Journal of Southeast University (English Edition) Vol. 24, No. 3, pp. 281 - 283 Sept. 2008 ISSN 1003—7985

Meta-analysis of potentially confounding effect of class size on associations between object-oriented metrics and maintainability

Lu Hongmin Zhou Yuming Xu Baowen

(School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: This paper uses three size metrics, which are collectable during the design phase, to analyze the potentially confounding effect of class size on the associations between object-oriented (OO) metrics and maintainability. To draw as many general conclusions as possible, the confounding effect of class size is analyzed on 127 C ++ systems and 113 Java systems. For each OO metric, the indirect effect that represents the distortion of the association caused by class size and its variance for individual systems is first computed. Then, a statistical meta-analysis technique is used to compute the average indirect effect over all the systems and to determine if it is significantly different from zero. The experimental results show that the confounding effects of class size on the associations between OO metrics and maintainability generally exist, regardless of whatever size metric is used. Therefore, empirical studies validating OO metrics on maintainability should consider class size as a confounding variable.

Key words: object-oriented; metrics; validation; class size; confounding; maintainability

In the validation of object-oriented (OO) metrics, it is common to use a univariate model to determine whether an individual metric is statistically related to important external quality attributes in the expected direction[1]. In Ref. [2], however, El-Emam et al. argued that such a validation method did not take into account the confounding effect of class size. In their research, before controlling for class size (measured by source lines of code, SLOC), the investigated OO metrics were strongly associated with fault-proneness. However, after controlling for class size, the associations between investigated metrics and fault-proneness disappeared. They therefore recommend that future validation studies should always control for size. In Ref. [3], Zhou et al. found that class size also has a strong confounding effect on the associations between OO metrics and change-proneness.

In this paper, we investigate the potentially confounding effect of class size on the associations between OO metrics and maintainability. We use three size metrics, which are available during the design phase, to investigate this subject. To draw as many general conclusions as possible, we first extract OO metrics from a large number of systems. Then, we analyze the collected data using meta-analytical techniques.

Received 2008-04-15.

Biographies: Lu Hongmin (1975—), female, graduate; Xu Baowen (corresponding author), male, doctor, professor, bwxu@ seu. edu. cn.

Foundation item: The National Natural Science Foundation of China (No. 60425206, 60633010).

Citation: Lu Hongmin, Zhou Yuming, Xu Baowen. Meta-analysis of potentially confounding effect of class size on associations between object-oriented metrics and maintainability[J]. Journal of Southeast University (English Edition), 2008, 24(3): 281 - 283.

1 Confounding Effect and Model

Suppose that X is the independent variable, Y is the dependent variable, and Z is the third variable causally associated with Y and non-causally or causally associated with X. Then, we can describe the relationships between these variables by the following equations[4]:

$$Y=\beta_0^* + \tau X + e^* \tag{1}$$

$$Y=\beta_0 + \tau' X + \gamma Z + e \tag{2}$$

$$Z=\beta_0^{**} + \beta X + e^{**} \tag{3}$$

where β_0^*, β_0 and β_0^{**} are respectively the population regression intercepts and e^*, e and e^{**} are respectively the residuals. For simplicity of presentation, we use population parameters in Eqs. (1), (2) and (3) recognizing that in practice the population values are replaced by sample values.

From Eqs. (1), (2) and (3), we can conclude that

$$\tau = \tau' + \beta\gamma \tag{4}$$

It can be seen that the distortion of the association between X and Y due to Z is $\beta\gamma$, and τ is different from τ' whenever both β and γ are non-zero. τ is called the total effect, τ' is called the direct effect, and $\beta\gamma$ is called the indirect effect.

2 Research Method

2.1 Data source

In this study, we download 127 C ++ systems and 113 Java systems from http://sourceforge. net/. The selection criteria are as follows: 1) Software must be written in pure Java or C ++; 2) Software with a large number of downloads (such information is available) has priority to be selected. The purpose of the selection criteria is to make selected systems as representative as possible of open source software. Tab. 1 shows some descriptive statistics for the C ++ and Java systems.

