. POPs are a metric suitable for estimating the size of object oriented software, based on the behaviors that each class is delivering to the system along with top level inputs describing the structure of a system. However there is no real mapping of SLOC to POPs exists.

II. PROPOSED MODEL RELATING POP METRIC WITH SOFTWARE SIZE

As already stated that POP is better size indicator for object oriented system thus the practitioners may use this metric to estimate the effort required to complete the project by using the COCOMO II model [3, 10]. The model uses Kilo Source lines of code for effort estimation therefore a technique is required to convert POP to Estimated KLOC, defined here as EKLOC. This has been proposed to be done here using simple linear regression analysis [1].

The Linear regression equation is of the form given in equation (1).

$$Y = b_0 + b_1 * X \qquad (1)$$

Where b_0 is the intercept and b_1 is the slope. These are also known as regression constants. Here X is an independent variable and Y is dependent on X. The slope and intercept can be calculated by the formula given in equation (2) and (3).

$$b_0 = (\text{avg } Y) + b_1 (\text{avg } X) \qquad (2)$$

$$b_1 = \frac{(\sum X * Y) - \left[\frac{(\sum X) * (\sum Y)}{n}\right]}{(\sum X^2) - \left[\frac{(\sum X)^2}{n}\right]} \qquad (3)$$

Here the estimation of KLOC is desired in terms of POP therefore POP is considered as independent variable (i.e. X) and the KLOC is considered as the dependent variable (i.e. Y). So the equation (1) will turn to equation (4).

$$KLOC = b_0 + b_1 * POP \qquad (4)$$

Then b_0 and b_1 are calculated as in equation (5) and (6).

$$b_0 = (\text{avg } KLOC) + b_1 (\text{avg } POP) \qquad (5)$$

$$b_1 = \frac{(\sum POP * KLOC) - \left[\frac{(\sum POP) * (\sum KLOC)}{n}\right]}{\left(\sum POP^2\right) - \left[\frac{(\sum POP)^2}{n}\right]} \qquad (6)$$

Where n is the number of projects and the KLOC obtained from the equation is termed as Estimated KLOC i.e. EKLOC. Mapping is an operation that associates each element of a given set (the domain) with one or more elements of a second set (the range). The two entities i.e. POP and EKLOC are being mapped as per equation 4 in Fig.1.

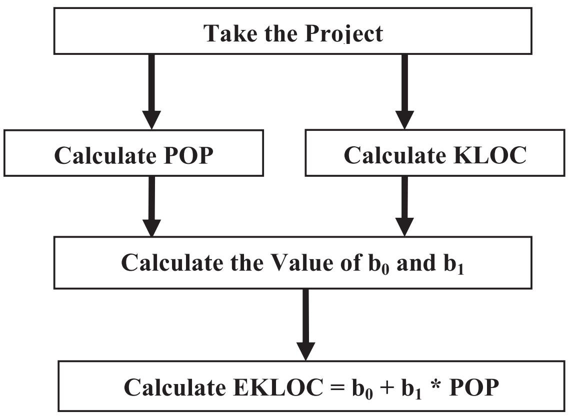

Figure 1: Steps required for the proposed mapping process

III. Methodology Used

A several versions of the project called “The Lightweight Java Game Library (LWJGL)” are selected for the mapping process [11]. This project is a solution aimed directly at professional and amateur Java programmers alike to enable commercial quality games to be written in Java. LWJGL provides developers access to high performance cross platform libraries such as OpenGL (Open Graphics Library), OpenCL (Open Computing Language) and OpenAL (Open Audio Library) allowing for state of the art 3D games and 3D sound.

Additionally LWJGL provides access to controllers such as Gamepads, Steering wheel and Joysticks. The KLOC and POP values of the chosen nine various versions of this project are obtained through an APA tool [9].

Project No.	LWJGL(Version)	KLOC	POP
1	lwjgl_0.9	14.119	2419.3199
2	lwjgl_0.92	18.262	3176.9945
3	lwjgl_0.93	19.366	3283.3734
4	lwjgl_0.94	20.624	4074.6891
5	lwjgl _0.96-2	23.58	5349.5434
6	lwjgl_0.98-1	24.64	5345.8681
7	lwjgl_1.0	30.097	6045.3585
8	lwjgl_1.0 rc-1	30.542	5867.1778
9	lwjgl_1.1.1	33.277	6153.8996

Table 1: KLOC and POP values of 9 versions of Project (LWJGL) measured through APA Tool

Table 1 shows the values of POP and KLOC measured from the tool. The values of b_0 and b_1 is then calculated by using these values of POP and KLOC for 9 different versions and then these values along with POP are substituted into the equation (4) to evaluate finally the Estimated KLOC i.e. EKLOC.

