

페이지랭크 알고리즘을 이용한 결합도 척도의 개선

박철현, 류성태, 이은석
 성균관대학교 정보통신공학부
 pch851130@skku.edu, xenz@skku.edu, leees@skku.edu

Using PageRank Algorithm to Improve Coupling Metrics

Cheol-Hyun Park, Sung-Tae Ryu, Eun-Seok Lee
 Dept. of Electrical and Computer Engineering, Sungkyunkwan University

요 약

소프트웨어 품질 측정은 소프트웨어 공학의 필수적인 요소이다. 소프트웨어 품질 척도 중 하나인 결합도는 모듈간의 얼마나 강하게 연결되어있는지를 나타낸다. 결합도는 소프트웨어의 결합-경향성, 모듈화, 재사용성, 변경-경향성 등 다양한 목적으로 사용된다. 기존의 결합도 척도들은 메소드 호출 횟수에 의해서 결정되는데, 이는 메소드의 가중치를 고려하지 않기 때문에 결합도를 정확히 측정 하지 못한다. 본 논문은 페이지랭크 알고리즘을 이용하여 메소드의 가중치를 측정하고, 이를 이용한 결합도 척도 개선 방법에 대해 제안한다. 본 논문의 유효성을 검증하기 위하여, 4 개의 오픈 소스 프로젝트를 대상으로 기존의 방법과 개선된 방법으로 결합도 척도 3 개를 측정하였다. 개선된 결합도 3 개는 유지보수의 척도로 사용되는 변경-경향성(Change-Proneness)과의 상관관계가 기존의 결합도 척도에 비하여 눈의 띄게 향상되었다. 따라서 개선된 결합도 척도는 소프트웨어 품질을 더 정확하게 측정할 수 있다.

1. 서론

소프트웨어 품질 측정은 소프트웨어 공학의 필수적인 요소가 되었다. 특히 소프트웨어 품질 척도 중에서 결합도는 소프트웨어 모듈간의 얼마나 강하게 연결되어있는지를 나타낸다[1]. 결합도는 소프트웨어의 결합-경향성, 모듈화, 재사용성, 변경-경향성 등 다양한 목적으로 사용된다. 본 논문에서는 결합도 척도 개선 방법에 대해서 제안한다.

기존의 결합도 측정은 주로 메소드 호출 횟수에 의해서 결정되었다[2]. 이러한 결합도 측정은 메소드의 가중치를 고려하지 않고 모든 메소드가 동일한 가중치를 가진다는 가정하에 측정한다. 그러나 소스 코드에는 다른 메소드들보다 상대적으로 중요한 메소드가 존재하기 때문에 메소드의 가중치를 고려하여 결합도를 측정하는 것이 더 정확하다고 할 수 있다.

본 논문에서는 페이지랭크 알고리즘을 이용하여 메소드의 가중치를 측정하고, 이를 이용하여 결합도 척도를 개선하는 방법을 제안한다[3]. 본 논문의 유효성을 검증하기 위하여, 4 개의 오픈 소스 프로젝트에 적용하였다. 그리고 기존의 방법과 개선된 방법으로 결합도 척도 3 개를 측정하였다. 개선된 결합도 3 개는 유지보수의 척도로 사용되는 변경-경향성(Change-Proneness)과의 상관관계가 기존의 결합도 척도에 비하여 눈의 띄게 향상되었다[4]. 본 연구 결과를 통해 개선된 결합도가 기존의 결합도보다 더 정확하게 소프트웨어 품질 측정이 가능함을 확인하였다.

본 논문은 다음과 같이 구성된다. 2 장에서는 결합도 관련 연구들을 소개하고, 3 장에서는 본 논문이 제

안하는 메소드의 가중치 측정과, 결합도 척도 개선 방법에 대해 설명한다. 4 장에서는 적용 사례를 통하여 제안한 방법의 유효성을 확인하였고, 5 장에서는 결론을 내린다.

2. 관련 연구

본 장에서는 기존의 결합도 척도를 소개하고, 메소드 가중치를 측정하기 위해서 사용한 페이지랭크 알고리즘에 대해서 설명한다

1) 결합도 척도

기존의 결합도는 소프트웨어의 구조적인 관계를 고려한 구조적 척도가 주를 이룬다.

