

객체지향 메트릭을 이용한 결합 예측 모형의 실험적 비교 (A Comparative Experiment of Software Defect Prediction Models using Object Oriented Metrics)

김 윤 규 ^{*} 김 태 연 ^{††}
(Yun Kyu Kim) (Tae Yeon Kim)

채 흥 석 ^{***}
(Heung Seok Chae)

요약 검증과 확인을 통한 소프트웨어의 효율적인 관리를 지원하기 위하여 객체지향 메트릭 기반의 결합 예측 모형이 많이 제안되고 있다. 제안된 모형은 주로 로지스틱 회귀분석으로 개발하였다. 그리고 개발된 모형의 결합 예측 정확도는 60~70%이었다. 본 논문에서는 기존 결합 예측 모형의 효과를 확인하기 위하여 이를립스 3.3을 대상으로 개발된 모형과 유사한 방법으로 실험을 하였다. 실험 결과 모형의 정확성은 약 40%이었다. 이는 주장된 예측력보다 많이 낮은 수치이었다. 또한 단순 로지스틱 회귀분석이 다중 로지스틱 회귀분석보다 높은 예측력을 보였다.

키워드 : 메트릭, 결합 예측 모형, 객체지향 시스템

Abstract To support an efficient management of software verification and validation activities, many defect

- 본 연구는 저식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(IITA-2009-(C1090-0902-0032))
- 이 논문은 제34회 추계학술대회에서 '객체지향 메트릭을 이용한 결합 예측 모형의 실험적 비교'의 제목으로 발표된 논문을 확장한 것임

^{*} 학생회원 : 부산대학교 컴퓨터공학과
kim_yunkyu@pusan.ac.kr

^{††} 비회원 : 부산대학교 컴퓨터공학과
tykim@pusan.ac.kr

^{***} 정회원 : 부산대학교 컴퓨터공학과 교수
hschae@pusan.ac.kr

논문접수 : 2009년 1월 19일

심사완료 : 2009년 6월 12일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제15권 제8호(2009.8)

prediction models have been proposed based on object oriented metrics. They usually adopt logistic regression analysis. And, they state that the correctness of prediction is about 60~70%. We performed a similar experiment with Eclipse 3.3 to check their prediction effectiveness. However, the result shows that correctness is about 40% which is much lower than the original results. We also found that univariate logistic regression analysis produces better results than multivariate logistic regression analysis.

Key words : Metric, Defect prediction model, Object oriented system

1. 서 론

소프트웨어의 품질을 높이기 위한 방법으로 메트릭을 이용한 결합 예측 모형들이 제안되었다[1,3-5]. 결합 예측 모형은 소프트웨어 메트릭을 사용하여 예측 대상 시스템의 결함 유무를 클래스 단위로 평가한다. 결합 발생이 높은 클래스를 포함하는 특정 모듈에 더 많은 자원과 관리를 한다면 개발 단계 초기에 소프트웨어의 결함을 발견할 수 있기 때문에 개발 이후 단계보다 수정하는데 적은 비용이 발생된다.

객체지향 소프트웨어의 구조적 속성을 측정 하기 위하여 객체지향 소프트웨어 메트릭이 연구되었다. Chidamber and Kemerer(CK) 메트릭, Metrics for Object Oriented Design(MOOD) 메트릭이 대표적인 객체지향 소프트웨어 메트릭이다[2].

그리고 메트릭 측정값으로 결함을 예측하기 위하여 통계적 기법으로 회귀분석이 주로 사용되고 있으며 최근에는 로지스틱 회귀분석을 이용한 예측 모형이 다수 발표되었다. 그러나 대규모 시스템을 대상으로 기존 예측 모형들을 비교 실험한 연구는 부족하다. 결합 예측 모형이 실제 개발 환경에서 활발히 사용되기 위해서는 예측 정확도에 관한 실증적 연구 결과가 뒷받침되어야 한다. 본 논문에서는 결합 예측 모형간의 정확도를 비교 실험 하였다. 기존의 결합 예측 모형을 범용적인 예측 모형으로 가정하고 대규모 오픈 소스 소프트웨어인 이를립스에 적용하였다. 그리고 결합 예측 모형에 필요한 클래스별 결합 정보는 버그추적시스템을 통하여 수집하였다.

