

ACCURACY IMPROVEMENT OF OBJECT ORIENTED SOFTWARE DESIGN MEASUREMENT USING ANALYTIC NETWORK PROCESS

Petrus Mursanto
Enterprise Computing Lab
Faculty of Computer Science, Universitas Indonesia
Kampus UI, Depok, INDONESIA 16424
E-mail: santo@cs.ui.ac.id

ABSTRACT

The application of Analytic Network Process (ANP) has provenly increased accuracy in measuring object oriented design quality compared to the Analytic Hierarchy Process (AHP). The accuracy improvement has been shown by the result of software design quality measurement over a number of object oriented (OO) Java applications as the experimental samples. The samples consist of three set of student works, each of which has different level of complexity in implementing object oriented (OO) characteristics. Their quality is ranked using AHP and ANP based on MOOD's metrics as the multi criteria. Another multi criteria decision making (MCDM) tool called Promethee has also been applied for comparison purpose. The result shows that relative quality of the samples using ANP and Promethee are relatively consistent, whereas AHP's result shows some inconsistency. Moreover, final priority values resulting from ANP has been more accurately representing OO quality embedded in the softwares. The result of experiments conducted in this paper has supported the robustness of ANP as opposed to AHP in dealing with inter-dependent criteria, such as the characteristics of OO metrics.

Keywords: ANP, AHP, Promethee, MCDM, Object Metrics.

1. Introduction

The object paradigm in software development has shifted the way of programmers to implement software systems and designers to evaluate the system quality. Object oriented (OO) software system design must conform to the properties of good OO characteristics (Briand, Wüst, Daly, and Porter, 2000). Many research groups have developed metrics for measuring quality of OO design. Good OO designs tend to have the same characteristics which are represented by a set of metric values. There are two metric sets that are widely used in implementing object software; they are Metrics for Object Oriented Software Engineering (MOOSE) (Chidamber and Kemerer, 1993) and Metrics for Object Oriented Design (MOOD) (Abreu and Carapuça, 1994). The metrics represent quantitative values that describe inter-dependency and inter-correlation of OO design components within software system.

However, with a set of metric values given, we are not able to decide whether or not the system software has a good design. There should be a method to interpret those metric values to derive an estimate conclusion of the quality. This paper presents one way among others to interpret a bunch of OO metrics data obtained from measuring a number of Java software applications. The metrics are treated as multi criteria in defining OO quality. Since the characteristic of OO metrics are inter-dependent one to the others, Analytic Network Process (ANP) is a suitable tool to serve the purpose. The application of ANP and its effectiveness are evaluated through a series of experiments reported in this paper.

2. Previous Works

The adoption of the OO paradigm is to produce better quality and more maintainable software. The main structural mechanisms of this paradigm, namely, inheritance, encapsulation, information hiding and polymorphism, are the keys to foster reuse and to achieve easier maintainability. However, the use of language construction that supports those mechanisms can be more or less intensive, depending mostly on the designer ability. Hence, we can expect various of different quality products to emerge, as well as different productivity gains (Abreu and Melo, 1996).

Java is the most popular OO programming widely used nowadays. A measurement tool is required to derive quantitative values representing quality of Java applications. AHP and ANP call the value as **priority**. Christariny developed a Metrics Calculator for Java program based on Abreu's metrics in (Christariny, 2004). Nurmaya improved the usability of the calculator and utilized it for explorative experiments on the applicability of the Confidence Interval Good OO Design (CIGOOD) (Nurmaya, 2007). She revealed the drawback of CIGOOD, i.e. the measurement result is much dependent on the quality of sample programs taken as the statistical population of the method. The use of OO metrics as weighted parameters in multi criteria decision using the AHP (Saaty, 2000) was proposed by (Hermawan, 2007) and (Dulianto, 2008). (Rahman, 2009) developed an automated tool so that weight of parameters is adjustable to suit the aspect of design being evaluated. The use of AHP has been explored and compared with another MCDM tool, called Promethee in (Mursanto and Sari, 2011). However, the success of AHP does not always apply to arbitrary sample of applications under observation.

