

객체지향 시스템에서 간접 의존성을 포함한 결합도 메트릭

유 문 성*

Coupling Metrics Including Indirect Dependency for Object-Oriented Systems

Yoo, Moon Sung

〈Abstract〉

Nowadays software developers are moving from conventional software process technologies to the object-oriented paradigm. To develop the object-oriented softwares efficiently, various software metrics have been suggested. Coupling refers to the degree of independence between components of the system. It has long been well known that good software practice calls for minimizing coupling interaction. Many researches have been studied coupling metrics of the object-oriented systems. We review Chidamber and Kemerer's work & Li's work.

In this paper, we study the coupling of the overall structures of object-oriented systems by analyzing the class diagram of UML. We propose four coupling metrics for object-oriented softwares. First, we use an established coupling metric for object-oriented systems as a basic coupling metric. Then we modify the basic coupling metric by including indirect coupling between classes. We also suggest two relative coupling metrics to measure coupling between subsystems. We investigate the theoretical soundness of the proposed metrics by the axioms of Briand et al. Finally, we apply the presented metrics to a practical case study.

This coupling metric will be helpful to the software developers for their designing tasks by evaluating the coupling metric of the structures of object-oriented system and redesigning tasks of the system.

Key Words : Object-oriented Systems, Object-oriented Softwares, Coupling, Software Metrics, UML

I. 서론

객체지향 시스템은 전통적인 절차적 언어를 사용하여 소프트웨어를 개발하는 것 보다 소프트웨어의 재사용성 등을 높였기[1] 때문에 현재 개발 되고 있는 소프트웨어

는 거의 객체지향 시스템을 사용하고 있다. 객체지향 소프트웨어를 좀 더 효율적으로 설계하고 품질을 높이기 위하여 여러 가지 소프트웨어 품질 메트릭이 제안되었다 [2-12]. 그 중 결합도는 클래스 간의 의존성 정도를 측정하는 메트릭이다. 클래스의 재사용은 결합도가 낮을수록 재사용되기가 용이하므로 시스템을 되도록 결합도가 낮게 구축하는 것이 필요하다.

* 상지대학교 컴퓨터정보공학부 교수(교신저자)

본 연구에서는 객체지향 시스템의 결합도를 측정하기 위하여 객체 지향 분석 및 설계를 위한 표준적 표기법인 UML(Unified Modeling Language)을 사용하여 시스템을 분석한다. UML 중에서 시스템의 정적인 구조 설계를 하는데 사용되는 클래스 다이어그램[13]을 분석하여 객체지향 시스템의 결합도를 여러 가지 방법으로 측정한다. 설계 단계에서 결합도를 측정함으로써 개발 중인 시스템을 재평가하고 필요시 재설계를 함으로써 구현의 오류를 줄이고 개발 비용을 줄일 수 있을 뿐만 아니라 소프트웨어의 유지보수, 리팩토링, 시스템 분해 등에 효율적으로 사용될 수 있다.

본 논문에서는 먼저 객체지향 시스템에서 많이 적용하는 기존의 결합도 메트릭을 기본적 메트릭으로 사용하고 이를 이용하여 간접적으로 의존하는 클래스의 결합도를 반영한 메트릭과 상대적 비교를 할 수 있는 메트릭을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 기존의 객체 지향 시스템과 컴포넌트 기반 소프트웨어의 결합도 메트릭을 소개한다. 3장에서는 객체 지향 시스템 관련 새로운 결합도 메트릭을 제안한다. 4장에서는 3장에서 제안한 결합도 메트릭이 이론적으로 바람직한 것인가를 Briand 등이 정의한 공리에 적용하여 조사한다. 5장에서는 제안한 클래스 간의 결합 메트릭 적용 사례를 제시하고 6장에서는 결론을 맺는다.

II. 관련연구

객체지향 시스템 품질 메트릭은 객체지향 시스템의 특성인 상속, 정보은닉, 캡슐화, 객체 추상화 등과 같은 특성을 반영해야 한다. 객체지향 시스템이나 컴포넌트 기반 시스템을 대상으로 한 여러 가지 소프트웨어 품질 메트릭이 제안되었다[2-12]. 그 중 대표적인 Chidamber and Kemerer의 메트릭(C&K 메트릭)과 Li의 메트릭은 다음과 같다.

S. R. Chidamber와 C. F. Kemerer는 여섯개의 메트릭을 제안하였으며, 그것들은 클래스당 가중치 메소드 값(WMC), 상속 트리의 깊이(DIT), 자식 클래스의 수(NOC), 객체간의 결합도(CBO), 클래스에 대한 응답도(RFC), 메소드들의 응집도(LCOM) 이다[2].