2.2 Dependent and independent variables

In this study, system maintainability is quantified via a maintainability index M. M is a combination of widely-used and commonly-available metrics that affect maintainability[5]. More precisely, M is defined as follows:

$$M = 171 - 5.2\ln(\text{aveV}) - 0.23\text{aveV}(g') - 16.2\ln(\text{aveLOC}) + 50\sin(\text{sqrt}(2.4\text{perCM}))$$

where aveV is the average Halstead's volume per module[6], aveV(g') is the average extended cyclomatic complexity per module, aveLOC is the average count of lines of

Tab. 1 Descriptive statistics for the number of classes

Language	N	Minimum	Maximum	Mean	Median	Standard deviation
C++	127	63	1 566	236.74	170	229.88
Java	113	64	4 030	485.94	291	556.04

source code per module, and perCM is the average percentage of lines of comments per module.

The independent variables consist of six OO metrics and three class size metrics. The six OO metrics are WMC, DIT, RFC, NOC, CBO and LCOM, which capture the concepts of cohesion, coupling, and inheritance. The definitions of these OO metrics can be found in Ref. [7]. The three class size metrics are NM, NA and NMA, which are available during the design phase. NM is the number of methods in a class; NA is the number of attributes in a class, and NMA is the number of methods and attributes in a class, excluding inherited ones.

2.3 Data analysis methods

The analysis of the confounding effect of the third variable is based on the linear regression equations of standardized variables:

$$\tau_s = \tau_s' + \beta_s \gamma_s \tag{5}$$

where τ_s, τ_s', β_s and γ_s are the standardized regression coefficients that respectively correspond to τ, τ', β and γ in Eqs. (1), (2) and (3). In this context, the total, direct, and indirect effects are denoted by τ_s, τ_s' and $\beta_s\gamma_s$, respectively.

For each OO metric, we first compute the indirect effect and its variance for individual systems. Then, we use a statistical meta-analysis technique to compute the average indirect effect over all systems and determine if it is significantly different from zero.

2.3.1 Calculating indirect effect for individual systems

Since β_s is equal to the correlation coefficient between X and Z, and since γ_s is equal to the partial regression coefficient relating Z and Y, controlling for X, the indirect effect $\beta_s\gamma_s$ can be equivalently represented as

$$\beta_s\gamma_s = \frac{\rho_{XZ}(\rho_{ZY} - \rho_{XY}\rho_{XZ})}{1 - \rho_{XZ}^2} \tag{6}$$

where ρ_{XY}, ρ_{XZ}, and ρ_{ZY} are respectively the population Pearson correlation coefficients between X and Y, between X and Z, and between Z and Y.

Let $\mathrm{ES}_{\mathrm{indirect}}$ be the sample estimate of $\beta_s\gamma_s$ obtained by respectively substituting the sample Pearson correlation coefficients r_{XY}, r_{XZ}, and r_{ZY} for ρ_{XY}, ρ_{XZ}, and ρ_{ZY} in Eq. (6). The variance of $\mathrm{ES}_{\mathrm{indirect}}$ is

$$\mathrm{var}(\mathrm{ES}_{\mathrm{indirect}}) = \boldsymbol{a}\boldsymbol{\Phi}\boldsymbol{a}^{\mathrm{T}} \tag{7}$$

where the vector

$$\boldsymbol{a} = \left\{\frac{r_{XZ}^2 r_{ZY} + r_{ZY} - 2r_{XZ}r_{XY}}{(1 - r_{XZ}^2)^2}, \frac{-r_{XZ}^2}{1 - r_{XZ}^2}, \frac{r_{XZ}}{1 - r_{XZ}^2}\right\} \tag{8}$$

and the variance-covariance matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathrm{var}(r_{XZ}) & \mathrm{cov}(r_{XZ}, r_{XY}) & \mathrm{cov}(r_{XZ}, r_{ZY}) \\ \mathrm{cov}(r_{XZ}, r_{XY}) & \mathrm{var}(r_{XY}) & \mathrm{cov}(r_{ZY}, r_{XY}) \\ \mathrm{cov}(r_{XZ}, r_{ZY}) & \mathrm{cov}(r_{ZY}, r_{XY}) & \mathrm{var}(r_{ZY}) \end{bmatrix} \tag{9}$$