2.1. Object Oriented Metrics

There are two major sets of metrics for OO, i.e. *Metrics for Object Oriented Design* (MOOD) to represent quality of an integrated system (Abreu, Esteves and Goulao, 1996), and *Method For Object Oriented Software Engineering* (MOOSE) to measure quality of each class as the component of OO system (Chidamber and Kemerer, 1993). MOOSE presents the following metrics: Weighted Method per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling between Object Classes (CBO), Response Set for A Class (RFC), and Lack of Cohesion in Methods (LCOM). Whereas MOOD consists of the following metrics: Method Hiding Factor (MHF), Attribute Hiding Factor (AHF), Method Inheritance Factor (MIF), Attribute Inheritance Factor (AIF), Polymorphism Factor (POF), and Coupling Factor (COF). A metrics calculator has been developed for extracting the above metric values from a collection of Java class or jar files (Christariny, 2004). Thanks to the metric calculator, the use of AHP has been explored in (Mursanto, Hermawan and Jatmiko, 2008). In this paper we focus on the application of MOOD for measuring students works quality.

2.2. Promethee

Promethee stands for Preference Ranking Organization Method for Enrichment Evaluation, is a multi criteria decision method that has produced six extensions: partial ranking, complete ranking, interval based ranking, continuous case, MCDA with segmentation constraints, and a representation of human brain (Brans and Mareschal, 1994). Paper of (Mursanto, Hermawan and Jatmiko, 2008) explores the application of Promethee I (partial ranking) and II (complete ranking).

Once a set of alternatives and criterias have been determined, decision makers choose one out of six preference functions for each of the criteria and define required parameters (thresholds and min-max) as well as criteria weights to determine preference index for every pair of criteria alternatives. There are six preference functions to choose from (Brans and Vincke, 1985). Promethee I or partial ranking is achieved by computing strength and weakness of the alternatives. Figure 1a depicts an example of Promethee I diagram. Promethee II or complete ranking is achieved by calculating net flow for each of the alternatives by reducing weakness from its strength. Figure 1b describes Promethee II diagram for corresponding partial ranks in Figure 1a.

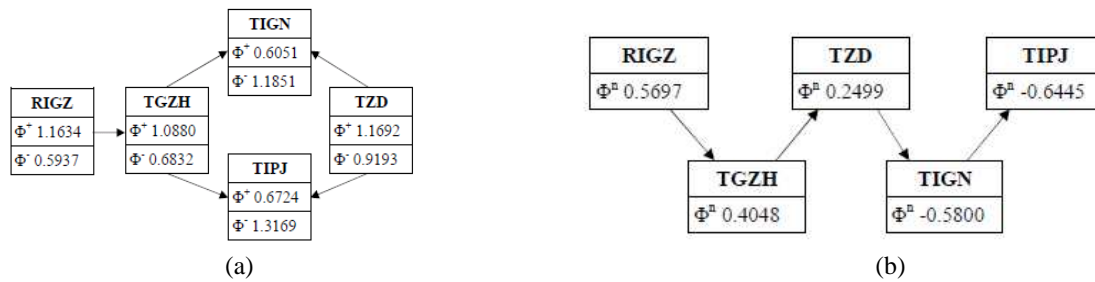


Fig 1. Promethee netflow of (a) Partial ranking and (b) Complete ranking

Unlike AHP, Promethee has no guidance to determine the weights. It does not have any support for criteria hierarchy either (Brans and Vincke, 1985). Too many non-intuitive inputs are also required in Promethee such as suggested by (Hostmann, 2005). However, Promethee is good in involving decision makers to simulate ranking process by using different weights and preference functions. It also has simpler calculation and is easier to understand compared to AHP, as this may increase decision makers' confidence (Niknafs, Charkari, and Niknafs, 2008).

AHP is the correct method for decision making tool involving many elements with hierarchical structure as shown in Figure 2a. However, we have learnt that metrics are inter-related and not fully independence one to the others. This situation has motivated us for applying Analytic Network Process (ANP) in addition to AHP. The network constructed from metrics correlation and inter-dependence is shown in Figure 2b which is derived from Halim's Thesis (Halim, 2012). We conducted re-measurement over the same samples of OO software systems using ANP. Promethee was also applied to the same samples. The result is compared with the ones brought from previous measurements using AHP and ANP. This paper reports the overall comparison of results produced by AHP, ANP and Promethee.

3. Scenario of Experiments

A number of Java applications are used as experimental samples in this research. The samples are student works in Fundamental of Programming class at Faculty of Computer Science, Universitas Indonesia. This course, taught in the first semester, introduces students to the concept of object oriented programming with Java. Students are assigned with programming works in stages according to the material covered in class. The assignment complexity is gradually increasing and covering encapsulation (E), inheritance (I) and polymorphism (P). From each stage, thirty student works are used as the samples. For the sake of clarity in this paper, they are called A, B, C, ... up to Z, AA, BB, CC, DD. The three sets of works are named after their accumulated complexity, i.e. E, EI, and EIP. MOOD values are measured from the three sets of samples; they are presented in Table 1. The OO metrics are treated as multi criteria in AHP and ANP; their weights are adjusted according to the main OO characteristics being implemented for every stage. The hierarchy structure of AHP and network structure of ANP is presented in Figure 2. The priority preference of hierarchical components is presented in Table 2.

