

ISSN 2287-4348
Vol. 5 No. 1

한국스마트미디어학회 & 한국전자거래학회
2016 춘계학술대회 논문집

P R O C E E D I N G S

일시 : 2016. 04. 29 (금) ~ 30 (토)
장소 : 신라대학교 종합강의동

주최 : 한국스마트미디어학회
한국전자거래학회
신라대학교

포스터 발표순서 (4월 30일 토요일 13:00-14:30)

P18 **제목 :** K-평균 클러스터링 기반 앞서귀 질병 검출
272Page **저자 :** 박정현, 이성근, 고진광(순천대)

P19 **제목 :** Spark기반의 농업 빅데이터 분석 플랫폼 설계
275Page **저자 :** 뉘엔 신 녹, 뉘엔 반 퀴엣, 김경백(전남대)

P20 **제목 :** P07(의사소통능력):프로그램 학습성과 평가 체계 모형 연구
279Page **저자 :** 오수열(목포대)

P21 **제목 :** IoT 기반 지능형 환기 히팅 시스템 설계
283Page **저자 :** 조동욱, 김민지, 이성근(순천대), 김강이((주)제노텍)

P22 **제목 :** 저전력 관련 코드 메카니즘과 소프트웨어 가시화 접목
286Page **저자 :** 이근상, 김영철(홍익대)

P23 **제목 :** 역공학을 통한 소스 코드로부터 유스케이스 설계 추출
289Page **저자 :** 권하은, 박보경, 김영수, 박지훈, 김영철(홍익대)

P24 **제목 :** 기존 오픈 소스 도구들 비교 분석을 통한 정적 분석 및 가시화 도구 구축
292Page **저자 :** 서채연, 박보경, 변은영(홍익대), 박용범(단국대), 김영철(홍익대)

P25 **제목 :** 프로파일러를 이용한 소프트웨어 메모리 성능 가시화 방법
296Page **저자 :** 강건희, 이진협(홍익대), 이근상(전북 TP), 김영철(홍익대)

P26 **제목 :** 의사결정기법(AHP)을 통한 한국형 테스트 성숙도 모델 요소의
적절성 검증에 관한 연구
298Page **저자 :** 박보경, 변은영(홍익대), 김기두(한국정보통신기술협회), 김영철(홍익대)

역공학을 통한 소스 코드로부터 유스케이스 설계 추출

권하은^{1*}, 박보경^{2*}, 김영수^{4*}, 김영철^{3*}

¹홍익대학교 컴퓨터정보통신공학과 소프트웨어공학연구소

e-mail : {¹kwon, ²park, ³bob}@selab.hongik.ac.kr

Nipa 소프트웨어공학센터^{4*}

ysgold@nipa.kr^{4*}

Extracting Use Case Design from Source Code based on Reverse engineering

Haeun Kwon^{1*}, Bokyoung Park^{2*}, R. Youngchul Kim^{3*}, Young S. Kim^{4*}

¹SE lab, Hongik University,

Nipa SE center^{4*1)}

요 약

국내 소프트웨어 개발의 대부분이 중소기업에 이루어 있으나[1], 중소기업의 경우 미비한 설계 문서를 토대로 개발을 진행하거나, 설계 문서가 낙후되어 코드가 문서의 불일치가 발생하고 있다. 이로 인해 유지보수 비용의 증가가 야기된다[2]. 이를 해결하기 위해 역공학 기반 소스 코드로부터 모든 설계 문서 추출이 필요하다. 과거 연구[3,4,5]에서는 Tool-chain을 구성하여 구조적 관점과 행위적 관점에서 설계 문서를 추출하였다. 본 논문은 이를 확장하여 추출된 설계 도면으로부터 유스케이스 다이어그램을 추출한다. 이런 설계로 리팩토링을 수행한다. 즉 '설계→구현→설계'의 순환을 제안한다. 이를 통해 설계에 충실한 코드를 얻는 것이 가능하다. 향후는 Function Point 기반 소프트웨어 비용 산정 기법 대신에 Use Case Point (UCP) 적용과 비용 산정 후에 검증에 기초가 될 것이다.

