웹 기반의 Java 바이트 코드의 이해를 지원하는 XML 문서 생성

나강숙[U] 이재현 유철중 장옥배
전북대학교 컴퓨터과학과
벽성대학교 전자계산학과
kasna@cs.chonbuk.ac.kr
jhlee@mail.byuksung-c.ac.kr
{cjyoo, okjang}@moak.chonbuk.ac.kr

Web-based XML Document Generation Supporting Java Byte Code Understanding

Kang-Suk Na[U] Jae-Hyun Lee Cheol-Jung Yoo Ok-Bae Chang
Dept. of Computer Science, Chonbuk National University
Dept. of Computer Science, Byuksung College

요 약

본 논문은 웹 기반의 Java 바이트 코드의 이해를 지원하는 XML(eXtensible Markup Language) 문서를 생성하는 것을 목적으로 한다. 기존 XML 문서는 사용자가 임의로 태그를 생성하여 확장할 수 있는 장점이 있는 반면에 프로그램에 대한 태그의 정적인 정보만을 제공하는 단점이 있다. 따라서 정적인 정보만을 제공하는 XML 문서에 Java 바이트 코드를 Javap로 역어셈블(disassemble)하여 얻을 수 있는 메소드 호출의 동적인 정보를 추가할 필요가 있다. 본 논문은 이러한 Java 바이트 코드에 대해 동적· 정적인 정보가 포함되어 있는 XML 문서를 웹 상에서 클라이언트에게 다운로드(download)할 수 있는 기능을 제공하여 Java 프로그램의 이해도를 증진시키는데 그 의의가 있다.

1. 서론

Java 소스 코드의 로직(logic)을 분석하는 경우에 프로그래머가 Java 프로그램을 이해하기 위해서는 상당한 전문적인 프로그램 능력과 시간이 요구된다. 프로그램에 대한 능력이 부족한 사용자 관점에서 보았을 때 Java 소스 코드에 대한 로직이 너무 복잡하고 별도로 문서화에 대한 정보가 제공되지 않기 때문에 프로그램을 이해하는데 더 어려움이 있다[1].

본 논문에서는 Java 소스 코드보다 상세한 정보가 있는 Java 바이트 코드인 클래스 파일을 최근에 인터넷의 차세대 언어로 주목받고 있는 XML 문서로 변환하여 제공하려고 한다. XML 문서에는 Javap로 역어셈블된 동적인 정보와 XML문서 고유의 정적인 정보가 모두 포함된다. 또한 웹 기반 환경에서 클라이언트가 업로드(upload)한 Java 바이트코드인 클래스 파일을 자동변환된 XML 문서로 다운로드할 수 있는 기능을 제공하고자 한다. 따라서 프로그래머는 Java 프로그램을 분석하는 경우에 동적· 정적인 정보가 있는 XML 문서에 의해 프로그램을 쉽게 이해할 수 있고 웹 기반 환경이기 때문에 정보를 상호 공유할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 먼저 2장의 관련연구에서는 XML에 대해 설명하고 Java 소스 코드를 컴파일해서 얻을 수 있는 Java 바이트 코드에 대해 기술한다. 다음으로 Java 바이트 코드인 클래스 파일을 보다 상세한 정보로 얻을 수 있는 Javap라는 역어셈블러에 대해 살펴본다. 3장에서는 본 논문에서 제시한 Java 바이트 코드를 XML 문서로 자동변환 해주는 시스템의 설계 구성도와 웹 기반 환경에서 클라이언트가 클래스 파일을 업로드한 후 자동 변환된 XML 문서를 다운로드하는 과정을 설명한다. 마지막으로 4장에서는 결론 및 향후 연구과제를 제시한다.

2. 관련연구

2.1 XML

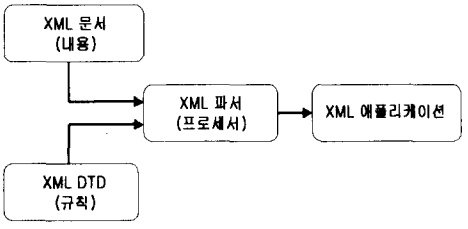
XML은 HTML(HyperText Markup Language)과 같이 고정된 형식이 아닌 확장이 가능한 마크업 언어로서 본질적으로 다른 언어를 기술하기 위한 메타언어이다. XML은 복잡한 SGML(Standard Generalized Markup Language)과 고정된 태그를 사용하는 HTML의 단점을 보완하기 위해 개발되어 보다 단순하면서도 태그를 확장할 수 있는 기능을 제공한다.

XML로 할 수 있는 애플리케이션은 다음과 같다. XML이 풍부한 정보를 기술하기 때문에 웹 상에서 비즈니스를 관리하는 경우 보다 나은 지적 전송 기법을 제공하는

전자상거래가 있다. 재정분야에서는 몇몇 회사가 사용하는 정보를 상호 교환하는 애플리케이션을 나타내는 OFX(Open Financial eXchange) 마크업 언어를 생성한다. 그리고 온라인 뱅킹 애플리케이션과 관심있는 영역을 선택한 클라이언트의 생각을 서버가 판단해서 이에 해당하는 정보를 푸시(push)하는 PointCast Network 브라우저 애플리케이션이 있다. 그밖에 메타 태그(Meta tag)를 이용하는 개선된 검색 엔진과 웹을 통해 사무실에서 가정용 주방기기나 욕실 및 가전전등을 제어하는 제어 시스템이 있다.

