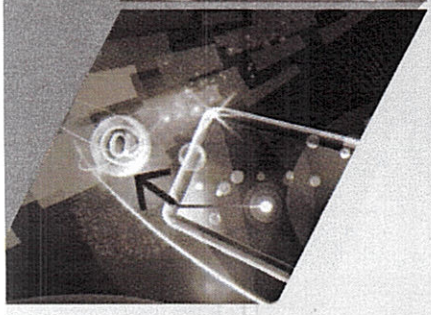
한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

http://www.kiise.or.kr

2014 한국컴퓨터종합학술대회
Korea Computer Congress 2014

"사물 인터넷 시대의 SW 기술"



2014년 6월 25일(수)~27일(금)
부경대학교&해운대그랜드호텔

- 자연어처리 및 정보검색 최근 동향 워크샵(6.25)
- HPC 최근 연구 동향 워크샵(6.25)
- 데이터베이스 최근 연구 동향 워크샵(6.25)
- GPU Standard API Workshop(6.25)
- 인공지능 최근 연구 동향 워크샵(6. 26)

|후 원|

플래티넘 NAVER KOFST

골 드 SAMSUNG 삼성전자 SAMSUNG 삼성테크윈 btc 부산관광공사

실 버 SAMSUNG 삼성SDS 신한데이타시스템 SHINHAN DATA SYSTEM

브론즈 hancom 한글과컴퓨터 KCC정보통신 We pave your way. Suresoft

193. 절차식 언어 기반의 코드 정적 분석을 위한 툴 체인 사례 연구 ..... 강건희 · 이근상 · 김동호 · 황준순 · 김영수 · 박용범 · 김영철 559
194. CVL 기반의 소프트웨어 프로덕트 라인을 위한 테스트 전략 ..... 이주석 · 서용진 · 천은영 · 김수지 · 김진아 · 김현수 562
195. 코드 기반 소프트웨어 변경 영향 분석 도구 비교 ..... 전형민 · 김태연 · 채흥석 565
196. 시큐어 임베디드 소프트웨어 개발을 위한 위협 모델링 기법 ..... 이진호 · 황대연 · 최진영 568

언어공학

197. 구기반 통계적 모델을 이용한 한국어 형태소 분할 및 품사 태깅 ..... 나승훈 · 김영길 571
198. [우수논문] Structural SVM 기반의 한국어 의미역 결정 ..... 이창기 · 임수종 · 김현기 574
199. 통합적 방식을 이용한 한국어 문맥의존 철자오류 교정규칙의 재현율 향상 ..... 최현수 · 윤애선 · 권혁철 577
200. [우수논문] 어절 N-gram을 이용한 문맥의존 철자 오류 교정 ..... 김민호 · 최성기 · 권혁철 580
201. 의미사전 기반의 한국어-한국수화 기계번역 ..... 정현영 · 민은주 · 안성주 · 오영준 · 이종혁 583
202. 시간과 문체 자질을 이용한 한국어 게시판에서의 저자 식별 ..... 김경훈 · 윤희근 · 박성배 586
203. 식품 안전 관련 트윗의 의도 분류를 위한 기계학습 알고리즘 성능 비교 연구 ..... 염하늘 · 황명권 · 황미녕 · 정한민 589
204. 한국어 의미역 말뭉치 구축을 위한 반자동 태깅 도구 개발 ..... 배장성 · 오준호 · 박천음 · 최경호 · 이창기 592
205. 순차적 레이블링을 이용한 한국어 의미격 인식 ..... 임수종 · 김현기 595
206. SVM 기반의 Mention Pair Model을 이용한 한국어 상호참조해결 ..... 최경호 · 박천음 · 이창기 598
207. 조건부 랜덤 필드를 이용한 절 경계 발견 ..... 신재훈 · 이종혁 601
208. RankSVM을 이용한 음성채팅시스템의 성능 향상 ..... 안혁주 · 송영길 · 김학수 604
209. 한국어 어휘 의미망을 이용한 통계적 문맥 의존 철자오류 교정 성능 향상 ..... 서한영 · 최성기 · 권혁철 607
210. 서답형 문항의 채점을 위한 모델 자동 구축 방법 ..... 장은서 · 강승식 610
211. CRF를 이용한 특허 개체명 인식 ..... 이태석 · 전홍우 · 강승식 612
212. Distant Supervision을 이용한 관계 추출과 개체명 인식의 상호 보완 기법 ..... 이우철 · 강상우 · 서정연 614
213. 예시문서의 주제어 그래프를 이용한 구조기반 문서 탐색 ..... 탁해성 · 조환규 617
214. 빅(Big)데이터 감성단어 분석을 통한 시청률 분석 ..... 권상희 · 최윤정 620
215. 다국어 서브토픽 마이닝의 결과 시각화를 위한 검색 의도 기반 서브토픽 분류 방법 제안 ..... 김세종 · 이종혁 623

