The Third AFSS(1998). 302=307

The Analysis of Significance of the Reusability Decision Metrics using Rough Set

Wan-Kyoo Choi[†], Young-Nam Na[††], Sung-Joo Lee[†], Hwan-Mook Chung[†††]

ABSTRACT

Software reuse is a well-known method to increase the productivity of software, nevertheless it is not employed well on real world. One of the important factors that this problem occurs is programers' distrust in the existing components. Therefore in this paper, to increase the reliability of reusability decision, we proposed a method which can analyze significance of the reusability decision metrics using Rough Set.

1. Introduction

Recently, software's size has been increased more and more, and consequently human resource and cost required to maintenance software have been relatively increased. To resolve this problem, many researches about software reuse have been in progress. Software reuse is a method that reuse the existing software as part of the new software.

Previous researches illustrated that the software reuse increased the software productivity[5]. But, because of the following factors, software reuse is not employed well. First, because most of projects have the restricted budget and development schedule, programmers do not have enough time to search and create the reusable softwares. Second, programmers do not trust the codes which others generate. Third, programmers have difficulties of understanding and adapting the existing codes. Fourth, the standard system of the reuse management and support does not exist.

To solve problems described above, many researches have been in process, which dissolve users' distrust by guaranteeing software's quality or produce the standard system of the reuse management and support[1][3][9][10].

But most of existing researches about guarantee of software quality measure the software reusability by combining the existing metrics or decide it based on degree corresponded to the enumerated evaluation criteria.

These researches don't reflect how much effect each metric has on software's reusability decision and measurement. It means that some of factors may be more important on deciding and measuring the software reusability.

Therefore, in this paper, we proposed

[†] Dept. of Computer Science, Chosun University
[††] Chosun University Technical College
[†††] Faculty of Electronic & Information Engineering, Catholic Univ. of Taegu-Hyosung

method that can analyze an significance of the reusability measurement and decision factors based on rough set theory. An analysis of significance of them can increase reliability of the existing reusability evaluation models.

Rough set theory provides efficient algorithms finding hidden pattern in data, finds minimal sets of data, evaluates significance of data, generates sets of decision rules from data, is easy to understand and offers straightforward interpretation of obtained results[8].

In section 2, we describe and analyze the existing reusability evaluation models previously published and their problems. In section 3, we describe method that evaluates significance of data. In section 4, we analyze and produce significance of each decision metric in the existing reusability decision model[8]. In section 5, we present conclusion and future studies.

2. The existing assessment models

The reusability assessment model determines whether software components can be reused or not using several quality measurement metrics. The representative models are CARE system[3] and McCall's Reuse Quality Factors[2].

2.1 CARE System

CARE system defines functional usefulness, reuse costs and quality as the quality factors, which affect the reusability. Functional usefulness includes the commonality and the variety of functions performed by components. Reuse costs are the costs extracting the component from the old system, packing it into a reusable component, modifying it, and integrating it into the new system. The concept of quality includes correctness, readability, testability, ease of modification, and performance that are important for component reuse. These were measured by four metrics as following: Halstead's volume, complexity of module to be measured by McCabe's cyclomatic number, component regularity to be measured by the closeness of the estimate to the actual size, reuse frequency according to the number of calls addressed to a component.

2.2 McCall's Reuse Quality Factors

McCall defines 'generality', 'modularity', 'self documentation', 'hardware independency', and 'software system independency' as the quality factors which have an effect on the software reuse. Generality, which is the most fundamental property, is called the degree to be utilized not only for specific application areas and domains but also for general purposes. Since the basic unit of the software reuse is module, each module should have such fundamental conditions as abstraction and information hiding and furthermore, the maximum cohesion and the minimum coupling. A modules to be reused have nothing to do with other softwares and hardwares to be executed. And for the reuse of software components, it is necessary to inform the correct function, usage, and interface of modules.

2.3 Problems of the existing models

CARE system measured functional usefulness, reuse cost, and quality of modules selected for reuse, employing four metrics. McCall's reuse quality factors measured reusability, using the five metrics.

Really, we can not produce the software that satisfy all the evaluation criteria and the measurement metrics proposed by the existing models. Therefore we have to decide what measurement metrics are more important. But the existing models can not evaluate significance of them.

Therefore, in this paper, we propose method that can evaluate and utilize significance of them.

