

# 역공학 기반의 객체지향 다이어그램 추출기법에 관한 연구

박상훈\*, 김연형\*, 류성열\*

\*송실대학교 컴퓨터학과

e-mail:se\_hoon@selab.ssu.ac.kr

## A Study on Reverse Engineering Based Object Orient Diagram Extract Technique

Sang-Hoon Park\*, Yun-Hyung Kim\*, Sung-Yul Rhew\*

\*Dept of Computer at Graduate School, Soong-Sil University

### 요 약

레거시 소프트웨어를 재사용하기 위해서는 소스코드의 구조와 행위, 연관관계를 파악하는 것이 중요하다. 소스코드 분석을 통해 레거시 소프트웨어의 구조를 얻어내고 이에 대한 수정을 통해 소스코드에 반영하고자 하는 연구가 활발히 진행되고 있다. 그러나 기존의 역공학 도구들은 레거시 소프트웨어를 객체지향 다이어그램으로 정확히 추출해내지 못하였고 또한 호환성이 떨어지는 문제점을 가지고 있다. 본 연구에서는 기존의 역공학 툴들의 문제점을 보완하고 호환성을 높이며 재사용성을 향상시키기 위한 방법을 제시한다. 자바 소스코드로부터 UML클래스 다이어그램을 추출하기 위해 본 논문에서는 자바 소스코드의 파싱을 통해 클래스 다이어그램 생성에 필요한 AST를 추출하여 XML로 저장함으로써 상호운영성을 높이며 클래스간의 연관관계를 상세히 표현하기 위한 방법을 제시한다.

..

### 1. 서론

오늘날의 기업환경은 아주 빠르게 급변하고 있다. 빠르게 변화하는 비즈니스 요구사항을 충족하기 위해 소프트웨어 산출물들은 빠른 적응성을 가져야 한다. 이에 따라 기존 시스템을 변경하고 개발할 수 있는 자동화된 도구의 필요성이 증대되고 있다.

기존 시스템을 변경하기 위해서 우선 시 되는 것은 소스코드의 이해이다. 그러나 소스코드를 이해하는 것은 시간이 많이 소요되며 아주 지루한 일이다. 이러한 현상은 원본 설계자에 의해 소스코드가 변경되지 않기 때문이다. 주로 소스코드를 수정하거나 변경하는 것은 유지보수자의 몫이기 때문이다[1]. 따라서 보통의 프로그래머는 다른 사람의 소스코드를 해독하는데 엄청난 시간을 소비한다.

역공학 기술은 분석가들에게 기본적인 프로그램 아키텍처와 행위를 발견하기 위하여 사용되는 설계 모델을 산출하는 것을 가능하게 한다. 설계 모델을 바탕으로 전체적인 시스템의 그림을 그릴 수 있다.

또한 역공학 기술은 쉬운 코드의 재공학과 진화적인 개발을 하기 위한 중요한 역할을 한다.

소스코드를 나타내는 것은 소프트웨어 분석단계에서 가장 중요한 문제이다. 전통적인 표현방법은 Abstract Syntax Trees(AST)를 통해서 구현되었다. 그러나 XML 도구와 기술의 발달로 인하여 XML 기술을 사용하는 것이 변환이 쉽고, 소스코드를 변경하거나 이해하기 쉽다.

이 논문은 표준화된 역공학 프로세스와 XML 기반기술을 사용하여 자동화된 소프트웨어 환경의 XML 인코딩(encoding)의 강력함과 단순함을 강조한다.

### 2. 관련연구

#### 2.1 역공학(Reverse Engineering) & 재공학(Re-Engineering)

소프트웨어 역공학(Reverse Engineering)이란, 자동화 도구(Case Tool), 수작업 등을 통하여 소프트

웨어의 물리적 계층(Physical Layer)의 표현을 데이터 요소와 프로세스등을 설명해 주는 명세 계층(Specification Layer)의 표현으로 재구성하는 과정 혹은 목표시스템을 시스템의 구성요소와 그들간의 관계를 분석하고 좀더 추상화된 다른 표현으로 나타내는 과정을 말한다.

역공학의 노력은 일반적으로 소프트웨어 시스템의 재문서화나 재공학을 준비하기 위한 시스템의 문제 가능성을 인식하는데 집중되고 있다. 역공학은 존재하는 설계문서, 소스 코드, 소프트웨어의 실행, 개발자와 운영자의 인터뷰를 통하여 다양한 정보 출처로부터 레거시 정보를 습득, 추상화 수준의 뷰를 제공하는 도구 등으로부터 다양한 방법을 통하여 수행 시스템 분석, 설계의 문서를 추출 할 수 있다.