인공지능

216. Automated Quality Analysis for Classification of Rice Grains ..... Joystna.G.Anandache · Sung-Hwan Jung 626
217. 그리드 단체 위의 Dirichlet 분포에서의 MCMC 표집 ..... 신봉기 629
218. [우수논문] 통계적 얼굴 모델을 이용한 부분 폐색을 가진 얼굴 검출 ..... 서정인 · 박혜영 632

절차식 언어 기반의 코드 정적 분석을 위한 툴 체인 사례 연구

강건희[O1] 이근상[1] 김동호[1] 황준순[1] 김영수[2] 박용범[3] 김영철[1]
[1]홍익대학교 컴퓨터정보통신공학과 소프트웨어공학연구실
[2]정보통신산업진흥원 소프트웨어공학센터 [3]단국대학교 컴퓨터과학과
[1]{kang, yi, ray, hwang, bob}@hongik.ac.kr, [2]ysgold@nipa.kr, [3]ybpark@dankook.ac.kr

A Practical Study on Tool Chain for Code Static Analysis on Procedural Language

Geon-Hee Kang[O1] Keunsang Yi[1] DongHo Kim[1] Junsun Hwang[1] Youngsoo Kim[2] Young B. Park[3] R. Young Chul Kim[1]
[1]SE Lab, Dept. of Computer Information Communication, Hongik University
[2]National IT Industry Promotion Agency
[3]Dept, of. Computer Science, Dankook University

요 약

국내의 소프트웨어 산업계는 고 품질을 개발/테스트 프로세스, 성숙도 측정 등에 초점을 두고 있지만 실제 산업현장에서는 코드중심으로 개발되고 있다. 기존 코드중심의 개발은 설계를 포함하지 않으므로 코드 내부의 복잡도가 높다. 그러므로 산업현장에서는 프로세스, 성숙도 측정 등과 같은 SW 공학 기법보다는 코드의 가시화(visualization)가 더 중요하다. 본 논문에서는 절차식 언어인 C 코드의 가시화를 위해 기존 도구로 툴 체인 구성 방법을 제안한다. 제안한 방법은 NIPA의 SW Visualization 기법을 절차식 언어에 적용 및 확장한 것이다. 최종적으로 레가시 코드에 대해 역 공학 기법에도 적용이 가능하리라 본다. 앞으로 기존의 소스코드 구조의 가시화를 통한 소프트웨어 품질관리가 가능하다.

1. 서 론

정보통신산업진흥원(NIPA)의 '2013 SW공학백서[1]'에 따르면 프로세스·품질인력·공학기술을 종합한 국내 기업의 소프트웨어공학 품질수준은 100점 만점에 64.8점으로 최하위 등급인 '열등'을 간신히 벗어난 정도이다.

또한 국내 기업의 검증 역량이 부족한 이유는 그동안 소프트웨어 사업이 시스템통합(SI) 중심으로 수행 되어왔기 때문이다. 그래서 소프트웨어개발에서 소프트웨어 공학적으로 접근하지 못하였다. 구현에만 치우쳐져 있어 소프트웨어 품질의 중요성은 인지하지만 좋은 소프트웨어품질과는 동떨어져 있는 개발을 하고 있다[1].