CBO 는 클래스간의 호출관계를 이용하여 결합도를 측정한다[5][6]. RFC 는 클래스가 가진 메소드 개수와 호출하는 메소드 개수를 이용하여 결합도를 측정한다[5][6]. MPC 는 클래스가 호출하는 메소드 개수를 이용하여 결합도를 측정한다[7]. DAC 는 클래스 내에서 변수로 정의되는 클래스의 개수를 이용하여 결합도를 측정한다[7]. COF 는 클래스 간의 상속관계와 호출관계를 이용하여 결합도를 측정한다[8].

본 논문에서는 3 개의 척도를 개선하였다. 3 개의 척도(CBO, MPC, RFC)를 선택한 이유는, 메소드, 클래스 호출 횟수에 직접적으로 의존적인 척도로서, 가중치를 적용하기에 알맞기 때문이다.

2) 페이지랭크 알고리즘

페이지랭크(PageRank)는 월드 와이드 웹(World Wide Web)과 같은 하이퍼링크 구조를 가지는 문서에 상대적 중요도에 따라 가중치를 부여하는 방법이다[3].

이 알고리즘은 서로간에 인용과 참조로 연결된 임의의 묶음에 적용할 수 있기 때문에 메소드 호출 그래프에 적용하여 메소드 가중치를 측정하였다. 수식 1은 페이지랭크 알고리즘 계산 방법을 보여준다.

$$PR(A) = (1 - d) + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn)) \quad (1)$$

$d = 0.85$

T_i : 페이지 A를 참조하는 페이지

$C(T_i)$: 페이지 T_i 의 진출차수(outdegree)

그림 1은 페이지랭크 계산을 위해 임의로 생성한 웹 페이지 링크 그래프이다. 노드(웹 페이지), 에지(웹 페이지간의 링크 관계)로 구성되어 있다.

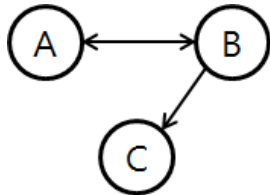


그림 1 웹 페이지 링크 그래프

각 페이지의 페이지랭크 초기값은 0으로 할당된다. 수식 1을 사용하여 페이지랭크 연산을 반복해서 수행하다가 페이지랭크 값이 안정상태가 되면 종료한다.

표 1 각 단계별 페이지랭크 연산 결과

iterate	PR(A)	PR(B)	PR(C)
1	0.15	0.28	0.27
2	0.27	0.38	0.31
3	0.31	0.41	0.33
4	0.33	0.43	0.33
5	0.33	0.43	0.33

3. 결합도 척도의 개선

본 장에서는 페이지랭크 알고리즘을 이용하여 메소드의 가중치를 측정하고, 이를 이용한 3개의 결합도 척도 개선 방법에 대해 제안한다.

그림 2은 결합도 척도 개선 프로세스를 보여준다. 메소드 호출 그래프에 페이지랭크 알고리즘을 적용하여 메소드의 가중치를 측정한다. 측정된 메소드의 가중치를 이용하여 결합도 척도를 개선할 수 있다.

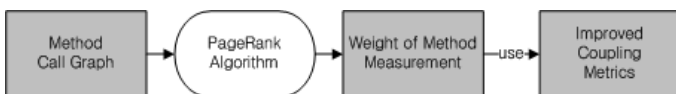


그림 2 결합도 척도 개선 프로세스

1) 메소드의 가중치 측정

그림 3은 결합도 측정을 위해 임의로 생성한 메소드 호출 그래프이다. 노드(메소드), 에지(메소드간의 호출 관계)로 구성되어 있다.

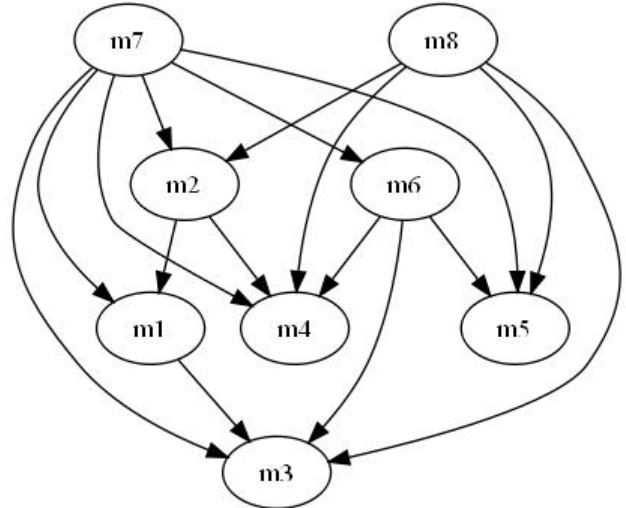


그림 3 메소드 호출 그래프

그림 3 메소드 호출 그래프에 2.2장과 같이 페이지랭크 알고리즘을 적용한다. 표 2는 페이지랭크 알고리즘을 적용하여 계산한 메소드 8개의 가중치를 보여준다. 이와 같이 페이지랭크 알고리즘을 이용하여 메소드의 가중치를 측정하는 것이 가능하다.

