

한국정보과학회
Korean Institute of Information Scientists and Engineers

제 20 권 제 1 호
Vol. 20 No. 1



2018

제 20 회 한국 소프트웨어공학 학술대회

논문집

Proceedings of the 20th Korea Conference on
Software Engineering (KCSE 2018)

- 일시: 2018년 1월 29일(월) ~ 1월 31일(수)
- 장소: 강원도 평창 한화리조트(휘닉스파크점)

주최: 한국정보과학회, 한국정보처리학회

주관: 한국정보과학회 소프트웨어공학 소사이어티
한국정보처리학회 소프트웨어공학연구회

후원:  SOLUTIONLINK,  redhat, (주)비트컴퓨터,
T3Q(주), (주)다한테크, 슈어소프트테크(주),
STA 테스트컨설팅(주), TTA 소프트웨어시험인증연구소,
지능형 블록체인 연구센터, 신뢰적 지능형 CPS 연구단,
SW 상시모니터링기술연구단,
소프트웨어 안전성 보증 연구센터

1월 31일 (수)

시 간		행 사 내 용			
		논문 발표 E			
		E1: 분석 및 평가 2 좌장: 홍신(한동대) 장소: 그랜드홀 2	E2: SW 품질 2 좌장: 고인영(KAIST) 장소: 세미나실 1	E3: 유지보수 좌장: 이선아(경상대) 장소: 세미나실 2	E4: 설계 & 모델링 2 좌장: 유준범(건국대) 장소: 세미나실 6
10:40-12:00 (80 분)	<p>GitHub 에서의 이슈 및 풀 요청 관련 활동이 소스 코드 품질에 미치는 영향 분석 [단편논문] 조영준, 고인영(KAIST)</p> <p>객체지향 재사용 매트릭스 [단편논문] 변은영, 박지훈, 김영철(홍익대)</p> <p>역공학 기반 코드 가시화를 통한 기능점수의 비용 추정 검증 방안 [단편논문] 문소영, 김영철(홍익대)</p> <p>Code X-Ray: LG Electronics' code quality inspection platform [산업체논문] Milhan Kim, Jewhi Ryu, Heewon Choi (LG 전자)</p>	<p>가중치 기반 센서 데이터 획득 효율성 향상 기법 [단편논문] 송명호, 김수동(숭실대)</p> <p>MR3 엔진 ASM 단계 성능 비교 [우수단편논문] Hoang Le, Sungwoo Park(포항공대)</p> <p>요구사항에 따른 이슈 분류 [단편논문] 심재경, 정세린, 이선아(경상대)</p> <p>딥러닝 모델 기반의 버그 담당자 자동 배정 시스템 성능 측정 [학부논문] 박해성, 김수빈, 이찬근, 채병훈(중앙대)</p>	<p>유전 프로그래밍을 이용한 버그 자동 정정 기법에 유사 버그 정정 활용 [일반논문] 정영준, 민경식, 양근석(서울시립대), 이정원(아주대), 이병정(서울시립대)</p> <p>Toward Measuring Quality Factors for Software Product Operation [일반논문] Horm Daneth, 홍장의(충북대)</p> <p>국방 소프트웨어 자원관리시스템의 소스 코드 품질을 위한 자바 파서 개발을 통한 적용 사례 [단편논문] 박보경, 문소영, 서채연, 김영철(홍익대) 김광남, 최영식, 신상훈(육군 정보체계관리단)</p>	<p>군집 비행을 위한 무인비행기 네비게이터 시스템 설계 [단편논문] 이성희, 김덕엽, 윤보람, 이우진(경북대)</p> <p>동적 데이터 획득 시스템 시각화 프레임워크 [단편논문] 임성민, 김수동(숭실대)</p> <p>클래스 변경 정보에 기반한 마이크로서비스 단위의 애플리케이션 구성 방법 [단편논문] 김대호, 김웅수, 윤동규(부산대), 박준석(물류혁신네트워킹연구소), 염근혁(부산대)</p> <p>금속재료 피로시험의 최적설계 소프트웨어의 클린코드 적용 사례 [산업체논문] 정인용, 박상건, 손현승(모아소프트), 김광진(알에스피)</p>	
12:00-12:30 (30 분)	<p>폐회식 장소: 그랜드홀 2</p>	<p>사회: 이병정 조직위원장(서울시립대)</p>			

* 위 일정은 사정에 따라 변경될 수가 있습니다.