이러한 현재 상황을 극복하고 소프트웨어품질을 향상시키기 위해서는 시스템 개발에 필요한 각종 설계, 문서 등을 코드로부터 추출할 수 있는 역 공학 기법의 적용이 필요하다. 기존의 레가시 코드를 통해 원래 시스템의 설계도면을 추출하고 역 공학적으로 고품질의 소프트웨어를 개발이 필요하다.

그래서 역 공학 기법을 통하여 기존코드를 분석하고 소프트웨어의 구조를 가시화하기 위해서 Source Navigator[4]에서 추출이 되는 소프트웨어정보에 대해 분석을 하였고 기존에 있던 툴 체인을 적용 타깃에 맞게 코드 수정을 하고자 한다. 결과적으로는 해당 프로그램 소스 코드에 대한 구조를 가시화 한 이미지를 얻고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 NIPA 소프트웨어 공학센터 에서 제안한 S/W 가시화(Visualization)와 역 공학을 설명 하고, 3장에서는 정적분석을 위한 툴 체인화를 설명을 언급한다. 4장에서는 툴 체인에 대한 적용사례를 보여준다. 5장에서는 결론 및 향후 연구에 대해서 언급한다.

* 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업원천기술개발사업[10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발]과 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2011601).

2. 관련연구

2.1 S/W 가시화(Visualization)

성공적인 소프트웨어개발 관리를 위해서는 소프트웨어 자체 즉 소스 코드와 소프트웨어 개발 프로세스에 대한 관리가 필요하다.

소프트웨어공학 프로세스는 이러한 관리를 위한 전통적인 방법이다. 그러나 소프트웨어공학 프로세스에 의한 소프트웨어개발 품질관리를 수행하기에는 그 방법이 너무나 전문적이기 때문에 인력과 비용이 많이 부족하다. 그래서 NIPA의 소프트웨어 Visualization은 소프트웨어의 역 공학을 통한 시각화와 문서화를 제안하였다.[2]

첫째, 시각화는 소프트웨어 개발의 가장 어려운 점인 소프트웨어 비가시성을 극복함으로서 전체 소프트웨어개발의 과정을 파악 할 수 있도록 하여 소프트웨어개발 품질관리를 수행하는 방안이다. 둘째, 문서화 는 기업 혹은 단체의 개발 노하우 관리 및 내부 인력간의 업무 이해도

향상과 특정 상황에서 외부와의 의사소통을 위한 방안이다.

결과적으로 SW Visualization은 소스코드와 개발 프로세스를 관리 하는 것이 목적이며, 시각화와 문서화로 소프트웨어개발 품질관리를 수행하게 된다[2].

2.2 역 공학(Reverse Engineering)

순 공학 프로세스가 요구사항 분석, 설계, 구현, 테스트, 유지보수의 순서를 따른다면 역 공학 프로세스는 반대로 구현된 프로그램에서 설계문서를 설계문서에서 요구사항을 추출한다[2].

3. 코드 정적 분석을 위한 툴 체인화

코드 정적분석을 위해 해당 프로그램 소스코드를 파서를 이용해서 분석한다. 하지만 실제로 파서를 구현을 하기는 쉽지 않다. 본 연구에서는 Source Navigator[4] 와 DOT[5], SQLite[6]를 이용하여 아래 그림1과 같은 구조로 툴 체인을 구성하였다.