XML 문서를 구조화하는 일반적인 방법으로 SGML 템플릿 형식인 DTD(Document Type Definition)를 사용해서 XML 파서가 DTD에서 수행되는 자료를 해석하는 방법과 XSL(XML Stylesheet Language)을 사용하는 방법이 있다.

[그림 1]은 XML 시스템의 구성요소를 나타내고 있다. XML 문서에는 실제정보인 내용이 들어 있고 XML DTD에는 XML 문서의 논리적 구조가 정의되어 있다. 프로세서인 XML 파서는 XML 파일과 DTD의 구문을 분석한다. XML 애플리케이션은 정보 교환을 위한 메타 데이터를 처리한다[2,3].



[그림 1] XML 시스템의 구성요소

2.2 Java 바이트 코드

Java 바이트 코드는 Java 소스 코드를 컴파일한 Java 클래스 파일이다. Java 컴파일러인 Javac로 컴파일하면 class라는 확장자가 붙는 파일이 생성된다. 이 파일은 Java 소스 코드 내에 정의된 클래스, 메소드, 필드에 대한 정보가 포함되어 있다. Java 바이트 코드는 [표 1]과 같은 구조로 되어 있으며 u1, u2, u4는 각각 unsigned 1, 2, 4 바이트 값을 의미한다. 모든 클래스 파일의 첫 4바이트는 0xCAFEBABE이고 두 번째와 세 번째는 Java 컴파일러의 주 버전, 부 버전을 나타낸다. Java 소스 코드의 모든 클래스와 스트링, 정수 등을 데이터의 길이가 정해져 있지 않은 항목들의 연속으로 되어 있는 constant_pool로 취급한다. access_flags는 변경자(modifier)의 종류를 나타내며 다음으로 자신의 super 클래스, 자기 자신을 나타내는 this 클래스와 this 클래스가 구현한 인터페이스와 필드, 그리고 메소드를 나타내고 있다. attribute_field에는 클래스 파일 이름을 저장한다[4,5].

[표 1] Java 바이트 코드의 구조

```
class_file {
        u4              magic;
        u2              minor_number;
        u2              major_number;
        u2              constant_pool_count;
        cp_info         constant_pool[constant_pool_count -1];
        u2              access_flags;
        u2              this_class;
        u2              super_class;
        u2              interfacecount;
        u2              interface[interface_count];
        u2              fields_counts;
        field_info      fields[fields_count];
        u2              methods_count;
        method_info     methods[methods_count];
        u2              attributes_count;
        attribute_info  attributes[attributes_count];
}
```

2.3 Javap

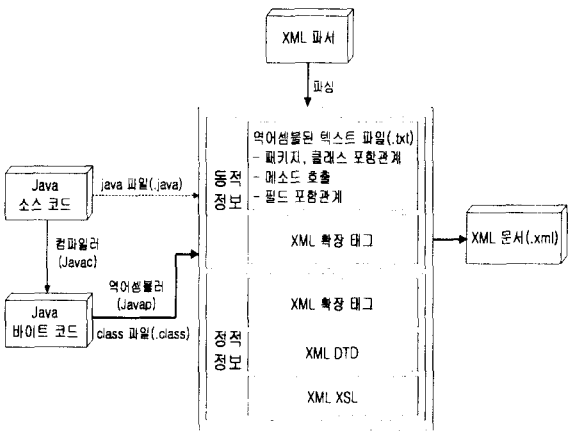
Javap는 바이트 코드로 되어 있는 클래스 파일을 보다 상세한 정보를 나타내는 텍스트 파일로 만드는 JDK(Java Development Kit)에서 지원하는 역어셈블러(disassembler)이다. 역어셈블된 텍스트 파일은 클래스와 메소드의 제어흐름을 분석할 수 있는 동적인 정보를 나타내고 있다. Javap는 Java 바이트 코드를 이해하기 쉬운 형태인 오피코드(Opcode)로 표현한다. 이러한 오피코드에 대한 정보를 [표 2]에서 나타내고 있다[5,6,7].