Hierarchical components of Encapsulation, i.e. MHF and AHF are equally weighted 50%. So are MIF and AIF in Inheritance. Whereas POF and COF are 100% weighted in Polymorphism and Coupling respectively. For each set, we have adjusted the weights of E, I, P and Coupling to represent the emphasized characteristics being evaluated, as shown in Table 2. Inconsistency ratio is maintained less than 0.10 for every configuration.

Table 1. Partial MOOD values of works samples

No.	Sample	MHF	AHF	MIF	AIF	POF	COF
1	A	0.109657	0.799194	0.815031	0.760884	0.297348	0.023508
2	B	0.120526	0.776517	0.859813	0.766206	0.314487	0.202284
3	C	0.032258	0.687500	0.828254	0.729957	1.490196	0.729473
...
29	BB	0.000000	0.820512	0.983651	0.926966	0.000000	0.831095
30	CC	0.038217	0.609589	0.968244	0.918798	3.000000	0.070000

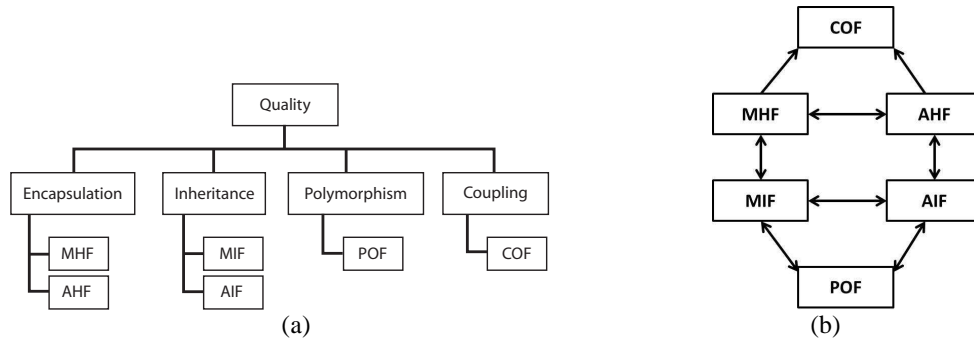


Figure 2.a) hierarchical structure in AHP and b) network structure in ANP

Table 2. Hierarchical weights of sample sets

Sample set	E	EI	EIP
Encapsulation	0.700	0.640	0.594
- MHF	0.500	0.500	0.500
- AHF	0.500	0.500	0.500
Inheritance	0.100	0.235	0.263
- MIF	0.500	0.500	0.500
- AIF	0.500	0.500	0.500
Polymorphism	0.100	0.063	0.081
- POF	1.000	1.000	1.000
Coupling	0.100	0.063	0.062
- COF	1.000	1.000	1.000
Inconsistency ratio	0.00	0.08	0.03

AHP and ANP require us to do pairwise comparison of the alternatives in respect to every criteria. For the purpose of comparison, the metric values do not properly fit to the scheme. It can be easily observed that inconsistency ratio is far beyond the threshold if we use the raw data in the comparisons. Hence, we have to convert them so that logically represent preferences of one to the others. For example, COF of B and C are 0.202284 and 0.729473 respectively. These values must be converted to represent preference degree of B over C and vice versa according to preference scheme of AHP. Although possible value of COF is ranging from 0 to 1, we should better use min and max value from the raw data. This gives us more contrast degree of preferences in case COFs cluster into certain values. In this experiment, COF min is 0.023508 and COF max is 0.831095. Hence the range value of COF must be interpolated into the range from 1 to 9. Since a good design has low coupling factor, or in other words lower COF is preferred, then the value is inversed between B and C.

$$\text{Pref B over C} = 1 + \frac{(0.729473 - 0.202284)}{(0.831095 - 0.023508)} \times 8 = 6.222362 \quad (\text{Eq. 1})$$

$$\text{Pref C over B} = \frac{1}{6.222362} = 0.160701 \quad (\text{Eq. 2})$$

This mechanism must be applied for every alternative with respect to each paired criteria in AHP. The same principle is conducted for every pairwise comparison to build the ANP supermatrix. The structure of ANP drawn using SuperDecision is shown in Figure 3.