소프트웨어 재공학(Reengineering)이란, 자동화 도구, 수작업 등을 통하여 기존의 소프트웨어 시스템의 기능 및 디자인을 수정, 성능 향상 및 유지 보수를 꾀하는 과정을 말한다. 재공학의 목표는 레거시 시스템의 이해를 도와 계속해서 레거시 시스템을 유지하거나 새로운 적용을 수행 할 수 있게 복잡성을 줄이는 것이다[2].

역공학은 구체화된 소스코드에서 추상화된 설계모델이나 요구사항을 추출하는 작업을 말하고 재공학은 하나의 구체화된 구현을 또 다른 구체화된 구현으로 변환하는 작업을 의미한다.

## 2.2. XML

XML은 World Wide Web Consortium(W3C)에 의하여 개발된 eXtensible Markup Language이다. Web상에서 구조화된 문서를 전송 가능토록 설계된 표준화된 텍스트 형식이다. XML은 SGML/HTML과 상호운용을 하며 SGML을 응용한 언어이다. 일반 Web 응용 S/W에 의해 처리되고 사용자-정의 문서를 다양하게 배치 가능하다. XML 문서들은 태그들에 의하여 에워싸이는 보통의 텍스트로 이루어진다. 다음은 단순한 보기를 제공한다.

```
<?xml version="1.0"?>
<!DOCTYPE book SYSTEM "book.dtd">
<CHAPTER>
  <TITLE>1. XML </TITLE>
  <BODY>
    <SECTION>
      <TITLE>(1) XML Introduce </TITLE>
```

```
<BODY>
```

```
XML is extensible.
```

```
</BODY>
```

```
</SECTION>
```

```
</BODY>
```

```
</CHAPTER>
```

XML 문서는 다양한 어플리케이션을 지원하고 가독성이 뛰어나며 간결하며 문서생성이 용이하다는 특징이 있다.

## 2.3 Java Markup Language (JavaML)

JavaML은 XML을 기반으로 자바 소스코드를 나타낸다. JavaML은 코드의 요소를 나타내는 태그에 의하여 규정되는 자바 프로그램이다. 역공학에 있어서 XML도구의 사용은 파싱과 분석을 더욱 쉽게 도와준다. JavaML은 자바 원시 코드의 구조를 파악하기 위하여 필요한 소프트웨어 도구와 프로그래밍 지원 도구의 대체책을 제공한다. JavaML은 역공학을 위하여 프로그램과 구조 정보에 유용한 접근법이다. 자바 문법의 지식을 통해, JavaML은 더욱 읽기 쉽고 이해가 쉽다.

## 2.4 UML & XMI

UML은 시스템을 설계하고 문서화하고, 가시화하기 위한 완벽한 모델링 언어이다. 특히, 클래스 다이어그램은 시스템의 정적인 구조를 모델링 하기 위해 사용된다. UML은 UML과 객체지향 언어들 사이에 단단한 맵핑 관계와 많은 프로그래머들에게 친밀하기 때문에 자바 역공학을 위한 추상적인 표현 방식으로 적당하다.

XML Metadata Interchange(XMI)는OMG에 의하여 개발된 표준이다. 정보 교환을 위해 XML의 일반적인 방식으로 UML과 병합한다. 결과는 디자인 정보의 상호교환을 위한 표준 서식을 만들어낸다. UML 다이어그램들을 통하여 나타내어지는 관계와 정보를 저장하고 변환하기 위하여 표준화된 방법을 제시한다. 이러한 형식은 소프트웨어 개발을 위한 다양한 방법의 프로세스를 도울 수 있다. XMI를 사용하면 역공학 도구에서 개발 환경으로 설계정보를 쉽게 재사용할 수 있다.

## 3. 선행연구의 문제점

레거시 시스템의 역공학을 통한 정보 추출은 현재로서는 분석가가 직접 소스를 분석하여 정보를 추출

하는 방법과 역공학 틀을 이용하여 분석 정보 및 다이어그램을 추출해 내는 방법이 있다.

첫번째 방법인 분석가가 직접 레거시 시스템의 소스를 분석하여 정보를 추출해 내는 방법은 레거시 시스템의 개발자의 부재, 매뉴얼의 부족, 개발 문서의 부족 등으로 인하여 시스템에 대한 이해도와 분석 기간 등 여러 부분에서 효율적이지 못한 부분이 있다.