*

1. 서 론

국내 소프트웨어 개발의 대부분이 중소기업에 이루어진다[1]. 그러나 중소기업의 경우 미비한 설계 문서를 토대로 개발을 진행하거나, 설계 문서가 낙후되어 코드가 문서의 불일치가 발생하고 있다. 이는 소프트웨어 구조 파악에 많은 시간을 요구하게 되며, 결과적으로 유지보수 비용의 증가를 일으킨다[2].

이를 해결하기 위해 역공학을 통한 코드로부터 설계 문서 추출이 필요하다. 과거 연구[3,4,5]에서는 여러 공개 소프트웨어를 통해 Tool-chain을 구성하여 구조적 설계 도면인 클래스 다이어그램과 패키지 다이어그램, 행위적 설계 문서인 순차 다이어그램을 추출하였다. 본 논문에서는 이를 확장하여, 추출된 설계 도면을 토대로 유스케이스를 추출한다. 이를 통해 순공학(설계→구현)을 통해서만 얻을 수 있는 정보와, 역공학(구현→설계)을 통해서만 얻을 수 있는 정보를 둘 다 얻는 것이 가능하다[6]. 최종적으로 '설계→구현→설계'의 순환을 통해 설계에 충실한 코드를 얻고자 한다.

본 논문의 구성은 다음과 같다. 1장에서는 연구배경에 대해 언급한다. 2장에서는 과거 연구인 구조적 및 행위적

설계 추출 프로세스에 대해 기술한다. 3장에서는 과거 연구를 확장하여 유스케이스 추출 매커니즘에 대해 언급한다. 4장에서는 이에 대한 적용 사례를 보인다. 마지막 5장에서는 결론과 향후 연구 방향에 대해 언급한다.