객체지향 재사용 매트릭스

변은영⁰¹ 박지훈² 김영철³

홍익대학교 소프트웨어공학연구소

{eybyun¹, pjh², bob³}@selab.hongik.ac.kr

Object Oriented Reusability Metrics

Eun Young Byun⁰¹, Ji Hoon Park², R. Young Chul Kim³

SE Lab, Dept. of Computer Information Communication, Hongik University

요 약

소프트웨어 산업에서의 잦은 업데이트와 유지보수 시 소모되는 시간/비용 절감 및 고품질화를 위한 방안으로 재사용은 매우 중요한 이슈이다. 이 논문은 레거시 시스템을 기반으로 확장 또는 기능이 개선된 새로운 시스템 개발에서 재사용 메커니즘을 고품질을 위한 하나의 방안으로 제안한다. 기존에 연구된 객체지향 패러다임에서 재사용성 매트릭스의 문제점(①동적 매트릭스 미비, ②CK 매트릭스, ③순환 복잡도, ④상속성, ⑤다형성)을 도출하고, 이를 개선 및 확장한 정적/동적 매트릭스를 정의한다. 또한, 정의한 매트릭스를 측정하여 재사용 모듈을 자동 식별하는 시스템을 구축한다. 이는 기존 레거시 시스템 모듈의 재사용성을 높임으로써, 신뢰성 및 생산성을 개선시키고 품질을 향상시킬 수 있다.

1. 서 론

최근 소프트웨어의 적용 분야와 규모가 다양화 및 대 규모화 되면서 시장 환경의 급속한 변화, 빈번한 사용자 요구사항 변경 요청 등이 야기되고 있다. 이로 인한 부족한 시간, 예산, 인력 문제들로 인해 고품질 소프트웨어 개발은 더욱 어려운 실정이다. 고품질의 소프트웨어를 위해, 국내 중소/중견/벤처/스타트업 기업들은 SW 테스팅에 초점을 두지만, 이 논문에서는 레거시 시스템의 모듈을 재사용하여 고품질화시키기 위한 재사용 매트릭스 및 자동식별 시스템을 제안한다.

기존 재사용 매트릭스 연구[1,2,3]는 정적 분석 기반의 재사용 모듈 식별이 이루어졌지만, 동적 분석 기반에서의 재사용성 식별은 미비하다. 또한, 객체지향 프로그램은 절차지향 프로그램과 차이를 갖지만 상속성, 다형성 등의 특성이 반영되지 않았다. 이를 위해, 확장 및 개선된 재사용 매트릭스를 정의하고 이를 자동 측정하는 시스템을 구축한다. 식별된 모듈을 재사용함으로써 레거시 시스템을 개선 및 확장하는 새로운 시스템의 도메인에서 빠르고 효율적인 개발이 가능하다.

본 논문의 구성은 다음과 같다. 2장은 관련연구로 기존 재사용 매트릭스의 문제점을 언급한다. 3장은 확장 및 개선된 정적/동적 재사용 매트릭스를 정의한다. 4장은 정의한 매트릭스 기반 자동 식별 시스템을 설명한다. 5장에서는 결론 및 향후 연구를 언급한다.

2. 관련연구

Marko[4]의 연구에서 언급된 기존 연구에서 다음의

문제점을 도출하였다. 1) 정적 관점에 편향되어 있고 동적 관점은 매우 미비하다. 동적 매트릭스는 Component Dynamic Complexity (CDC) 단 한가지만이 언급된다. 2) 재사용성 측정을 위해 대표적으로 사용되는 CK 매트릭스는 결합도와 응집도의 종류를 고려하지 않는다. 3) 순환 복잡도는 객체지향 프로그램의 복잡도를 측정하는데 한계가 있다. 이 논문에서는 이를 개선하기 위해, 성숙 결합도/응집도를 측정함으로써 재사용성을 측정하고자 한다. 객체지향 프로그램과 절차지향 프로그램의 주요 차이점은 상속성이다. 4)이로 인해, 객체지향 메커니즘에서의 결합도는 단순한 호출 관계만으로는 식별이 불가능하다. 특히 레거시 시스템의 재사용에서의 상속성과 같은 클래스간의 관계는 고려해야 하는 문제이다. 다형성도 객체지향의 중요한 특성이다. 5)오버라이딩에서의 동적 바인딩은 동적 분석으로만 식별이 가능하다.

