

PTL Volume 4(ISSN 2288-8195)

2016 (사)ICT플랫폼학회 하계학술발표논문집

- 일 시 : 2016년 7월 1(금) 13:00 ~ 18:00
- 장 소 : 중앙대학교
- 주 최 : (사)ICT플랫폼학회
- 후 원 : 아시아행정학회(AAPA), 세립TSG, 체리 네트워크,
굿모닝아이텍, 대신정보통신, 시스원, 지니네트웍스,
티맥스소프트, 한국 레드햇, 효성인포메이션시스템



(사)ICT플랫폼학회

목 차

세션 1

반도체 스마트 팩토리 구축을 위한 SOA 기반 IIoT 환경 설계-----	1
강윤희, 고승호, 강경우(백석대학교)	
WBAN 공존상황 인지 알고리즘 v0.1-----	6
김범석, 김석훈, 김대영(창신대학교)	
LPWAN을 위한 다중채널 MAC 프로토콜-----	9
김범석, 김석훈, 김대영(창신대학교)	
모바일 통신망을 이용한 태양광 가로등 제어 시스템 연구-----	12
변상봉(GERI)	
미니PC와 근거리 무선 통신 장치를 활용한 사용자 인지 기반 비접촉식 맞춤형 플랫폼 개발-----	15
이은지, 김나경, 김정우, 설상훈, 최정우, 최재봉(성균관대학교)	
IoT 관련 사고사례 및 대응 방안-----	21
홍기원, 김소람, 전용진, 김주영, 장서연, 김종성 (국민대학교)	
람다 아키텍처 기반 웰니스 IoT 환경 구성-----	24
강윤희(백석대학교)	

세션 2

국내 OTP 사고사례 및 현황-----	31
김한기, 김원석, 김종성(국민대학교)	
SW 가시화를 위한 자체 xCodeParser도구 비교-----	34
변은영, 문소영, 서채연, 김영철, 손현승(홍익대학교)	

SW 가시화를 위한 자체 xCodeParser 도구 비교

Comparison between xCodeParser and Open source tool for Software Visualization

변은영*, 문소영*, 서채연*, 김영철*, 손현승**

Eun-Young Byun*, S.Y. Moon*, C.Y. Seo*, R. YoungChul Kim* , Hyun-Seung Son**

요 약

소/중견 소프트웨어 개발 업체들은 전문 인력과 개발시간, 예산의 부족으로 코드 중심의 개발이 이루어지고 있다. 이러한 문제를 해결하기 위해서 소스 코드의 가시화와 품질 지표 적용이 연구되고 있다. 가시화를 통해, 우리 해법은 소프트웨어 구조를 쉽게 파악 및 설계(객체, 상호작용, 유스케이스) 추출을 보여 줄 수 있다. 그리고 품질지표를 적용해 소프트웨어의 품질을 판단하고 품질상의 문제점을 파악할 수 있다. 이를 기반으로 refactoring으로 고품질의 소프트웨어를 개발이 가능하다. 본 논문에서는 소스 코드의 정적분석 도구로 Open-Source인 Source Navigator(SN)를 사용한 경우와 자체 개발한 xCodeParser와 품질 추출 및 효율성을 비교한다.

For most software development in small/medium sized software companies, it is focused on code-centric development without sufficient mature developers, development time, and budget. To solve this problem, there are diverse researches to visualize source code and apply quality metrics. With software visualization, our solution will represent to easily understand software structure and extract design elements (Object, Interactions, Use Case) based on reverse engineering mechanism. We can apply software metrics into the code, and then identify the problems of software quality on the code. Through identifying metrics, it is possible to do refactoring the source(legacy) code for high quality of software. In this paper, it is compared the difference between Open source based "Source Navigator(SN) and our own developed xCodeParser for extracting the elements of software quality and efficiency.