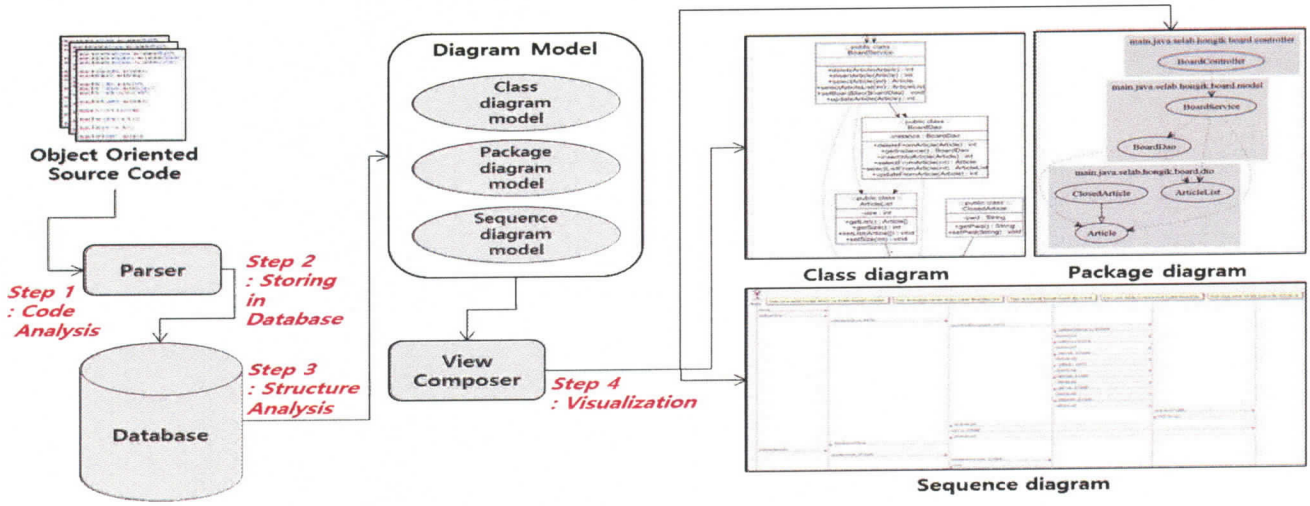
2. 구조적 및 행위적 설계 추출 프로세스

과거연구[3]에서는 여러 공개 소프트웨어를 엮어 Tool-chain을 구성하고, 객체지향 코드로부터 설계 도면을 추출하였다. 설계 도면으로는 구조적 설계 도면인 클래스 다이어그램과 패키지 다이어그램을 추출하였고, 행위적 설계 도면인 순차 다이어그램을 추출하였다. 그림 1은 이러한 설계 도면 추출 프로세스를 나타낸다. 전체 프로세스는 네 가지 단계로 구분된다. 첫 번째 코드 분석 단계에서는 파서(Parser)를 통해 코드를 분석하고 구성 요소로 분리한다. 두 번째 데이터베이스 저장 단계에서는 코드 구성 요소를 테이블에 분류한다. 세 번째 구조 분석 단계에서는 분류된 정보를 토대로 각 다이어그램 모델에 필요한 정보를 채워 넣는다. 마지막 가시화 단계는 가시화 도구(View Composer)를 통해 다이어그램 모델을 이미지 파일로 가시화한다.

3. 코드로부터 유스케이스 추출 매커니즘

앞에서 언급한 구조 분석 단계에서 생성되는 다이어그램 모델들을 통해 유스케이스를 추출한다. 각 다이어그램 모델은 다이어그램을 구성하는 정보들을 갖고 있는데, 이

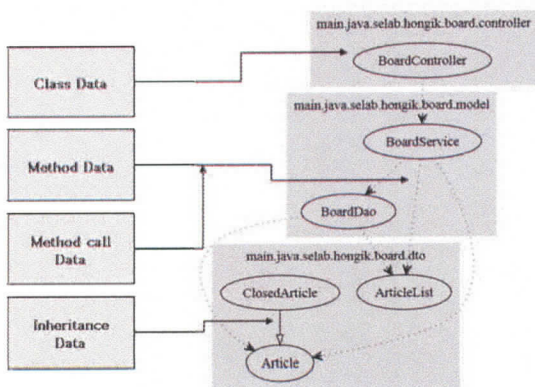
1) 본 연구는 2015년 교육부와 한국연구재단의 지역혁신 창의인력양성사업(NRF-2015H1C1A1035548)과 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2011601).



(그림 1) 구조적 및 행위적 설계 추출 프로세스 구조

러한 정보들을 토대로 유스케이스를 추출한다.

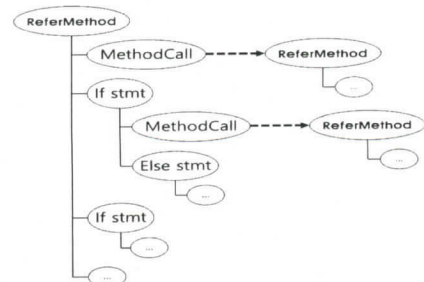
그림 2는 클래스 다이어그램과 패키지 다이어그램을 추출하기 위한 클래스 정보, 메소드 정보, 메소드 호출 정보, 상속 정보를 나타내고 있다. 그리고 이들 정보를 토대로 클래스, 패키지, 그리고 일반화 관계나 의존 관계 및 연고나 관계를 추출하는 것이 가능하다. 그림 3는 순차 다이어그램을 추출하는데 사용되는 메소드 호출 모델을 나타낸다. 먼저 ReferMethod 노드는 참조하는 메소드를 의미하고, MethodCall 노드는 참조되는 메소드를 의미한다. If stmt 노드와 Else stmt 노드는 분기 구문을 나타낸다. 메소드 호출 모델은 ReferMethod를 루트 노드로 하는 계층 구조를 가지는데, If stmt, Else stmt, Method Call 노드를 자식 노드로 가진다. 마찬가지로 If Stmt와 Else stmt 노드도 자식 모드를 가질 수 있으나, MethodCall은 자식 노드를 가질 수 없다. 대신에 다른 ReferMethod와 매핑될 수 있으며, 이로 인해 메소드 호출 트리를 탐색하기 위해 깊이 우선 탐색을 수행한다.