실험 결과 이를립스를 대상으로 결합을 예측한 결과는 기존 모형이 제시한 결합 예측력보다 낮게 측정되었다. 그리고 다중 회귀분석을 이용한 예측 모형 보다 단순 회귀분석을 사용하는 것이 높은 예측력을 보였다.

2. 배 경

2.1 CK 메트릭

CK 메트릭은 Chidamber와 Kemerer가 제안한 객체

지향 소프트웨어 메트릭이다[2]. 현재 실험적 연구에서는 소프트웨어의 결합을 예측하기 위하여 CK 메트릭을 사용하고 있다[3-5]. 개별 CK 메트릭의 정의는 다음과 같다.

- CBO(Coupling Between Object classes)는 클래스간에 사용되는 메소드나 인스턴스의 수를 계산하여 결합도를 측정하는 척도이다.
 - DIT(Depth of Inheritance Tree of a class)는 개별 클래스의 상속의 깊이를 측정하는 척도이다
 - LCOM(Lack of Cohesion on Methods)는 개별 클래스의 정의된 메소드에서 사용되는 필드를 계산하여 응집도를 측정하는 척도이다.
 - NOC(Number Of Children)는 개별 클래스를 상속하고 있는 자식 클래스의 개수 계산하여 복잡도를 측정하는 척도이다.
 - RFC(Response For a Class)는 클래스내의 전체 메소드의 집합과 다른 클래스 간의 통신의 양으로 복잡도를 측정하는 척도이다.
 - WMC(Weighted Methods Per Class)는 개별 클래스에 정의된 메소드의 복잡도를 측정하는 척도이다.
- ### 2.2 객체 지향 메트릭을 이용한 결합 예측 모형
- 대규모 시스템을 대상으로 객체지향 메트릭을 이용한 결합 예측 모형의 연구에 관하여 IEEE Transaction on Software Engineering에 발표된 논문을 중심으로 살펴보면 10년간 매년 한편 이상씩의 관련 논문이 소개되고 있다. 표 1은 대표적인 객체지향 메트릭을 사용한 결합 예측 모형 연구들이다. 표의 옆은 기존의 결합 예측 모형이고 행은 해당 모형의 특성을 나타낸다.

표 1 객체지향 메트릭을 이용한 결합 예측 모형

예측 모형	Olague[3]	Yuming[4]	Gyimothy[5]
실험대상	Mozilla Rhino	NASA KC1	Mozilla v.1.6
개발언어	Java	C++	C++
클래스 수	201	145	3192
규모	58.7 K 라인	40 K 라인	1 M 라인
정확성	82%	61.4%	72.6%
완전성	N/A	74.6%	65.2%

Olague 모형은 자바 언어로 개발된 Mozilla Rhino를 대상으로 개발되었다. 해당 예측 모형은 CK, MOOD, QMOOD 메트릭으로 예측력을 비교 실험하였다. 그 결과 예측력은 CK 메트릭을 이용한 모형이 비교적 높게 나왔으며 결합 예측의 정확성은 82%이었다. Yuming은 NASA의 위성용 소프트웨어를 대상으로 결합의 심각도를 세 단계로 나누어 예측 모형을 개발하였다. 그리고 해당 시스템의 실제 결합 비율은 42.6%이었다. Gyimothy는 대형 오픈 소스 소프트웨어인 모질라 1.6을 대상으로

예측 모형을 개발하였다.