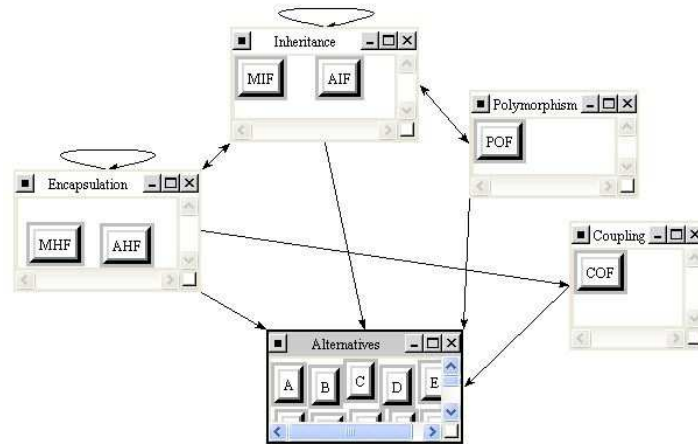


Figure 3. ANP struture is drawn with SuperDecision

Both AHP and ANP produce priorities of 30 program samples with respect to their quality. It turns out that the AHP's priorities are different from the ANP's for every set of samples E, EI and EIP. Since the Promethee has succeeded in producing the same relative quality in (Mursanto and Sari, 2011), we conducted remeasurement of the samples using Promethee. In this experiments, each set of samples was evaluated using Promethee based on weights produced by AHP and ANP.

4. Results and Discussion

The results of experiments for every set of samples are compiled in Table 3. Greyed rows are omitted from the Table to eliminate uninteresting parts of the results. In each column of the Table, student works are sorted according their relative qualities based on AHP, ANP and Promethee schemes. Highlighted cells are priorities of the samples sorted differently by AHP. We can see that for the whole sets, ANP always produces ranks in exactly the same order with Promethee. In other words, relative quality of the whole samples is consistently maintained with ANP as opposed to AHP. This result is not surprising as ANP is more appropriately applied for inter-correlated multi-criteria decision making. Such situation is well fitted by the charateristics of OO metrics.

Furthermore, we plot priority values that represent samples' relative qualities. For every scenario, samples are sorted by their values from largest to smallest. Figure 4 shows the plots of rank values for E and EI. It can be seen that ANP curve relatively equals to the Promethee's. This also happens for EIP although it is not displayed here. Although priority values are not exact quantitative representations of design quality, any sample's position in the rank remains constant relative to the others. This supports the argument that difference between two samples' rank values somehow represents relative degree of the quality between both samples.

5. Conclusion and Future Works

This paper reports reviews previous works on measuring quality of OO design based on Java codes. Several methods have been proposed to interpret values of MOOD metrics for deriving a single quantitative value that represent quality of the codes. The nature of OO metrics characteristic that is inter-correlated and inter-dependent one to the others had motivated us to evaluate the robustness of ANP compared to AHP.

Table 3. Rank Results of AHP, ANP and Promethee

Rank	E			EI			EIP		
	AHP	ANP	Prom	AHP	ANP	Prom	AHP	ANP	Prom
1	N	N	N	J	J	J	H	H	H
2	E	E	E	A	D	D	W	S	S
3	C	C	C	D	S	S	D	D	D
4	I	F	F	S	A	A	S	W	W
5	L	I	I	H	H	H	A	A	A
6	W	L	L	AA	AA	AA	J	J	J
7	P	P	P	W	W	W	B	B	B
...									
14	Q	Q	Q	Z	Z	Z	U	U	U
15	G	G	G	Q	U	U	Q	Q	Q
16	A	W	W	U	B	B	R	CC	CC
17	X	A	A	B	Q	Q	Y	R	R
18	F	X	X	L	L	L	CC	Y	Y
...									
22	J	K	K	P	P	P	E	E	E
23	AA	R	R	BB	K	K	X	X	X
24	K	B	B	K	BB	BB	T	T	T
25	B	M	M	E	E	E	G	G	G
26	Z	V	V	O	O	O	O	O	O
27	DD	DD	DD	M	F	F	I	M	M
28	V	O	O	V	V	V	V	V	V
29	O	Z	Z	F	M	M	C	BB	BB
30	R	J	J	T	T	T	K	K	K