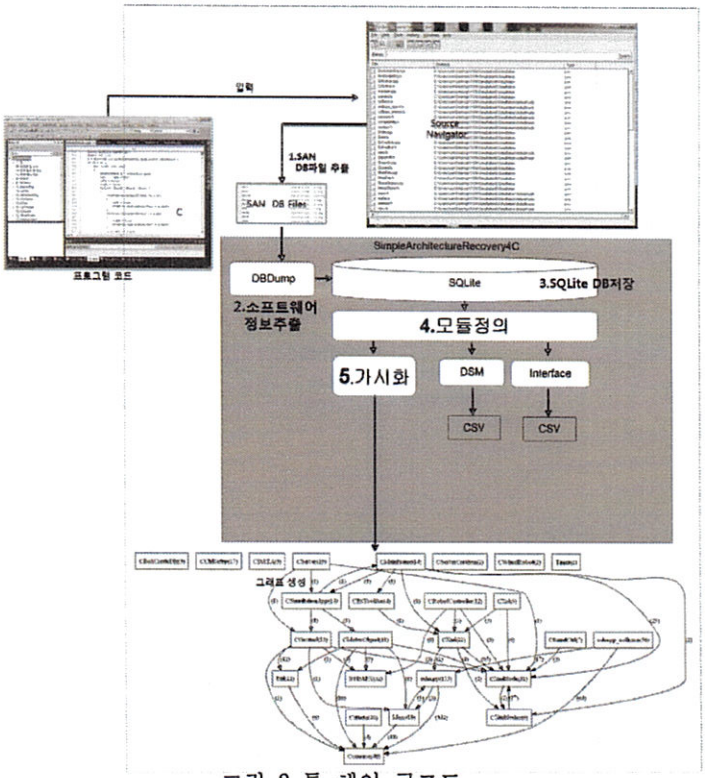


그림 2 툴 체인 구조도

코드정적 분석은 그림 2와 같이, SAN DB파일 추출-> 소프트웨어의 정보 추출 -> SQLite[6] DB저장-> 모듈지정-> 가시화 총 5단계를 거치게 된다.

- 1단계 : SAN DB파일 추출은 툴 체인을 통해 분석을 하려면 먼저 소프트웨어에 대한 정보가 필요하다. 소프트웨어의 정보는 프로그램 소스코드를 Source Navigator[4]에 입력하고 해당 프로그램 코드에 대한 전반적인 정보(함수의 정보, 지역변수, 전역변수, 파라미터 등)를 SAN DB파일로 출력된다.
- 2단계 : 소프트웨어의 정보추출은 Source Navigator에서 추출한 SAN DB파일이 바이너리로 구성되어, DBDUMP라는 파일을 이용하여 바이너리 정보를 다시 추출한다.
- 3단계 : 추출된 소프트웨어정보를 목적에 맞는 정보만을 얻기 위해 SQLite에 저장을 한다.
- 4단계 : 모듈정의 에서는 소프트웨어의 구조를 가시화하기 위해 툴 체인 내부에서 타깃에 대한 모듈을 기능적으로 정의를 한다.
- 5단계 : 가시화 에서는 SQLite에 저장된 정보로 DOT 스크립트를 만들어 툴 체인을 통해 실행하면 모듈 간의 관계 와 전체적인 타깃의 아키텍쳐 구조를 그리게 된다.

결과적으로 정적 분석을 위해 Tool들의 체인화를 통해 코드를 가시화 시켰다.

4. 사례연구

본 연구는 직접 개발된 6족 로봇 시뮬레이터를 적용타깃으로 사용한다. 6족 로봇 시뮬레이터에서는 다관절 로봇의 동작제어를 도구 내의 컨트롤러로 쉽게 할 수 있다. 그리고 실제로 로봇이 존재하지 않더라도 도구 내의 시뮬레이션을 통하여 실제 로봇의 동작을 예측이 가능하다. 즉, 시뮬레이션 로봇이 동작할 환경을 구축한 후 다관절 로봇의 모션을 입력하면 해당 환경에서 동작을 한다. 모델링 도구인 HIMEM과 네트워크로 연결이 가능하다. 이를 통해, Pre-Testing(선 테스팅) 방법을 제안했다[3]. 아래의 그림 3은 시뮬레이터의 실행화면이다.