표 1에 제시된 결합 예측 모형들은 정확성이 61.4%~82%의 범위에 있으며 이는 비교적 높은 예측력이 있음을 의미한다. 그리고 모형들은 모두 4만 라인 이상의 대규모 시스템을 대상으로 개발되었다.

2.3 로지스틱 회귀분석

로지스틱 회귀모형은 개별 클래스에 결함이 있을지 여부를 정하는데 적합하다. 식 (1)은 로지스틱 회귀모형을 나타낸다. X_i 는 독립변수인 객체지향 메트릭이며 C_i 는 X_i 의 회귀계수를 의미한다. 종속 변수인 π 는 클래스에서 결함이 발생할 확률을 의미하며 해당 확률이 특정 임계값 이상이면 해당 클래스는 결함이 있는 것으로 간주한다.

$$\pi(X_1, X_2, \dots, X_n) = \frac{e^{C_0 + C_1X_1 + \dots + C_nX_n}}{1 + e^{C_0 + C_1X_1 + \dots + C_nX_n}}$$

식 (1) 로지스틱 회귀모형

예를 들어 결합 예측 모형에 의해서 계산된 결합 발생 확률이 A 클래스에는 0.7이고 B 클래스는 0.3이라고 가정한다. 그리고 임계값은 0.5로 결합 예측 모형에서 정의되었다고 가정한다. 그러면 임계값을 기준으로 A 클래스는 결함이 있는 것으로 예측하고 B 클래스는 결함이 없는 것으로 예측한다.

3. 실험 설계

3.1 실험 대상

대규모 시스템을 대상으로 결합 예측 모형의 정확성과 완전성을 비교하기 위해서 실험 대상은 오픈 소스인 이클립스 3.3으로 선정하였다. 실험에서는 이클립스 3.3에 있는 패키지 중 org.eclipse 패키지에 있는 모든 하위 소스 파일만을 대상으로 한다. 표 2는 실험 대상인 이클립스 3.3 시스템의 규모와 결합 정보를 나타낸다.

표 2 이클립스 3.3의 규모 및 결합 정보

측정 대상	총 클래스	결합 클래스	결합 수	소스 라인
org.eclipse.*	11,505	1,725	4,133	1,455,395

3.2 결합에 대한 자료 수집

실험 대상의 개별 클래스가 결함인 경우는 이클립스 버그추적시스템에 등록된 자바 클래스 파일의 이름과 패키지명이 이클립스 소스파일과 동일할 때이다. 자료 수집 과정은 크게 결합 데이터 수집 단계와 정제 단계로 나누어진다.

우선 결합 수집 단계에서는 버그질라에 등록된 버그를 웹을 통해 파일 형태로 추출한다. 하나의 자바 클래

스에 여러 개의 결함이 존재하는 경우 각각을 1씩 더하였다. 수집 단계에서 버그 추적 시스템을 이용하여 3069 개의 버그 있는 자바 클래스를 수집하였다. 실험에서는 버그 추적 시스템의 등록된 버그 중 결함의 심각도에 상관없이 모든 버그를 대상으로 자료를 수집하였다. 다음으로 정제 단계에서는 버그로 수집된 클래스가 메트릭으로 측정한 클래스 중에 없다면 수집한 결함에서 제외했다. 결함으로 분류된 클래스의 수는 1725개이며 총 결함의 수는 4133개이다.

3.3 메트릭 측정

측정에 사용된 CK 메트릭은 대표적인 객체지향 메트릭으로 소프트웨어 모델의 크기와 복잡도를 측정하기 위해 사용된다. 또한 실험 과정에서는 결함을 발생 시킬 확률에 독립적인 영향을 주는 변수로 사용된다. 그리고 소프트웨어의 규모와 복잡도를 측정하는 전통적인 Line Of Code(LOC) 메트릭을 추가하여 CK 메트릭의 측정 결과값을 보완하도록 하였다. 본 실험에서 메트릭 측정은 Together 2007으로 하였다.