이러한 부분을 개선하기 위해서는 자동화 도구를 이용해야 한다. 현재 역공학을 위하여 사용하고 있는 자동화 도구의 종류는 3-4가지의 제품이 있으나 대부분 역공학에 필요한 정보들을 적절히 분석할 수 있도록 제시되지 못하고 있다.

또한 현재의 자동화 도구에서 추출되는 다이어그램의 종류도 클래스 다이어그램만이 유일한 실정이고 시퀀스, 유스케이스 다이어그램의 경우 추출이 불가능하다.

이러한 문제점은 빠르고 정확하게 레거시 시스템을 분석하기 위한 기본적인 추출정보에 대해서 미흡한 부분이 많다.

기존 Reverse 도구의 경우 객체지향 개발 방법론에 적절한 UML기반의 다이어그램을 추출해 내지 못하고 있다. 현재, 클래스 다이어그램의 경우 자동화 도구에 의해 추출이 가능하나, 시퀀스 다이어그램과 유스케이스 다이어그램은 자동화 틀에 의한 추출이 불가능 하다.

코드 파싱 정보에 대한 정확한 데이터 자료를 뽑아 내지 못하고 있으며, 이는 추후 다른 다이어그램 및 기타 분석 자료로서 사용되어 질 때 추후 활용도가 떨어진다.

자동화 도구의 제작을 위해서는 기본적으로 UML기반의 다이어그램의 정확한 추출기법이 필요하며, 또한 역공학 도구는 다이어그램뿐만 아니라 추출된 다양한 정보를 재사용할 수 있도록 가공되어야 한다. 하지만, 현재의 객체지향 방법론에 있어서 이러한 도구의 기능은 매우 빈약한 상태이며, 문서화에 대한 부분도 매우 부족한 현실이다.

<표 1> 기존 Reverse CASE 도구의 문제점 분석

	Rational Rose	Together	PLASTIC20 03
제작	Rational	Together	Plastic Soft
역공학 지원 여부			
자동화 추출 수준	Class Diagram	Class, Sequence Diagram (미흡)	Class Diagram
다이어그램 추출 정확도	Class Diagram ( )	Class Diagram() Sequence Diagram()	Class Diagram()
파싱정보 문서화 수준	X	X	X

4. 요구사항

앞 절에서 제시된 문제점들을 해결하기 위해 필요한 사항들을 정의하고, 그에 따른 요구사항을 정리하였다.

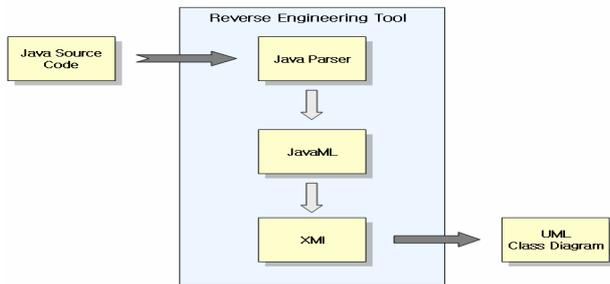
<표 2> 문제점에 따른 요구사항

문제점 파악	요구사항
소스코드의 파싱 정보를 활용하기 위하여 문서 또는 기타의 저장소에 소스코드의 파싱정보를 저장하여야 한다. 소스 코드의 정확한 파싱정보는 추후 다이어그램의 추적과 추출의 기본 정보가 된다.	소스코드의 파싱정보를 저장소 또는 XML문서로 저장하고 이를 활용할 수 있어야 한다. 파싱정보는 새로운 정보로의 재 창출을 위하여 판독이 가능한 정보로 저장되어야 한다.
UML 기반의 시퀀스 다이어그램과 유스케이스 다이어그램의 추출이 미흡 또는 불가능하다. UML 기반의 시퀀스 다이어그램 및 파싱 정보를 통하여 시퀀스 다이어그램과 유스케이스 다이어그램을 추출할 수 있는 기법이 없다.	역공학 도구에 의한 클래스 다이어그램뿐만 아니라 시퀀스, 유스케이스 다이어그램의 추출도 가능해야 한다. 기본적으로 각 다이어그램과의 관계와 파싱정보의 연관성, 추출기법을 정확하게 제시 되어야 한다.