3. 확장 및 개선된 정적/동적 재사용 매트릭스

기존의 문제점을 해결하고자, 확장 및 개선된 정적/동적 재사용 매트릭스를 정의한다. 레벨은 메소드, 클래스, 객체, 클래스-객체로 구분한다. 클래스-객체 레벨은 정적인 클래스와 동적인 객체 간의 동적 바인딩으로 현재 연구 중에 있다. 기존 매트릭스는 제외하고, 확장 및 개선된 매트릭스에 대해 설명한다.

3.1 메소드 레벨

- Coupling Between Method(CBM): 클래스 레벨에서 클래스간의 호출 횟수를 계산한 CBO(Coupling Between Object Classes)를 메소드 레벨로 확장한다.

- Method Coupling(MCO): 자료/스탬프/제어/외부/공유/내용 결합도를 구분하고 평균을 측정한다.
- Lack of Cohesion of Statement(LCOS): 클래스 레벨에서 클래스 내부 전역 변수에 접근성을 측정한 Lack of Cohesion of Methods(LCOM)을 메소드 레벨로 확장한다.
- Method Cohesion(MCE): 우연/논리/시간/절차/교환/순차/기능적 응집도를 구분하고 평균을 측정한다.
- Dynamic Coupling Between Method(DCBM): CBM을 동적 환경에서 측정한다.
- Dynamic Method Coupling(DMCO): MCO를 동적 환경에서 측정한다.

3.2 클래스 레벨

- Class Coupling(CCO): 클래스 내부 메소드가 외부 메소드들과 갖는 결합도 MCO를 합산한다.
- Class Cohesion(CCE): 클래스 내부의 메소드 간의 결합도 MCO를 합산한다.

3.3 객체 레벨

- Dynamic Coupling Between Object Classes (DCBO): CBO를 동적 환경에서 측정한다.
- Dynamic Object Coupling(DOCO): 클래스 내부 메소드가 외부 메소드들과 갖는 동적 결합도 DMCO를 합산한다.
- DORC(Dynamic Object Relation Coupling): 동적 환경에서 클래스의 상속, 참조, 호출 관계를 기반으로 동적 결합도를 측정한다.
- DOCE(Dynamic Object Cohesion): 클래스 내부의 메소드 간의 동적 결합도 DMCO를 합산한다.

4. 자동 재사용성 식별 시스템

4.1 시스템 구성도

재사용성을 측정하여 재사용 모듈을 자동 식별하는 시스템을 구축한다. 그림 1은 시스템 구조도이다. 이는

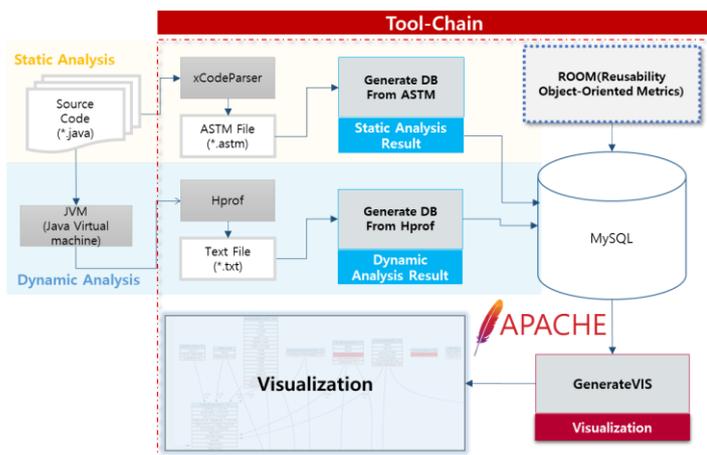


그림 1 시스템 구조도

자바 프로그램을 대상으로 개발되었으며, 정적 분석 도구 xCodeParser를 사용하여 ASTM파일을 추출하고, 동적 분석 도구 HPROF를 사용하여 프로그램 프로파일링 결과 텍스트 파일을 추출한다. 분석 결과를 데이터베이스에 저장하고 재사용 매트릭스를 측정하는 Generate DB from ASTM/Hprof를 구현한다. 그 결과는 Tabulator를 사용하여 웹 기반으로 가시화된다. 측정된 수치 및 재사용 모듈 식별 결과를 출력한다.