Key words : Open Source, Source Navigator, xCodeParser, Coupling

1. 서 론

소프트웨어 개발 프로세스는 요구사항 분석,

설계, 구현, 검증, 유지보수로 이루어진다. 하지만 현재 소프트웨어 개발은 전문 인력과 개발시간, 개발자의 부족으로 코드 중심의 개발이 이루어지

* 홍익대학교 소프트웨어공학연구소(uwuwu@naver.com)

** 교신저자: 홍익대학교 소프트웨어공학연구소 손현승 박사(hson@live.co.kr)

· 제1저자 (First Author) : 변은영

· 접수일자 : 2016년 6월 24일

고 있다. 그렇기 때문에 유지보수 단계에 소요되는 비용이 전체 개발 비용의 절반을 차지한다. 이러한 문제는 레가시 경우에 더욱 큰 문제로 설계의 부재와 코드 내부의 복잡도에 큰 영향을 주고 있다. 이를 해결하기 위해서 소스 코드의 가시화와 품질 지표의 적용에 초점을 둔 연구들이 이루어지고 있다. 가시화, 즉 역공학을 통해 소프트웨어 구조를 쉽게 파악과 설계의 부재를 해결할 수 있다. 그리고 품질지표를 적용해 소프트웨어의 품질과 어느 부분이 품질상의 문제점이 있는지를 파악할 수 있다. 이를 통해 고품질의 소프트웨어 개발이 가능하리라 본다.

소스 코드의 정적분석 도구로 Open-Source인 SN(Source Navigator)와 자체 개발한 xCodeParser를 사용한 경우를 비교한다[1,2,3].

본 논문의 구성은 다음과 같다. 2장 정적 분석 도구는 SN(Source Navigator)와 xCodeParser에 대해 설명한다. 3장은 Java 기반의 CouplingTest 프로젝트를 대상으로 하여 SN(Source Navigator)와 xCodeParser를 사용한 경우를 비교하여 기술한다. 마지막으로 4장에서는 결론 및 향후 연구를 언급한다.

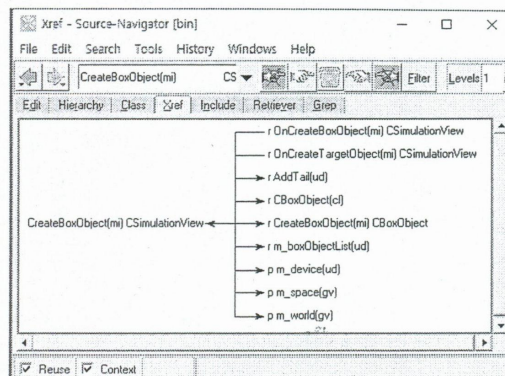
II. 정적 분석 도구

1. SN(Source Navigator)

Source Navigator[4]는 GNU General Public License를 준수하는 공개 소프트웨어이다. 이는 소스 코드 분석 도구로서 다양한 언어에서 지원한다. 또한 코드 구문을 해석하는 정적 분석으로 개발 중인 소스 코드도 분석이 가능하다.

소스 코드의 수정이나 클래스와 기능, 멤버들 간의 관계를 보여주고 호출 트리를 보여주는 기능을 한다. 따라서 소스 코드의 개선이나 유지 보수 작업에 있어 함수와 변수의 선언/구현 정보를 추출해 소스 코드의 전반적인 이해를 돕는다.

그림 1은 Source Navigator로 추출한 코드 요



소 간 참조 목록이다.

그림 1. Source Navigator로 추출한 코드 요소 간 참조 목록

그림 1은 CreateBoxObject 메소드에서 참조하는 코드 구성 요소 목록을 나타낸다. 가장 앞에 표시된 'r', 'w', 'rw', 'p' 는 각각 '읽기', '쓰기', '읽기쓰기', '접근 경우 없음' 등과 같은 접근 유형을 나타낸다. 그 뒤에 명시된 것은 접근하는 대상으로 괄호 내부에 'mi', 'gv', 'cl' 등을 통해 함수, 지역 변수, 클래스임을 나타낸다. 또한 함수일 경우는 해당 요소가 존재하는 클래스를 뒤에 명시한다.

2. xCodeParser

추상 구문 트리(Abstract Syntax Tree)는 프로그래밍 언어로 작성된 소스 코드의 추상 구문 구조의 트리이다. 이 트리의 각 노드는 소스 코드에서 생성하는 구조체를 의미한다. 추상적이라는 의미는 실제 구문에서 모든 세세한 정보를 나타내지 않고 중요한 부분들을 여러 개의 노드로 형상화 하여 나타내는 것을 의미한다.