표 2 메소드 가중치

PR(m ₁)	PR(m ₂)	PR(m ₃)	PR(m ₄)
0.26	0.20	0.47	0.34
PR(m ₅)	PR(m ₆)	PR(m ₇)	PR(m ₈)
0.25	0.17	0.15	0.15

2) 결합도 척도의 개선

본 장에서는 기존의 결합도 척도와 개선된 결합도 척도를 수식으로 기술한다. 각 척도에 접미사로 N이 추가되면, 개선된 결합도 척도를 의미한다.

CBO는 클래스 개수에 클래스 가중치를 곱하여서, 더 중요한 클래스를 많이 호출할수록 결합도가 높아지도록 개선되었다. MPC, RFC는 메소드 개수에 메소드 가중치를 곱하여서, 더 중요한 메소드를 많이 호출할수록 결합도가 높아지도록 개선되었다.

정의 1. CBO

$$CBO(C) = \sum_{C' \in SIC(C)} 1 \quad (2)$$

$$CBO_N(C) = \sum_{C' \in SIC(C)} W(C') \quad (3)$$

SIC(C): 클래스 C가 호출하는 클래스들의 집합
W(C): 클래스 C의 가중치

표 3 오픈 소스 프로젝트

프로젝트	사용된 버전	설명	클래스	라인	변경된 라인
Robocode[9]	1.3 버전→1.7 버전	프로그래밍 게임	166	39991	3659
JHotDraw[10]	7.1 버전→7.5 버전	자바 GUI 프레임워크	150	29667	3408
JabRef[11]	1.0 버전→2.7 버전	문헌 정보 관리 프로그램	75	29182	1232
Heritrix[12]	1.0 버전→1.4 버전	웹 크롤러	196	58288	3712

정의 2. MPC

$$MPC(C) = \sum_{m' \in SIM(C)} 1 \quad (4)$$

$$MPC_N(C) = \sum_{m' \in SIM(C)} W(m') \quad (5)$$

SIM(m): 메소드 m 이 호출하는 메소드들의 집합
W(m): 메소드 m 의 가중치

정의 3. RFC

$$RFC(C) = \sum_{m' \in SIM(C)} 1 + \sum_{i=1}^m 1 \quad (6)$$

$$RFC_N(C) = \sum_{m' \in SIM(C)} W(m') + \sum_{i=1}^m W(m_i) \quad (7)$$

4. 사례 연구

본 논문에서 제안한 결합도 척도 개선 방법을 검증하기 위해 4 개의 오픈 소스 프로젝트를 적용사례로 들었다. 기존의 방법과 개선된 방법으로 3 개의 결합도 척도를 측정하였고, 변경-경향성과의 상관계수 측정을 통해 제안한 방법의 우수성을 입증하였다.

1) 실험 환경

그림 4 는 본 논문에서 실험한 결합도 측정 실험 프로세스를 보여준다. ANTLR 를 이용한 파서는 소스 코드를 입력으로 받아 클래스 정보를 생성하고 이를 이용하여 각 클래스의 결합도를 측정한다[13].

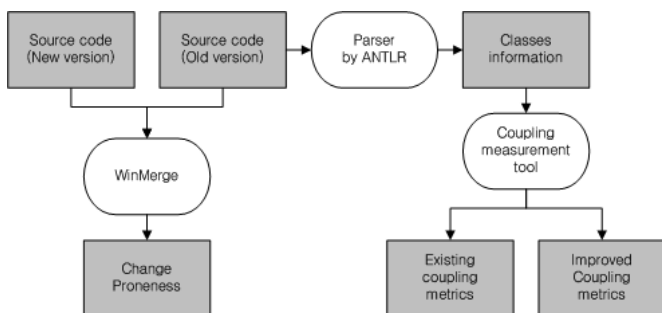


그림 4 결합도 측정 실험 프로세스

파일 비교 프로그램 WinMerge 를 이용하여 각 버전과의 소스 코드 변경 정도를 측정하고, 변경-경향성을 측정할 수 있었다[14]. 변경-경향성은 모듈에서 변경이 일어날 가능성을 말한다[15]. 변경-경향성을 측정하는 이유는 유지보수성(Maintainability)을 추정할 수 있고[7], 설계 의사 결정(Design decision making)에 도움을 주기 때문이다[16].

