

XML 역공학 시스템

박형철 · 간정현 · 장태진 · 이주연 · 이태혁 · 권순각
동의대학교 컴퓨터소프트웨어공학과

XML Reverse Engineering System

Hyungchul Park · Jeonhyeon Gan · Taejin Jang · Juyeon Lee · Taehyuk Lee · Soonkak Kwon
Department of Computer Software Eng., Dongeui University
E-mail : skkwon@deu.ac.kr

요 약

만들어진 소프트웨어 시스템을 역으로 추적하여 처음의 문서나 설계기법 등의 자료를 얻어 내는 역공학(Reverse Engineering)은 유지 보수 및 재사용 측면에서 필요하다. 본 논문은 역공학 개념을 이용하여 하나의 소스 파일에서 클래스 클래스내의 멤버 변수 및 메소드 서로 다른 클래스 사이의 연관 관계 등을 추출하여 xml 문서로 표시하여 저장하는 시스템을 구현한다 또한, 역공학한 프로젝트 내에서 선언된 변수와 함수들의 사용횟수를 분석하며 분석된 결과를 GUI 형태로 표시시켜 줌으로써 사용자의 편리성을 제공해준다

키워드

역공학, 패키지, xml, 클래스

I. 서 론

역공학(Reverse Engineering)은 소프트웨어 공학의 한 분야로 이미 만들어진 소프트웨어 시스템을 역으로 추적하여 처음의 문서나 설계기법 등의 자료를 얻어내는 과정을 의미한다 이것은 시스템을 이해하여 적절히 변경하는 소프트웨어 유지보수 과정에서 필수적으로 행하여져야 하는 사항이며 구현된 시스템의 분석을 통하여 확장을 통한 재사용에 중요한 의미를 갖는다.

만들어지는 있는 거의 대부분의 소프트웨어는 프로그래머 뿐만아니라 제품을 주문한 구매자도 내부적인 구조에 대해 전혀 신경을 쓰지 않고 프로젝트 결과물에 대해서만 중시하고 있는 경향이 있다. 그러다보니 다수의 팀원이 구성되어 하나의 소프트웨어 프로젝트를 진행하면서 팀원 내 할당받은 개개인의 방식으로 만든 후 끼워 맞추기 식으로 진행되어 사용되지도 않는 자원들이 나와 쓸모없는 자원을 낭비하는 경우가 많고, 독립적인 모듈별내 최적의 설계가 이루어지지 않거나, 모듈별 연관관계도 정확하게 정의되지 않는 경우가 많다.

따라서, 역공학 개념을 이용하면 하나의 소스 파일에서 뽑을 수 있는 정보 즉, 클래스, 클래스내의 멤버 변수 및 메소드, 서로 다른 클래스 사이의 연관 관계 등을 추출하여 특정 문서에 저장할 수 있

다. 이것을 통하여 역공학할 대상 프로젝트에서 선언된 변수와 함수들의 사용 횟수를 도출하여 검출 및 유지보수에 용이하도록 할 수 있고, 기존 도구에서 그려주는 클래스 다이어그램을 좀 더 자세히 그려줄 수 있을 것이다.