Li도 객체지향 시스템에 관한 여섯 개의 메트릭을 제안하고 있다. 클래스 상속 구조의 복잡도를 측정하는 NAC(Number of Ancestor Classes)와 NDC(Number of Descendent Classes), 클래스의 복잡도에 관한 NLM(Number of Local Methods)과 CMC(Class Method Complexity), 클래스 간의 결합도를 측정하는 CTA(Coupling Through ADT)와 CTM(Coupling Through Message passing)을 제안하였다[5].

이들 메트릭중 결합도와 관련되어 널리 사용되는 메트릭은 CBO(Coupling Between Objects)와 RFC (Response For a Class)이다. CBO(c)는 클래스 c에서 결합되어 있는 다른 클래스들의 수로 정의된다. 결합되어 있다는 것은 하나의 클래스가 다른 클래스의 메소드나 속성을 사용함을 말한다. RFC(c)는 클래스 c에서 있는 메소드 개수와 클래스 c에서 의존하는 다른 클래스의 메소드의 개수의 합이다.

기존의 객체지향 시스템에 관한 결합도에는 두 가지 문제점이 있다. 첫째, 기존의 결합도는 직접 의존성만 반영하여 클래스와 클래스에 직접 관련이 안 되어있더라도 관련성이 있는 클래스간의 간접 의존성을 무시하여 정확한 결합도를 측정하지 못하였다.

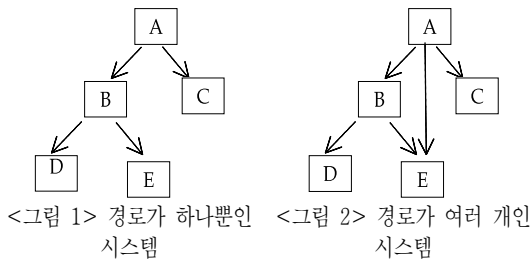
둘째, 기존 결합도는 절대적인 값 즉 클래스나 메소드의 개수 등 단순히 구성 요소의 크기만을 나타내므로 시스템간의 결합도를 상대적으로 비교할 수 없었다. 시스템이 클 경우는 일반적으로 절대적인 결합도의 값이 커지는데 그 시스템이 보다 작은 시스템에서 작은 절대적인 결합도의 값을 갖는 시스템보다 결합도가 높다고 말할 수 없다. 시스템간 결합도 비교를 하기 위해서는 절대적인 비교가 아닌 상대적인 비교가 필요하다.

III. 제안하는 객체지향 소프트웨어의 결합도

3.1 CBO와 간접 의존성을 포함한 CBOM

객체지향 시스템의 대표적인 결합도인 CBO를 기본으로 하여 몇 가지 수정된 결합도를 제안한다. CBO(c)는 클래스 c에서 직접 의존하는 클래스의 개수이다. 그런데 클래스 사이에는 직접 의존하는 것 이외에 간접 의존할 수도 있다. 즉 클래스 A가 클래스 B를 의존하고 클래스 B가 클래스 C를 의존하면 클래스 A는 클래스 C를 간접 의존하게 된다. 이 경우 클래스 A와 클래스 C 사이에는 결합력이 있으므로 이런 간접 의존에 대한 것도 고려하여 클래스 A의 결합도를 계산하여야 더 정확한 결합도를 계산할 수 있다.

예를 들어 [그림1]과 같이 클래스의 의존성을 나타낸 클래스 다이어그램으로 표시할 수 있는 객체지향 시스템을 가정하자. 그림에서 사각형은 클래스를 나타내고 연결선은 클래스 의존을 의미한다. CBO(A)는 A에서 직접 의존하는 B와 클래스 C를 계산하여 2가 되지만 B에서 직접 의존하는 D와 클래스 E도 클래스 A가 간접 의존하므로 그 결합도도 포함하여야 한다. 간접 의존하는 결합도는 의존하는 클래스와 의존되는 클래스가 멀리 떨어져 있을수록 작은 값을 가져야 하므로 두 클래스 사이의 경로의 길이가 d라고 할 때 $(\frac{1}{2})^{d-1}$ 로 한다. 만일 두 클래스 사이의 경로가 두 개 이상 존재하면 최단 경로(shortest path)를 두 클래스 사이의 거리 d로 한다.