본 논문에서는 변경-경향성을 수식 8 과 같이 정의하였다. 변경 되는 라인은 소스 코드 크기에 종속적이기 때문에, 전체 라인 수를 분모로 하여, 소스 코드 크기에 독립적인 변경-경향성을 정의하였다.

$$\text{변경-경향성} = \frac{\text{변경된 코드 라인 수}}{\text{전체 코드 라인수}} \quad (8)$$

측정된 변경-경향성과 결합도간의 피어슨 상관계수를 측정하여 상관 관계를 비교하였다.

2) 실험 결과

각 클래스 별로 기존의 결합도 3 개와 개선된 결합도 3 개를 측정하였다. 그리고 4.1 장에서 정의한대로 변경-경향성을 측정한 후, 기존의 결합도, 개선된 결합도와 각각 상관계수를 측정 및 비교하였다.

그림 5, 6, 7 은 각 결합도와 변경-경향성과의 상관계수 측정 결과를 보여준다. 그래프에서 각 프로젝트의 첫 번째 막대는 기존의 결합도와 변경-경향성과의 상관계수, 두 번째 막대는 개선된 결합도와 변경-경향성과의 상관계수를 나타낸다.

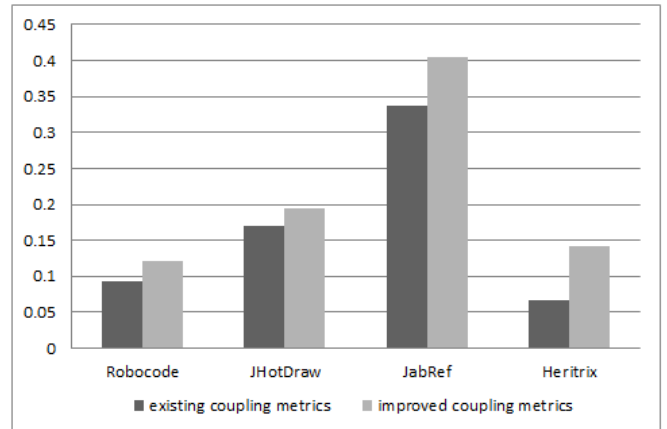


그림 5 CBO 와 변경-경향성과의 상관계수

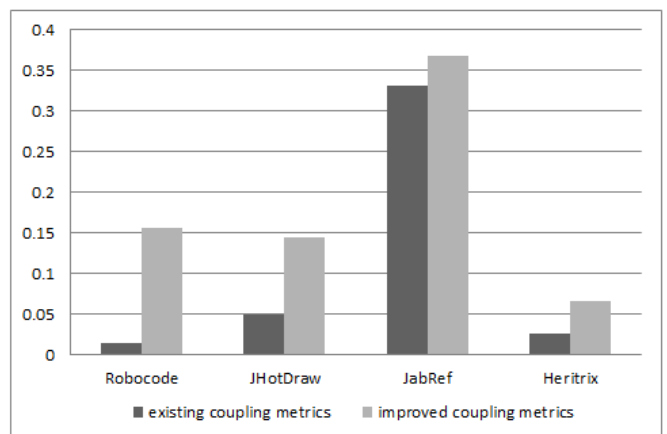


그림 6 MPC 와 변경-경향성과의 상관계수

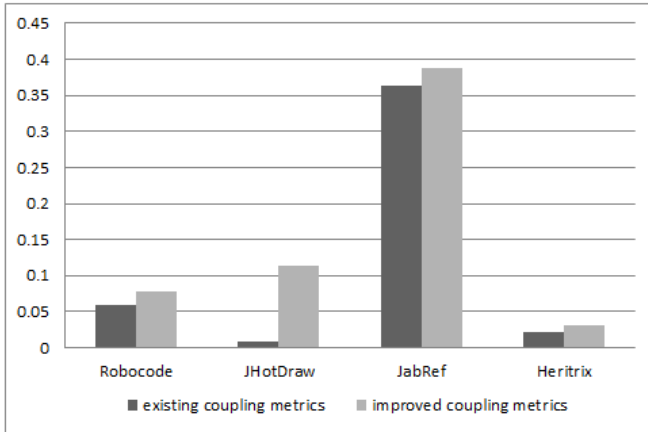


그림 7 RFC 와 변경-경향성과의 상관계수

하나의 모듈에 변화가 생겼을 때, 그것과 강하게 결합되어 있는 다른 모듈에게 변화는 전파된다. 따라서 모듈간의 높은 결합도를 가진다면, 변경이 일어날 가능성도 클 것이다. 그래서 결합도와 변경-경향성은 비례 관계이다. 그림 5, 6, 7의 상관 계수 측정 결과는 모두 비례 관계를 나타내는 양수 값이다. 이러한 결과값은 이 실험의 일관성을 보여준다.

4 개의 프로젝트의 모든 척도에서 상관 계수가 향상되는 것을 확인할 수 있다. 기존의 결합도 척도에 비하여 개선된 결합도 척도가 변경-경향성과의 상관 계수 값이 눈의 띄게 증가하였다. 따라서 개선된 결합도 척도는 변경-경향성을 예측하는 더 좋은 척도로서 사용될 수 있음을 알 수 있다

5. 결론

본 논문에서는 페이지랭크 알고리즘을 이용하여 메소드의 가중치를 측정하고, 이를 이용하여 결합도 척도의 개선 방법에 대해서 제안했다.

개선된 결합도 척도를 4 개의 오픈 소스 프로젝트에 적용하였고, 클래스의 변경-경향성과 결합도 측정값과의 상관 관계를 측정함으로써, 제안된 방법의 우수성을 실험적으로 확인할 수 있었다.

앞으로 진행할 연구로는, 본 논문에서 제안한 기법을 개선하여 다른 결합도 척도에 적용해보고, 또한 소스 코드 라인의 가중치를 측정하여 복잡도 척도를 개선해볼 예정이다.

참고문헌

- [1] W. Stevens, G. Myers, and L. Constantine, "Structured Design," IBM Systems J., vol. 13, no. 2, pp. 115-139, 1974.
- [2] L. C. Briand, J. W. Daly, and J. Wüst, "A Unified Framework for Coupling Measurement in Object-Oriented Systems," IEEE Transactions on Software Engineering, vol. 25, pp. 91-121, 1999.