4.2 적용 사례

가시화 타겟 코드는 연구실에서 개발한 ‘신재생 에너지 발전소 모니터링 시스템’으로 로컬과 서버의 통신을 위해 이중 데이터의 메타모델을 사용하는 시스템이다[5]. 그림 2는 이 시스템의 재사용 모듈 자동 식별 결과이다. 자동 재사용성 식별 시스템에 의해 메소드, 클래스, 객체 레벨의 매트릭스가 계산되고 그 결과는 바 형태로 나타내어 어떤 수치가 높은지를 구분할 수 있다. 또한, 이 수치를 기반으로 해당 모듈이 재사용에 적합한지 마지막 열에 표기한다. 이 기준은 사용자가 임의로 설정할 수 있으며, 이 사례에서도 임의로 사용자의 요구를 정의하여 결과를 식별했다.

The screenshot shows a web browser displaying a 'Reusability Score' page. It contains two tables: 'Method' and 'Class'. The 'Method' table lists methods like 'getTime', 'setTime', etc., with columns for CBM, MCO, LCOS, MCE, DCBM, DMCO, and Reuse. The 'Class' table lists classes like 'BackgroundImage', 'ButtonExec', etc., with columns for CCO, CCE, and Reuse. The 'Object' table at the bottom lists objects like 'ClockLabel', 'CommonData', etc., with columns for DCBO, DOCO, DORC, DOCE, and Reuse. Green bars and checkmarks indicate high scores or reusability, while red 'X' marks indicate low scores or non-reusability.

그림 2 재사용 모듈 자동 식별

5. 결론 및 향후 연구

최근 급속한 시장 환경 변화와 빈번한 요구사항 변경으로 소프트웨어의 생산성을 높이지 못하고 있다. 이는 소프트웨어 품질의 문제로 직결된다. 이 문제의 방안들로 재사용성, 유지보수성, 품질, 생산성, 효율성 등에 대한 다양한 연구가 이루어졌다. 그 중에서도 재사용성은 소프트웨어 시장의 잦은 유지보수로 인한 품질 저하 문제를 해결할 수 있다.

기존 연구들은 재사용성의 정적 지표에 대해 초점이고 동적 지표의 연구는 부족했다. 이를 해결하고자, 기존 지표를 기반으로 확장/개선된 정적/동적 지표를 제안하였다. 이를 위해, 오픈 소스 기반으로 구축된 재사용 모듈 자동 식별 시스템을 구축하였다. 이는 동적 분석 도구를 사용하여 정적 분석에서는 얻을 수 없는 데이터를 수집하여 재사용성을 측정한다. 레거시 시스템의 확장 및 기능 개선으로 새로운 시스템을 구축할 경우, 식별된 모듈을 재사용하여 개발 비용 및 시간 절감과 품질 개선을 기대한다.

Acknowledge

본 논문은 2015년 교육부와 한국연구재단의 지역혁신창의인력양성사업의 지원을 받아 수행된 연구임(NRF-2015H1C1A1035548).

참고문헌

- [1] W Frakes and C. Terry, "Software reuse: metrics and models," ACM Computing Surveys, Vol. 28, No. 2, pp. 415-435, June. 1996.
- [2] Chidamber Shyam and Kemerer Chris, "A metrics suite for object oriented design," IEEE Transactions on Software Engineering, Vol. 20, No. 6, June. 1994.
- [3] DocForge, (Dec. 2017.) "Code reuse".
- [4] Marko Mijac and Zlatko Stacic, "Reusability Metrics of Software Components: Survey" in Central European Conference on Information and Intelligent, Varazdin, Croatia, pp. 221-230, September, 2015.
- [5] Hyun Seung Son, So young Moon, R. Young Chul Kim, "Replacing Source Navigator with Abstract Syntax Tree Metamodel(ASTM) on the open source oriented tool chains SW Visualization," In International Conference on Convergence Technology, Vol. 5, No. 1, pp. 366-367, 2015.