표 3은 실험 대상으로 측정한 메트릭의 기술 통계량이다. DIT 메트릭인 경우 중앙 값은 2이며 제1사분위수(Q1)부터 제3사분위수(Q3)까지 50%의 클래스가 1과 3 사이 값을 가진다. DIT 메트릭 측정값은 다른 메트릭 측정 결과값보다 평균값에서 더욱 조밀하게 클래스들이 분포함을 알 수 있다.

표 3 실험 대상 시스템의 메트릭 기술 통계량

	WMC	LCOM	RFC	DIT	CBO	NOC	LOC
최대값	1561	500701	1213	11	231	210	7740
최소값	0	-1	0	0	0	0	1
중앙값	10	0	42	2	8	0	45
평균값	25.53	153	71.56	2.17	14.87	1.04	126.12
표준편차	58.25	4698	81.15	1.67	19.28	5.44	284.23
Q1	4	-1	21	1	3	0	17
Q3	26	27	99	3	19	0	130

3.4 결합 예측 모형의 적합도 평가

결합 예측 모형의 예측력 평가 방법은 적합도 분류를 이용한 정밀도와 정확도 그리고 완전성이 있다.

정밀도는 전체 클래스 중에서 예측 모형에 의하여 정확히 분류된 클래스가 차지하는 비율이다. 정확도는 예측 모형에 의하여 결합으로 분류된 전체 클래스 중에서 실제로도 결합인 클래스가 차지하는 비율이다. 완전성은 실제로 존재하는 총 결함 중에서 예측 모형에서 결합으로 분류한 클래스에 존재하는 결함의 비율이다.

표 4는 개별 클래스를 예측 모형에 의해 결합으로 분류된 클래스와 실제로 결합이 존재하는 클래스들을 표시하였다. 분류표의 열은 결합 예측 모형에 의해 클래스

의 결함을 예측하고 해당 결과에 따라 결합이 있는 클래스와 없는 클래스로 나타낸다. 분류표의 행은 실제로 결합이 있는 클래스를 기준으로 나타낸다. A1~A4까지 영역은 분류된 자바 클래스들의 수이고 B1과 B2는 해당 영역에 분류된 클래스들이 가지고 있는 실제 결합의 총합이다. 그러므로 정밀도와 정확도, 완전성을 다음과 같이 계산 할 수 있다.

표 4 적합도 분류표

		모형의 예측	
		결합 무 ($\pi < X$)	결합 유 ($\pi \geq X$)
실 제	결합 무	A1	A2
	결합 유	A3(B1)	A4(B2)

· 정밀도: $(A1 + A4) / (A1 + A2 + A3 + A4)$

· 정확도: $A4 / (A2 + A4)$

· 완전성: $B2 / (B1 + B2)$

정확성과 완전성은 절충관계(trade-off)가 있으므로 보통 절충점을 임계값으로 결정하여 모델의 예측력을 계산한다.

4. 실험 결과

4.1 다중 변수를 사용한 결합 예측 모형 분석

개별 클래스마다 결합이 발생될 확률(π)은 메트릭 계산 결과값과 결합 예측 모형이 정의한 회귀분석 수식으로 정해진다.

표 5는 다중 회귀분석을 이용하여 개발된 결합 예측 모형이다. 모든 예측 모형에서 사용된 메트릭은 CBO가 유일하였다.

표 5 다중 변수를 사용한 결합 예측 모형

	Olague	Yuming	Gyimothy
상수	-2.87	-0.232	1.241
CBO	0.06	0.206	0.774
DIT	0.65	NA	0.505
LCOM	-0.21	0.008	NA
LOC	NA	0.006	0.678
NOC	0.49	-1.091	NA
RFC	NA	-0.014	-0.214
WMC	0.06	NA	NA

해당 예측 모형을 가지고 이를립스의 개별 클래스를 대상으로 계산하여 클래스마다 결합이 발생될 확률(π)을 구했다. 그리고 임계값은 0부터 1까지 0.001씩 조정하여 절충점을 계산한 후에 그 중에서 가장 높은 예측력을 보인 지점으로 정하였다.

