

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

1 월 31 일 (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.

국방 소프트웨어 자원관리시스템의 소스 코드 품질을 위한 자바 파서 개발을 통한 적용 사례

박보경* 문소영* 서채연* 김영철*[○] 김광남** 최영식** 신상훈**

*홍익대학교 소프트웨어공학 연구실, **육군 정보체계관리단

*{park, msy, chyun, bob[○]}@selab.hongik.ac.kr, **kma6471@hanmail.net,

**pmhcysbj@hanmail.net, **godshin2000@naver.com

An Applied Practice through developing Java Parser for Code Quality of the National Defense Software Resource Management System

Bo Kyung Park*, So-Young Moon*, Chae Yun Seo*, R. Young Chul Kim*[○],

Kwang-Nam Kim**, Young-Sik Choi**, Sang-Hoon Shin**

*SELab., Hongik University, **ROK Army

요 약

현재 육군에서는 고품질의 소프트웨어 개발을 위한 품질관리 방법을 적용하는데 어려움을 겪고 있다. 이를 위해, 육군에서는 소프트웨어 품질 개선이 가능한 소프트웨어 가시화 방법을 적용하였다. 하지만 이 방법을 국방 소프트웨어 자원관리 시스템에 적용하였으나 다음과 같은 문제가 발생하였다. 먼저, 1) 기존의 소스 네비게이터는 전자정부프레임워크에서 정확한 분석을 수행하지 못한다. 2) 소스 네비게이터로 분석된 정보들 중 육군에 불필요한 데이터(reference_class, Instance 등)가 다수 존재한다. 본 논문에서는 국방 소프트웨어 자원관리시스템에 적합한 파서 개발 및 적용 방법을 제안한다. 개발된 파서를 적용하여 국방 소프트웨어 자원관리시스템의 가시화를 수행하였다. 개선된 소프트웨어 가시화 방법을 통해, 고품질 소프트웨어 개발 및 유지보수성 향상을 높이고자 한다.

1. 서 론

국방 소프트웨어는 첨단 무기체계의 핵심요소로 운용되고 있으며, 전차, 함정, 전투기 등 주요 무기체계에서 차지하는 비중이 점차 증가하고 있다. 소프트웨어 중요성이 증가함에 따라, 육군에서는 고품질의 소프트웨어 개발을 위한 품질관리 방법을 적용하고 있다. 하지만 다음과 같은 문제로 인해 품질관리 적용에 많은 어려움을 겪고 있다. 1) 1인 1건의 개발 추진으로 개발 자 성향에 따라 개발 패턴이 다르기 때문에 복잡도의 일관성이 부족하다. 2) 요구사항이 지속적으로 변경됨에 따라 요구 분석, 설계, 구현의 전 과정이 일치하지 않는다. 3) 개발 후 운용 지원 부서로 개발업무가 이관되기 때문에, 운용 유지 측면에서 다양한 개발 패턴을 유형별로 분석하고 조치하는데 많은 시간이 소요된다. 이러한 문제를 해결하기 위해서는 소프트웨어의 비가시성 문제 해결이 가능한 소프트웨어 가시화 방법이 필요하다. 소프트웨어 가시화는 소프트웨어 개발의 과정과 구조를 파악함으로써 전문적이지 않더라도 고품질을 위한 소프트웨어 개발과 품질 관리가 가능하다[1,2]. 하지만 육군에서는 기존의 소프트웨어 가시화 방법을 적용하였으나, 여러 문제점이 발생하였다. 현재 육군의 개발 환경은 전자정부

프레임 워크 기반이다. 하지만 코드 분석 도구인 소스 네비게이터는 전자정부프레임워크에서 정확한 분석을 수행하지 못한다. 그 예로 부분적으로 소스 코드의 구문 파싱이 안 되며, if 문 같은 제어문을 분석하지 못한다. 이러한 문제를 해결하기 위해서, 육군 자원관리 시스템 분석을 위한 파서(Parser)를 개발하였다. 역공학을 통해 소프트웨어 품질 향상이 가능하도록 개발한 자바용 파서를 적용하여 소스코드를 구문 분석하였다. 이 방법을 통해 고품질 소프트웨어 개발 및 유지보수성 향상을 기대할 수 있다.

2. 국방 소프트웨어 자원관리의 소프트웨어 가시화를 위한 파서(Java Parser for Army, JPA) 개발

기존의 소프트웨어 가시화 방법은 소스 네비게이터를 기반으로 소스 코드를 분석한다. 소스 네비게이터는 다양한 언어를 지원하며, 코드 구문을 해석하여 소스 코드를 분석한다. 기존 방법을 국방 소프트웨어 자원 관리 시스템에 적용해 보았으나 다음과 같은 문제점이 발생하였다. 국방 소프트웨어 자원관리 시스템은 전자정부 프레임워크 기반이기 때문에 Annotation의 일부 파싱 오

* 본 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A3B03035421)