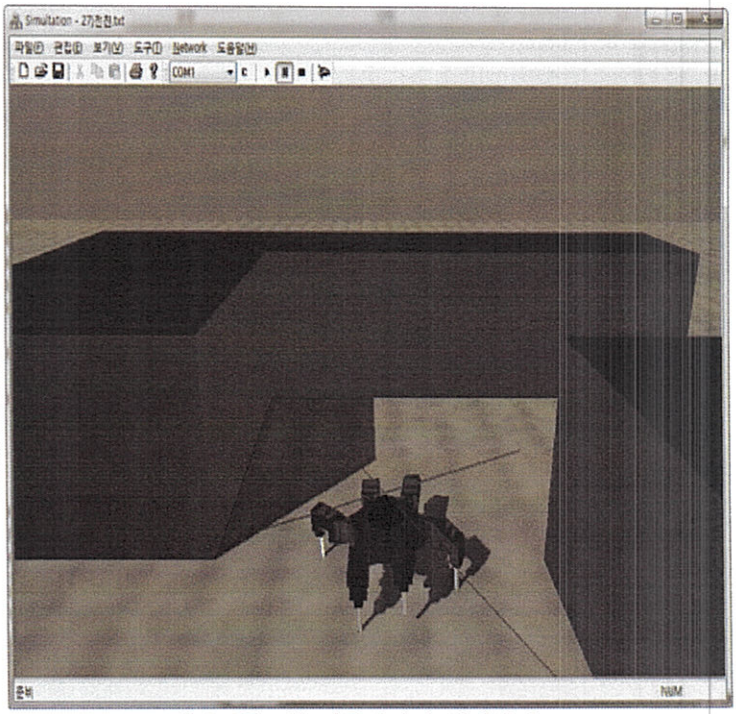


그림 3 6족로봇 시뮬레이션 실행화면

그림4,5는 그림 2의 5번 가시화 단계의 SQLite

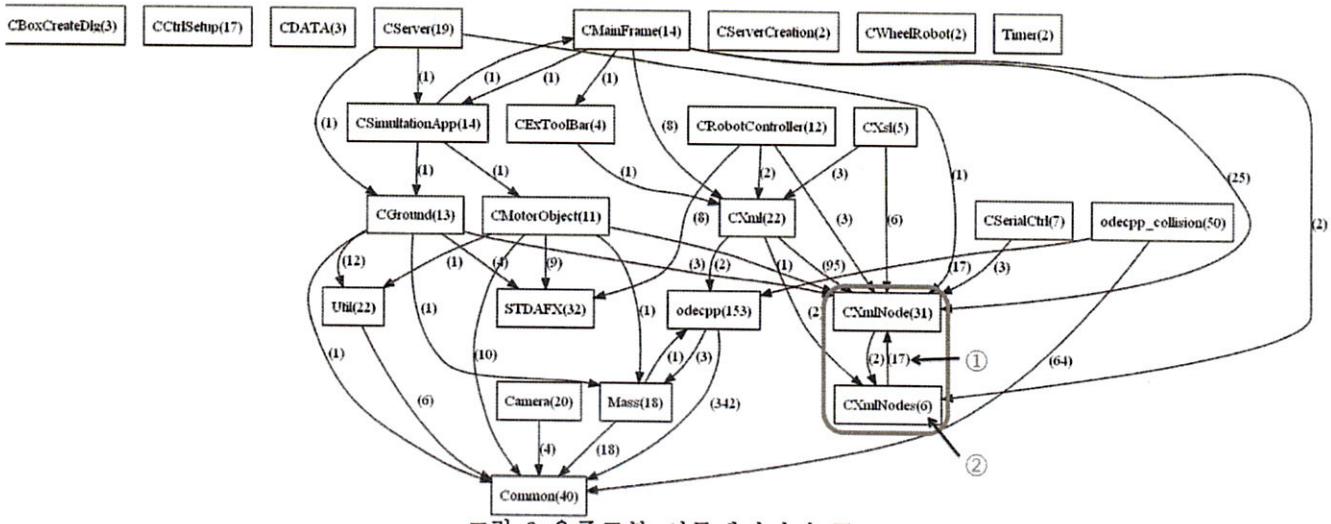


그림 6 육족로봇 시뮬레이터의 구조도

DB내에서 모듈 간의 관계를 숫자로 표현하기 위해 정보를 추출하기 위해 사용하는 쿼리문 이다.

```
String module_query = "select MODULE_NAME, count(*) as CNT
from MODULE_MAPPING where " +
" PROJECT_NAME='" + project_name +"' and ARCHITECTURE_NAME='"
+ architecture_name
+ "' group by MODULE_NAME";
```