본 논문은 Java로 구현된 소스 파일을 역공학하고 결과를 xml 문서로 저장하는 시스템을 구현한다

II. 시스템 흐름

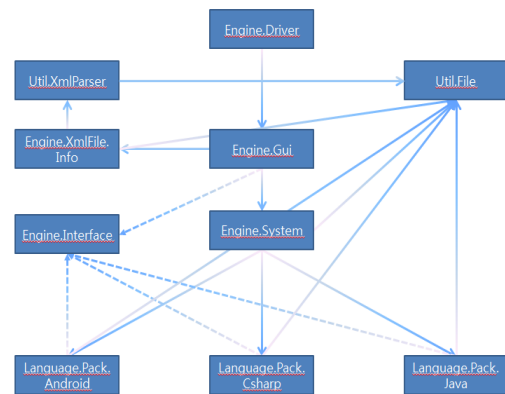


그림 1. 전체 패키지 흐름도

TargetLanguage는 Java의 경우 그림1의 맨 밑 Language.Pack의 3가지중 Java팩이 선택될 것이고 Engine.System에서는 언어에 맞는 역공학이 수행될 것이다. 그리고 Engine.Gui에서는 Util.XmlParser를 통해 분석되어서 Engine.XmlFile.Info로 넘어온 값들과 Engine.System에서 넘어온 값들이 사용자가 보기 쉽게 보여질 것이다. 그리고 Util.File은 API개념으로 쓰이며 파일에 대한 처리부분을 맡고 있다 Engine.Driver는 프로그램 총괄적 구동 패키지이다

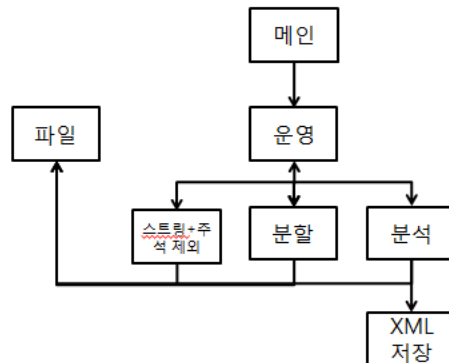


그림 2. 자바팩 시스템 흐름도

네모를 하나의 클래스로 보면 메인 클래스에서 프로그램을 시작하고 운용 클래스는 스트링 주석 제외, 분할, 분석클래스로 나누어 관리한다

스트링+주석 제외 클래스에서는 주석을 제거하고 스트링 변수 값을 제거한다 분할 클래스에서는 한 클래스 파일 안에 여러 개의 클래스가 선언되어 있을 경우 따로 파일을 만들어 저장을 한다

분석 클래스에서는 크게 클래스 타입클래스, 인터페이스, 이넘, 변수 타입(자료형 변수, 인스턴스 변수), 함수 타입(일반 메소드, 추상 메소드)로 생각하고 정보들을 저장한다.

그리고 저장된 정보를 상속 받은 것인지 인터페이스를 통해 구현 것인지에 대한 관계에 대해 도출하여 저장한다.

파일 클래스는 API개념으로 쓰이고 파일 열기 닫기, 쓰기, 읽기, 찾기 등의 기능을 가지고 있다

이후 사용 횟수클래스에서 선언된 변수들과 함수들의 사용 횟수들을 도출하여 보고서를 작성해주고 클래스 다이어그램을 그러 클래스 간의 연관 관계를 보기 쉽게 해준다.

III. 클래스 분석 방법 및 GUI

3-1 제거

예를 들어 String str = "class"; 라고 선언하였을 경우 class라는 단어를 보고 클래스를 분할하기 때

문에 위험 요소가 되어 문장을 분석할 때 방해 요소를 사전에 제거할 필요가 있다.

그리고 프로젝트를 분석할 때 한글 주석// ~~\n)과 다중 주석(/* ~~~ */)에 대해서는 분석할 필요가 없기 때문에 여기서 사전에 제거한다

3-2 클래스 분할

클래스 분할은 하나의 자바 파일 안에는 내부 클래스라던가 한 클래스 안에 여러 클래스가 선언될 수 있는데, 각각의 독립된 클래스로 분할하는 역할을 한다. 분할된 각각의 클래스들은 임시 파일로 저장되며 이 임시 파일들이 클래스 분석 때 쓰인다

3-3 구문 분석

그림3은 구문분석의 흐름을 보여주는 알고리즘에 대한 그림이다 'public class A{'란 문장을 예로 들어 설명하였는데 한글자씩 읽어들이 다음 공백 혹은 특수문자와 같은 글자 외의 것들이 오면 글자 외의 것들이 오는 부분 앞까지가 한 단어로 인식하게 되는 것을 보여준다.