The variance of the sample Pearson correlation r_{ab} between two variables a and b is

$$\mathrm{var}(r_{ab}) = \frac{(1 - r_{ab}^2)^2}{N} \tag{10}$$

where N is the sample size, while the covariance of two sample Pearson correlations in which there is a common variable, say r_{ab} and r_{ac}, is

$$\mathrm{cov}(r_{ab}, r_{ac}) = \frac{\frac{1}{2}(2r_{bc} - r_{ab}r_{ac})(1 - r_{ab}^2 - r_{ac}^2 - r_{bc}^2) + r_{bc}^3}{N} \tag{11}$$

2.3.2 Calculating average indirect effect over all systems

There are two kinds of meta-analyses: fixed-effect models and random-effect models[8]. In the fixed-effect models, all effect sizes are assumed to be estimates of a common population effect size. Consequently, results cannot be generalized beyond the included systems. In contrast, in the random-effect models, a random-effect variance results from drawing systems from a universe of possible systems in addition to the variation due to the sampling of subjects in the original studies. Because of this, random-effect models not only permit one to draw inferences about studies that have been made but also to generalize to studies that might be made in the future. In our study, the indirect effect estimates from individual systems are, hence, aggregated using random-effects models, with the formulae provided by Hedges and Vevea[8].

3 Experimental Results

Tab. 2 summarizes the results from our data set. For each individual OO metric: 1) The third column shows the average total effect τ_s computed using the random-effect models and its confidence interval (shown in parentheses) from the model where we do not control for class size; 2) The fourth, fifth and sixth columns show the average direct effect τ_s' computed using the random-effect models and its confidence interval (shown in parentheses) from the models where we do control for NM, NA, and NMA, respectively; 3) The seventh, eighth and ninth columns show the average indirect effect $\beta_s\gamma_s$ computed using the random-effect model and its confidence interval (shown in parentheses).

Tab. 3 shows the direction of the confounding effect of class size for individual OO metrics. Here, "+" means a positive confounding effect, "−" means a negative confounding effect, and 0 means class size has no confounding effect. As can be seen, the confounding effect of class size on the associations between OO metrics and maintainability in general exist, regardless of whatever size metric is used.

Tab. 2 Results of testing for confounding effect

System	Metric	Not controlling for size	Controlling for size			Testing for confounding		
			NM IMP	NAIMP	NMA	NM IMP	NAIMP	NMA
		r_w	r_w	r_w	r_w	Δr_w	Δr_w	Δr_w
C++ software	CBO	-0.281 7	-0.307 5	-0.244 0	-0.287 5	0.030 9	-0.033 5	0.012 2*
	LCOM	-0.089 4	-0.051 7	0.103 6	0.024 5*	-0.025 8	-0.172 0	-0.093 5
	RFC	-0.131 4	-0.123 9	-0.082 3	-0.075 7	-0.005 0*	-0.034 2	-0.042 5
	WMC	-0.357 9	-0.715 2	-0.356 0	-0.612 1	0.356 0	0.002 7*	0.254 8
	NOC	0.121 4	0.134 5	0.111 7	0.133 1	-0.005 1	0.004 1	-0.004 1
	DIT	-0.080 2	-0.081 7	-0.096 2	-0.084 3	0.003 0*	0.007 7	0.004 7
Java software	CBO	-0.321 9	-0.410 3	-0.330 4	-0.389 1	0.084 3	0.008 3	0.064 9
	LCOM	-0.087 5	-0.103 6	0.130 3	-0.043 8	0.017 0*	-0.217 2	-0.042 3
	RFC	-0.079 3	-0.086 0	-0.011 6*	-0.026 2*	-0.000 07*	-0.053 7	-0.043 1
	WMC	-0.220 4	-0.801 3	-0.233 1	-0.588 6	0.578 9	-0.016 1*	0.368 9
	NOC	0.194 5	0.200 9	0.185 9	0.199 9	-0.004 9	0.005 2	-0.003 8
	DIT	-0.076 5	-0.067 7	-0.057 4	-0.059 6	-0.003 1	-0.010 3	-0.008 2