표 6은 다중 회귀분석을 이용한 결합예측 모형의 적합도 분류표이다.

표 6 다중 변수를 사용한 결합 예측 모형의 적합도 분류

(a) Olague 모형		(b) Yuming 모형			
실제	모형의 예측		실제	모형의 예측	
	결합 무 ($\pi < 0.602$)	결합 유 ($\pi \geq 0.602$)		결합 무 ($\pi < 0.993$)	결합 유 ($\pi \geq 0.993$)
결합 무	8037	1743	결합 무	8765	1015
결합 유	1386 (3455)	339 (678)	결합 유	1224 (2748)	501 (1385)

Olague 모형인 경우 임계 값을 0.602로 설정하였다. 실험 결과 9423개의 클래스가 결합 없는 것으로 예측 되었고 2082개의 클래스는 결합이 있는 것으로 예측 되었다. 예측 모형은 전체 클래스 중에서 18.1%를 결합으로 분류하였다. 실제 소스 코드에는 결합 있는 클래스의 비율이 15%로 낮기 때문에 Olague 모형은 결합이 없는 클래스를 잘못 예측한 수가 상대적으로 많았다. 이는 Olague 모형의 결합 예측력을 낮게 하는 요인으로 분석된다.

Yuming 모형인 경우 π 가 0.99일 때 임계 값이다. 9989 개의 클래스는 결합이 없는 것으로 예측되었고 1516개의 클래스는 결합이 있는 것으로 예측하였다. 해당 모형은 전체 클래스 중에서 13%를 결합 있는 클래스로 분류하였다. 결합 있는 클래스로 맞게 예측한 클래스 중에 포함된 결합의 총합은 Yuming 모형인 경우 1385개였고 Olague 모형인 경우에는 678개이다. 이는 Yuming 모형이 Olague 모형보다 예측률이 높은 것으로 분석된다.

표 7은 결합 예측 모형의 평가표이다. Yuming 예측 모형이 Olague보다 정밀도, 정확도 그리고 완전성이 높다. 그리고 Gyimothy 모형인 경우 π 가 1에 매우 근접한 값으로 구해졌기 때문에 정확도와 완전성간의 절충점을 정하지 못했다. 따라서 표에서는 NA로 표시 하고 결합 예측 모형들과의 비교에서 제외하였다.

표 7 다중 변수 결합 예측 모형의 평가

예측 모형	π	정밀도	정확도	완전성
Olague	0.602	72.8%	16.3%	16.3%
Yuming	0.99	80.5%	33.0%	33.5%
Gyimothy	NA	NA	NA	NA

4.2 단일 변수를 사용한 결합 예측 모형 분석

표 8은 단순 회귀분석을 이용하여 개발된 결합 예측 모형이다. 결합 예측 모형에서 사용된 메트릭은 개별 클래스에 결합이 발생될 확률(π)에 독립변인이다. 그리고 예측 모형은 독립변인으로 CK 메트릭과 LOC 메트릭을 사용하지만 그 중에서 p-value가 0이고 R^2 값이 다른 독립변인 중에서 가장 큰 값을 가질 때 해당 메트릭을

표 8 단순 회귀분석을 사용하여 개발된 결합 예측 모형

모형	상수	CBO	LOC	WMC	P-value	R^2
Olague	-2.59	0.019	NA	NA	0	NA
Gyimothy	-0.252	NA	NA	1.069	0	0.321
Yuming	-0.160	NA	0.010	NA	0	0.390

단순 회귀분석 모형의 독립변인으로 선정했다. 표의 Olague 모형은 연구[3]에서 R^2 값을 명시하지 않았기 때문에 P-value가 0인 값 중에서 실험 대상에 가장 높은 정확도와 완전성을 가진 모형으로 선정했다.