(그림 2) 클래스 다이어그램과 패키지 다이어그램 추출 매커니즘

이러한 메소드 호출 트리를 패키지 단위로 구분하고, 이를 세 가지 유스케이스의 유형으로 분류하였다. 유스케이

스의 유형은 유스케이스 다이어그램에서 액터만 접근 가능한 유스케이스, 다른 유스케이스를 통해서만 접근 가능한 유스케이스, 그리고 액터와 유스케이스 모두로부터 접근 가능한 유스케이스로 분류한다. 먼저 액터에서만 접근 가능한 유스케이스는 ReferMethod 노드가 다른 어떤 MethodCall 노드와 매핑되지 않는 경우에 해당한다. ReferMethod 노드가 MethodCall 노드와 매핑된다는 것은 다른 모듈로부터 호출되는 것을 의미하므로 액터만 접근하는 경우에는 매핑되어서는 안 된다. 이는 유스케이스 관계 중 연관(Association) 관계에 해당된다. 나머지 두 가지 유스케이스 유형은 각각 포함(Include), 확장(Extend), 일반화(Generalization) 관계와 연관이 있다. 이는 보다 상세한 구분 방법이 필요하다. 본 논문은 첫 번째 유형의 유스케이스를 추출만 구현한다. 이를 위해 소스 코드로부터 추출되는 모든 메소드 호출 트리에 대해 ReferMethod 노드와 MethodCall 노드의 매핑을 수행한다. 그 결과 다른 어떤 MethodCall 노드와도 매핑되지 않는 ReferMethod 노드를 추려낸다. 이는 소스 코드 내부가 아닌 외부에서 호출되는 메소드이다. 다시 말해, 외부 요인에 의해 최초로 호출되는 메소드이므로, 유스케이스의 한 가지 유형으로 볼 수 있다. 최종적으로 추려낸 유스케이스를 그림 1의 View Composer가 인식 가능한 스크립트로 명시하고, 이를 다이어그램으로 가시화한다.



(그림 3) 메소드 호출 트리 구조

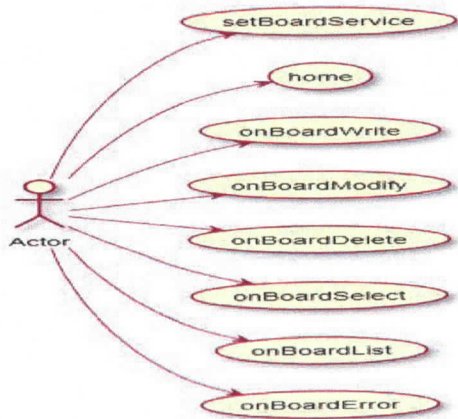
4. 적용 사례

제안한 프로세스를 게시판 어플리케이션에 적용한다. 그리고 게시판 어플리케이션 개발 당시 작성된 유스케이스 다이어그램과 추출된 유스케이스 다이어그램을 비교분석한다. 그림 4는 설계 도면으로 작성된 게시판 어플리케이션의 유스케이스 다이어그램이며 'Write', 'Attach', 'Delete', 'Modify', 'List', 'Read', 'Search'의 일곱 개의 유스케이스를 갖는다. 각각의 유스케이스는 '글 쓰기', '파일 첨부', '글 삭제', '글 수정', '목록 확인', '글 읽기', '검색'의 기능을 의미한다.



(그림 4) 설계 당시 유스케이스

그림 5는 제안한 프로세스를 통해 추출된 유스케이스 다이어그램이다. 표 1은 그림 3의 유스케이스와 그림 4의 유스케이스를 기능 별로 비교분석하여 정리한 것이다. '글 쓰기', '글 수정', '글 삭제', '글 읽기'의 기능에 대해서는 1:1로 유스케이스가 매핑된다. '목록 확인'의 경우에는 추출된 유스케이스가 두 개의 유스케이스가 추출되는데, 이는 코드 상에서 '목록 확인'에 대한 기능을 수행하는 메소드가 중복해서 존재하기 때문이다. 반면에 '검색'과 '파일 첨부' 기능은 코드로 구현되지 않았기 때문에 유스케이스가 추출되지 않았다. 마지막으로 '에러 페이지' 기능은 개발 과정에서 에러 로그(Log)를 확인하기 위한 기능으로 설계 당시에는 고려하지 않았던 것이다.