The Table 1 values have been plotted as shown in Fig.2 in order to get the values of slope (b_1) and intercept (b_0).

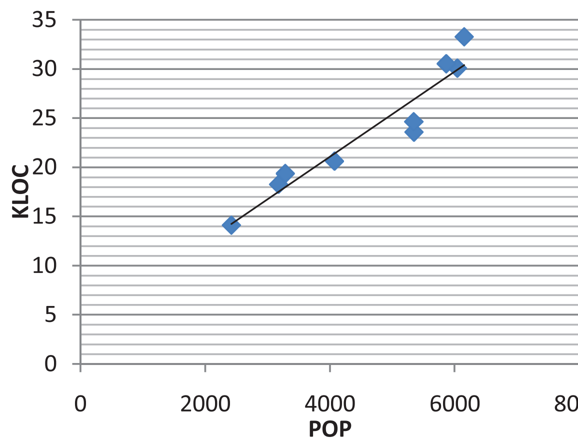

Figure 2: A Graph between POP and KLOC of 9 versions of project (LWJGL)

To strengthen the above relation obtained between EKLOC and POP, more versions of the projects should be analyzed and incorporated in the calculation of b_0 and b_1. The results should improve with increase in number of versions incorporation in regression analysis. In order to validate this, the more versions have been incorporated in calculation step by step.

Hence one more version of the project is analyzed and resulting values of POP and KLOC have been incorporated as shown in Table 3 in order to obtain the refined estimated value of KLOC.

Projects	Actual KLOC	POP	Estimated KLOC (EKLOC)
1	14.119	2419.3199	14.0096637
2	18.262	3176.9945	17.4549853
3	19.366	3283.3734	17.9387147
4	20.624	4074.6891	21.5370102
5	23.58	5349.5434	27.3340675
6	24.64	5345.8681	27.317355
7	30.097	6045.3585	30.4980996
8	30.542	5867.1778	29.6878708
9	33.277	6153.8996	30.9916611
10	**33.31**	**6166.1952**	**31.0475721**
b_0 = 3.008457 b_1 = 0.004547 EKLOC = 3.008457 + 0.004547 * POP			

Table 3: KLOC, POP and Estimated KLOC values of the 10 versions of project (LWJGL)

The respective graph for refined values of b_0 and b_1 are shown in Fig. 3.

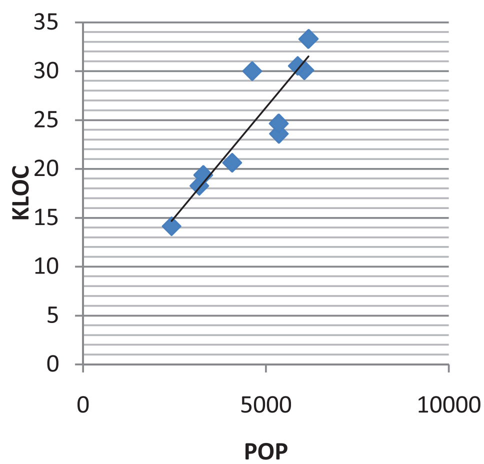

Figure 3: A Graph between POP and KLOC of 10 versions of project (LWJGL)

In the same way one more version of the project is analyzed and resulting values of POP and KLOC have been incorporated in Table 4 in order to obtain the refined estimated value of KLOC.

The b_0 and b_1 values are refined further.

Projects	Actual KLOC	POP	Estimated KLOC (EKLOC)
1	14.119	2419.3199	13.9983754
2	18.262	3176.9945	17.487573
3	19.366	3283.3734	17.9774627
4	20.624	4074.6891	21.6215823
5	23.58	5349.5434	27.4924648
6	24.64	5345.8681	27.4755395
7	30.097	6045.3585	30.6967907
8	30.542	5867.1778	29.8762436
9	33.277	6153.8996	31.1966377
10	33.31	6166.1952	31.2532606
$b_0 = 2.857069$ $b_1 = 0.004605$ EKLOC = 2.857069 + 0.004605 * POP			

Table 4: KLOC, POP and Estimated KLOC values of the 11 versions of project (LWJGL)

Continuous incorporation of more versions of the project and hence further refinement in values of b_0 and b_1 take Estimated KLOC to more close to the actual KLOC value of that version. The results may be seen through a graph plotted between Estimated KLOC and POP as shown in Fig.4.