3. Rough Set Theory

3.1 Rough Sets

Rough sets proposed by Z.Pawlak can be described as approximate inclusion of sets and is a new mathematical approach to vagueness and uncertainty[5]. Rough sets concept can be applied to automatic classification, pattern recognition, learning algorithms, etc.

In fig.1, an approximation space A is the ordered pair A=(U, R) where U is the universe and R is an indiscernibility (equivalence) relation. If $(x_1, x_2) \in R$, x_1 and x_2 are indiscernible in A. Equivalence classes of relation R are called elementary sets. Any finite union of elementary sets in A is called a definable sets in A.

In rough set theory, elements which can be with certainty classified as elements of X by employing the given knowledge can be expressed approximately by rough sets called upper and lower approximation of X, positive, negative and boundary region of X.

$$R^*(X) = \cup\{Y \in U/R : Y \cap X \neq \varnothing\}$$
$$R_*(X) = \cup\{Y \in U/R : Y \subseteq X\}$$
$$POS_R(X) = R_*X$$
$$NEG_R(X) = U - R^*X$$
$$BN_R(X) = R^*X - R_*X$$

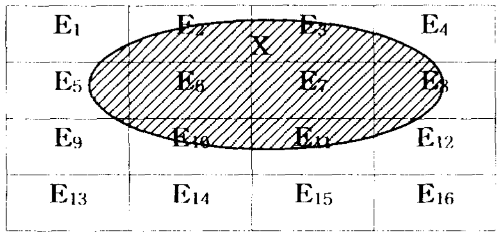

Fig.1 Approximation space A

In fig.1, each set is as following:

$$U = \{E_1, E_2, \ldots, E_{15}, E_{16}\}$$
$$R^*(X) = \{E_1, E_2, \ldots, E_{11}, E_{12}\}$$
$$R_*(X) = \{E_6, E_7\}$$
$$POS_R(X) = \{E_6, E_7\}$$
$$NEG_R(X) = \{E_{13}, E_{14}, E_{15}, E_{16}\}$$
$$BN_R(X) = \{E_1, \ldots, E_5, E_8, \ldots, E_{12}\}$$

3.2 Knowledge Representation

Knowledge is necessarily connected with the variety of classification patterns related to specific parts of the real and abstract world, called the universe of discourse. These classification pattern can be easily expressed using decision table. Decision table, T=(U, (C, D)) is Knowledge Representation System with distinguished condition attributes(C) and decision attributes(D).

3.3 Significance of attributes

If all elementary categories of knowledge C can be defined in terms of some elementary categories of knowledge D, knowledge C is superfluous within the knowledge base in the sense that D provides all characterization of C. This is expressed by dependency of knowledge and as following;

$$k = \gamma_C(D) = \frac{|POS_C(D)|}{|U|} \quad \text{-----(eq.1)}$$

$|POS_C(D)|$: cardinal of POSc(D)

$|U|$: cardinal of U

In decision table, some of all the given attributes may have greater significance for decision-making. This is expressed by significance of attributes and as following;

$$S_C = \gamma_C(D) - \gamma_{C-C'}(D) \quad \text{---------(eq.2)}$$

4. Analysis of significance of the reusability decision metrics

The Adaptable Reusability Decision Model(A-RDM)[8] decide reusability, combining LOC(Lines of Code), cyclomatic number, volume, difficulty and effort.

A-RDM chooses a boundary of above five metrics as like table.1, employing the evaluation criteria proven by many researches and experiments in industry studies. It constructs decision table as like table.2, classifying 175 modules within run-time library of C compiler using table.1.

Table.1 Classifying criteria

criteria / metric	VS	LS	LC	VC
LOC	<50	50~100	100~150	>150
Cyclomatic Number	<10	10~20	20~50	>50
Volume	<200	200~1000	1000~10000	>10000
Difficulty	<10	10~30	30~100	>100
Effort	<50000	50000~100000	100000~300000	>300000

note) VS : Very simple LS : Little Simple
LC : Little Complex VC : Very Complex

Table 2. Decision table for reusability decision

U	a	b	c	d	e	reusable
1	0	0	0	0	0	yes
2	0	0	0	1	0	yes
3	0	0	0	2	0	yes
4	0	1	0	1	0	yes
5	1	1	0	2	0	yes
6	1	1	0	2	1	yes
7	1	1	0	1	0	yes
8	2	1	1	2	2	yes
9	0	1	0	2	0	yes
10	1	0	0	0	0	yes
11	2	2	0	2	2	yes
12	1	0	0	2	0	yes
13	1	1	1	2	2	yes
14	1	0	0	2	2	yes

< a=Lines of Code, b=Cyclomatic number, c=Volume, d=Difficulty, e=Effort >

In table.2 C={a, b, c, d, e} is set of condition attributes and D={reusable} is set of decision attribute.