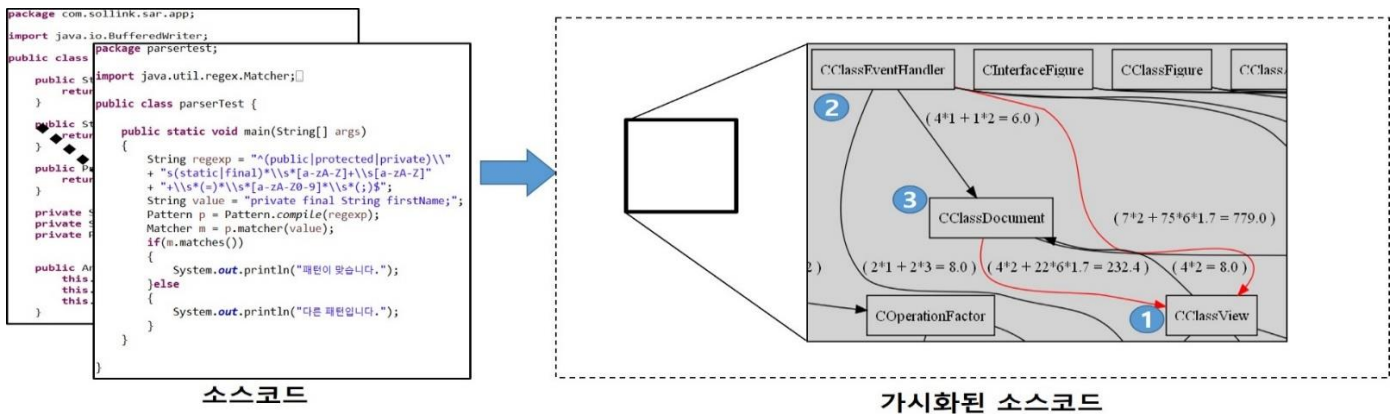


그림 2. 소프트웨어 가시화 결과

류가 발생하였다. 두 번째로 소스 네비게이터로 분석된 정보들은 SQLite에 저장되는데, DB에 축적된 데이터 중 불필요한 reference_class 및 Instance 정보가 다수 존재한다. 또한 저장된 데이터는 16진수 코드 값 및 # 등의 특수 문자로 저장되기 때문에 데이터를 분석하는데 많은 시간이 소요된다. 따라서 국방 소프트웨어 자원관리 시스템에 적합한 별도의 파서 개발이 필요하다.

2.1 파싱을 위한 정규 표현식

그림 1에서 소스 코드는 정규 표현식을 사용하여 분석한다. 자바에서 정규 표현식을 사용하기 위해서는 Pattern과 Matcher 클래스 객체가 필요하다[3]. Pattern 클래스의 compile() 메소드를 사용하여 정규표현식을 Pattern 객체로 생성한다. 컴파일된 된 정규식 패턴은 Matcher 객체로 생성되며, 입력된 문자열이 정의한 패턴에 일치하는지 여부를 판단하게 된다.

```

public boolean isAnnotation(String stringLine) {
    String packagePattern = "(@[A-Z]{1}[a-zA-Z]+\\s*(\\|O)*\\s*[=|'\"a-zA-Z|\\s|_0-9]*\\s*(\\|))\"";
    Pattern p = Pattern.compile(packagePattern);
    Matcher m = p.matcher(stringLine);
    boolean patternExist = m.find();
    if(patternExist)
    {
        System.out.println("패턴이 Amotatio Pattern Line : "+m.group(0));
    }
    return patternExist;
}
    
```

그림 1. 정규 표현식의 예제

2.2 비교 결과

표 1. 기존 시스템과 개발된 시스템 비교 결과[2]

항목	기존 시스템	개발된 시스템
산출물 검증	부분 가능	가능
소스코드 가시화	불가	가능
개발일정 관리	수작업	자동화
개발자의 Bad Smell 식별	불가	가능
복잡도 측정	불가	가능
소스코드 품질 개선	미비	가능

표 1은 기존에 구축한 시스템과 소프트웨어 가시화

구축을 통한 비교 결과이다. 기존 시스템에서 부분적으로 가능하거나 불가능했던 항목들이 JPA가 적용된 시스템에서는 산출물 검증, 복잡도 측정 등이 가능하였다.

그림 2는 소프트웨어 가시화 결과이다. 이 그래프는 JPA로 분석된 데이터를 기반으로 출력된 가시화 결과이다. 네모는 클래스를 나타내며, 클래스와 클래스를 연결하는 화살표는 클래스 간의 결합도를 의미한다. 그림 2에서 클래스 간의 높은 결합도를 살펴보면 다음과 같다. ① CClassView와 ② CClassEventHandler 간의 결합도 값은 779이며, ③ CClassDocument와 ① CClassView 간의 결합도는 232.4이다. 이 결합도들은 다른 클래스 간의 결합도 수치보다 높음을 알 수 있다. 따라서 이 2가지 결합도는 많은 코드 수정이 필요하다.

3. 결 론

본 논문은 국방 소프트웨어 자원관리 시스템의 아키텍처 가시화를 위한 파서 개발 및 적용방법을 제안한다. 기존 연구에서 사용하던 소스 네비게이터는 소스코드 파싱 오류로 인해 annotation에서 method 인식 오류 및 제어문 미인식 등의 문제가 발생하였다. JPA 파서는 정규표현식을 이용하여 소스 코드를 분석하기 때문에 소스 네비게이터의 문제점을 보완할 수 있었다. 이 파서를 적용함으로써, 국방 소프트웨어 자원관리 시스템에 필요한 소프트웨어 가시화에 적용하여 올바른 소프트웨어 개발 및 유지보수성 향상을 기대할 수 있다. 또한 개발된 파서는 각 개발 환경에 맞게 변경 가능하기 때문에, Java 외에도 다른 언어에 적용 가능하다.

참고문헌

[1] NIPA SW 공학센터, “SW 개발 품질관리 매뉴얼 (SW Visualization)”, 2013.12.
 [2] 박보경, 문소영, 서채연, 김영철, “국방 소프트웨어 자원관리시스템의 소스 코드 품질 향상을 위한 소프트웨어 가시화 적용 구축 사례”, 2017 한국정보과학회 동계학술대회, 2017.
 [3] <https://docs.oracle.com/javase/tutorial/essential/regex/>