

# 러프 집합과 퍼지 집합에 기반한 프로그램 재사용 가능성도 측정

## Program Reuse Possibility Measurement Based on Rough and Fuzzy Sets

김혜경† · 김미경†† · 최완규 · 이성주

† 조선대학교 전자계산학과

†† 춘천기능대학 정보기술학과

Hye-Kyoung Kim† · Mi-Kyoung Kim†† · Wan-Kyoo Chio · Sung-Joo Lee

† Dept Computer Science, Chosun University

† † Dept Information Technology, Chunchon Poli-technical College

### ABSTRACT

소프트웨어의 유지보수면에서, 재사용이 매우 중요시되고 있는 가운데, 사용자가 최소 노력으로 필요한 컴포넌트들을 선택 및 유사 컴포넌트들을 평가할 수 있는 방법이 필요하다.

따라서, 본 논문에서는 현업에서의 많은 연구와 실험을 통해서 그 타당성이 검증된 척도들을 측정 인자로 설정, 여기에 러프 집합으로써 각 측정 인자들의 중요도를 측정하고, 이러한 측정값들을 Sugeno의 퍼지 적분으로써 종합하여 컴포넌트의 재사용 가능성을 평가하여 재사용이 용이한 컴포넌트순으로 사용자에게 제공할 수 있는 재사용 가능성도 평가 방법을 제안한다.

## I. 서론

프로그램 재사용 기술은 소프트웨어 공학의 한 분야로써 컴퓨터 프로그램을 개발하고, 유지·보수하는데 소요되는 노력과 비용에 영향을 주는 소프트웨어의 특징을 식별하고, 분류하며, 측정하는 분야로 많은 연구들이 수행되어 왔다[4].

소프트웨어 재사용에서 컴포넌트의 신뢰성 향상을 위해서 컴포넌트의 품질 보증에 관한 연구가 소프트웨어

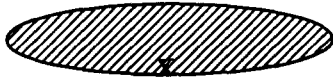
재사용 분야에서 우선적으로 이루어져야 한다. 컴포넌트의 품질 보증은 재사용 가능한 컴포넌트를 식별하여 사용자에게 제공하므로써, 컴포넌트 사용자들의 불신을 해소시키고 재사용율을 제고하여 소프트웨어의 생산성과 신뢰성을 향상시킨다.

그러나 현실적으로 재사용할 컴포넌트들이 개발자의 목적과 일치하는 컴포넌트이기 보다는 구현 사항만 다르

고 기능적으로 유사한 컴포넌트인 경우가 많다. 이 경우에 개발자는 최소 적용 노력이 소요되는 컴포넌트를 선택해야 한다. 컴포넌트들의 단순한 나열은 기존 컴포넌트에 대한 이해와 적용을 어렵게 하여 개발자로 하여금 기존 컴포넌트의 재사용을 단념시킬 수 있다. 그러므로, 본 연구에서는 이런 문제점을 해결하기 위해 현업에서의 연구와 경험을 통해서 증명된 객관성 있는 척도들을 측정인자로 설정하고, 러프집합을 이용하여 각 측정인자들의 중요도를 구하여 가중치로 하고[3], Sugeno의 퍼지적분을 이용하여 측정값들을 종합함으로써[11], 컴포넌트들의 재사용 가능도를 평가할 수 있는 새로운 척도를 제안한다.

## II. 러프 집합 과 퍼지 척도

1982년 Pawlak에 의해 제안된 러프 집합 이론은 애매한 범주들에 대한 새로운 수학적 접근 방법이다[6]. 러프 집합은 자동 분류, 패턴 인식, 학습 알고리즘 등 다양한 분야에 적용될 수 있다[7,10].



E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>
E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>8</sub>
E <sub>9</sub>	E <sub>10</sub>	E <sub>11</sub>	E <sub>12</sub>
E <sub>13</sub>	E <sub>14</sub>	E <sub>15</sub>	E <sub>16</sub>

그림 1. 근사공간 A

그림 1의 근사공간 A와 같이 주어진 지식을 이용하여 명확히 분류될 수 없는 대상들의 집합은 X의 상한 근사(upper approximation :  $R^*(X)$ ), 하한 근사(lower approximation :  $R_*(X)$ ), 상한과 하한 근사는 X의 긍정 영역(positive region : POSR(X)), 부정 영역(negative region : NEGR(X)), 경계 영역(borderline region : BNR(X)) 이라 불리는 러프 집합으로 다음과 같이 표현된다.