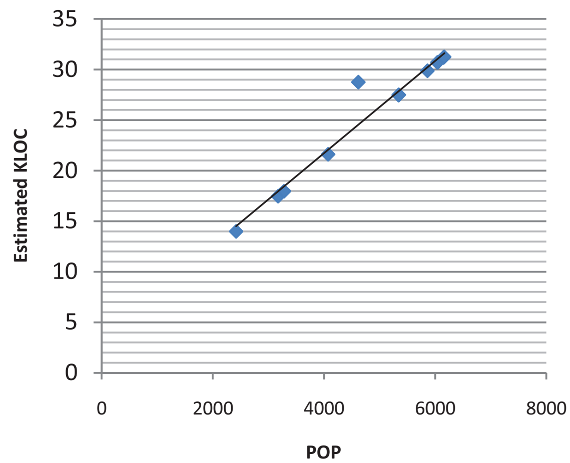

Figure 4: A Graph between EKLOC and POP for 11 versions of chosen project (LWJGL)

IV. ANALYSIS AND RESULTS

In above study, it may be observed that each time when a new project is evaluated the values of b_0 and b_1 are updated which in turn updates the regression equation EKLOC = $b_0 + b_1$ * POP. The POP value of the projects is then substituted in the equation to find the Estimated KLOC.

It is found that the values of Estimated KLOC also changes each time and comes nearer to the actual KLOC values of the projects which signifies that the accuracy of the equation increases every time as the number of projects increases.

Therefore from the above analysis it is clear that a mapping can be done between KLOC and POP by using simple linear regression equation and to improve the accuracy more and more number of the projects should be added.

V. CONCLUSIONS

From the results presented in this study it is found that Predictive Object Point Metrics can be used to estimate the size of the Object Oriented Software Systems. Here the nature of the relationships between KLOC and POP in object-oriented (OO) systems has been understood. This study depicts the relationship between POP and Kilo Lines of Code (KLOC) to get estimation of size of software through POP. After sizing software, Effort prediction may be done using EKLOC and consequently schedule and cost may also be estimated for software projects.

Another future study prospect would be to find the direct relation between POP and Software size.

REFERENCES

[1] Online Statistics Education: An Interactive Multimedia Course of Study, Developed by Rice University (Lead Developer), University of Houston Clear Lake, and Tufts University URL:

http://onlinestatbook.com /2/regression/intro.html
[2] Arlene F. Minkiewicz, "Measuring Object Oriented Software with Predictive Object Points" PRICE Systems, L.L.C. 609-866-6576, 1997.
[3] B. W. Boehm, Estimating Software Costs, Prentice Hall, N.J., 1981,
[4] T. C. Jones, Estimating Software Costs, McGraw-Hill, New York, 1998.
[5] J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, vol. 9,no. 6, pp. 639–648, 1983.
[6] F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", Communications of the ACM, vol.30, no. 5, pp. 416-429, 1987.
[7] Kitchenham, "Using Function Points for Software Cost Estimation- Some Empirical Results", 10thAnnual Conference of Software Metrics and Quality Assurance in Industry, Amsterdam, the Netherlands, 1993.
[8] T. R. Judge, A. Williams ," OO Estimation – an Investigation Of The Predictive Object Points (POP) Sizing Metric in an Industrial Setting", Parallax Solutions Ltd, Coventry, UK.
[9] Shubha Jain, Vijay Yadav and Prof. Raghuraj Singh, (2013). "OO Estimation Through Automation of Predictive Objective Points Sizing Metric", International Journal Of Computer Engineering and Technology (IJCET) Volume 4, Issue 3, May- June (2013), pp. 410-418.
[10] Center for Software Engineering, (2000) "COCOMO II Model Definition Manual," Computer Science Department, University of Southern California, Los Angeles, Ca.90089, Available At:http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.ht ml
[11] "The Lightweight Java Game Library (LWJGL) Project", Available at: http://sourceforge.net/ projects / java-game-lib/files/? Source = navbar