Significance of each attribute in table.2 is as following.

$POS_C(D)$ = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
$POS_{C-\{a\}}(D)$ = {2, 3, 6, 11, 13}
$POS_{C-\{b\}}(D)$ = {1, 6, 7, 8, 10, 11, 13, 14}
$POS_{C-\{c\}}(D)$ = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
$POS_{C-\{d\}}(D)$ = {6, 8, 11, 13, 14}
$POS_{C-\{e\}}(D)$ = {1, 2, 3, 4, 7, 8, 9, 10, 11, 13}

$\gamma_C(D) = 14/14 = 1$
$S_a = \gamma_C(D) - \gamma_{C-\{a\}}(D) = 9/14$
$S_b = \gamma_C(D) - \gamma_{C-\{b\}}(D) = 6/14$
$S_c = \gamma_C(D) - \gamma_{C-\{c\}}(D) = 0$
$S_d = \gamma_C(D) - \gamma_{C-\{d\}}(D) = 9/14$
$S_e = \gamma_C(D) - \gamma_{C-\{e\}}(D) = 4/14$

Fig.2 Calculation of significance of attribute

Consequently, in A-RDM, significance of each metric used to decide modules' reusability is as like table.3

Table.3 Significance of each attribute

Attribute	a	b	c	d	e
Significance	0.6428	0.4285	0	0.6428	0.2587

Table.3 illustrates that when A-RDM decides modules' reusability, volume is not important and LOC(lines of code) and difficulty is most important.

5. Conclusion

Software reuse is a well-known method to increase the productivity of software, nevertheless it is not employed well on real world. One of the important factors that this problem occurs is programers' distrust in the existing components.

Therefore, in this paper, we proposed a method which can analyze significance of the reusability decision metrics, in order to increase the reliability of the reusability decision.

Analysis of significance of metrics has advantages as following.

First, as the existing quality measurement models introduce and utilize it, their reliability can be increased.

Second, when evaluating effort of understanding and adapting of the existing software components, it can is used as weight-value for evaluation.

A continued research about method that can utilize the analyzed significance of metrics rather than only propose it is needed. Therefore, we continue researching the method that can evaluate a degree of reusability of the existing software components employing the significance of metrics

Referrence

[1] Peter Freeman, *TUTORIAL : Software reusability'*, pp.2-8, pp.10-23, pp.50-56, pp.106-116, pp.151-154, pp. 155-178, IEEE, Washington, 1987.

[2] Rogers S.Pressman, *Software engineering : A Practitioner's approach*, pp.452-458, McGraw-Hill, New York, 1987.

[3] Caldiera, G. and V.R. Basili, "Identifying and Qualifying Reusable Software Components", IEEE Computer, pp.61-70, Feb, 1991.

[4] Tracz W., *Tutorial : SOFTWARE*

REUSE : EMERGING TECHNOLOGY, pp.18-22, IEEE, 1988.

[5] Pawlak Z., "Rough sets", International Journal of Computer and Information Sciences, 11, pp.341-356, 1982.

[6] Pawlak Z., *Rough Sets-Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[7] K.H.Lee et.,, Software Engineering, pp.303~315, Cheongmoongak Press, Seoul, 1993.

[8] S.J.Lee, Adaptable Reusability Decision Model based on Rough theory, Catholic University of Taegu-Hyosung, Dep. of Computer Science and Statistic, Ph. d. dissertation, 1998.

[9] H.S. Jang, M.G.Park, "On the Quantitative Metrics of Software Reusability", The Transaction of the Korean Information Processing Society, Vol. 5, No 2, Feb., pp.176-184, 1995.

[10] S.J.Lee et., Rough Set and Application, Chosun University Publisher, KawngJu, 1998.