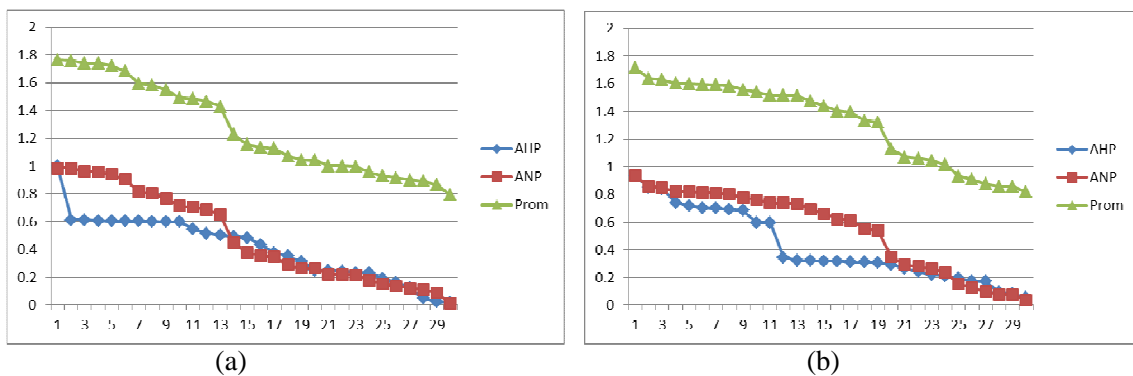


Figure 4. Plot of ranks for (a) E and (b) EI

We also applied Promethee scheme, whose results are used as comparison. It has been shown that ANP produced more stable and consistent priorities compared to AHP with reference to what produced by Promethee. More over, the difference between priority values also represents degree of quality difference between the samples. Overall, the result of experiment conducted in this paper supports the argument that ANP is the proper tool for multi criteria decision making in which inter-dependency among the criteria exists.

Although ANP and Promethee produce the same relative rank over the whole sample sets, the effect of applying ANP's weighting scheme to Promethee must be further examined. Hence, the results are yet to be validated.

Further experiments are required to verify ANP's robustness in dealing with other OO metrics such as MOOSE as well as using other MCDM tools. The mechanism to interpolate metric values for AHP and ANP preference scheme must also be further investigated.

REFERENCES

Abrue F.B., Esteves R., Goulao M. (1996) The Design of Eiffel Programs: Quantitative Evaluation Using the MOOD Metrics, *Proceeding of TOOLS*, USA.

Abreu F.B., Melo W.L. (1996). Evaluating the Impact of Object-Oriented Design on Software Quality, *3rd International Software Metrics Symposium*, Germany.

Brans, J.P., B. Mareschal (1994). Chapter 5: Promethee Methods, available on <http://www.inf.unideb.hu/valseg/dolgozok/anett.racz/docs/DSS/Promethee.pdf>.

Brans, J.P., Ph.Vincke, (1985). A Preference Ranking Organisation Method: (The PROMETHEE Method for Multiple Criteria Decision-Making), *Management Science* 31 (6).

Chidamber, Shyam, Kemerer, Chris F. (1993). A Metrics Suite for Object-Oriented Design, M.I.T. Sloan School of Management E53-315.

Christariny (2004). Metrics Calculator for Object Oriented System. Final Project – Faculty of Computer Science, Universitas Indonesia.

Fernando Brito e Abreu, Rogério Carapuça (1994). Object-Oriented Software Engineering: Measuring and Controlling the Development Process. *4th Int Conference on Software Quality*. McLean, VA, USA.

Des Dulianto (2008). The Application of AHP and Software Metrics in Defining Object Oriented Software Quality. Master Thesis – Faculty of Computer Science, Universitas Indonesia.

Halim, A. (2012). Internal Quality Measurement Model for Prediction of Fault-Prone UML Design, Master Thesis – Faculty of Computer Science, Universitas Indonesia.

Hostmann, M. (2005). Decision Support for River Rehabilitation. *Dissertation at Swiss Federal Institute of Technology*, Zurich.

Lionel C. Briand, Jürgen Wüst, John W. Daly, D. Victor Porter (2000). Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software*, 51 (3), 245–273.

Niknafs, A., N.M. Charkari, A.A. Niknafs, (2008). A PROMETHEE-based Recommender System for Multi-sort Recommendations in On-line Stores. Tarbiat Modares University.

Nurmaya (2007) The Measurement of Software Design using Metrics for Object Oriented Design (MOOD). Final Project – Faculty of Computer Science, Universitas Indonesia.

Petrus Mursanto, Efan Hermawan, and Wisnu Jatmiko (2008). Measuring Relative Quality of Object Oriented Software Design using Analytic Hierarchy Process. *Proc. Indonesia-Japan Joint Scientific Symposium (IJSS)*, 89-94.

Petrus Mursanto and Wida Sari (2011). Defining Relative Qualities of Object Oriented Design Implementations using AHP and Promethee. *ISAHP, Sorrento – Italy*.