(그림 5) 추출된 유스케이스

(표 1) 유스케이스 비교

기능	기존 유스케이스	복원된 유스케이스
글 쓰기	Write	onBoardWrite
글 수정	Modify	onBoardModify
글 삭제	Delete	onBoardDelete
글 읽기	Read	onBoardSelect
목록 확인	List	home, onBoardList
검색	Search	-
파일 첨부	Attach	-
에러 페이지	-	onBoardError

5. 결론 및 향후 연구

본 논문의 목적은 객체지향 코드로부터 자동으로 설계 문서를 복원하는 것에 있다. 이를 위해 역공학 기법을 통해 구조적 관점과 행위적 관점의 두 가지 관점에서 코드를 분석하고, 최종적으로 유스케이스 다이어그램을 추출한다. 또한 유스케이스를 세 가지 유형으로 구분하였고, 이중 한 가지 유형에 대한 유스케이스 추출 사례를 보였다. 추후에는 나머지 두 가지 유형의 유스케이스 추출에 대한 방법과 포함 관계, 일반화 관계와 같은 유스케이스 간 관계를 추출하는 방법에 대해 연구하고자 한다. 향후, Use Case Point (UCP) 적용과 비용 산정 후에 검증에 연구할 예정이다.

참고 문헌

[1] NIPA SW공학센터, 「SW공학백서」, 2014.
 [2] Nicolas Anquetil, Timothy Lethbridge, "Extraction Concepts from File Names; a New File Clustering Criterion", IEEE, 1998.
 [3] 권하은, 박보경, 김영철, "객체 기반 프로그래밍으로부터 구조적 및 행위적 설계 자동 추출", 「한국스마트미디어 추계학술발표논문집」, Vol.4, No.2, pp.129-131, 2015.
 [4] 권하은, 박보경, 이근상, 박용범, 김영수, 김영철, "코드 가시화부터 모델링 추출을 통한 역공학 적용", 한국정보처리학회, 제 21권, 제 2호, pp.650-653, 2014.
 [5] 박보경, 권하은, 양효석, 문소영, 김영수, 김영철, "객체지향 코드의 정적 분석을 위한 Tool-chain화 사례 연구", 「한국정보과학회 학술대회 논문집」, pp.463-465, 2014.
 [6] Chikofsky, Elliot J., and James H. Cross, "Reverse engineering and design recovery: A taxonomy." Software, IEEE 7.1, pp.13-17, 1990.

한국스마트미디어학회 & 한국전자거래학회
2016 춘계학술대회 학술발표 논문집
Proceedings of KISM & SEB Spring Conference 2016

제 5권 제 1호
2015년 4월 25일 발행

발행인 / 차준섭, 김훈태 대회장

편집인 / 김병기, 이홍주, 이성근, 김영철 학술위원장

발행처 / (사) 한국스마트미디어학회

광주 남구 송암로 60 광주CGI센터 기업동 309호 (송하동)

전화 : 062)655-3507 / 팩스 : 062)655-3510

홈페이지 : www.kism.or.kr

E-Mail : kism1122@kism.or.kr

디자인 및 편집 / 장 영 우(한국스마트미디어학회 연구원)

후원 / LG CNS, 아이티센, NICE R&C, 콤텍시스템, 비온시이노베이터

한국IT비즈니스진흥협회, 소프트캠프(주), (주)피앤피시큐어, 함소아한의원

중앙대 의료보안연구소

2016
SPRING
CONFERENCE
OF
KISM & SEBS



Smart Media
KOREAN INSTITUTE OF SMART MEDIA



Society for
e-Business Studies



NICE R&C 주식회사
NICE Research & Consulting, Inc



BonC Innovators



함소아 한의원