예를 들어 Gyimothy 예측 모형의 독립변인으로는 WMC가 사용되었고 WMC의 계수는 1.069이고 상수는 -0.252이다. 그리고 WMC 메트릭의 P-Value는 0이고 R^2 는 0.321이었다. p-value가 0인 의미는 Gyimothy 모형의 개발 환경에서 WMC 메트릭이 사용된 회귀방정식이 유의함을 나타내고 R^2 가 크다면 해당 변인은 π 에 좋은 예측인자로 분석할 수 있다.

표 9는 해당 결합 예측 모형별로 실험 대상의 클래스를 임계값에 따라 분류한 표이다. Yunming 모형인 경우 결합으로 예측한 클래스는 1277개이고 결합이 아닌 것으로 예측한 클래스는 10228개이다. 해당 모형의 정밀도는 93.2%이고 정확도와 완전성은 40.7%이었다.

표 9 단일 변수를 사용한 결합 예측 모형의 적합도 분류

(a) Olague 모형		(b) Yuming 모형			
실제	모형의 예측		실제	모형의 예측	
	결합 무 ($\pi < 0.124$)	결합 유 ($\pi \geq 0.124$)		결합 무 ($\pi < 0.932$)	결합 유 ($\pi \geq 0.932$)
결합 무	8926	854	결합 무	9023	757
결합 유	1215 (2579)	510 (1554)	결합 유	1205 (2448)	520 (1685)

(c) Gyimothy 모형

실제	모형의 예측	
	결합 무 ($\pi < 0.998$)	결합 유 ($\pi \geq 0.998$)
결합 무	4189	5591
결합 유	300 (673)	1425 (3460)

표 10에 의하면 모든 결합 예측 모형이 다중 변수를 사용한 회귀 모형보다 예측력이 높았다.

표 10 단일 변수 결합 예측 모형의 평가

예측 모형	π	정밀도	정확도	완전성
Olague	0.124	82.0%	37.5%	37.5%
Yuming	0.932	82.9%	40.7%	40.7%
Gyimothy	0.998	48.8%	20.3%	83.7%

표 10에서는 단일 변수로 LOC를 사용한 Yuming 모형이 가장 높은 결합 예측률을 보였다. 또한 표 9(b)를 보면 Yuming 모형은 총 1277개의 클래스를 결합 있는 클래스로 분류하였고 이는 전체 클래스 중에서 11%에 해당하는 클래스를 결합으로 분류한 것이다. 그리고 결합을 예측한 클래스의 수는 예측 모형 중에서 가장 적었고 예측력은 가장 높았다. 이는 Yuming 모형이 비교 모형 중에서 결합을 예측하는데 가장 유용함을 알 수 있다. 표 9(c)를 보면 비록 Gyimothy 모형이 가장 많이 결합 있는 클래스를 예측 했지만 총 7015개의 클래스를 결합 있는 클래스로 분류하였고 이는 전체 클래스 중에서 약 60.9%를 결합 있는 것은 예측한 것이다. 이는 추출된 클래스 수와 완전성은 비례하게 높아짐을 의미한다. 그 결과 정확성은 낮고 완전성은 높아진 것으로 분석된다.

실험 결과 단일 변수를 사용한 모든 예측 모형들이 다중 변수를 사용한 예측 모형보다 높은 정확성과 완전성을 보였다. 이는 단일 변수를 사용한 예측 모형이 다중 변수를 사용한 예측 모형보다 적은 개수로 결합 있는 클래스를 분류하지만 결합 클래스의 예측력이 더 높음을 의미한다.

그리고 실험에서 적용한 모든 결합 예측 모형은 16.3%에서 40.7%까지 정확성을 보였다. 이는 기존 결합 예측 모형들이 제시한 시스템에서 측정한 값보다 낮은 값이다.