그림 4 모듈정보 쿼리문

그림 4는 툴 체인 내부에서 정의된 모듈의 정보(모듈의 이름, 모듈이 맵핑되는 수)를 불러오는 쿼리문 이다.

```
String dependency_query = "select AM.MODULE_NAME as FROM_MODULE,
BM.MODULE_NAME as TO_MODULE , count(*) as CNT "
+ "from MODULE_MAPPING as AM, LINK, MODULE_MAPPING as BM where "
+ "AM.PROJECT_NAME ='" + project_name +"' and
AM.ARCHITECTURE_NAME ='"+architecture_name+"' and "
+ "BM.PROJECT_NAME = AM.PROJECT_NAME and BM.ARCHITECTURE_NAME =
AM.ARCHITECTURE_NAME and "
+ "LINK.PROJECT_NAME ='" + project_name +"' and "
+ "AM.TYPE_NAME = LINK.FROM_NAME and "
+ "LINK.TO_NAME = BM.TYPE_NAME "
+ "and AM.MODULE_NAME <> BM.MODULE_NAME "
+ "group by AM.MODULE_NAME, BM.MODULE_NAME";
```

그림 5 의존도 쿼리문

그림5는 AM모듈과 BM모듈 사이에 서로 얼마나 호출이 됐는지의 횟수정보를 불러오는 쿼리문이다.

그림 6은 6족로봇 시뮬레이터의 구조도이다. 이 구조도는 타깃의 확실한 모듈이 정의가 되지 않았기 때문에 툴 체인 내부에 임의의 모듈을 정의를 해놓고 DOT Script를 이용하여 출력시킨 이미지이다. 이미지의 모듈간의 숫자는 원래 모듈간의 결합도와 응집도를 표현하기 위한 숫자이다. 그림 6의 ①의 (17) 은 CXmlNodes에서 CXmlNode 모듈의 호출횟수 즉 의존도를 표현한 것이고, ②의 CXmlNodes의 (6)은 CXmlNodes이 맵핑되는 횟수를 표현한 숫자이다.

5. 결론 및 향후 연구

기존의 레가시 코드를 툴 체인을 통해서 임의의 모듈을 정의를 하고 소프트웨어의 구조를 가시화 하였다. 그리고 모듈 간의 관계를 대략적인 숫자로 정의를 하였다. 향후 연구에서는 현재 타깃에 대한 불확실하게 모듈이 정의가 되어서 확실한 모듈을 정의가 필요하다. 그리고 타깃의 아키텍쳐 구조를 한층 더 자세하게 가시화 시킬 예정이다. 이를 통해 현재 소프트웨어의 모듈간의 호출 횟수와 모듈이 맵핑되는 횟수를 품질지표로 정하였다. 그래서 결합도와 응집도 뿐만 아니라 고품질화를 위해 새로운 품질 지표를 추가 할 예정이다.

참고문헌

[1] 이상은, "2013 SW 공학 백서", 정보통신 산업진흥원 소프트웨어공학센터, 2013
[2] 이상은, "SW개발 품질관리 매뉴얼(SW Visualization)", 정보통신 산업진흥원 소프트웨어공학센터, 2013
[3] 김재수, "모델링 및 시뮬레이션 기반의 소형 이종 이동체 임베디드 시스템", 박사학위논문, 홍익대학교, 2009
[4] emthornber, "Source Navigator NG", http://sourcenav.-sourceforge.net/
[5] graphviz, "DOT tutorial and specification", http://www.graphviz.org/Documentation.php
[6] SQLite, "About SQLite", http://www.sqlite.org/about.html