인식된 단어들은 예약어와 키워드를 정의해 둔 파일을 통해 각 단어가 의미하는 값들을 알아내서 분석한다. 분석한 단어들에 해당하는 값들을 저장 후 라인에 대한 속성 즉 클래스, 인터페이스, 이넘, 메소드, 추상 메소드, 자료형 변수, 인스턴스 변수, 연관 관계인지를 특수문자(소괄호, 중괄호, 세미콜론)의 존재 유무를 통해 파악함으로써 현재 라인에 대한 구문 분석을 완료한다

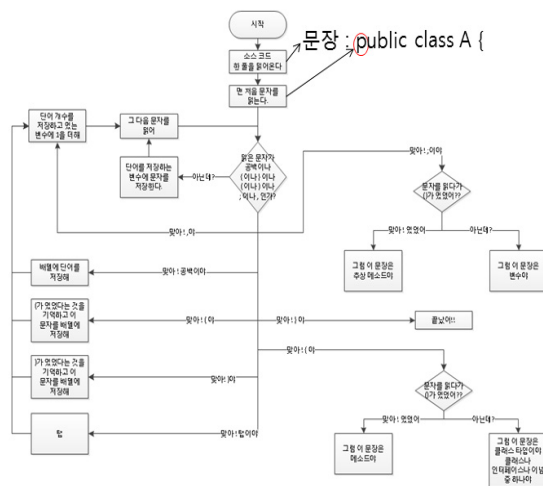


그림 3. 구문 분석 알고리즘

3-4 xml 문서 저장

구문분석을 통해서 나온 값들을 순차적으로 xml구조로 저장하는데, 노드의 이름과 구문분석을 통해

저장된 값들은 조건문을 거쳐서 구성하고, 4개의 깊이로 구성한다.

```

<패키지>
  <패키지 이름>
</패키지 이름>
  <클래스 타입>
    <클래스 타입 정보>
  </클래스 타입 정보>
  <메소드>
    <메소드 정보>
      <매개변수 정보>
    </매개변수 정보>
  </메소드 정보>
  <메소드>
  <변수>
    <변수 정보>
  </변수 정보>
  <변수>
  <관계>
    <관계 정보>
  </관계 정보>
  </관계>
</클래스 타입>
</패키지>
    
```

그림 4. xml 구조

- 깊이 1은 패키지이다.
 - 깊이 2는 패키지 이름, 클래스 타입(클래스, 인터페이스, 이념)이다.
 - 깊이 3은 클래스 타입 정보(extends, implement, visibility, static, final 등), 메소드, 변수이다.
 - 깊이 4는 메소드 정보, 변수 정보, 관계 정보이다.
 - 깊이 5는 매개변수정보(visibility, return type, static, final 등) 이다.
- 다음과 같은 깊이 구조로 xml 문서에 저장 한다.

3-5 연관 관계 도출

xml에 저장된 정보들 중에서 extends를 통해 상속 받아야 할 클래스 이름과 implements를 통해 구현해야 할 클래스들의 이름을 통해 부모 클래스에서 상속받아온 함수 또는 변수인지, 인터페이스를 통해 추상 메소드를 오버라이드하여 구현해야할 함수인지 등을 찾아 정보를 저장한다.

3-6 GUI

XML로 저장되면 원하는 정보만 추출 혹은 찾기가 편리하지만 한눈에 정보를 보이게는 무리가 있다 이를 보완하기 위해 xml기반으로 GUI를 만들게 되었다. 그림5는 GUI로 만들어진 프로젝트를 찾는 탐색기 부분이다.