### 5. 객체지향 다이어그램 생성

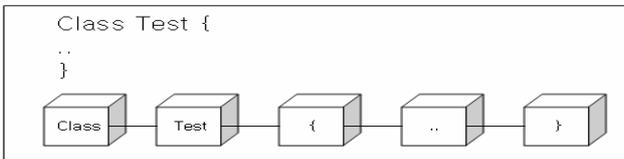
본 논문에서는 이러한 문제점을 개선하고자 소스에서 파싱된 정보를 통해, 추출된 정보의 재가공 및 다이어그램의 추출기법에 대한 연구를 통하여 추후 자동화 도구를 통하여 클래스 다이어그램뿐만 아니라, 시퀀스, 유스케이스 다이어그램도 추출 가능할 수 있는 기법에 대한 연구를 논의한다.

우선 Class Diagram을 생성하기 위한 전체흐름을 (그림1)에서 나타내었다. 자바 소스코드에서 UML Class Diagram을 생성하기 위한 기법은 다음과 같다.



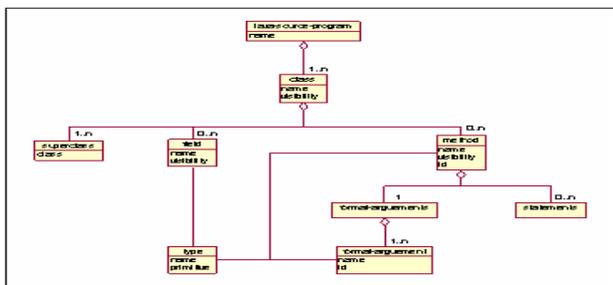
(그림 1) Class Diagram의 생성

(1) Java 문법을 포함한 Java Parser를 통해 Java 소스코드를 순회하여 소스코드를 (그림 2)의 토큰단위로 분해한다.



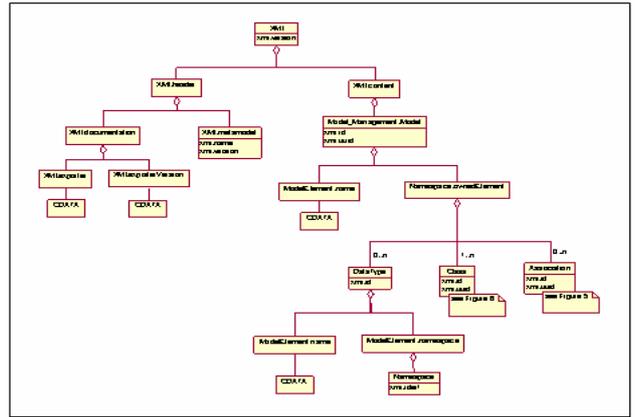
(그림 2) Java Parser에 의한 토큰으로 분해

(2) 각 토큰을 JavaML 엘리먼트와의 맵핑을 통해 알맞은 태그를 덧붙여 JavaML로 변환한다.



(그림 3) JavaML 엘리먼트 구조

(3) JavaML의 태그들과 XMI 태그를 맵핑하여 XMI 파일을 생성한다.



(그림 4) XMI 엘리먼트 구조

XMI 파일은 UML 클래스 도형을 위한 모든 디자인 정보를 포함하므로 이를 UML Class Diagram으로 변환한다.

### 6. 결론 및 향후연구

본 논문에서는 기존의 역공학 도구의 문제점을 분석하고 이를 개선하기 위한 방안을 모색해 보았다. 레거시 시스템의 소스를 토큰 단위로 파싱하여 구조와 기능, 관계를 파악하기 위한 방법을 제시하여 파싱된 소스의 분석을 통하여 객체지향 개발 방법에 필요한 UML 다이어그램 추출 방법을 연구하였다. 그 중 하나로 클래스 다이어그램 생성에 대한 기법을 제시하였다. 향후 레거시 시스템을 정확히 분석하여 새로운 환경에 적절하게 대응할 수 있는 완벽한 역공학 도구의 개발에 대한 연구를 진행하려고 한다.

### 참고문헌

[1] E. J. Chikofsky and J. H. Cross II. "Reverse Engineering and Design Recovery: A Taxonomy". IEEE Software, Jan 1990.  
 [2] 최일우, "레거시 시스템 진화를 위한 효율적 재공학 프로세스", 정보처리학회 논문지 D, 2003.8  
 [3] 한무희, "Java 코드 분석기법을 이용한 UML 클래스 다이어그램 생성 방법", 정보과학회 2003추계 학술발표, 2003  
 [4] A. V. Aho, R.Sethi and J. D. Ullman. Compilers: Principles, Techniques and Tools. Addison-Wesley, 1986.  
 [5] Gregory McArthur, "An Extensible Tool for Source Code Representation Using XML", IEEE WCRC02, 2002