추상화 구문 트리는 종종 Parser에 의한 산출물로 나오고 의미 분석과 코드 생성의 입력으로 사용된다.

하지만 AST(Abstract Syntax Tree)는 다양한 프로그래밍 언어에 따라 다양한 형태로 생성된다. 이 문제를 해결하기 위해 OMG(Object Management Group)는 AST의 표준화를 위한 Abstract Syntax Tree Metamodel(ASTM)을 정

의했다.

ASTM(Abstract Syntax Tree Metamodel)은 하나의 표준화된 추상 구문 트리를 통해 다양한 언어의 구조 표현이 가능하다. 그러므로 다양한 프로그래밍 언어들이 동일한 소프트웨어 언어 구조 및 개념을 나타내게 된다.

xCodeParser[1]는 다양한 언어의 프로그램을 입력 받아 그 결과로 표준화된 추상 구문 트리인 ASTM을 추출할 수 있다. 또한 서로 다른 언어로 구성된 프로그램 간에 변환이 가능하다.

III. 도구 비교

Source Navigator와 xCodeParser를 이용한 가시화 단계는 그림 1과 같이 Step2인 Source Analysis를 제외하고 구조가 같다.

Source Navigator는 2단계에서 SNDB 파일들을 추출한다. SNDB 파일들은 바이너리 파일이기 때문에 읽을 수 있는 정보로 변환하기 위해 DBdump를 이용한다. DBdump로 추출된 텍스트 정보를 SQLite 데이터베이스에 저장한다.

xCodeParser는 2단계에서 ASTM 파일들을 추출한다. ASTM 파일은 XML 형태로 추출되며 이 정보를 이용하여 SQLite 데이터베이스에 저장한다.

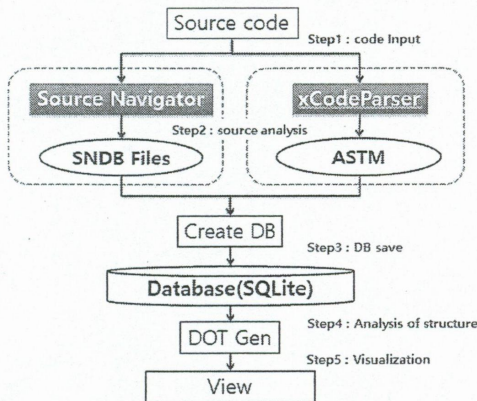


그림 1. 정적 분석 도구에 따른 가시화 구조

비교에 사용된 대상 코드는 Java 기반의 CouplingTest 코드이다. 이 코드는 정적 분석 도

구에 따라 나오는 산출물인 Coupling을 비교하기 위해 Coupling에 필요한 요소를 제외한 다른 기능은 포함하지 않는다.

1. Data Coupling

그림 2는 Data Coupling에 대한 테스트 코드이다. 그림 5는 xCodeParser를 이용한 가시화이다. 그림 6은 Source Navigator를 이용한 가시화이다. 그림 5, 6에서 Data Coupling은 같은 결과가 나온다.

```

1 DataCoupling.java
2 public class DataCoupling {
3     DataCallee c;
4     void Data_Caller(){
5         int i=1;
6         c.Data_Callee(i);
7     }
8 }
9
10
11

1 DataCallee.java
2 public class DataCallee {
3     void Data_Callee(int i){
4         System.out.println(i);
5     }
6 }
7
8
9
10
11
    
```

그림 2. Data Coupling 테스트 코드

2. Stamp Coupling

그림 3은 Stamp Coupling에 대한 테스트 코드이다. 그림 5, 6에서 Stamp Coupling은 같은 결과가 나온다.

```

1 StampCoupling.java
2 public class StampCoupling {
3     StampCallee s;
4     void Stamp_Caller(){
5         s.Stamp_Callee("Stamp");
6     }
7 }
8
9

1 StampCallee.java
2 public class StampCallee {
3     void Stamp_Callee(String c)
4     System.out.println(c);
5 }
6
7
8
    
```

그림 3. Stamp Coupling 테스트 코드

3. Control Coupling

그림 4는 Control Coupling에 대한 테스트 코드이다. Control Coupling의 결과는 그림 5, 6에서 다른 결과가 나온다.