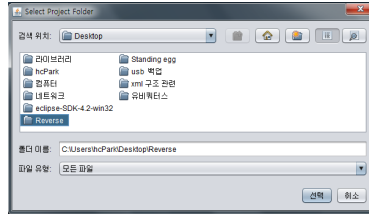


그림 5. Xml 기반 GUI

기본에 구현했던 방법은 경로를 직접 입력 받도록 하였는데 이와 같은 방법으로 하게되면서 보다 편하고 정확하게 되었다. 그림 6은 분석 후 트리형식으로 정리된 패키지별 클래스이다

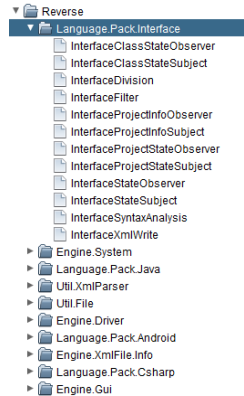


그림 6.

패키지별로 정리가 되어서 보기가 훨씬 쉽게 되었고 트리형식으로 저장하여 그림에서 보는 것과 같이 패키지를 클릭하게 되면 패키지별 클래스가 바로 눈에 나타나게 된다.



그림 7. 변수와 메소드 표시

그림6에서 클래스를 클릭하게 되면 그림7의 내용처럼 분석된 내용이 변수와 메소드로 나타나게 된다

IV. 구현내용

```

- <attribute>
  <visibility>private</visibility>
  <instance>String</instance>
  <name>readAll</name>
</attribute>
- <method>
  <visibility>public</visibility>
  <returnType>void</returnType>
  <name>dir</name>
  - <parameter>
    <instance>String</instance>
    <name>path</name>
    <instance>String</instance>
    <name>extension</name>
  </parameter>
</method>
- <relation>
  <name>br</name>
  <new>new</new>
  <instance>BufferedReader</instance>
  <Relationship>dependency</Relationship>
</relation>
</relationLevel>
- <class>
  <visibility>public</visibility>
  <class>class</class>
  <name>FileManagement</name>
</class>
</classLevel>
<package>package</package>
<package_name>Util.File</package_name>
<import>import</import>
<import_name>java.io.*</import_name>
<import>import</import>
<import_name>java.util.*</import_name>
</packageLevel>

```

그림 8. xml로 저장된 값

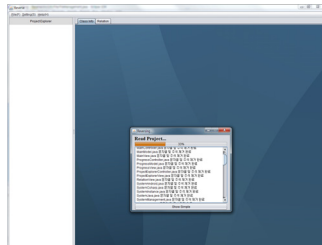


그림 9. gui 파일 로딩 및 분석

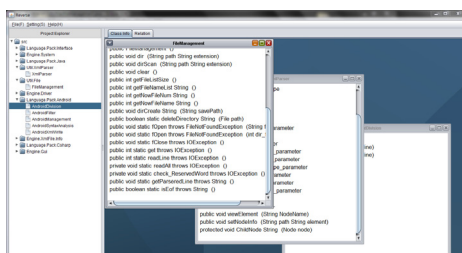


그림 10. gui 분석된 내용

V. 결 론

역공학은 프로그램을 주문한 구매자와 그 프로그램을 직접 구현한 개발자가 놓치고 있던 불필요한 자원 낭비를 줄이고, 이미 만들어진 시스템을 역으로 추적하여 소스 코드에서 필요한 최대한의 정보를 추출하여 소프트웨어의 재사용성을 높이고 유지보수를 용이하게 하는데 목적이 있다. 본 프로젝트에서 구현하고 있는 역공학 시스템은 예약어 판별을 용이하게 하기 위해 주석과 스트링을 제거하는 단계 내부 클래스와 하나의 클래스 안에 선언된 다수의 클래스를 독립된 클래스로 분할하는 단계 라인을 받아와서 해쉬맵을 통한 분석단계, 분석단계를 통해 생성된 값들을 XML에 저장하는 단계, 저장된 XML을 통한 연관 관계를 도출하여 저장하는 단계로 구성된다. 또한, GUI로 화면에 표시도 가능하다.

참고문헌

- [1] Kathy Sierra, Bert Bates / HeadFirst JAVA / 2011. 04. 28
- [2] 남궁 성 / Java의 정석 / 2009. 01. 20