$$R^*(X) = U\{Y \in U/R : Y \cap X \neq \emptyset\}$$

$$R_*(X) = U\{Y \in U/R : Y \subseteq X\}$$

$$\begin{aligned} \text{POSR}(X) &= R_*X, & \text{NEGR}(X) &= U - R^*X \\ \text{BNR}(X) &= R^*X - R_*X \end{aligned}$$

그림 1에서 각 집합은 다음과 같다

$$U = \{E_1, E_2, \dots, E_{15}, E_{16}\}$$

$$R^*(X) = \{E_1, E_2, \dots, E_{11}, E_{12}\}$$

$$R_*(X) = \{E_6, E_7\}, \text{ POSR}(X) = \{E_6, E_7\}$$

$$\text{NEGR}(X) = \{E_{13}, E_{14}, E_{15}, E_{16}\}$$

$$\text{BNR}(X) = \{E_1, \dots, E_5, E_8, \dots, E_{12}\}$$

퍼지척도는 특정 원소가 여러 개의 집합 중에서 어떤 집합에 소속하는지에 대한 가능성이 애매할 때 이 소속 정도를 나타내는 척도이다. 특히 퍼지척도를 이용한 Sugeno의 퍼지 적분은 어떤 대상이 여러 항목에 대해서 평가가 되고 각 평가 항목의 중요도가 차이가 있을 때 이들 평가 값들을 종합하는데 유용하게 이용될 수 있다 [11].

## III. 프로그램 재사용 가능도 측정

### 1. 측정 매트릭스

재사용 가능 컴포넌트로 분류되는 컴포넌트들은 정확성, 효율성, 유지 보수성, 신뢰성 등이 있어야 하고 컴포넌트의 품질이 높아야 한다. 높은 품질의 컴포넌트는 크기가 작고, 구조가 간단하며, 문서화가 잘 되어있고, 동일한 언어로 작성되어야 한다[2].

따라서, 본 연구에서는 현업에서의 연구와 실험을 통해서 제시된 품질 분류 기준들의 척도인 LOC(lines of code), McCabe의 cyclomatic number, Halstead의 volume, difficulty, effort를 측정 인자로 설정하였다 [1][2][5][9]. 산업현장에서의 실험적인 연구들은 코드의 라인 수는 50-100라인 범위, Cyclomatic 복잡도는 10이하, Difficulty는 10이하, 프로그램 노력은 10000이하가 적합하다고 제시하였다[1,2,5,9]. 따라서 본 논문에서는 이들의 기준을 종합하여 프로그램 분류 기준을 표 1과 같이 설정하였다.

표 1. 소프트웨어 분류 기준

Domain Measure	0	1	2	3
LOC	<50	<100	<150	>150
Cyclomatic Number	<10	<20	<50	>50
Volume	<1000	<4000	<10000	>10000
Difficulty	<10	<25	<50	>100
Effort	<10000	<100000	<300000	>300000

표 1에 근거하여 각 측정 인자들의 퍼지 소속함수는 다음 식과 같은 S자형 함수로 정의한다.

$$\mu(x) = \begin{cases} 1 & , x \leq x_0 \\ 1 - 2 \left( \frac{x - x_0}{x_1 - x_0} \right)^2 & , x_0 < x \leq \frac{x_0 + x_1}{2} \\ 2 \left( \frac{x - x_0}{x_1 - x_0} \right)^2 & , \frac{x_0 + x_1}{2} < x \\ 0 & , x > x_1 \end{cases} \quad (1)$$

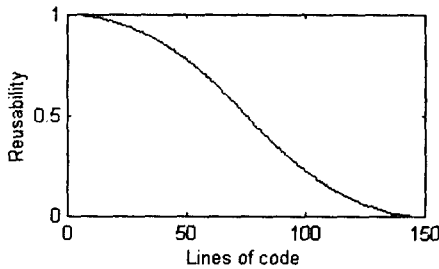


그림 2. LOC의 소속함수

## 2. 척도의 중요도

지식 표현 시스템에서 지식 C(조건 속성)의 모든 원소 범주들이 지식 D(의사결정 속성)의 원소 범주들로 정의될 수 있으면, 지식 D가 지식 C의 모든 특성을 제공하므로 지식 C는 지식 베이스에서 필요 없다. 이것은 지식간의 종속성으로 표현되며, 다음 식과 같다[1,10].

$$k = \gamma_C(D) = \frac{|POS_C(D)|}{|U|} \quad (2)$$

$|POS_C(D)|$  :  $POS_C(D)$ 의 원소의 개수

$|U|$  : 전체 집합의 원소의 개수

지식표현 시스템에서 주어진 모든 조건 속성들 중 특

정 속성들은 의사결정에 있어서 다른 속성들 보다 더 중요할 수 있다. 이것은 속성의 중요도로 표현되며, C의 부분 집합 C'의 중요도는 다음 식과 같다[7,10].