그림 4. Control Coupling 테스트 코드

Control Coupling의 정의는 "특정 모듈의 결과가 분기 조건"이 된다.

하지만 Source Navigator는 해당 정의대로 Control Coupling을 분석하기에 적합하지 않다. 그러므로 정의를 "특정 모듈의 return 값이

```

1 ControlCoupling.java
2 public class ControlCoupling {
3     ControlCallee con;
4     boolean a = true;
5
6     void Control Caller(){
7         con.Control_Callee(a);
8     }
9 }
10
11 ControlCallee.java
12 public class ControlCallee {
13     void Control_Callee(Boolean a){
14         if(a.equals("TRUE")) System.out.println("TRUE");
15         else System.out.println("FALSE");
16     }
17 }
    
```

boolean인 경우"로 제한하여 분석하였다.

반면에 xCodeParser는 함수의 내부에 조건문들을 분석할 수 있어 정확한 ControlCoupling 정보를 얻을 수 있다.

4. External Coupling

External Coupling의 정의는 Source Navigator를 사용한 경우와 xCodeParser를 사용한 경우가 달라서 비교하는데 어려움이 있다. 그러므로 결합도 정의의 표준화가 필요하다.

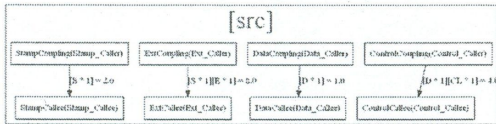


그림 5. xCodeParser를 이용한 가시화

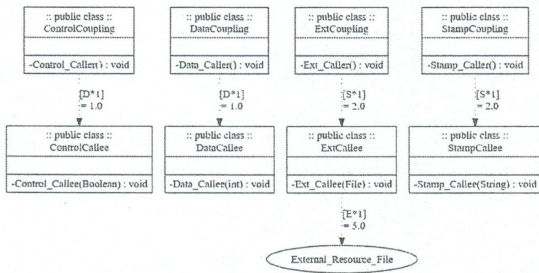


그림 6. Source navigator를 이용한 가시화

IV. 결론 및 향후연구

본 연구에서는 소스 코드의 정적 분석 도구인 Source Navigator와 xCodeParser를 비교해 보았다. Parser에 따라서 가시화 결과가 다르다는 것

을 확인할 수 있다. Source Navigator는 코드 요소의 종류에 따라 여러 개 파일이 나오는 반면에 xCodeParser는 한 개의 파일에 산출물이 나온다. 그 내부를 살펴보면 Source navigator는 형식에 따라 제한된 정보가 나온다. 하지만 xCodeParser는 소스 코드의 모든 정보를 추출해 낼 수 있다.

지금까지 연구로는 소스 코드 가시화에 있어서 xCodeParser의 기능을 충분히 사용하지 못했다. 향후에는 xCodeParser를 이용해 결합도 기반의 소프트웨어 품질 측정을 계속적으로 연구할 계획이며 응집도 기반의 소프트웨어 품질 측정도 연구할 계획이다.

ACKNOWLEDGMENT

본 연구는 2015년/2016년 지속사업인 교육부와 한국연구재단의 지역혁신창의인력양성사업(NRF-2015H1C1A1035548)과 2016년 한국정보통신기술협회(TTA)의 연구 개발 지원으로 수행함.

참고문헌

- [1] Hyun Seong Son, So young Moon and R. Young Chul Kim, "Replacing Source navigator or with Abstract Syntax Tree Metamodel(A STM) on the open source oriented tool chains SW Visualization," The 5th International Conference on Convergence Technology 2015, vol. 1, no. 1, pp. 366-367, June. 2015.
- [2] Geon-Hee Kang, Hyun Seung Son and R. Young Chul Kim, "A Practical Study on Code Static Analysis through Open Source based Tool Chains," 한국정보과학회 컴퓨팅의 실제 논문지, vol. 21, No. 2, pp. 148-153, Feb. 2015.
- [3] Haeun Kwon, Bokyung Park and R. Youngchul Kim, "Automatically Extracting Structural and Behavioral Designs From Object Oriented Programming," 한국스마트미디어학회, Vol. 4, No. 2, pp. 129-131, Oct. 2015.

[4] <http://sourcenv.sourceforge.net/>



(사)ICT플랫폼학회 하계학술발표논문집