[표 2] Java 바이트 코드의 오피코드

오피코드	기 능
Nop	어떤 작업도 수행하지 않음
Const	상수값을 오퍼랜드 스택에 저장
Bipush	스택에 부호화된 byte형 상수를 푸시(push)함
Sipush	스택에 부호화된 short형 상수를 푸시(push)함
Ldc	스택에 있는 상수를 로드(load)함
Getfield	스택에서 레퍼런스를 꺼내옴
Setfield	필드의 값을 초기설정해줌
Invokevirtual	메소드를 호출하는 경우에 사용함
Invokespecial	상위클래스에 있는 메소드를 호출하는 경우에 사용함
invokestatic	정적 메소드를 호출하는 경우에 사용함
Invokeinterface	인터페이스에 선언되어 있는 메소드를 호출하는 경우에 사용함

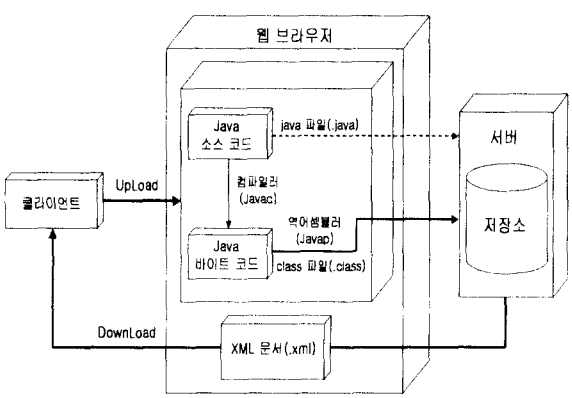
3. 웹 기반의 Java 바이트 코드의 이해를 지원하는 XML 문서 생성

본 논문의 전체적인 설계 구성도는 [그림 2]와 같다. Javap가 Java 바이트 코드를 역어셈블해서 얻은 텍스트 파일에는 패키지, 클래스의 포함관계와 메소드 호출관계 및 필드의 포함관계가 동적인 정보로 표현되어 있다. 여기에 XML 확장 태그와 DTD, XSL과 같은 정적인 정보를 추가하게 된다. 이러한 동적· 정적인 정보를 XM 파서가 파싱을 하고 나면 프로그래머에게 최종의 XML 문서를 제공하고 Java 프로그램을 보다 쉽게 이해할 수 있는 장점이 있다[8].



[그림 2] 설계 구성도

[그림 3]은 웹 기반의 Java 바이트 코드의 이해를 지원하는 XML 문서를 생성하는 과정을 나타내고 있다. 웹 상에서



[그림 3] 웹 기반의 Java 바이트 코드의 이해를 지원하는 XML 문서 생성

클라이언트가 Java 바이트 코드인 클래스 파일을 업로드한 후에 서버는 자동 변환된 XML 문서를 서버측이 다시 클라이언트가 다운로드할 수 있는 기능을 제공한다. 클라이언트는 웹 상에서 다운로드한 XML 문서로 Java 바이트 코드보다 나은 동적· 정적인 정보를 얻을 수 있다. 따라서 웹 기반 환경에서 정보를 상호 공유할 수 있고 Java 프로그램을 더 쉽게 이해할 수 있다.

4. 결론 및 향후 연구과제

본 연구에서는 웹 환경에서 클라이언트에게 Java 바이트 코드의 이해를 지원하는 XML 문서를 자동으로 생성하여 제공하는 것을 제안하였다. Java 프로그램을 분석하는 경우에 복잡한 프로그램에 대한 로직과 문서화에 대한 정보가 부족하여 프로그램을 이해하는데 어려움이 있다. 따라서 프로그래머에게 Java 바이트 코드를 역어셈블해서 얻은 동적인 정보와 XML 확장 태그, DTD, XSL에 의한 정적인 정보를 제공할 필요가 있다. 또한 이 모든 정보를 웹 기반에서 정보 공유 측면으로 클라이언트에게 제공하려고 한다. 따라서 클라이언트가 다운로드한 XML 문서로 Java 프로그램을 보다 쉽게 이해할 수 있는 장점이 있다.

향후 연구방향은 설계한 내용을 객체지향언어인 Java로 구현하는 것이 필요하다. 그리고 복잡한 로직이 있는 Java 프로그램을 쉽게 이해하려는 프로그래머측면에서 본 논문에서 제시한 XML 파서가 분석하는 동적· 정적인 정보 외에 추가할 수 있는 정보에 대한 연구와 클라이언트/서버 환경에서 저장소(repository)의 자료를 관리하는 과정에 대한 연구가 필요하다.

참고문헌

[1] Douglas Kramer, "API documentation from source code comments : a case study of Javadoc", Proc. ACM SIGDOC of 17th Intl. Conf. on Computer documentation, pp. 147~153, New Orleans, LA USA 12~14 September, 1999.

[2] Frank Boumphrey, Olivia Direnzo, Jon Duckett, Joe Graf, Paul Houle, Dave Hollander, Trevor Jenkins, et al., "Professional XML Applications", p.53, Wrox, 1999.

[3] Alex Ceponkus and Faraz Hoodbhoy, "Applied XML", p.68, John Wiley & Sons, 1999.

[4] Bill Venners, Inside the Java Virtual Machine, McGraw-Hill, New York, 1998.

[5] 류동향, 정민수, "자바 바이트 코드의 분석기의 설계 및 구현", 한국정보과학회 봄 학술발표논문집, pp. 77~79, Vol. 25, No. 1, 1998.

[6] 박옥자, "Java 프로그램의 품질평가를 지원하는 메트릭스 측정 시스템", 전북대학교 대학원 석사 논문, 1999.

[7] Sun Microsystems, Javap, http://java.sun.com/products

[8] XML_Parsers, http://www.xml.com/pub/Guide/