$$S_{C'} = \gamma_C(D) - \gamma_{C-C'}(D) \quad (3)$$

## 3. 프로그램 재사용 가능성도 측정

Sugeno의 퍼지적분은 어떤 대상을 여러 항목에 대해서 평가되고 각 평가 항목의 중요도에 차이가 있을 때, 이들 각 항목에 대한 평가 값을 퍼지척도를 사용하여 종합하는 방법으로 사용될 수 있으며[11], 주관적인 판단이 개입되는 평가 문제에서도 유용하게 이용될 수 있다.

특히, 집합 X가 유한 집합이고  $E \in P(X)$ 일 때, Sugeno의 퍼지적분은 다음과 같이 구한다.  $x_i \in X (i=1, 2, \dots, n)$ 에 대해서  $h(x_i) \leq h(x_{i+1})$  이라 하고,

$E_i = \{x_k \mid k=i, i+1, \dots, n\}$  이라 한다면, 평가함수 h의 상대적 중요도 함수 g에 대한 Sugeno의 퍼지적분은 다음과 같다.

$$\begin{aligned} Pcm(A) &= \int_X h(x) \cdot g(\cdot) \\ &= \text{Sup}_{E \in X} \text{Min} [ \text{Min} h(x), g(E) ] \end{aligned} \quad (4)$$

## IV. 실험 및 분석

본 논문에서는 C언어 Run time 라이브러리의 274개 모듈들을 대상으로 하여 프로그램 재사용 가능성을 측정하였다.

### 1. 상대적 중요도

대상 프로그램들을 표 1의 분류 기준에 따라 분류하여 표 2와 같이 지식표현 시스템을 구성하였다. 표 2에서  $C = \{LOC, Cyc, Vol, Dif, Eff\}$ 이고,  $D = \{reusability\}$ 이다. 이때 각 척도들의 상대적 중요도는 표 3과 같다.

표 2. 재사용 결정을 위한 의사 결정표

U	LOC	Cyc	Vol	Dif	Eff	reusability
1	0	0	0	0	0	1
2	2	0	0	0	0	1
3	0	0	0	1	0	1
4	1	0	0	2	1	1
5	2	0	1	2	1	1
6	2	0	1	2	2	1
7	1	0	0	0	0	1
8	1	0	0	1	0	1
9	0	0	0	1	1	1
10	0	0	0	2	1	1
11	2	0	0	2	1	1
12	0	0	0	2	0	1
13	2	1	1	2	1	1
14	1	0	0	1	1	1
15	2	0	0	1	0	1
16	2	2	1	2	2	1
17	1	1	0	2	0	1
18	1	0	1	2	1	1
19	1	1	1	2	1	1

\* LOC=Lines of Code, Cyc=Cyclomatic number,  
Vol=Volume, Dif=Difficulty, Eff=Effort

표 3. 각 척도들의 상대적 중요도

척도	중요도
Lines of code	0.34091
Cyclomatic number	0.13636
Volume	0.09091
Difficulty	0.25000
Effort	0.18182

## 2. 실험 결과

표 1에서 제시하고 있는 기준 (30, 10, 1000, 10, 10000)에 대한 프로그램 재사용 가능성은 0.82000 이고, 기준 (50, 20, 4000, 25, 100000)에 대한 프로그램 재사용 가능성은 0.65909이다.

다수의 프로그램들을 실험한 결과 측정인자들과 재사용 가능성과의 관계 중 프로그램 재사용 가능성의 분포는 그림 3과 같다.

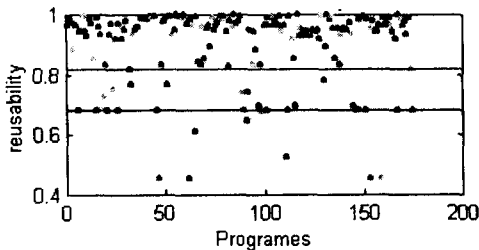


그림 3. 프로그램 재사용 가능성의 분포

위 실험결과에서 보듯이, 재사용 가능성이 높은 프로그램은 기준값보다 측정값들이 작았다.

또한 네 개의 정렬 모듈들에 대한 재사용 가능성을 구

현 사항만 다르고 기능적으로 유사한 네 개의 정렬 모듈에 대한 재사용 가능성을 측정하였다.

네 개의 모듈에 대한 측정값들은 표 4에 주어진다.