Notes: r_w is the weighted mean Pearson's correlation coefficient with MI; Δr_w is the weighted mean difference between Pearson's correlation coefficients with and without control size; * means $p > 0.05$.

Tab. 3 Direction of confounders

Metric	Size as confounder for C++/Java		
	NM	NA	NMA
CBO	-/-	+/-	0/-
LCOM	+/0	-/-	-/+
RFC	0/0	+/+	+/+
WMC	-/-	0/0	-/-
NOC	-/-	+/+	-/-
DIT	0/+	-/+	-/+

4 Conclusion

This paper uses statistical meta-analysis technique to investigate the potentially confounding effect of class size on the associations between OO metrics and maintainability. The results show that the confounding effect of class size on maintainability generally exists and suggests that empirical validation of OO metrics on maintainability should also consider class size as a confounding variable.

References

[1] Basili V R, Briand L C, Melo W L. A validation of object-oriented design metrics as quality indicators [J]. *IEEE Transactions on Software Engineering*, 1996, **22**(10): 751-761.

[2] El-Emam K, Benlarbi S, Goel N, et al. The confounding effect of class size on the validity of object-oriented metrics [J]. *IEEE Transactions on Software Engineering*, 2001, **27**(6): 630-650.

[3] Zhou Y, Leung H. Examining the potentially confounding effect of class size on the associations between object-oriented metrics and change-proneness [R]. Hong Kong: Department of Computing of Hong Kong Polytechnic University, 2008.

[4] MacKinnon D P, Lockwood C M, Hoffman J M, et al. A comparison of methods to test mediation and other intervening variable effects [J]. *Psychological Methods*, 2002, **7**(1): 83-104.

[5] Welker K, Oman P, Atkinson G. Development and application of an automated source code maintainability index [J]. *Journal of Software Maintenance: Research and Practice*, 1997, **9**(3): 127-159.

[6] Halstead M H. *Elements of software science*[M]. New York: Elsevier, 1977.

[7] Chidamber S R, Kemerer C F. A metrics suite for object-oriented design [J]. *IEEE Transactions on Software Engineering*, 1994, **20**(6): 476-493.

[8] Hedges L V, Vevea J L. Fixed- and random-effects models in meta-analysis [J]. *Psychological Methods*, 1998, **3**(4): 486-504.

类规模对面向对象度量与易维护性关联关系潜在混和效果元分析

卢红敏　周毓明　徐宝文

(东南大学计算机科学与工程学院,南京 211189)

摘要:用在设计阶段可以收集到的3个规模度量分析类规模对面向对象度量与易维护性之间关联关系的混和效果. 为得到尽可能通用的结论,在127个C++系统和113个Java系统上分析类规模的混和效果. 首先,对每个面向对象度量,在单个系统上计算代表由类规模造成的关联扭曲的间接效果和它的方差. 然后,利用统计的元分析方法计算这些系统上的平均间接效果并判断它是否显著地区别于零. 实验结果表明,不论使用哪个类规模度量进行分析,类规模对面向对象度量与易维护性之间关联关系的混和效果通常是存在的. 因此,验证面向对象度量与易维护性之间关系的实验研究应将类规模视为一个混和变量.

关键词:面向对象;度量;验证;类规模;混和;易维护性

中图分类号:TP311.5