CBO(c)에 간접 의존에 대한 것도 고려하여 CBOM(c)를 제안한다. CBOM(c)는 c에서 간접 의존하는 클래스가 있으면 c에서 그 클래스 대한 최단 경로(shortest path)를 찾아 그 길이 d를 계산한다. c에서 길이가 d인 클래스의 개수를 n(d), d의 최대값을 max라고 하면

$$CBOM(c) = \sum_{d=1}^{\max} (\frac{1}{2})^{d-1} \cdot n(d) \text{ 라고 정의한다. 그리고}$$

시스템에 있는 모든 클래스에 대한 합계를 그 시스템에 대한 CBO, CBOM로 나타낸다.

$$CBO = \sum_{c \in C} CBO(c), \quad CBOM = \sum_{c \in C} CBOM(c)$$

[그림1]에서 CBO(A)=2, CBO=4, CBOM(A)=3, CBOM=5 이다. 한 클래스에서 다른 클래스로의 경로가 2개 이상인 경우의 예로서 [그림2]로 표시한 시스템에서는 CBO(A)=3, CBO=5, CBOM(A)=3.5, CBOM=5.5이다. A에서 D의 의존 경로가 2개 있는데 경로의 길이가 작은 경로를 선택한다.

3.2 상대적인 결합도, RCBO와 RCBOM

절대적인 결합도만으로는 시스템간의 결합도를 상대적으로 비교할 수 가 없기 때문에 시스템의 크기와 관계 없이 여러 시스템의 결합도를 비교하기 위해서는 상대적인 결합도가 필요하다. 클래스의 개수가 |C|인 시스템에서는 클래스끼리 의존할 수 있는 최대의 수는 $|C|^2 - |C|$ 이다. CBO와 CBOM에 대한 상대적인 결합도를 구하기 위해서는 RCBO와 RCBOM의 값을 $|C|^2 - |C|$ 로 나누면 된다. 이렇게 되면 그 값은 0에서 1의 값을 갖게 되고 결합도의 상대적인 비교가 가능하게 된다. 다음과 같이 CBO와 CBOM의 상대적인 결합도 RCBO와 RCBOM을 정의한다.

$$RCBO = \frac{\sum_{c \in C} CBO(c)}{|C|^2 - |C|}$$

$$RCBOM = \frac{\sum_{c \in C} CBOM(c)}{|C|^2 - |C|}$$

[그림1]에서 $RCBO = \frac{4}{20}$, $RCBOM = \frac{5}{20}$ 이고

[그림2]에서 $RCBO = \frac{5}{20}$, $RCBOM = \frac{5.5}{20}$ 이다.

IV. 제안한 메트릭의 이론적 검증

메트릭에 대한 이론적 검증은 Briand 등이 제안한 엄격한 수학적 공리를 사용하였다[4]. 이 공리는 복잡도 (complexity), 응집도(cohesion), 결합도(coupling) 등 여러 가지 소프트웨어 메트릭을 정의할 때 바람직한 기준을 제공한다. Briand 등이 원래 정의한 것은 모듈 시스템에서의 메트릭 성질에 관한 것인데 객체지향 시스템에서는 클래스가 모듈의 역할을 한다. 아래에 5가지로 기술한 Briand가 정의한 결합도가 가져야 할 바람직한 성질을 가지고 본 연구의 객체지향 시스템에 대한 결합도 메트릭의 이론적 타당성을 검증하였다.

coupling 1: Nonnegativity

CBO와 CBOM은 0이상이고 |C|는 양수이므로 CBO와 CBOM, RCBO와 RCBOM은 모두 음수가 될 수 없다.

coupling 2: Null Values

만약 클래스들 간에 의존이 없다면 CBO와 CBOM은 0이다. 그러면 상대적인 결합 메트릭 RCBO와 RCBOM도 0이 된다.

coupling 3: Monotonicity

클래스 내부 구성 요소에는 변화 없이 클래스들 사이의 상호 작용만 증가한다면 클래스 간에 의존수가 증가하는 것이므로 CBO와 CBOM도 증가한다.

$$RCBO = \frac{\sum_{c \in C} CBO(c)}{|C|^2 - |C|} \text{ 과 } RCBOM = \frac{\sum_{c \in C} CBOM(c)}{|C|^2 - |C|} \text{ 도}$$

모두 분모는 변하지 않는데 분자만 증가하므로 RCBO와

RCBOM도 증가한다.

coupling 4: Merging of Services

두 클래스 C1과 C2에 대하여, 두 클래스를 합한 새로운 클래스를 C 라고 하자. 그러면 다음과 같은 수식이 성립한다.

$$|CBO(C)| = |CBO(C1)| + |CBO(C2)| - |CBO(C1 \cap C2)| \leq |CBO(C1)| + |CBO(C2)|$$

CBOM도 마찬가지로 이 성질을 만족한다.