표 4. 각 모듈에 대한 측정값

입력값 모듈명	LOC	CYC	VOL	DIF	EFF
A	21	4.00000	271.04513	22.50000	6098.51531
B	37	9.00000	641.89475	75.00000	48142.10626
C	27	5.00000	562.46053	26.86667	15111.43946
D	26	7.00000	450.40259	39.00000	17565.70115

각 모듈에 대해 퍼지 소속함수를 적용한 결과 각 모듈에 대한 소속등급은 표 5와 같다.

표 5. 각 모듈에 대한 소속등급

입력값 모듈명	LOC	CYC	VOL	DIF	EFF
A	0.91180	0.98720	0.99853	0.89875	0.99917
B	0.72620	0.93520	0.99367	0.12500	0.94850
C	0.85420	0.98000	0.99367	0.85564	0.99493
D	0.86480	0.96080	0.99594	0.69580	0.99314

표 4와 표 5의 측정치를 Sugeno의 퍼지적분에 적용한 결과치, 즉 재사용 가능성은 표 6과 같다.

표 6. 각 모듈에 대한 재사용 가능성

모듈명	reusability
A	0.89875
B	0.72620
C	0.85420
D	0.75000

만약, 위와 같은 재사용에 관한 정보가 없다면, 개발자는 적합한 모듈을 찾기 위해서 모든 모듈을 검색해야 할 것이다. 이 경우에, 재사용 가능성은 표 7과 같다. 그러나 표 6의 정보를 이용한다면, 개발자는 최소의 노력으로 모듈 A를 곧바로 이용할 수 있기 때문에 개발자는 더 이상 다른 모듈을 조사할 필요가 없어진다. 이는 기존 컴포넌트들을 재사용하는 데 소요되는 노력을 상당히 감소시키게 되어 소프트웨어 개발의 효율 증대를 기대할 수 있게 한다.

표 7. 전체 재사용 가능성도

LOC	CYC	VOL	DIF	EFF	전체 재사용가능도
111	25	1925.803	163.3667	86917.76218	0.22727

## V. 결론

따라서, 본 논문에서는 컴포넌트의 재사용 가능성을 평가하여 재사용이 용이한 컴포넌트순으로 사용자에게 제공할 수 있는 새로운 재사용 가능성도 평가 방법을 제안하였다.

실험 결과 제안된 척도는 다음과 같은 장점이 있음을 알 수 있었다. 첫째, 제안된 측정 방법은 기존의 단순한 측정 요소를 고려한 척도들보다는 일차적으로 검증된 척도들을 종합하여 평가하므로 전체적으로 프로그램과의 상관관계가 높다. 둘째, 각 측정인자들의 분포와 프로그램들의 재사용율 분포가 대응됨을 알 수 있다. 셋째, 재사용에 관한 순서 정보를 제공할 수 있다.

## 참고 문헌

- [1] L.J.Arthur, "Measuring Programmer Productivity and Software Quality", John Wiley & Sons, New York, pp.138-142, 1985.
- [2] Caldiera, G. and V.R. Basili, "Identifying and Qualifying Reusable Software Components", IEEE Computer, Feb, pp.61-70, 1991.
- [3] W.K.Choi, et.al, " The Analysis of Significance of the Reusability Decision Metrics using Rough Set", The Third AFFS'98, pp.302-307, 1998.
- [4] Lesiie J. Waguespack and Sunil Baldani, "Software Complexity Assessment : An Introduction and Annotated Bibliography", ACM SIGSOFT, Sftware Eng. Notes, Vol.12, No.4, pp.52, Oct., 1987.
- [5] Lewis John, Henry Salie, "A Methodology for Integrating Maintainability Using Software Metrics", Proceedings: Conference on Software Maintenance, Miami, Florida, IEEE, Oct 16-19, pp.32-39, 1989.

- [6] Pawlak Z., "Rough sets", International Journal of Computer and Information Sciences, 11, pp.341-356, 1982.
- [7] Pawlak Z., *Rough Sets-Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.
- [8] Warren Harrison et.al, "Applying Software Complexity Metricc to Prograam Maintenance", Computer Journal, Vol.15, pp.65-79, Sep., 1982.
- [9] Horst Zuse "Software Complexity-measures and methods", Walter de Gruyter, Berlin · New York, pp.25-37, 1991.
- [10] 이성주 외, "러프 집합과 응용", 조선대학교출판국, 1998.
- [11] 이형광, 오길록, "퍼지이론 및 응용", 홍릉과학출판사, pp. 1.1-1.40, 9.1-9.36, 1991.