- [3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford Digital Libraries Working Paper, 1998.
- [4] Koru, A.G., Liu, H.: Identifying and characterizing change-prone classes in two large-scale open-source

products. JSS (2007).

- [5] S.R. Chidamber and C.F. Kemerer, "Towards a Metrics Suite for Object Oriented Design," A. Paepcke, ed., Proc. Conf. Object-Oriented Programming: Systems, Languages and Applications, OOPSLA' 91, Oct. 1991. Also published in SIGPLAN Notices, vol. 26, no. 11, pp. 197-211, 1991.
- [6] S.R. Chidamber and C.F. Kemerer, "A Metrics Suite for Object Oriented Design," IEEE Trans. Software Eng., vol. 20, no. 6, pp. 476-493, 1994.
- [7] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability," J. Systems and Software, vol. 23, no. 2, pp. 111-122, 1993.
- [8] F. Abreu, M. Goulão, and R. Esteves, "Toward the Design Quality Evaluation of Object-Oriented Software Systems," Proc. Fifth Int'l Conf. Software Quality, Austin, Texas, Oct. 1995.
- [9] Robocode. Available From : <http://robocode.sourceforge.net/>>2001.
- [10] JHotDraw. Available From : <http://www.jhotdraw.org/>>1997.
- [11] JabRef. Available From : <http://jabref.sourceforge.net/>>2003.
- [12] Heritrix. Available From : <http://crawler.archive.org/>>2004.
- [13] ANTLR. Available From : <http://www.antlr.org/>>1990.
- [14] WinMerge. Available From : <http://winmerge.org/>>2000.
- [15] D. L. Parnas. Some software engineering principles. pages 257-266, 2001
- [16] E. Arisholm and D. I. K. Sjoberg, "Towards a framework for empirical assessment of changeability decay," J. Syst. Softw., vol. 53, pp. 3-14, 2000