그러나 RCBO이나 RCBOM에 대해서는 |C|의 값이 |C1|+|C2|보다 1 감소하여 새로운 클래스의 분자, 분모 값이 모두 감소하므로 이 성질을 항상 만족하지는 않는다.

coupling 5: Disjoint Service Additivity

두 클래스 C1과 C2에 대하여, 두 클래스를 합한 새로운 클래스를 C 라고 하자. 그러면 다음과 같은 수식이 성립한다.

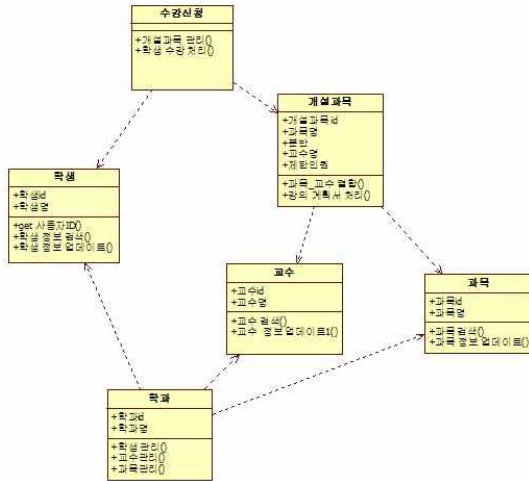
$$|CBO(C)| = |CBO(C1)| + |CBO(C2)| - |CBO(C1 \cap C2)|$$

그런데 $|CBO(C1 \cap C2)| = 0$ 이므로 $|CBO(C)| = |CBO(C1)| + |CBO(C2)|$ 이 성립한다. CBOM도 마찬가지로 이 성질을 만족한다.

그러나 RCBO이나 RCBOM에 대해서는 |C|의 값이 |C1|+|C2|보다 1 감소하므로 이 성질을 만족하지 않는다.

요약하면 Briand 등의 메트릭에 대한 공리를 본 논문에서 제안한 메트릭에 대해 조사하였더니 의존 수만 계산하는 CBO와 CBOM은 모든 조건을 만족하였지만 상대적 결합도를 측정하는 RCBO와 RCBOM은 coupling 4와 coupling 5를 만족하지 못하였다. Briand 등의 연구에서도 밝혔듯이 바람직한 메트릭이 모든 이론적 성질을 꼭 만족해야만 하는것은 아니다[14].

V. 사례연구



<그림 3> 수강 신청 시스템의 클래스 다이어그램 (의존도 표시)

<표 1> 서비스, 기호 정의표

클래스명	수강 신청	학생	개설 과목	교수	과목	학과
기호	A	B	C	D	E	F

제안한 결합도 메트릭을 수강 신청 시스템에 적용하였다. 이 시스템의 클래스 다이어그램은 <그림 3>에 나타내었는데 클래스 사이의 의존도를 표시하였다. 각 클래스에 대하여 결합도 계산시 편리를 위하여 클래스마다 기호를 <표 1>과 같이 정의하였다.

클래스A에 대한 각 결합도 메트릭을 다음과 같이 계산한다. A에서 직접 의존하는 클래스는 B, C 이므로 CBO(A)=2이다. RCBOM(A)를 구하기 위해서 클래스 A에서 간접 의존하는 클래스를 찾아야 한다. A에서 최단 경로가 2인 간접 의존 클래스는 C에서 의존하는 D, E가 있다. 그러므로 RCBOM(A)= 2+0.5×2=3이 된다. 다른 클래스에 대한 RCBO와 RCBOM 값도 동일한 방식으로 구하고 모든 클래스에 대한 합계가 RCBO와 RCBOM가 된

다. 클래스 개수가 모두 6개 이므로

$$RCBO = \frac{\sum_{c \in C} CBO(c)}{6^2 - 6}, \quad RCBOM = \frac{\sum_{c \in C} CBOM(c)}{6^2 - 6} \text{ 이고}$$

이들의 값을 계산한 결과를 <표 2>에 제시하였다.

<표 2> 수강 신청 시스템에 대한 결합도 메트릭 값

메트릭	CBO	CBOM	RCBO	RCBOM
값	7	8	0.233	0.267

결과를 보면 직접 의존성만 고려한 결합도 CBO보다 간접 의존성 까지 고려한 CBOM은 당연히 더 큰 값을 가지며 상대적인 결합도의 값은 0과 1사이의 값을 가진다. 상대적인 결합도를 다른 시스템이나 현 시스템을 수정할 경우 수정한 시스템의 결합도와 비교하여 어느 시스템이 결합도 측면에서 더 바람직한가를 판단할 수 있다.