5. 관련연구

Briand은 동일한 개발팀에서 개발된 두 개의 시스템을 대상으로 모형의 예측력을 비교하였다[6]. 해당 연구에서는 다양한 평가 방법을 사용하여 두 시스템 간의 예측력을 비교하였다. 실험 결과 시스템 간의 차이로 인하여 결합 예측 결과를 영향이 있음을 주장하였다. 해당 연구에서 사용된 시스템은 자바기반이며 텍스트 편집을 위한 웹용프로그램이기 때문에 다양한 시스템간의 범용적인 예측력 비교에 대한 논의가 미흡하다. 본 연구에서는 다양한 특성을 지닌 시스템간의 비교 분석을 위하여 개발 언어가 다르거나 사용목적인 다른 도메인 영역의 시스템을 대상으로 모형의 범용적인 적용을 실험하였다.

Khoshgoftaar는 버전이 다른 시스템을 대상으로 결합 예측 모형 기법간의 예측력을 비교하였다[7]. 비록 많은 비교 실험을 하였지만 동일 제품군에 대한 결합 예측 모형 비교를 하였기 때문에 모형의 범용적인 적용에 대한 논의가 없다. 본 실험에서는 서로 다른 3개의 시스템을 대상으로 모형의 예측력을 비교하였다.

6. 결론 및 향후 연구

객체지향 설계 기법을 이용하여 개발된 소프트웨어의 품질을 높이기 위한 방법 중에는 소프트웨어 메트릭을

사용한 결합 예측 모형이 있다. 본 논문의 목적은 기존의 결합 예측 모형들을 범용적인 예측 모형으로 사용이 가능한지 확인하고 실제로 대규모 프로젝트를 대상으로 결합 예측력의 정확성을 비교하기 위함에 있다. 결합 예측 모형으로는 최근 3년 사이에 제안된 Olague, Yuming 그리고 Gyimothy 모형을 적용하였다. 선정된 결합 예측 모형은 이를립스 3.3의 소스에 대해서 낮은 결합 예측력을 보였다. 그리고 모형은 다중 변수보다는 단일 변수 메트릭을 사용한 것이 높은 정확도와 완전성을 보였다.

향후 연구에서는 Family-Product간의 결합 예측력을 분석 하여 같은 소프트웨어의 특성을 가진 제품 군에는 결합 예측 모델을 적용하는 것이 정확도와 완전성을 높이는지 분석할 것이다. 그리고 정확도를 높이기 위해서는 결합 예측 모형을 전체 측정 대상보다는 결합이 많은 부분을 선별하여 사용하는 것이 합당한지 비교할 것이다.

참 고 문 헌

- [1] K. EL Eman, W. Melo and J.C. Machado, "The Prediction of Faulty Classes Using Object-Oriented Design Metrics," *Journal of Systems and Software*, vol.56, no.1, pp.63-75, 2001.
- [2] S.R. Chidamber, C.F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol.20, no.6, pp.476-493, 1994.
- [3] H. M. Olague, L. H. Etkorn, S. Gholston et al., "Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes," *IEEE Transactions on Software Engineering*, vol.33, no.6, pp.402-419, 2007.
- [4] Z. Yuming, and L. Hareton, "Empirical Analysis of Object-Oriented Design Metrics for Predicting High and Low Severity Faults," *IEEE Transactions on Software Engineering*, vol.32, no.10, pp.771-789, 2006.
- [5] T. Gyimothy, R. Ferenc, and I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Transactions on Software Engineering*, vol.31, no.10, pp.897-910, 2005.
- [6] L. C. Briand, W. L. Melo, and J. Wust, "Assessing the applicability of fault-proneness models across object-oriented software projects," *IEEE Transactions on Software Engineering*, vol.28, no.7, pp.706-720, 2002.
- [7] T. M. Khoshgoftaar, and N. Seliya, "Fault Prediction Modeling for Software Quality Estimation: Comparing Commonly Used Techniques," *Empirical Software Engineering*, vol.8, no.3, pp.255-283, 2003.