VI. 결론

본 논문은 객체지향 시스템에 대한 새로운 결합도 메트릭을 제안하였다. 기존 객체지향 기반의 결합도 메트릭에서 가장 많이 사용하는 CBO를 기본으로 직접 의존 외에 간접 의존까지 고려한 결합도 CBOM을 제안하였다. 또한 CBO와 CBOM은 서브 시스템간 결합도의 상대적인 평가를 할 수 없으므로 상대적인 결합도 메트릭인 RCBO와 RCBOM을 제안하였다.

본 논문에서 제안한 결합 메트릭이 이론적으로 바람직하다는 것을 검토하기 위하여 Briand등이 제안한 결합도 성질에 따라 조사하였고 제안한 결합 메트릭의 실용성을 검증하기 위하여 적용 사례를 제시하였다. 새로운 메트릭의 제안으로 객체지향 시스템의 결합도를 간접 의존성까지 반영하여 보다 정확하게 측정할 수 있게 되었으며 시스템간의 결합도를 상대적으로 평가할 수 있게 되었다.

앞으로의 연구과제는 이 메트릭을 사용하여 객체지향

시스템 소프트웨어의 유지보수, 리팩토링, 시스템 분해 등에 효율적으로 사용될 수 있음을 실제 사례를 통하여 성능을 실증적으로 검증하는 것이다.

참고문헌

- [1] M. Weisfeld, The object-oriented thought process, 3rd ed., Addison-Wesley, 2008.
- [2] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object-oriented design," IEEE Transactions on Software Engineering, Vol. 20, No. 6, 1994, pp. 476- 493.
- [3] N. E. Fenton and S. L. Pfleeger, Software metrics: a rigorous and practical approach, International Thomson Computer Press, London, 1996.
- [4] L. C. Briand, S. Morasca, and V. R. Basili, "Property-based software engineering measurement," IEEE Transactions on Software Engineering, Vol. 22, No. 1, 1996, pp. 68-86.
- [5] W. Li, "Another metric suite for object-oriented programming," Journal of Systems and Software, Vol. 44, No. 2, 1998, pp. 155-162.
- [6] K. El-Emam, "Object-oriented metrics: A review of theory and practice," Advances in software engineering, Springer-Verlag, 2002, pp. 23-50.
- [7] G. Gui and P. Scott, "Coupling and cohesion measures for evaluation of component reusability," Proceedings of the 2006 international workshop on Mining software repositories, 2006, pp. 18-21.
- [8] H. Washizaki, T. Nakagawa, Y. Saito, and Y. Fukazawa, "A coupling-based complexity metric for remote component-based software systems toward maintainability estimation," IEEE Trans. Softw. Eng, Vol. 14, No. 9, 1988, pp. 1357-1365.
- [9] 박성희, 홍의석, 우치수, 김태균, "객체 지향 프로그램에서 응집도, 결합도 측정 메트릭 집합," 정보과학회논문지(B), 제. 25권, 제12호, 1998, pp. 1779-1787.
- [10] 이종석, 우치수, "객체 지향 시스템에서의 클래스 응집도와 결합도 메트릭," 정보과학회논문지 : 소프트웨어 및 응용, 제27권, 제6호, 2000, pp. 595-606.
- [11] 최미숙, 이종석, 이서정, "효율적인 시스템 설계를 위한 클래스 간의 결합 척도," 인터넷정보학회논문지, 제9권, 제5호, 2008, pp. 85-97.
- [12] 화지민, 이숙희, 권용래, "객체 지향 시스템에서의 클래스 간 의존성 강도 측정을 위한 커플링 척도," 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 제14권, 제1호, 2008, pp. 81-85.
- [13] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley Professional, 2005.
- [14] L. C. Briand and J. Wüst, "Empirical studies of quality models in object-oriented systems," Advances in Computers, Vol. 56, 2002, pp. 97-166.

■ 저자소개 ■



유 문 성
Yoo, Moon Sung

2000년 9월~현재
상지대학교 컴퓨터정보공학부 교수

관심분야 : 소프트웨어공학, 웹소프트웨어 개발
모바일 소프트웨어

E-mail : msyoo@sangji.ac.kr

논문접수일 : 2011년 8 월 10 일
수 정 일 : 2011년 9 월 2 일
계재확정일 : 2011년 9 월 6 일